

William Stallings

Arquitetura e Organização de Computadores

8ª Edição

retrospectiva



Organização e arquitetura

- Arquitetura são os atributos visíveis ao programador.
 - Conjunto de instruções, número de bits usados para representação de dados, mecanismos de E/S, técnicas de endereçamento.
 - Por exemplo, existe uma instrução de multiplicação?
- Organização é como os recursos são implementados.
 - Sinais de controle, interfaces, tecnologia de memória.
 - p.e., existe uma unidade de multiplicação no hardware ou ela é feita pela adição repetitiva?

**ARQUITETURA E ORGANIZAÇÃO
DE COMPUTADORES**

- Toda a família Intel x86 compartilha a mesma arquitetura básica.
- A organização é diferente entre diferentes versões.

Estrutura e função

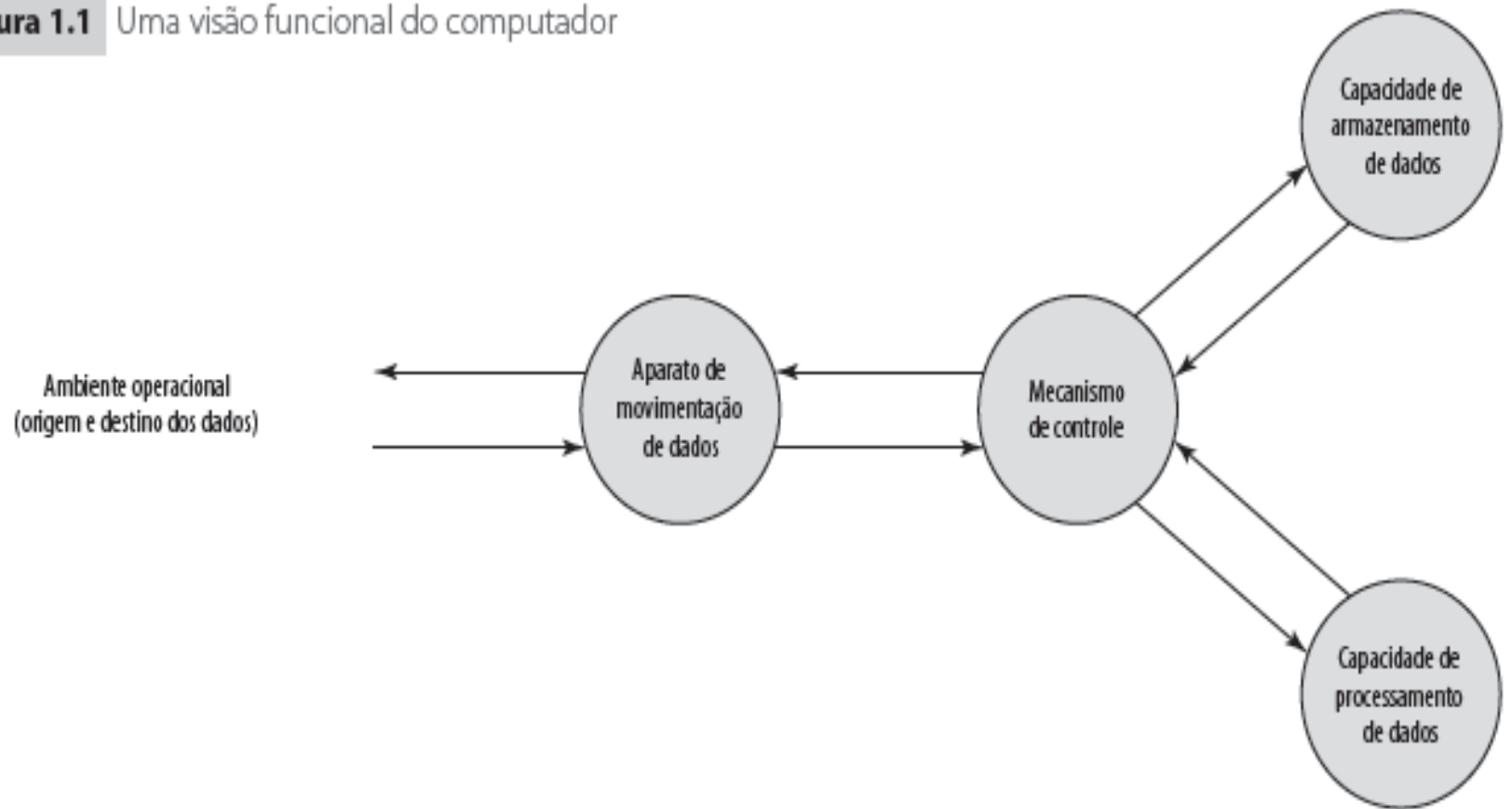
- Estrutura é o modo como os componentes são inter-relacionados.
- Função é a operação individual de cada componente como parte da estrutura.

Função

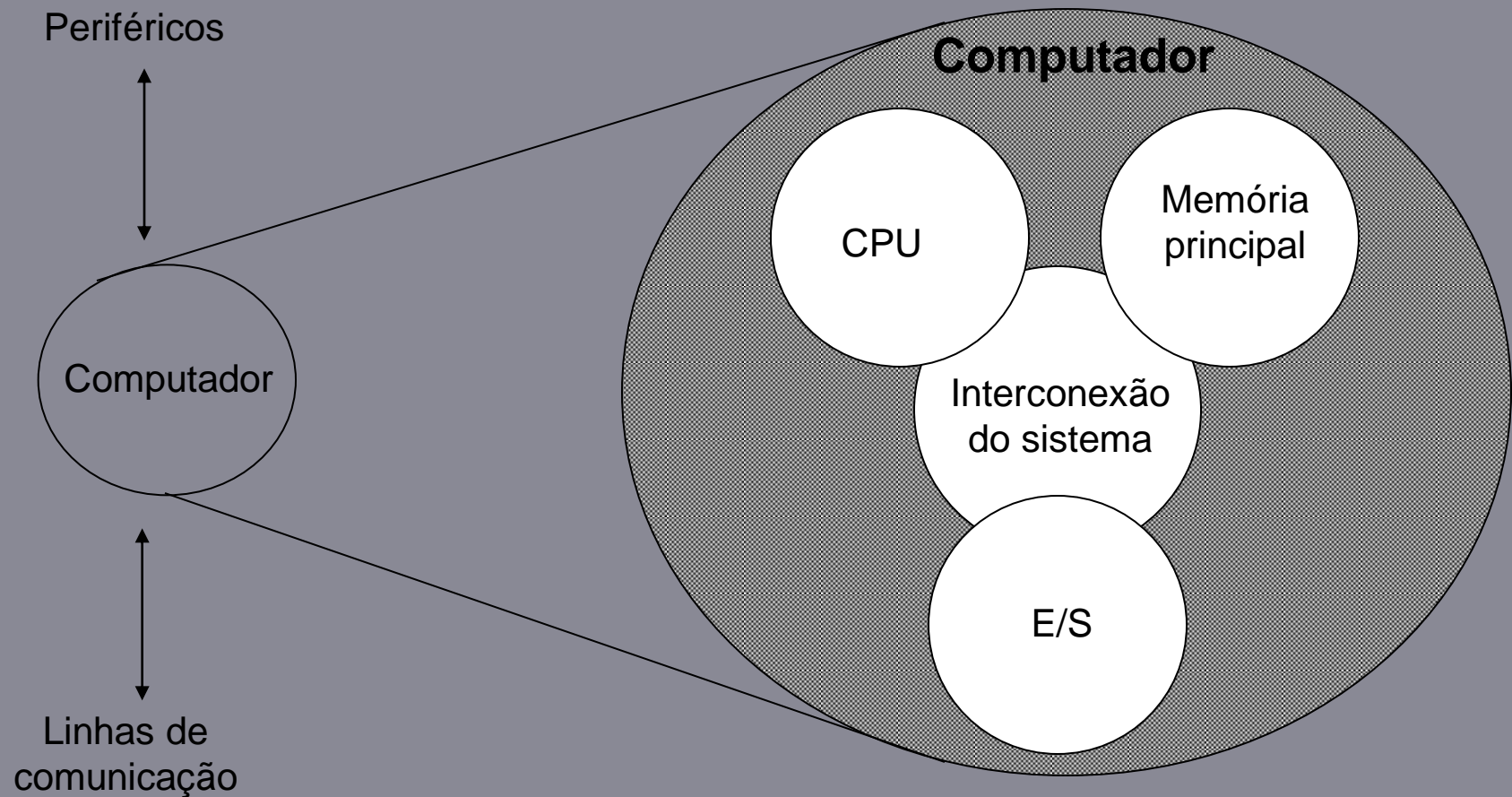
- Todas as funções do computador são:
 - Processamento de dados.
 - Armazenamento de dados.
 - Movimentação de dados.
 - Controle.

Visão funcional

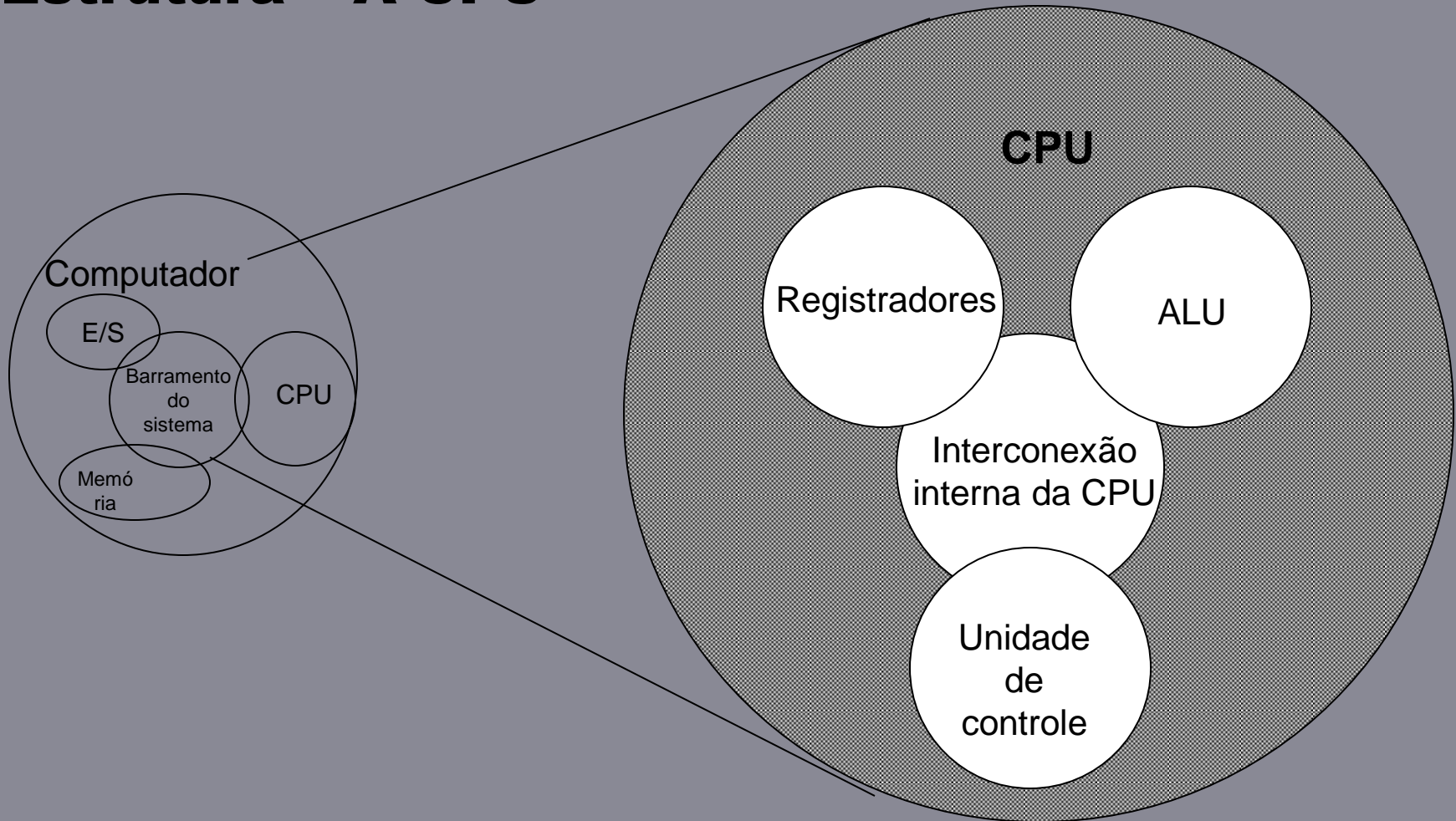
Figura 1.1 Uma visão funcional do computador



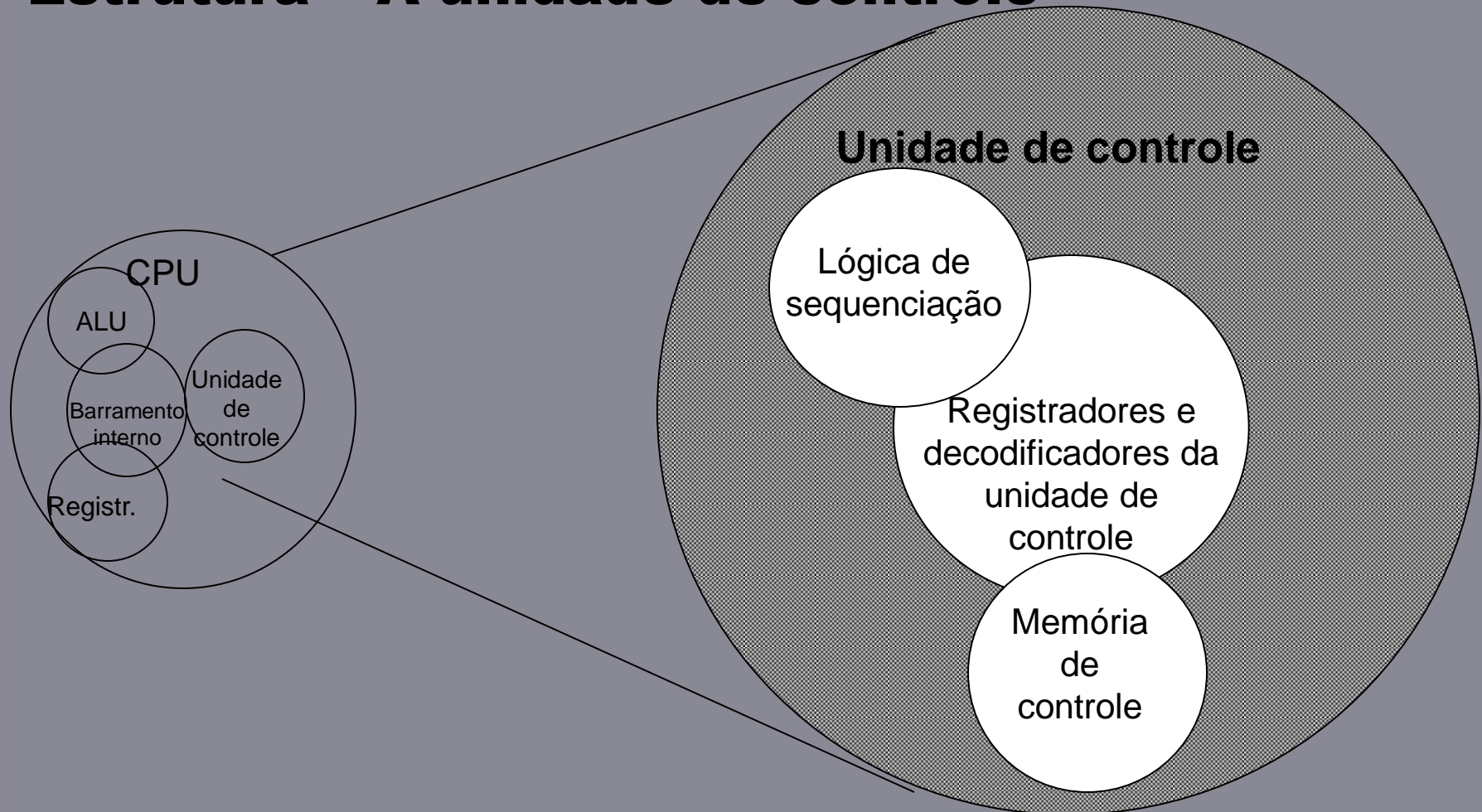
Estrutura – Alto nível



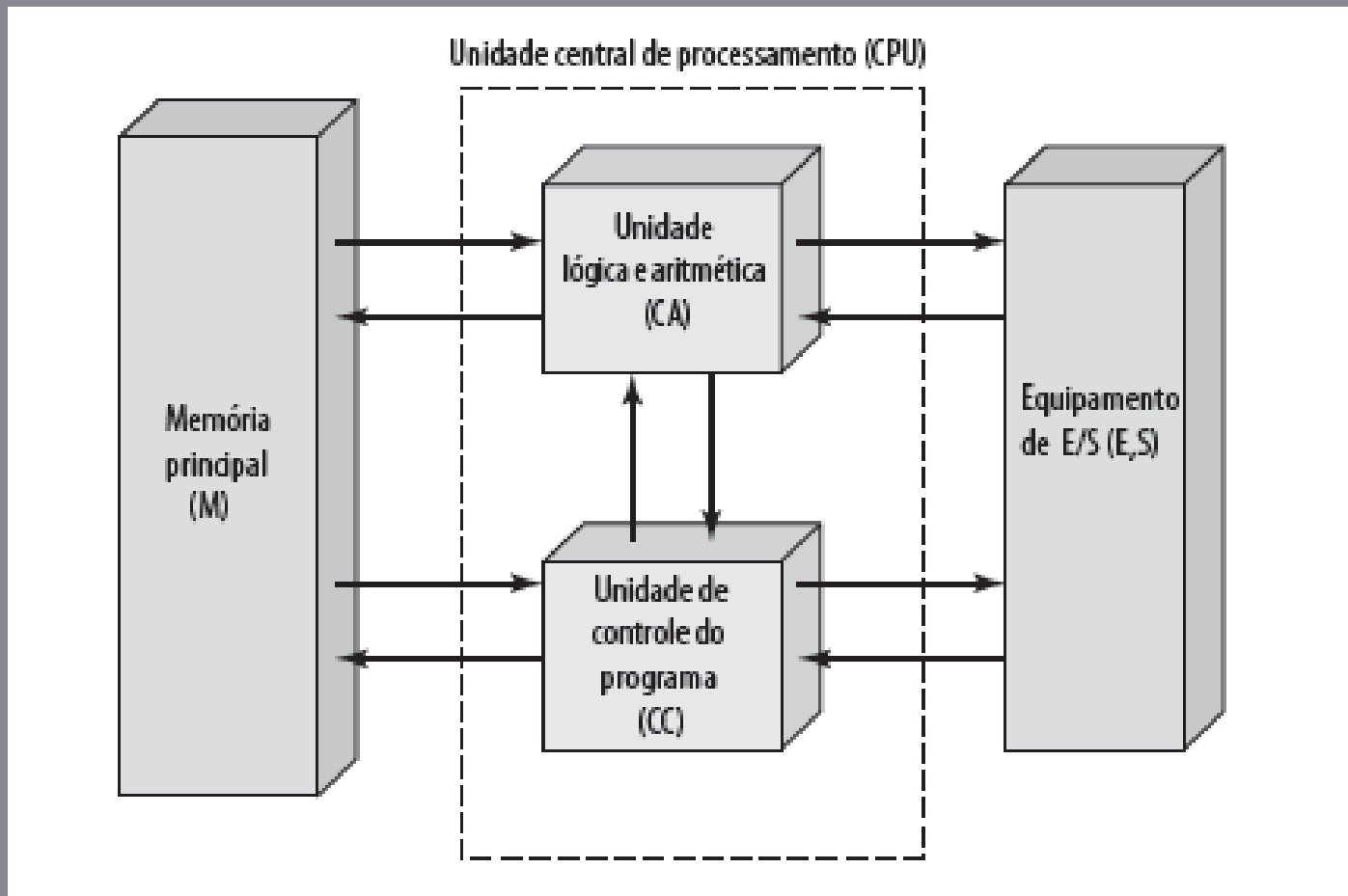
Estrutura – A CPU



Estrutura – A unidade de controle



Estrutura da máquina de von Neumann

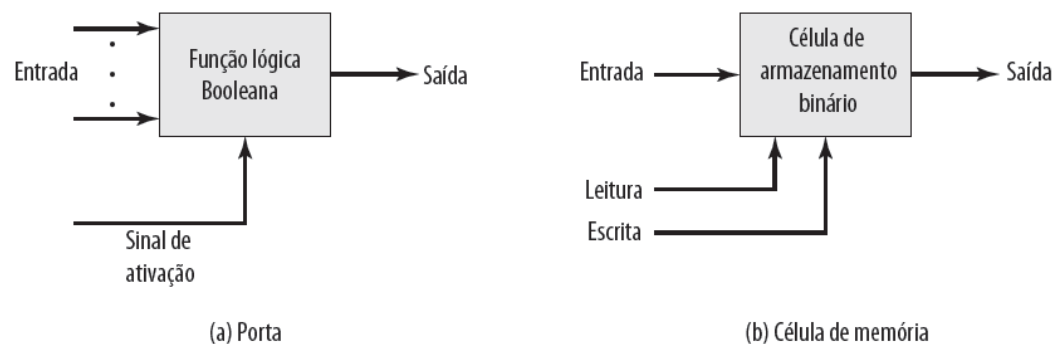


Terceira Geração: circuitos integrados

A Microeletrônica

- Literalmente – “pequena eletrônica”.
- Um computador é composto de portas, células de memória e interconexões.

Figura 2.6 Elementos fundamentais do computador



Circuitos aritméticos

Somador simples

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Vai-um

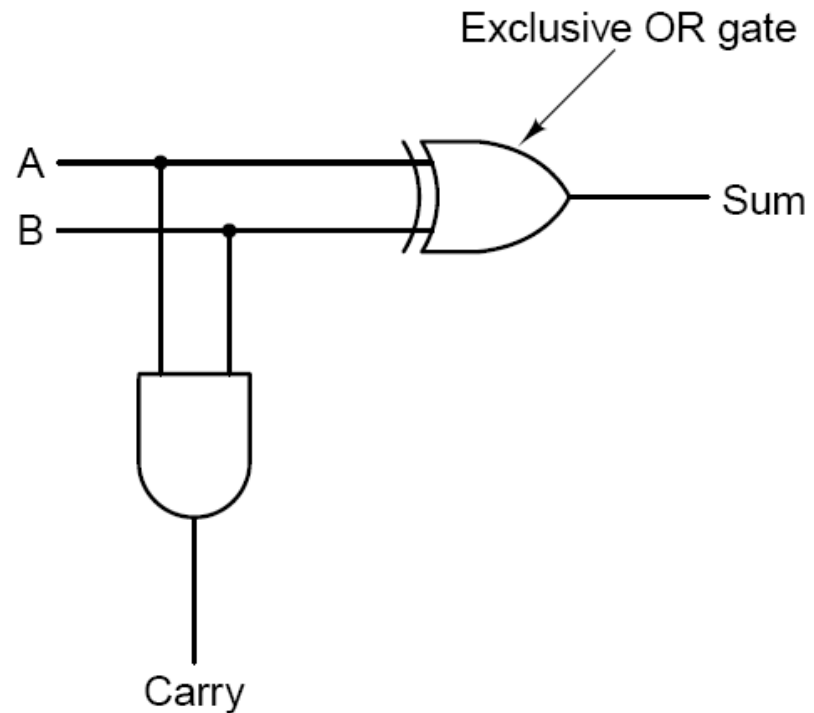


Figure 3-17. (a) Truth table for 1-bit addition. (b) A circuit for a half adder.

Memória

Não existe computador sem memória

A memória é útil para armazenar dados e instruções

Como representamos a memória em seu nível mais elementar para o nosso curso: o nível das portas ??

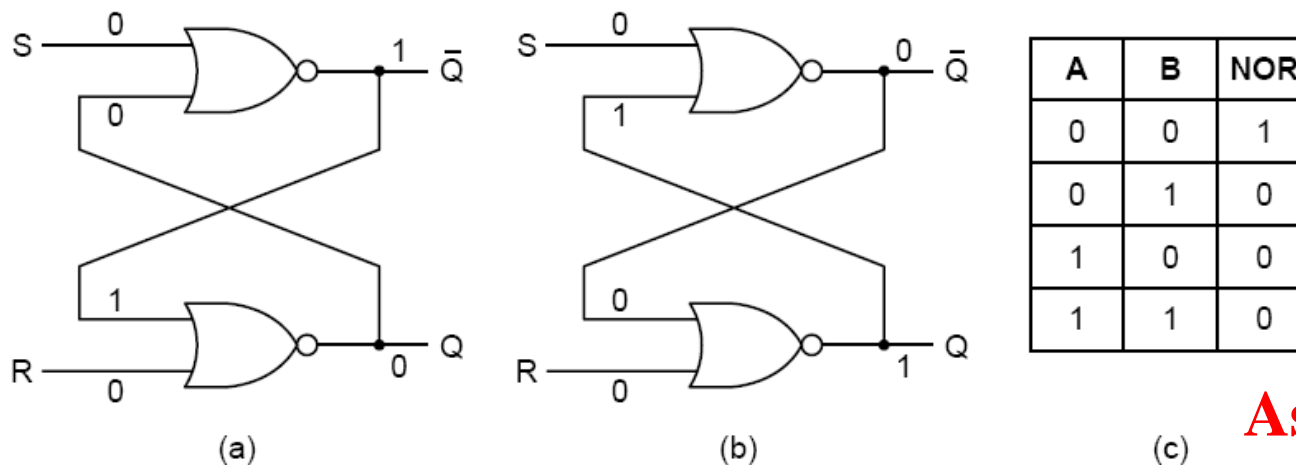


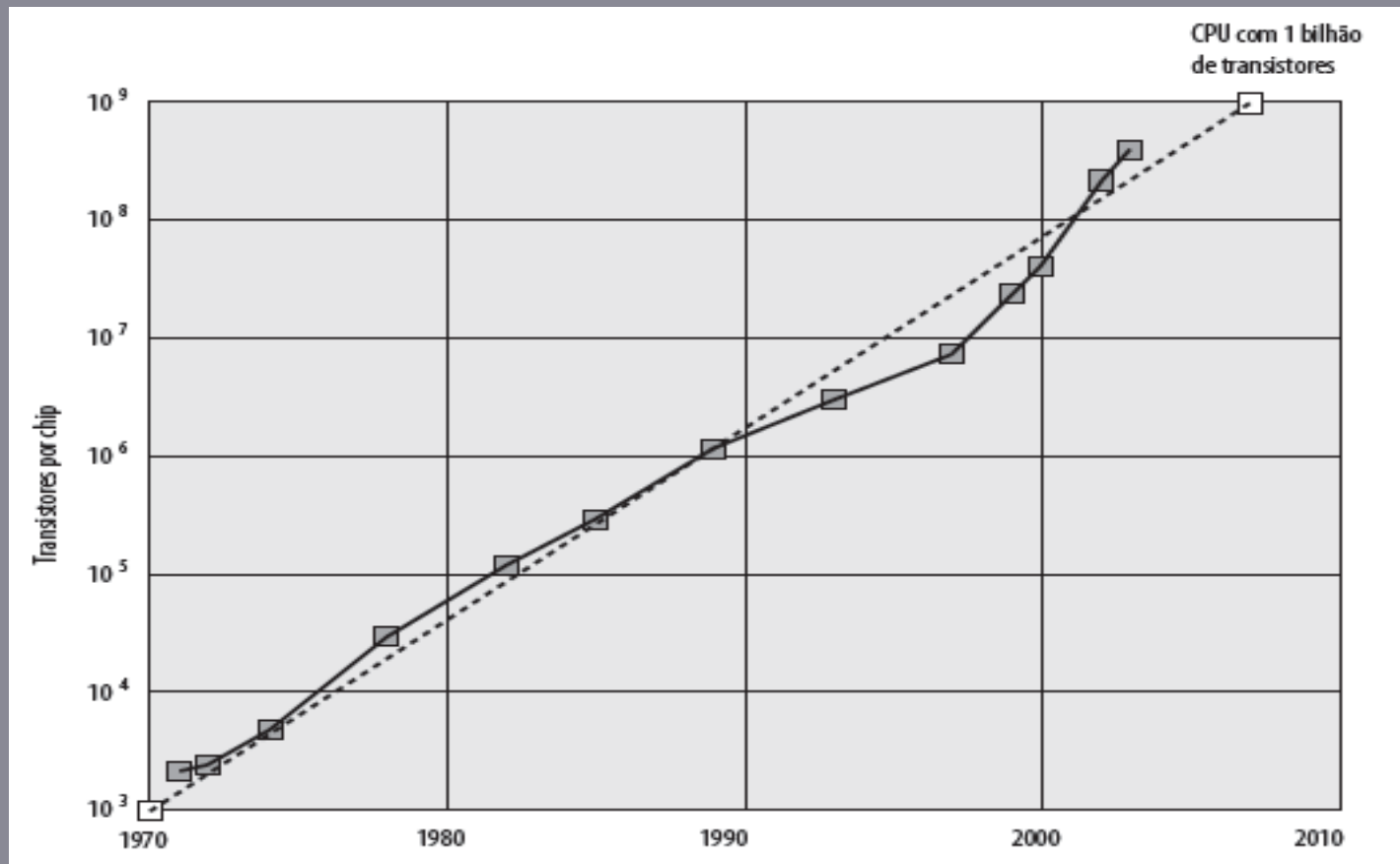
Figure 3-22. (a) NOR latch in state 0. (b) NOR latch in state 1. (c) Truth table for NOR.

**Assim se projeta
desde registradores
até memórias de
grande capacidade**

Lei de Moore

- Maior densidade de componentes no chip.
- Gordon Moore – cofundador da Intel.
- Número de transistores em um chip dobrará a cada ano.
- Desde 1970, isso diminuiu um pouco.
 - Número de transistores dobra a cada 18 meses.
- Custo de um chip permaneceu quase inalterado.
- Maior densidade de empacotamento significa caminhos elétricos mais curtos, gerando maior desempenho.
- Menor tamanho oferece maior flexibilidade.
- Redução nos requisitos de potência e resfriamento.
- Menos interconexões aumenta a confiabilidade.

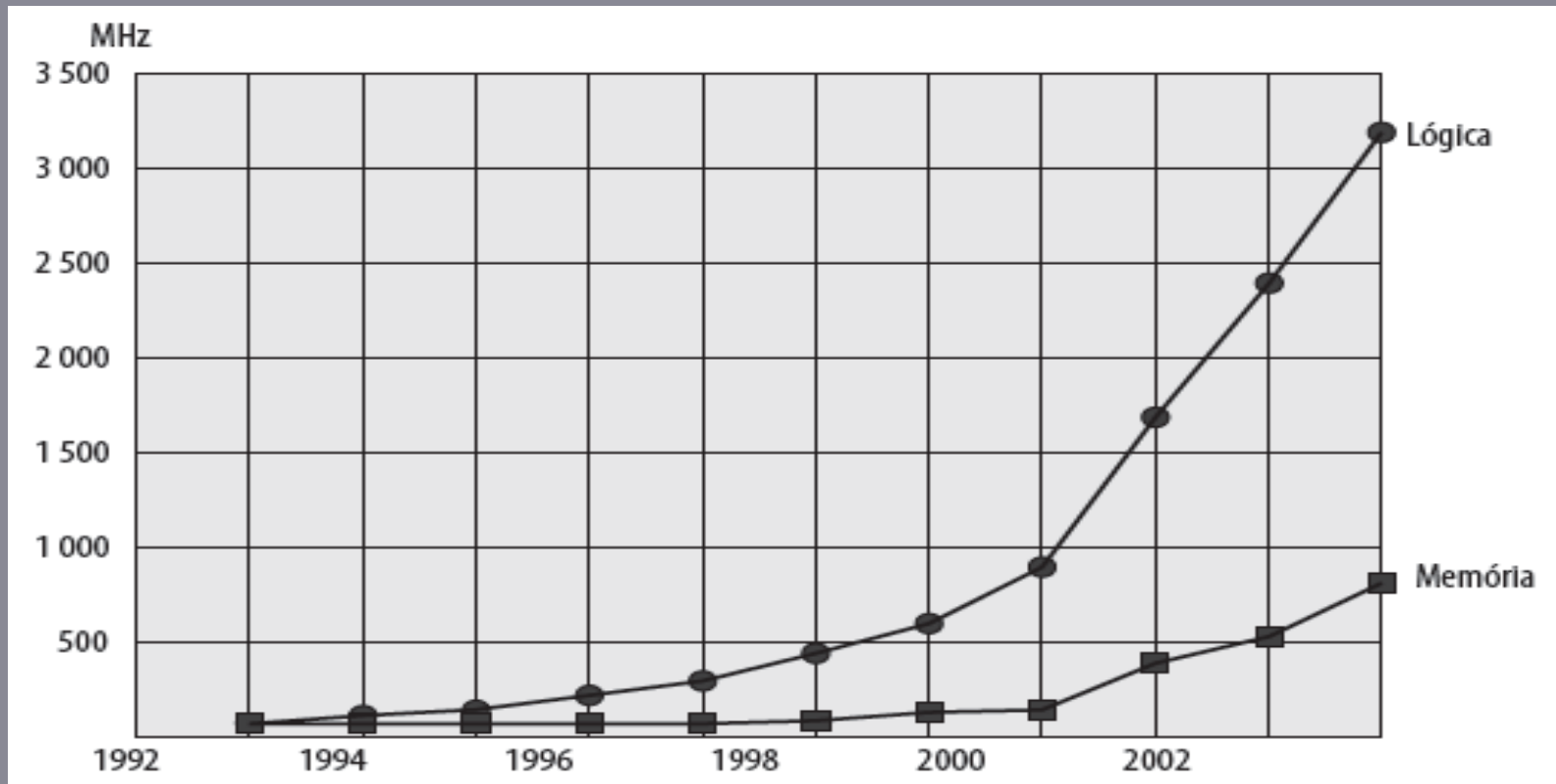
Crescimento na contagem de transistores da CPU



Balanco do desempenho

- Aumento da velocidade do processador.
- Aumento da capacidade de memória.
- Velocidade da memória fica para trás da velocidade do processador.

Diferença de desempenho entre lógica e memória



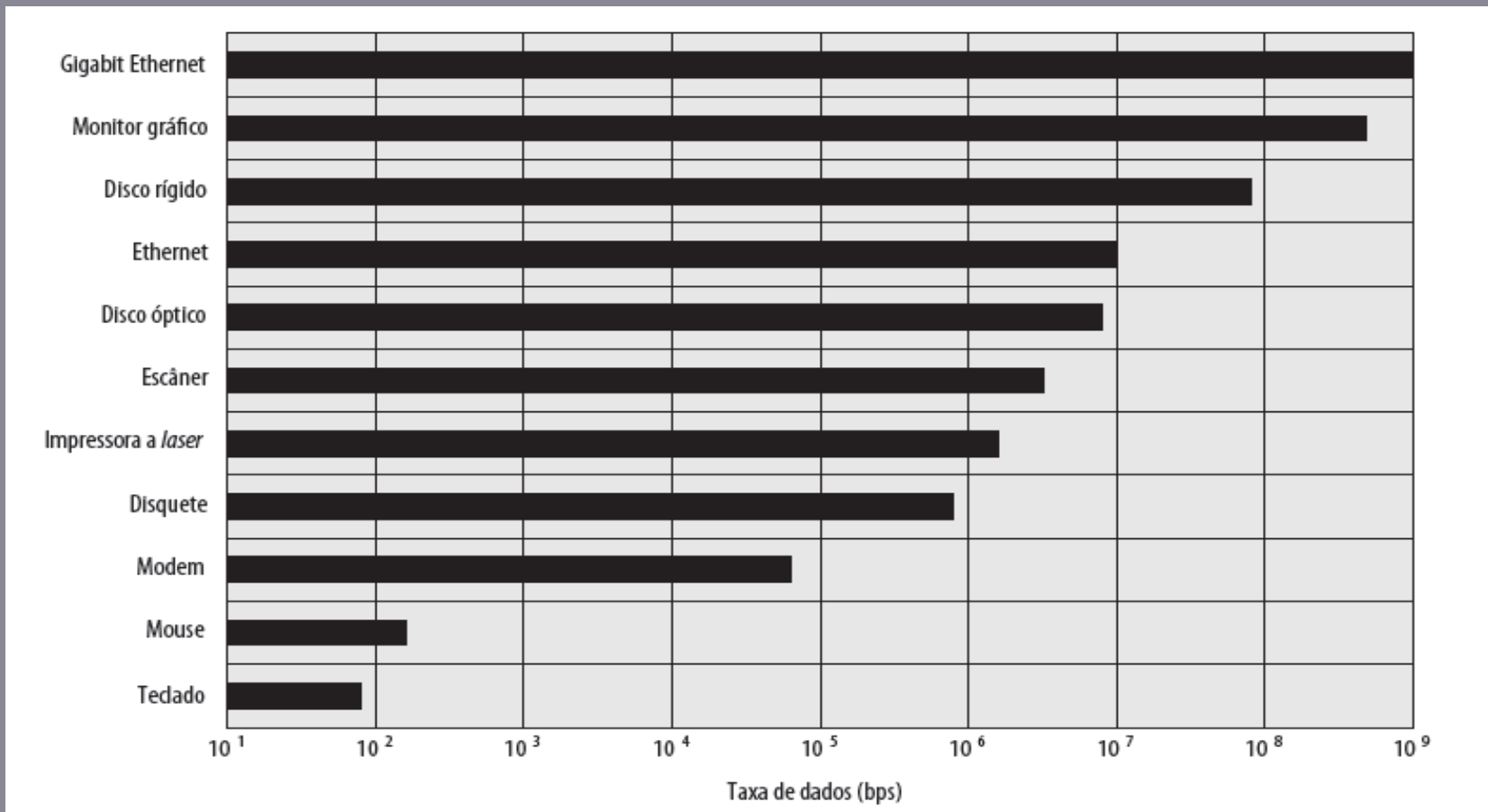
Soluções

- Aumentar número de bits recuperados de uma só vez.
 - Tornar DRAM “mais larga” ao invés de “mais profunda”.
- Mudar interface da DRAM.
 - Cache.
- Reduzir frequência de acesso à memória.
 - Cache mais complexa e cache no chip.
- Aumentar largura de banda de interconexão.
 - Barramentos de alta velocidade.
 - Hierarquia de barramentos.

E O MUNDO EXTERNO



Taxas de dados típicas dos dispositivos de E/S



Soluções:

Caching.

Buffering.

Barramentos de interconexão de maior velocidade.

Estruturas de barramentos mais elaboradas.

Configurações de processador múltiplo.

A chave é o balanço

- Componentes do processador.
- Memória principal.
- Dispositivos de E/S.
- Estrutura de interconexão.

Melhorias na organização e na arquitetura do chip

- Aumentar velocidade de hardware do processador.
 - Deve-se fundamentalmente ao encolhimento do tamanho das portas lógicas no chip.
 - Mais portas, reunidas mais de perto, aumentando a taxa de clock.
 - Redução no tempo de propagação dos sinais.
- Aumentar tamanho e velocidade das caches.
 - Dedicando parte do chip do processador para cache.
 - Tempos de acesso à cache caem significativamente.
- Mudar organização e arquitetura do processador.
 - Aumenta velocidade de execução efetiva.
 - Paralelismo.

Aumento da capacidade de cache

- Normalmente, dois ou três níveis de cache entre processador e memória principal.
- Densidade de chip aumentada.
 - Mais memória cache no chip.
 - Acesso mais rápido à cache.
- Chip Pentium dedicou cerca de 10% da área do chip à cache.
- Pentium 4 dedica cerca de 50%.

Lógica de execução mais complexa

- Permite execução de instruções em paralelo.
- Pipeline funciona como linha de montagem.
 - Diferentes estágios de execução de diferentes instruções ao mesmo tempo ao longo do pipeline.
- Superescalar permite múltiplos pipelines dentro de um único processador.
 - Instruções que não dependem uma da outra podem ser executadas em paralelo.

Nova técnica – múltiplos cores

- Múltiplos processadores em único chip.
 - Grande cache compartilhada.
- Dentro de um processador, aumento no desempenho proporcional à raiz quadrada do aumento na complexidade.
- Se o software puder usar múltiplos processadores, dobrar o número de processadores quase dobra o desempenho.
- Assim, use dois processadores mais simples no chip ao invés de um processador mais complexo.
- Com dois processadores, caches maiores são justificadas.
 - Consumo de potência da lógica de memória menor que lógica do processamento.

JÁ QUE FALAMOS DE PARALELISMO !!!

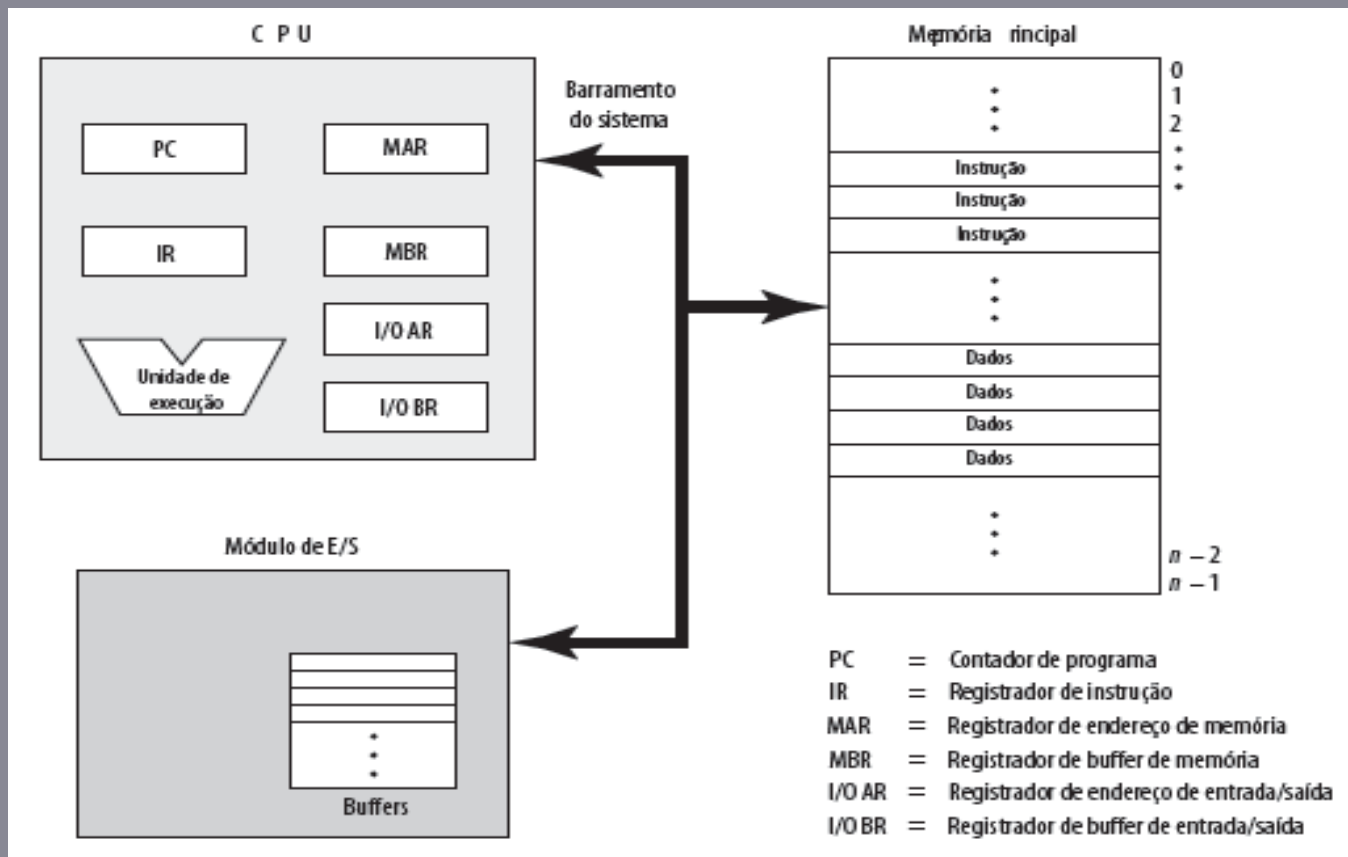
WILLIAM STALLINGS

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

Lei de Amdahl

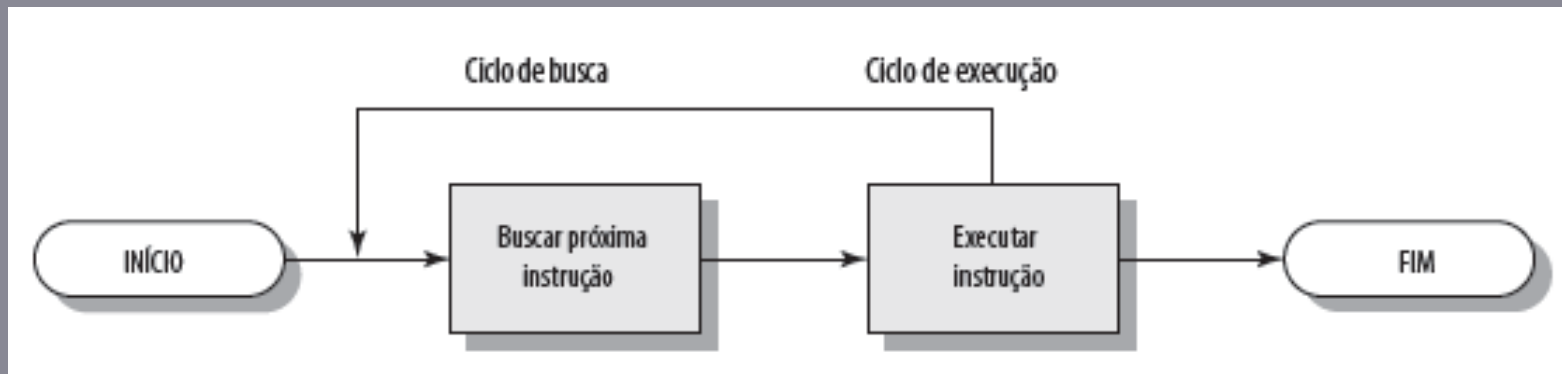
- Gene Amdahl [AMDA67].
- *Speedup* em potencial do programa usando múltiplos processadores.
- Concluiu que:
 - Código *precisa ser paralelizável*.
 - Speedup é limitado, gerando retornos decrescentes para uso de mais processadores.
- Dependente da tarefa:
 - Servidores ganham mantendo múltiplas conexões em múltiplos processadores.
 - Bancos de dados podem ser divididos em tarefas paralelas.

Componentes do computador: visão de alto nível



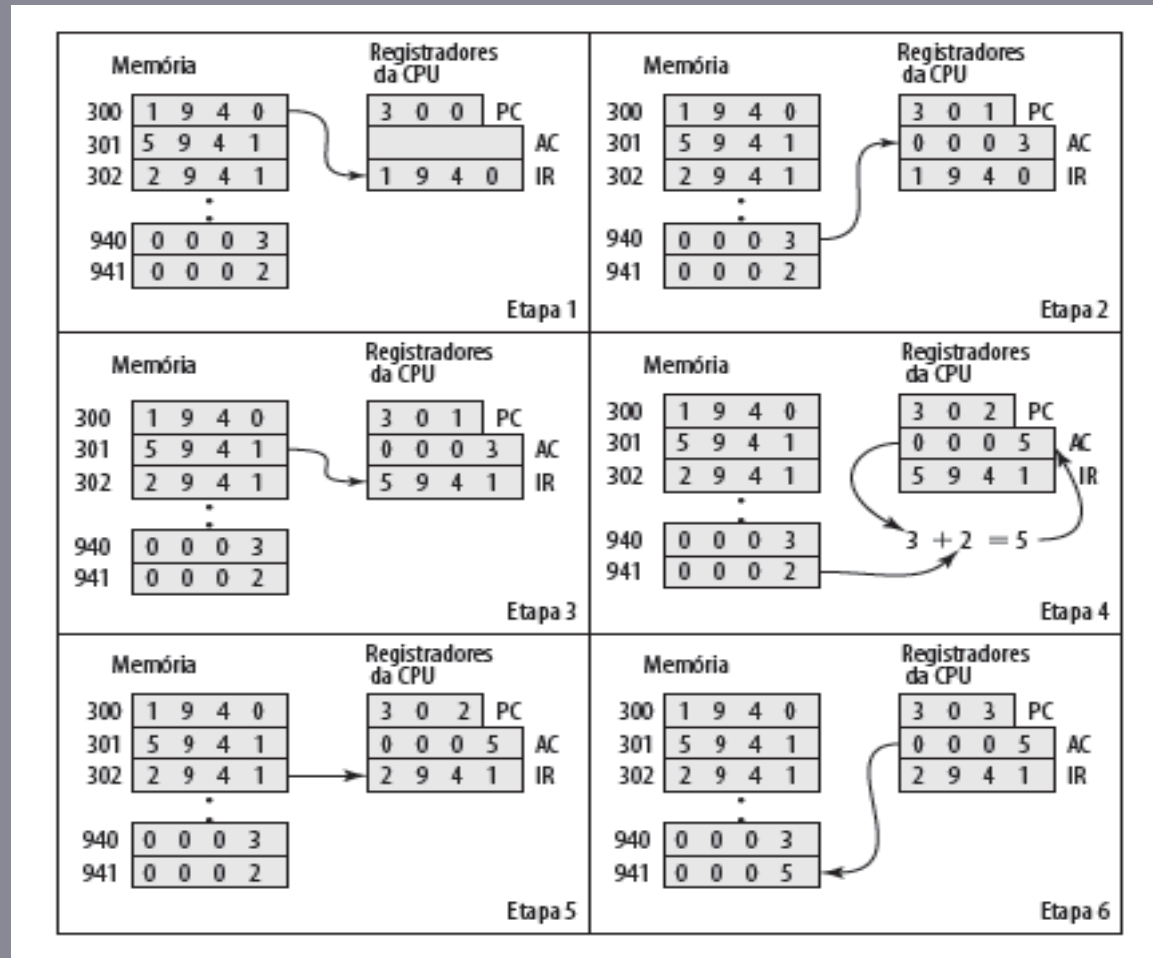
Ciclo de instrução

- Duas etapas:
 - Busca
 - Execução



ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

Exemplo de execução de programa



**AGORA VAMOS AOS DETALHES,
POIS SENÃO O CURSO NÃO É DE CIÊNCIA DA
COMPUTAÇÃO**

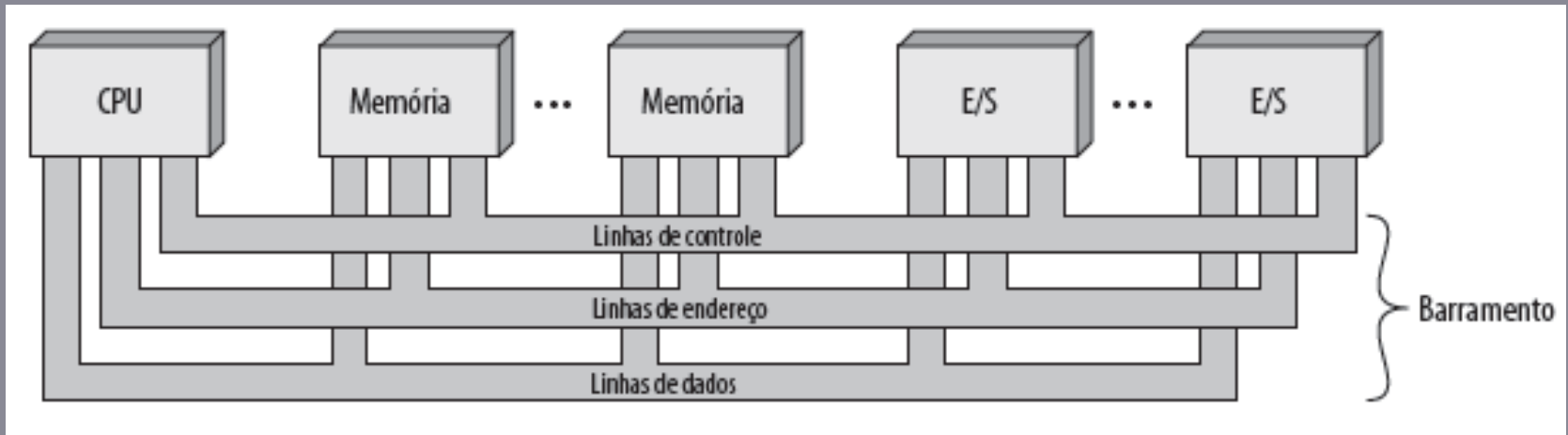
COMO INTERLIGAR TUDO (MEMÓRIA,
PROCESSADOR,
DISCOS, IMPRESSORAS, ETC...) ???

BARRAMENTOS

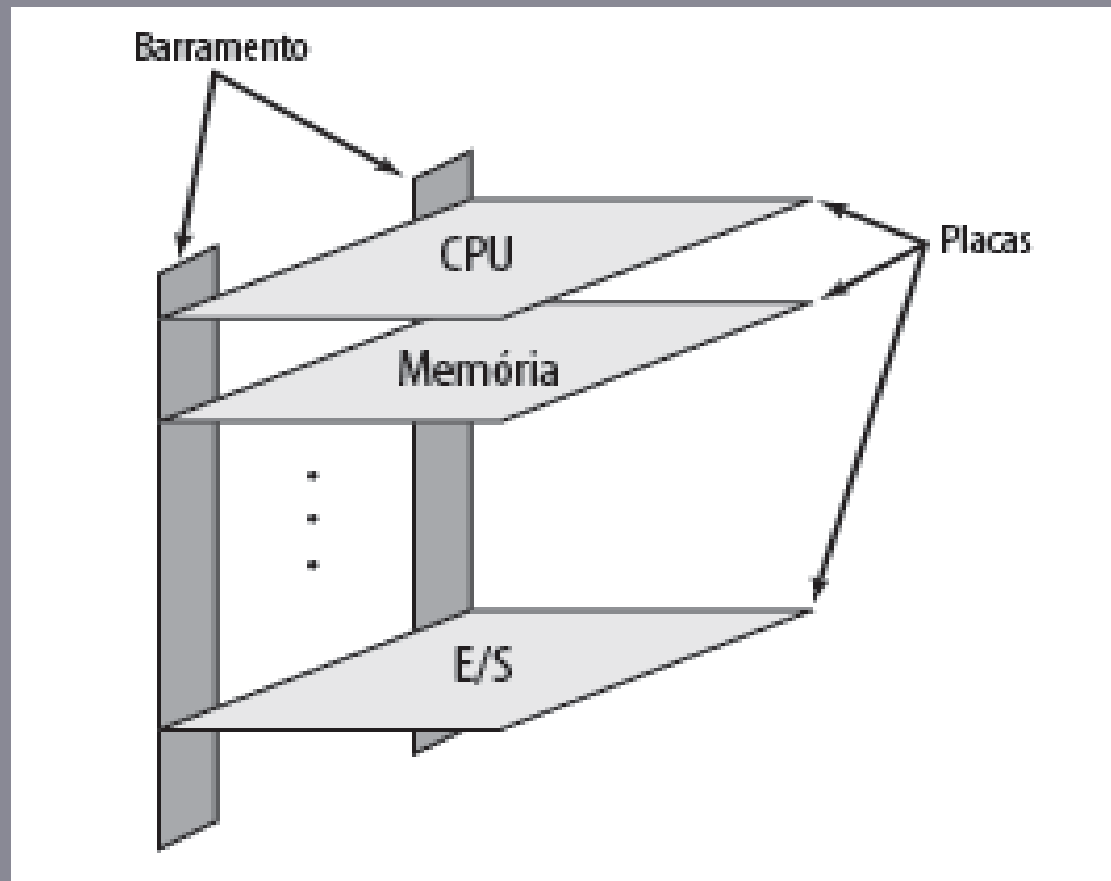
O que é um barramento?

- Um caminho de comunicação conectando dois ou mais dispositivos.
- Normalmente, broadcast.
- Frequentemente agrupado.
 - Uma série de canais em um barramento.
 - P.e., barramento de dados de 32 bits são 32 canais de bits separados.
- Linhas de potência podem não ser mostradas.

Esquema de interconexão de barramento



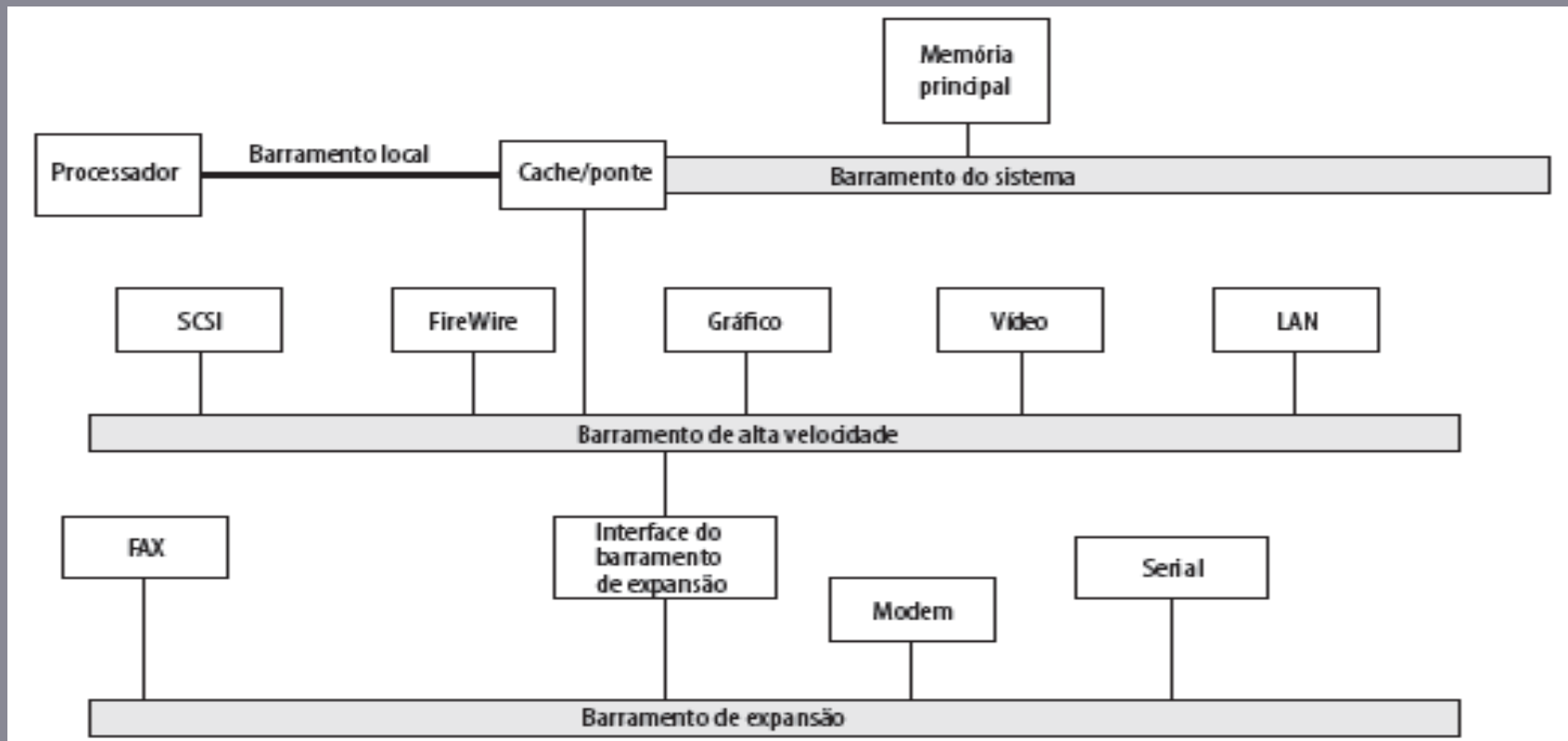
Realização física da arquitetura de barramento



Problemas do barramento único

- Muitos dispositivos em um barramento levam a:
 - Atrasos de propagação
 - Longos caminhos de dados significa que a coordenação do uso do barramento pode afetar contrariamente o desempenho.
 - Se a demanda de transferência de dados agregada se aproxima da capacidade do barramento.
- A maioria dos sistemas utiliza múltiplos barramentos para contornar esses problemas.

Arquitetura de alto desempenho



Tipos de barramento

- Dedicado:
 - Linhas separadas para dados e endereço.
- Multiplexado.
 - Linhas compartilhadas.
 - Linha de controle válida de endereço ou dados.
 - Vantagem – menos linhas
 - Desvantagens:
 - Controle mais complexo.
 - Desempenho máximo.

Arbitração de barramento

- Mais de um módulo controlando o barramento.
- P.e., CPU e controlador de DMA.
- Apenas um módulo pode controlar barramento de uma só vez.
- Arbitração pode ser centralizada ou distribuída.

Como a comunicação efetivamente ocorre
com o processador ??

INTERRUPÇÕES

Interrupções

- Mecanismo pelo qual outros módulos (p.e. E/S) podem interromper a sequência de processamento normal.

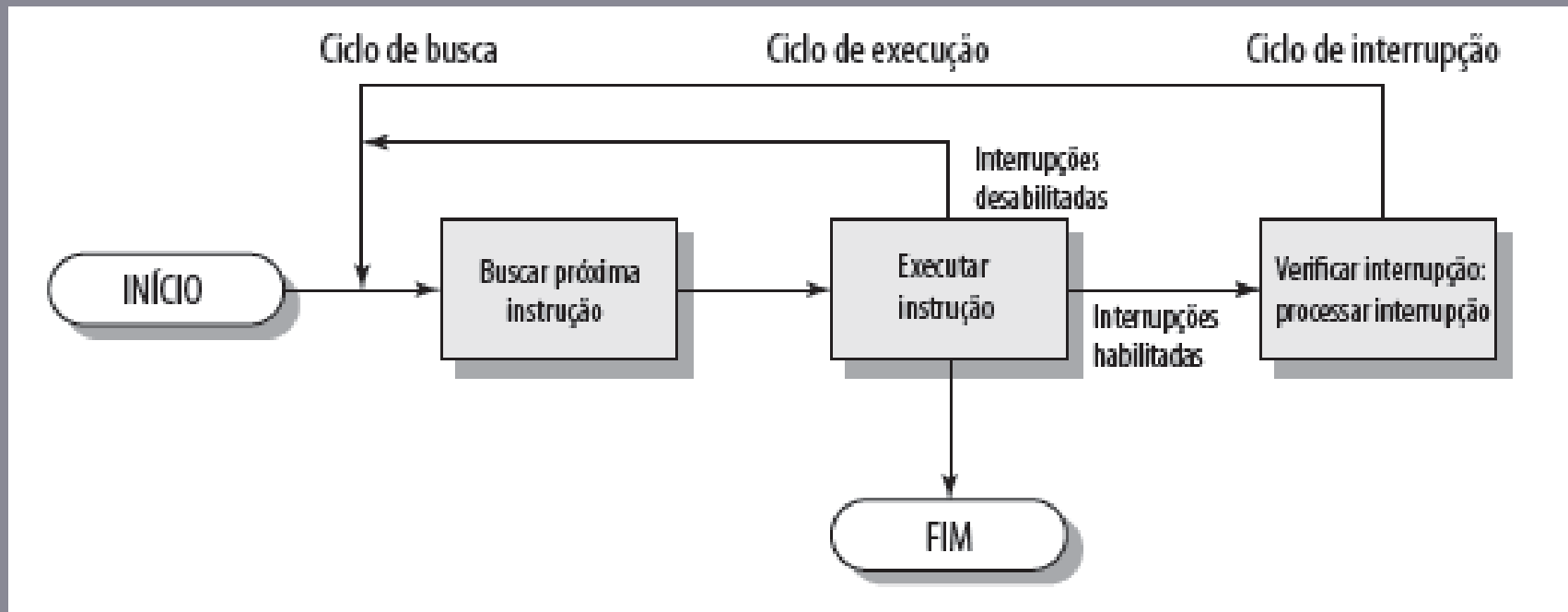
Tabela 3.1 Classes de interrupções

Programa	Gerada por alguma condição que ocorre como resultado da execução de uma instrução, como o <i>overflow</i> aritmético, divisão por zero, tentativa de executar uma instrução de máquina ilegal ou referência fora do espaço de memória permitido para o usuário.
Timer	Gerada por um timer dentro do processo. Isso permite que o sistema operacional realize certas funções regularmente.
E/S	Gerada por um controlador de E/S para sinalizar o término normal de uma operação ou para sinalizar uma série de condições de erro.
Falha de hardware	Gerada por uma falha como falta de energia ou erro de paridade de memória.

Ciclo de interrupção

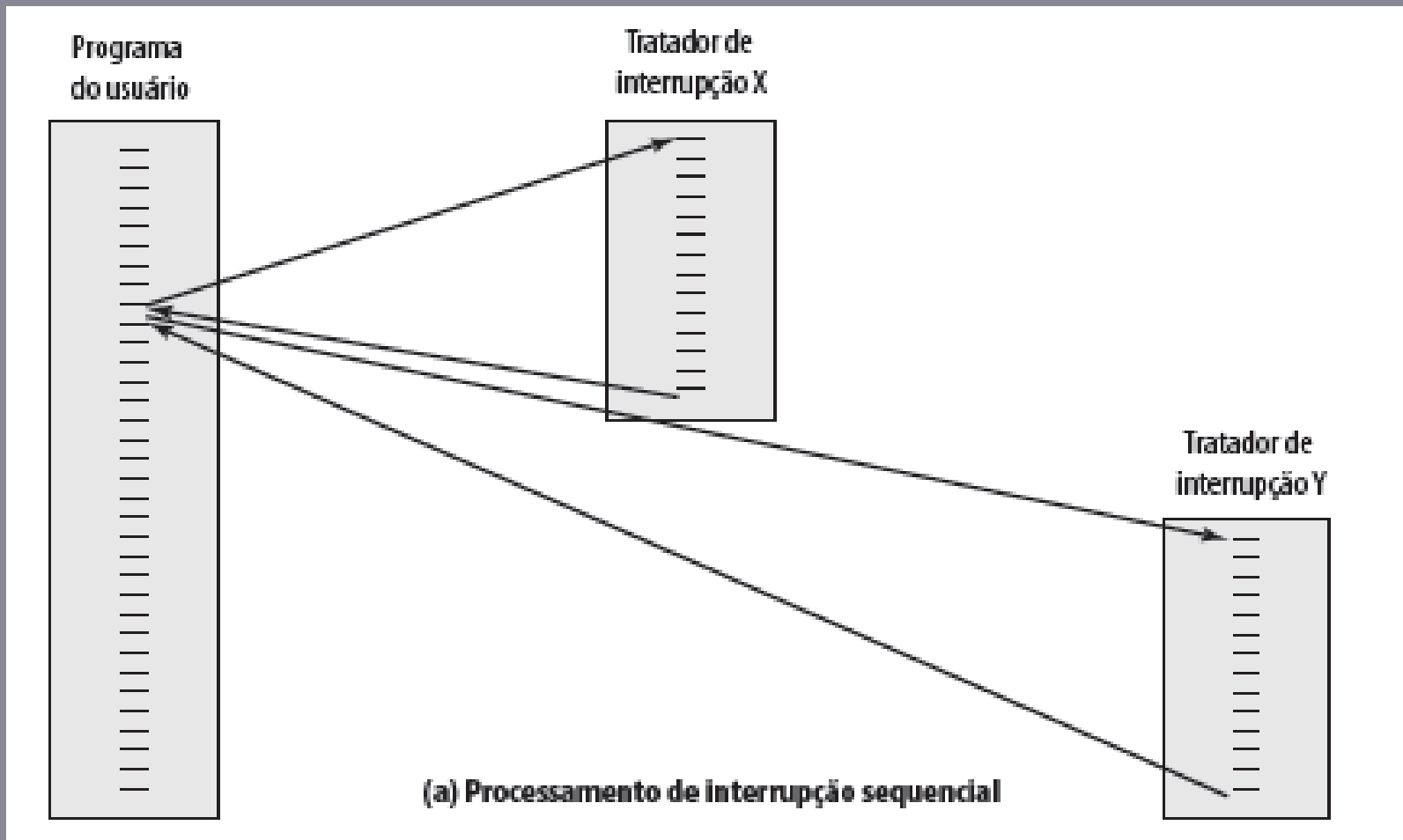
- Adicionado ao ciclo de instrução.
- Processador verifica interrupção.
 - Indicado por um sinal de interrupção.
- Se não houver interrupção, busca próxima instrução.
- Se houver interrupção pendente:
 - Suspende execução do programa atual.
 - Salva contexto.
 - Define PC para endereço inicial da rotina de tratamento de interrupção.
 - Interrupção de processo.
 - Restaura contexto e continua programa interrompido.

Ciclo de instrução com interrupções

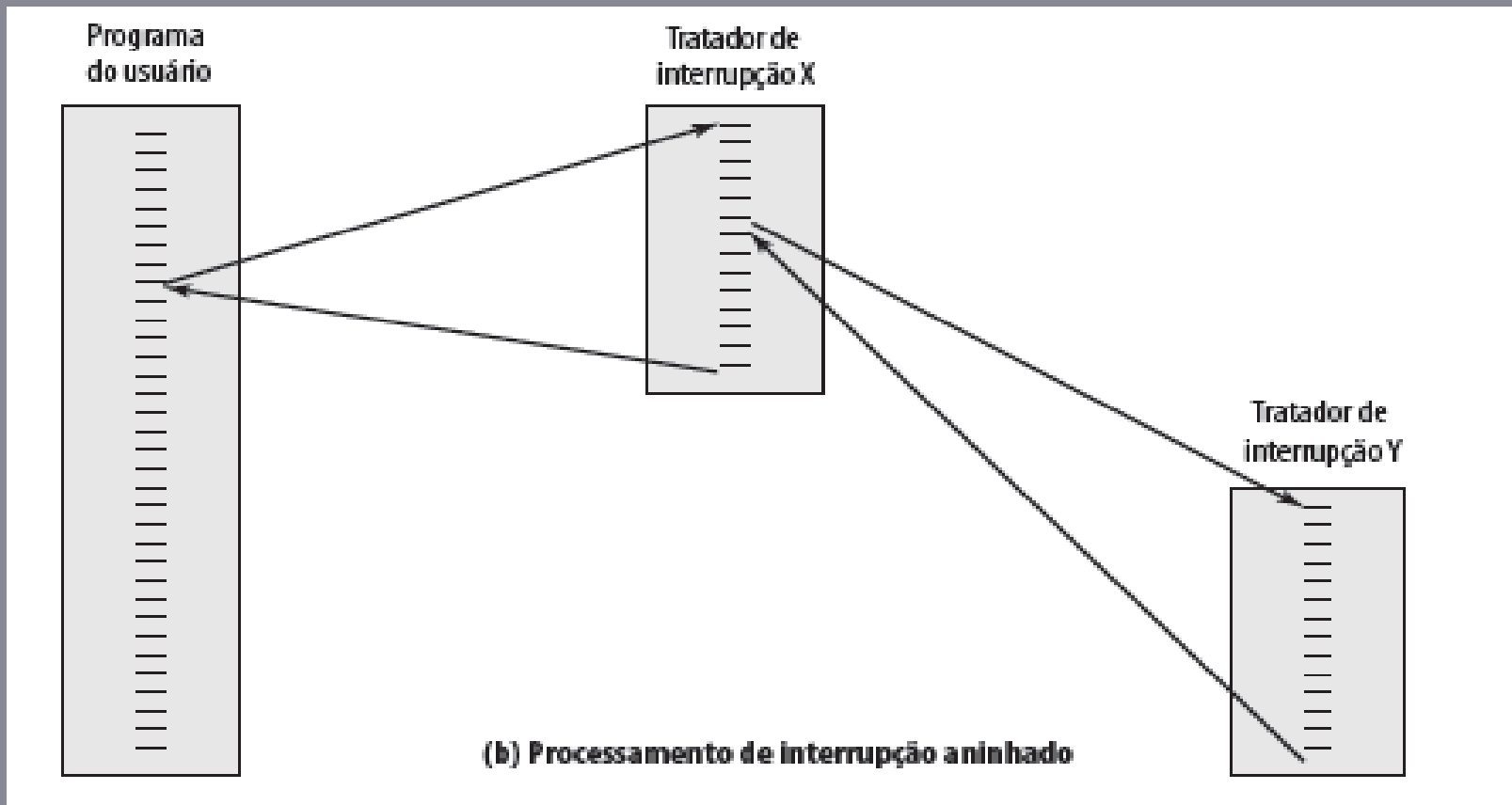


E SE TIVERMOS MÚLTIPLAS INTERRUPÇÕES??

Múltiplas interrupções – sequenciais



Múltiplas interrupções – aninhadas



MEMÓRIA INTERNA

CACHE

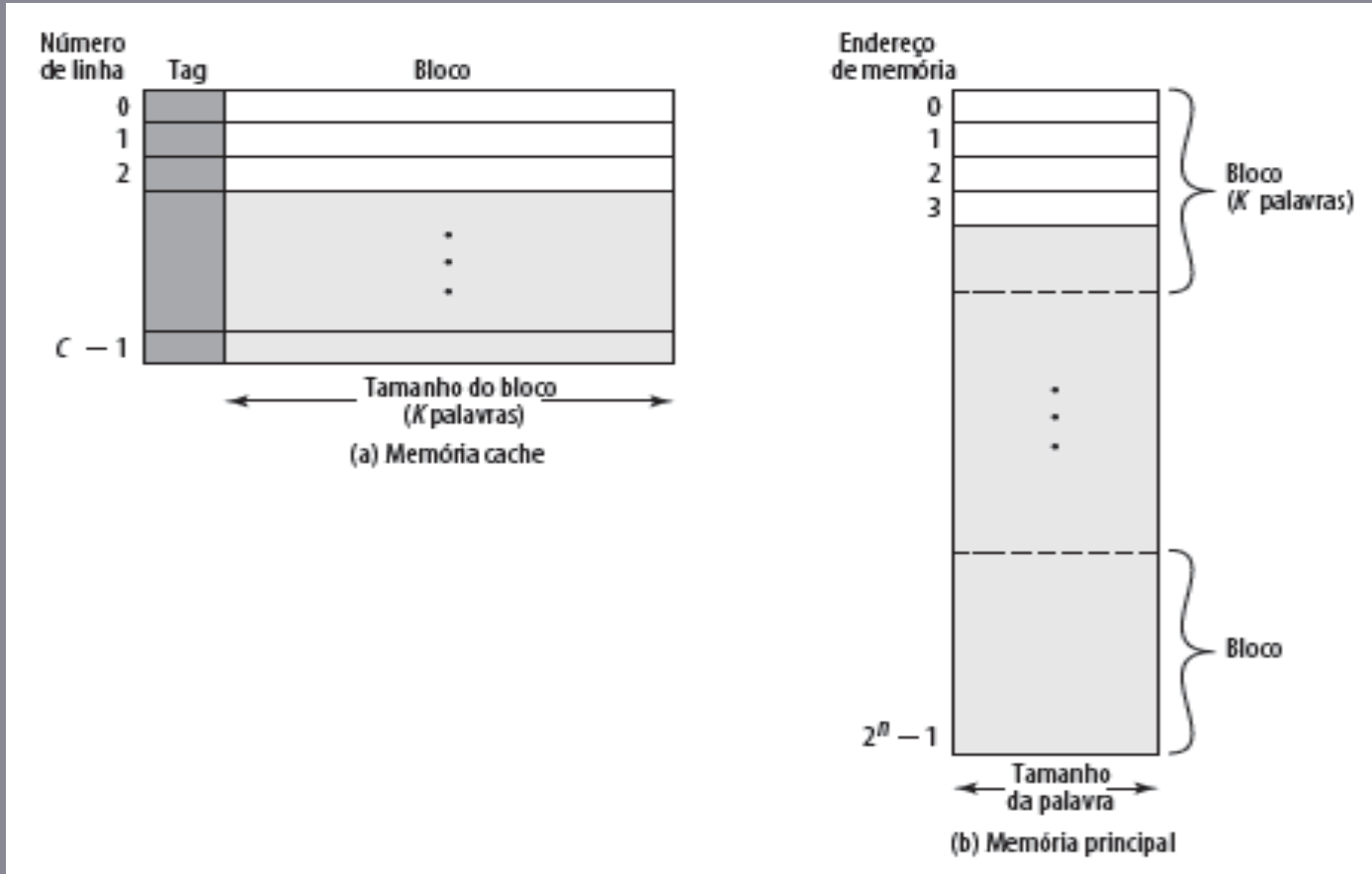
Desempenho

- Tempo de acesso (latência):
 - Tempo entre apresentar o endereço e obter os dados válidos.
- Tempo de ciclo de memória:
 - Tempo que pode ser exigido para a memória se “recuperar” antes do próximo acesso.
 - Tempo de ciclo é acesso + recuperação.
- Taxa de transferência:
 - Taxa em que os dados podem ser movidos.

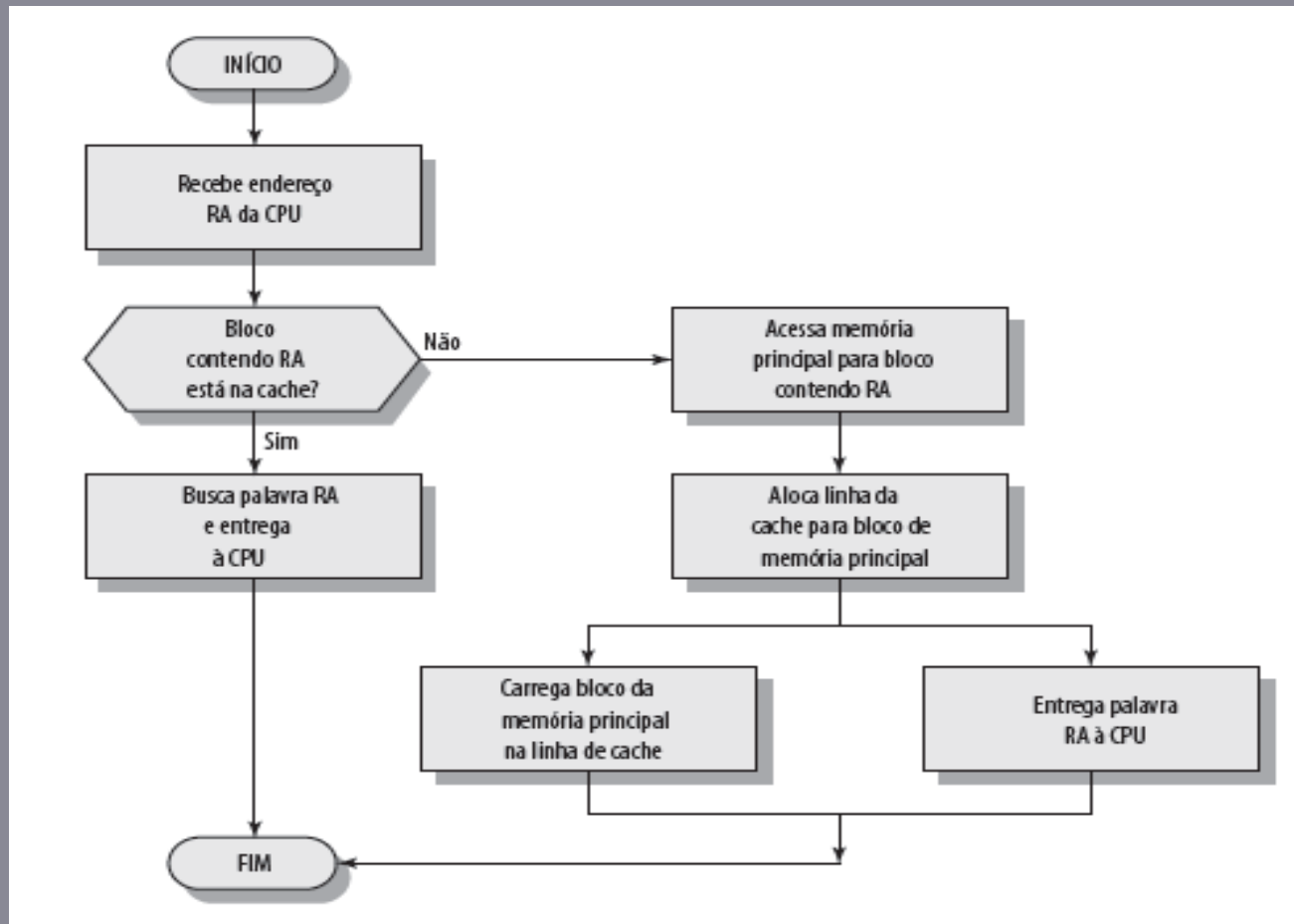
Localidade de referência

- Durante o curso da execução de um programa, as referências à memória tendem a se agrupar.
- P.e., loops.

Estrutura de cache/memória principal



Operação de leitura de cache – fluxograma



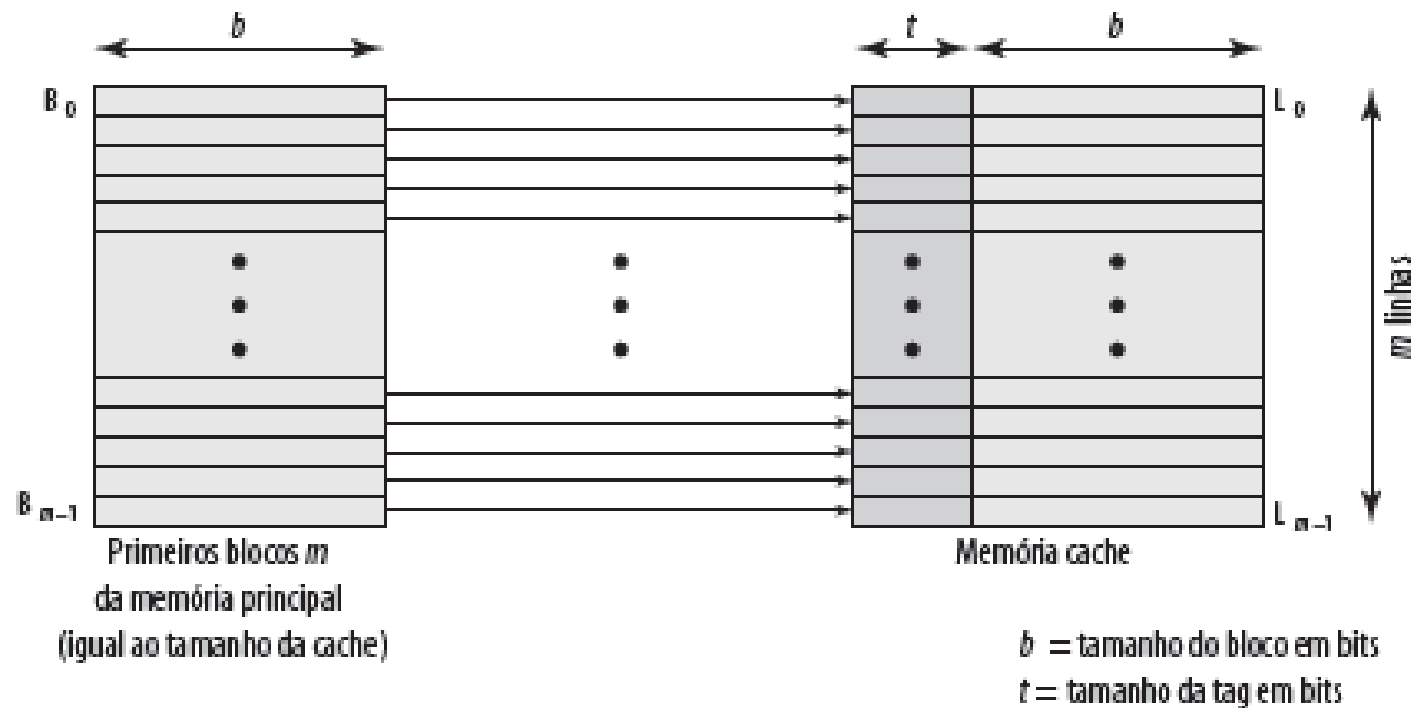
Endereçamento de cache

- Onde fica a cache?
 - Entre processador e unidade de gerenciamento de memória virtual.
 - Entre MMU e memória principal.
- Cache lógica (cache virtual) armazena dados usando endereço virtual.
 - Processador acessa cache diretamente, não através da cache física.
 - Acesso à cache mais rápido, antes da tradução de endereço da MMU.
 - Endereços virtuais usam o mesmo espaço de endereços para diferentes aplicações.
 - Deve esvaziar cache a cada troca de contexto.
- Cache física armazena dados usando endereços físicos da memória principal.

Tamanho não importa

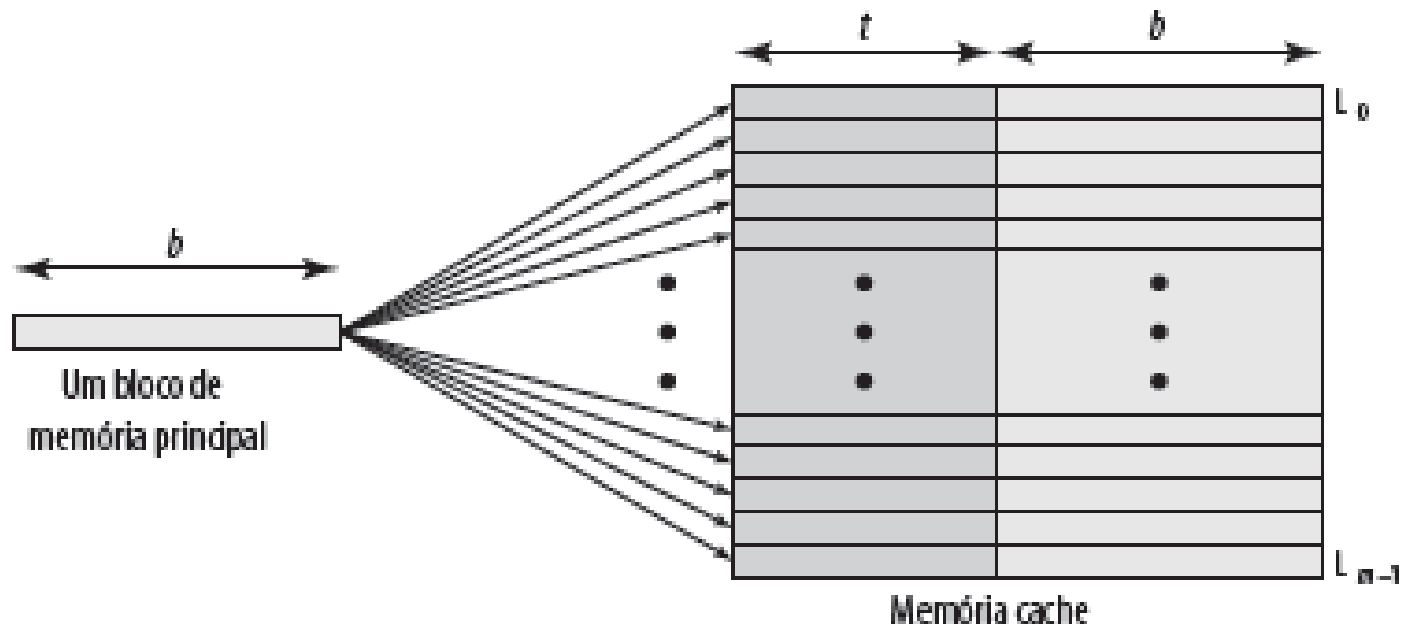
- Custo:
 - Mais cache é caro.
- Velocidade:
 - Mais cache é mais rápido (até certo ponto).
 - Verificar dados na cache leva tempo.

Mapeamento direto da cache para memória principal

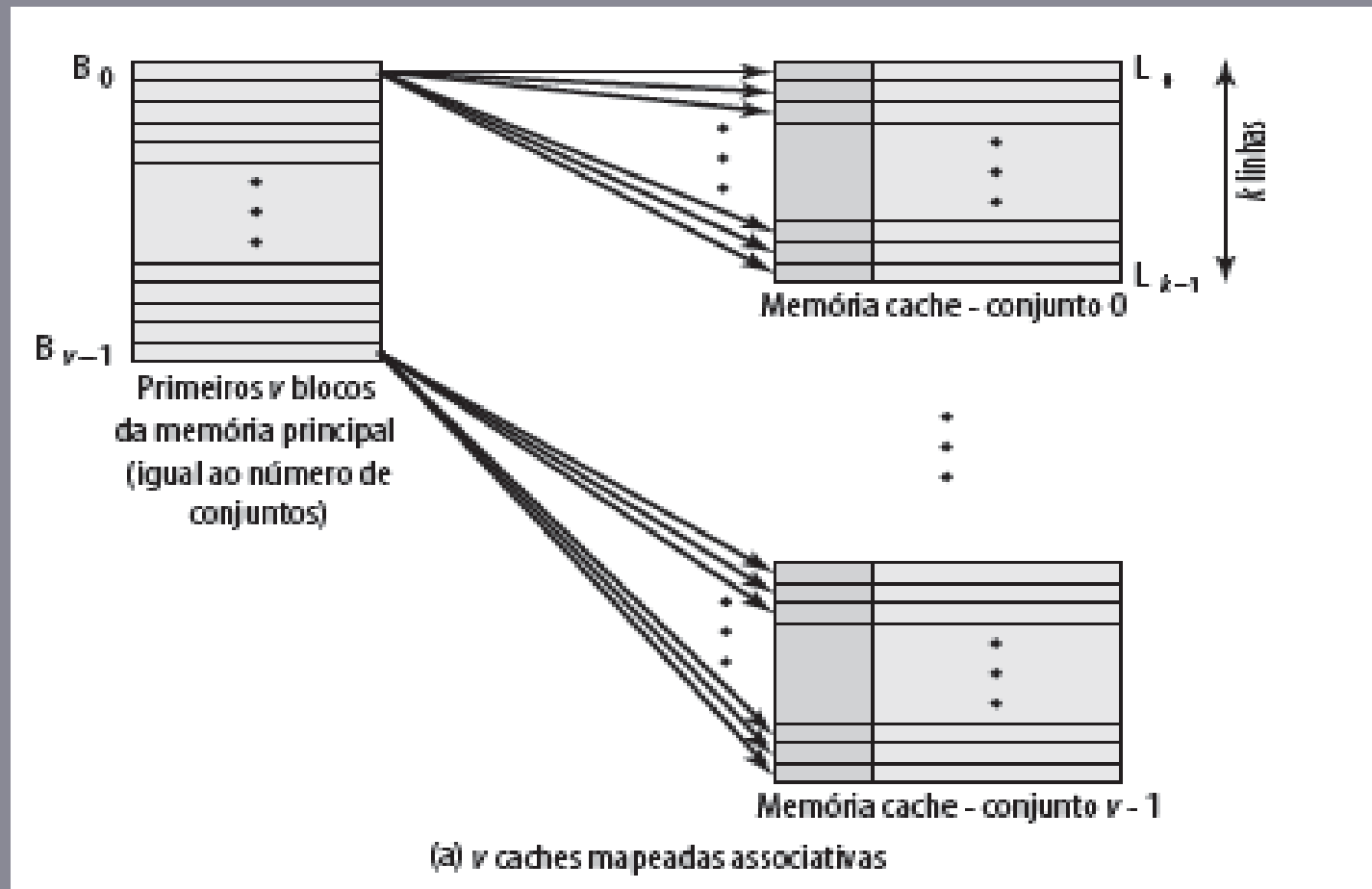


(a) Mapeamento direto

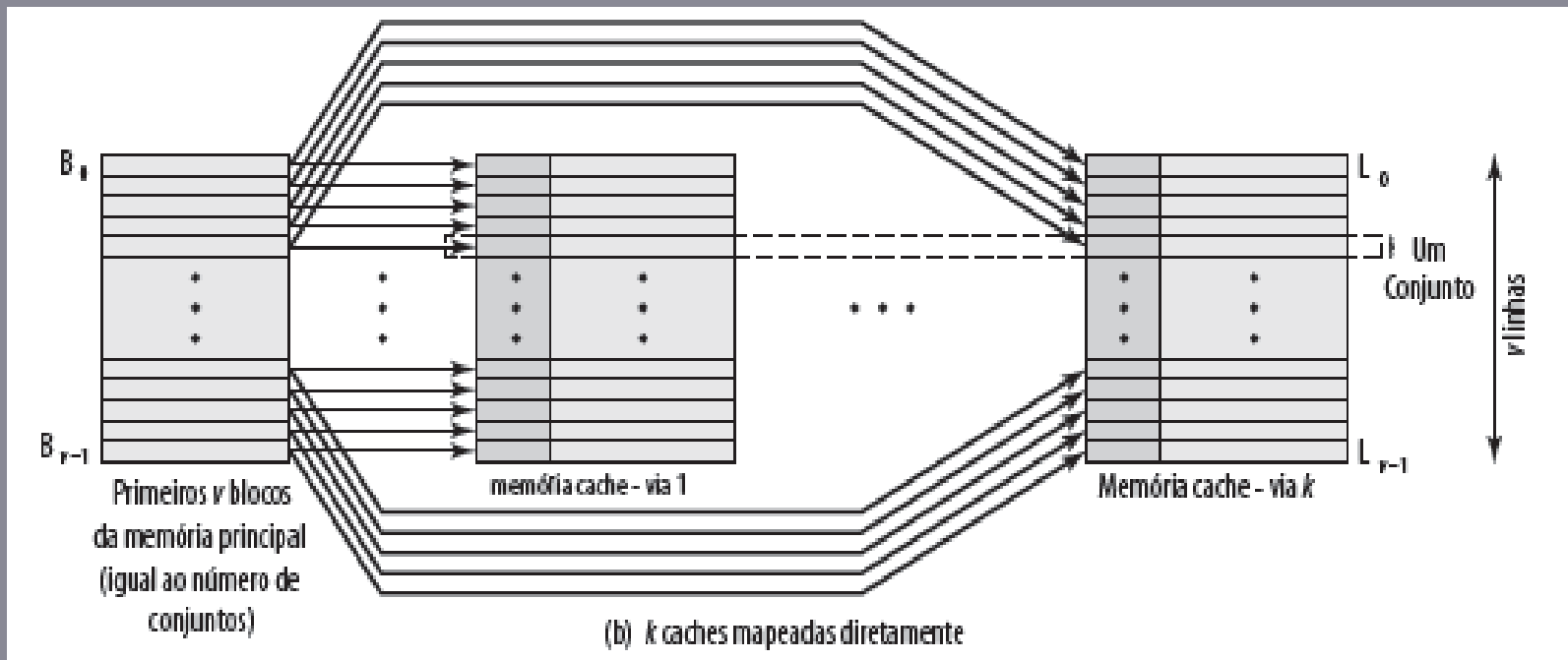
Mapeamento associativo da cache para a memória principal



Mapeamento da memória principal para cache: associativo com v linhas



Mapeamento da memória principal para cache: associativo com k linhas



Algoritmos de substituição

MAPEAMENTO DIRETO NÃO TEM ESCOLHA

Associativa e associativa em conjunto

- Algoritmo implementado no hardware (velocidade).
- Least Recently Used (LRU).
 - Substitui o bloco com maior tempo sem ser referenciado/utilizado. MAIS POPULAR ENTRE OS DEMAIS
- First In First Out (FIFO).
 - Substitui bloco que está na cache há mais tempo.
- Least Frequently Used (LFU).
 - Substitui bloco que teve menos acertos.
- Aleatório.

Caches unificadas *versus* separadas

- Uma cache para dados e instruções ou duas, uma para dados e uma para instruções.
- Vantagens da cache unificada:
 - Maior taxa de acerto.
 - Equilibra carga entre buscas de instrução e dados.
 - Apenas uma cache para projetar e implementar.
- Vantagens da cache separada:
 - Elimina disputa pela cache entre a unidade de busca/decodificação de instrução e a unidade de execução.
 - Importante no pipeline de instruções.

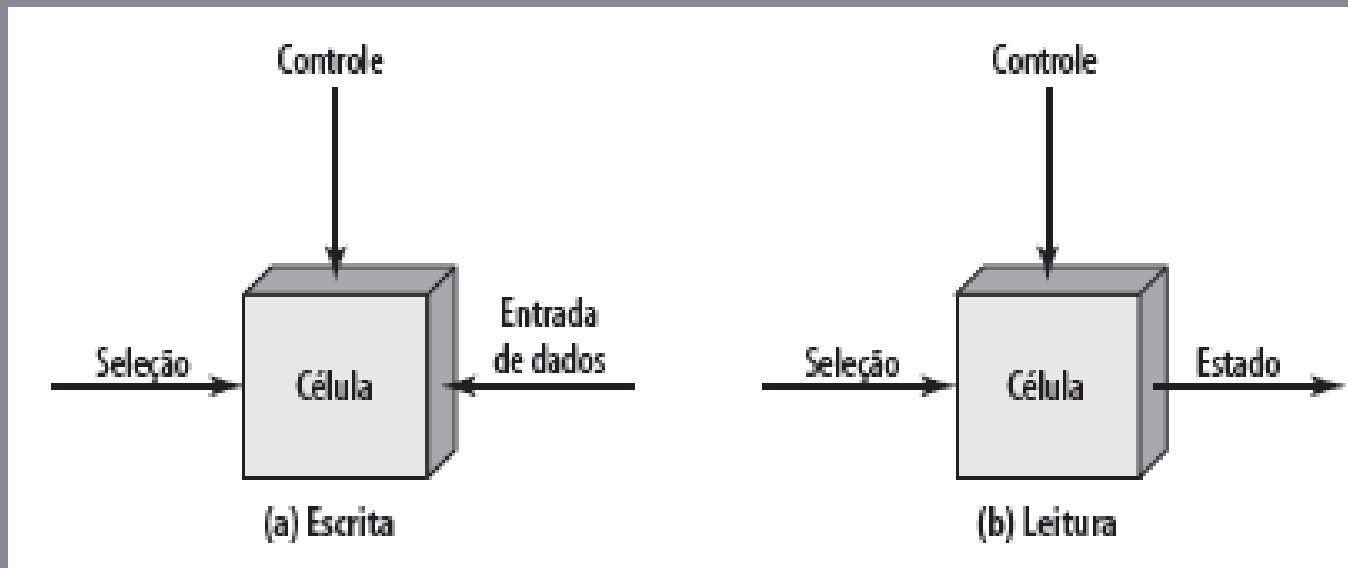
MEMÓRIA INTERNA

MEMÓRIA PRINCIPAL: RAM, ROM E OUTRAS !!!

Tipos de memória de semicondutor

Tipo de memória	Categoria	Apagamento	Mecanismo de escrita	Volatilidade
Memória de acesso aleatório (RAM)	Memória de leitura-escrita	Eletricamente, em nível de byte	Eletricamente	Volátil
Memória somente de leitura (ROM)	Memória somente de leitura	Não é possível	Máscaras	Não volátil
ROM programável (PROM, do inglês <i>programmable ROM</i>)			Eletricamente	
PROM apagável (EPROM, do inglês <i>erasable PROM</i>)	Memória principalmente de leitura	Luz UV, nível de chip		
PROM eletricamente apagável (EEPROM, do inglês <i>electrically erasable PROM</i>)		Eletricamente, nível de byte		
Memória flash		Eletricamente, nível de bloco		

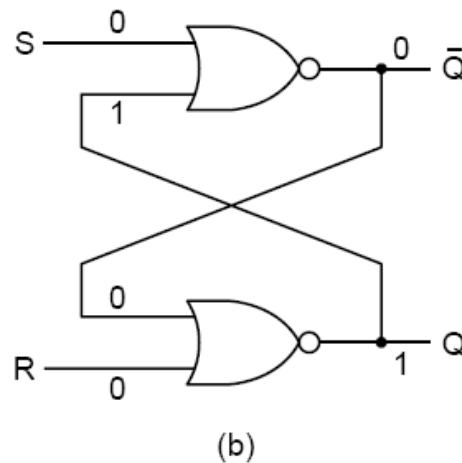
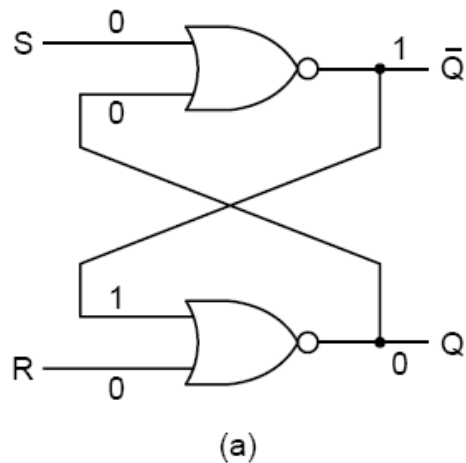
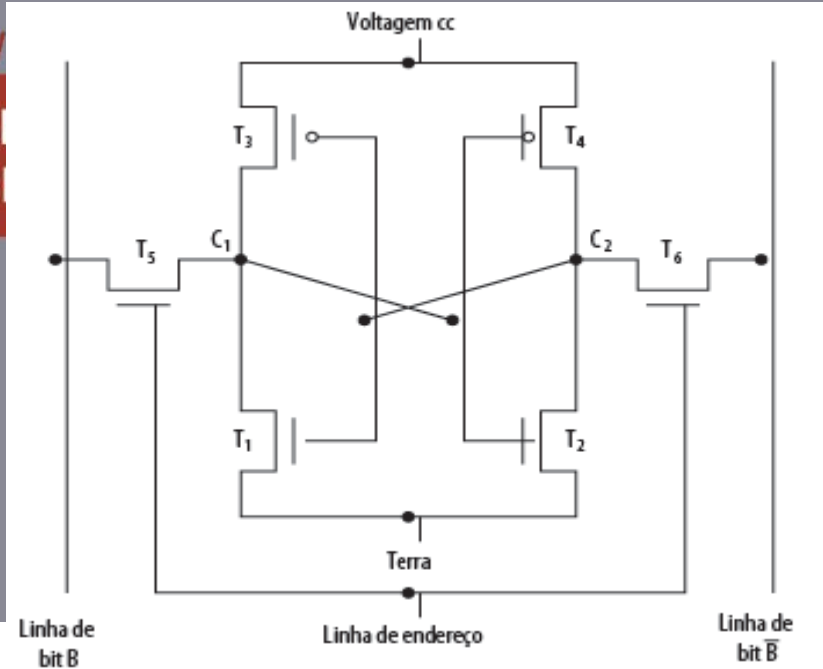
Operação da célula de memória



Estrutura da RAM dinâmica



Estrutura da RAM estática



A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

(c)

Figure 3-22. (a) NOR latch in state 0. (b) NOR latch in state 1. (c) Truth table for NOR.

SRAM *versus* DRAM

- Ambas voláteis.
 - É preciso energia para preservar os dados.
- Célula dinâmica:
 - Mais simples de construir, menor.
 - Mais densa.
 - Mais barata.
 - Precisa de *refresh*.
 - Maiores unidades de memória.
- Estática:
 - Mais rápida.
 - Cache.

Tipos de ROM

- Gravada durante a fabricação:
 - Muito cara para pequenas quantidades.
- Programável (uma vez):
 - PROM.
 - Precisa de equipamento especial para programar.
- Lida “na maioria das vezes”:
 - Erasable Programmable (EPROM).
 - Apagada por UV.
 - Electrically Erasable (EEPROM):
 - Leva muito mais tempo para escrever que para ler.
 - Memória flash:
 - Apaga memória inteira eletricamente.

DRAM síncrona (SDRAM)

- Acesso sincronizado com clock externo.
- Endereço é apresentado à RAM.
- RAM encontra dados (CPU espera na DRAM convencional).
- Como a SDRAM move dados em tempo com o clock do sistema, CPU sabe quando os dados estarão prontos.
- CPU não precisa esperar, e pode fazer alguma outra coisa.
- Modo de rajada permite que SDRAM defina fluxo de dados e o dispare em bloco.
- DDR-SDRAM envia dados duas vezes por ciclo de clock (transição de subida e descida).

DDR – SDRAM

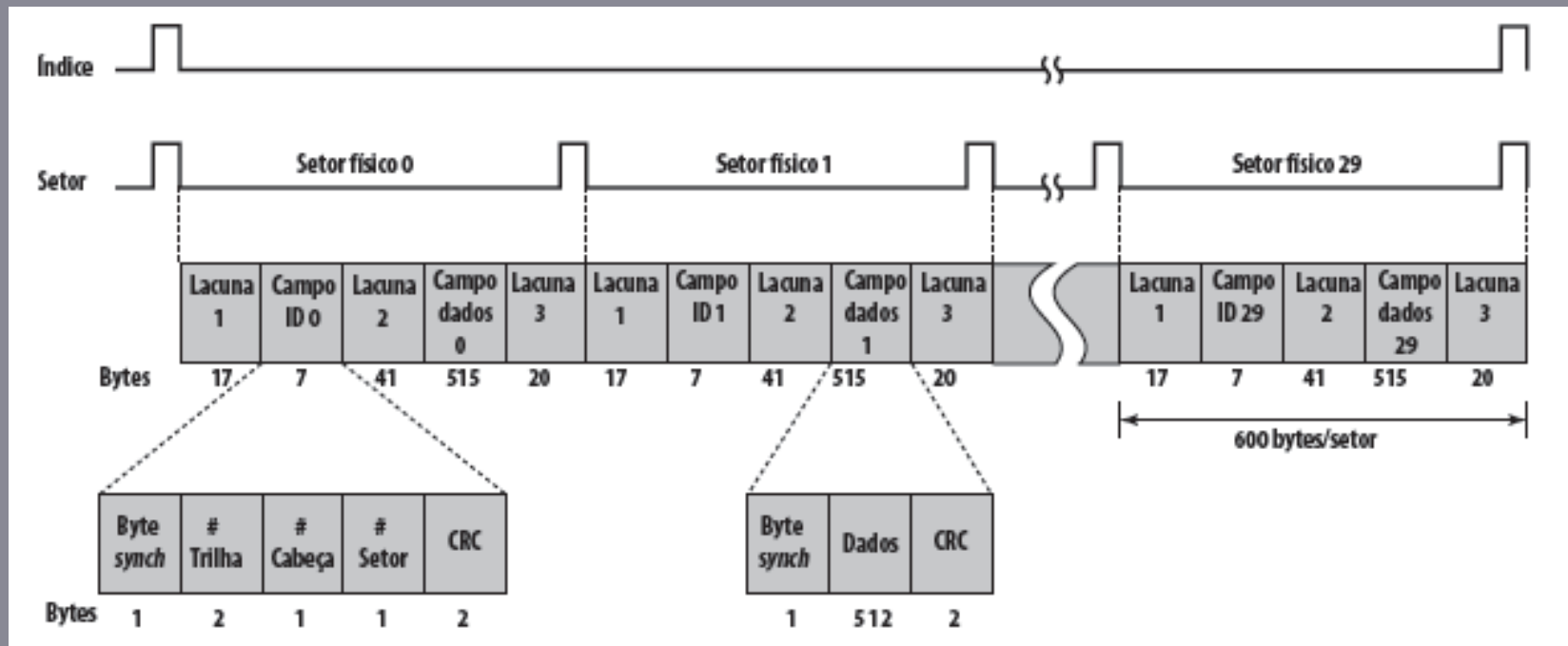
- SDRAM só pode enviar dados uma vez por ciclo de clock.
- Double-data-rate SDRAM pode enviar dados duas vezes por ciclo de clock.
 - Transição de subida e transição de descida.

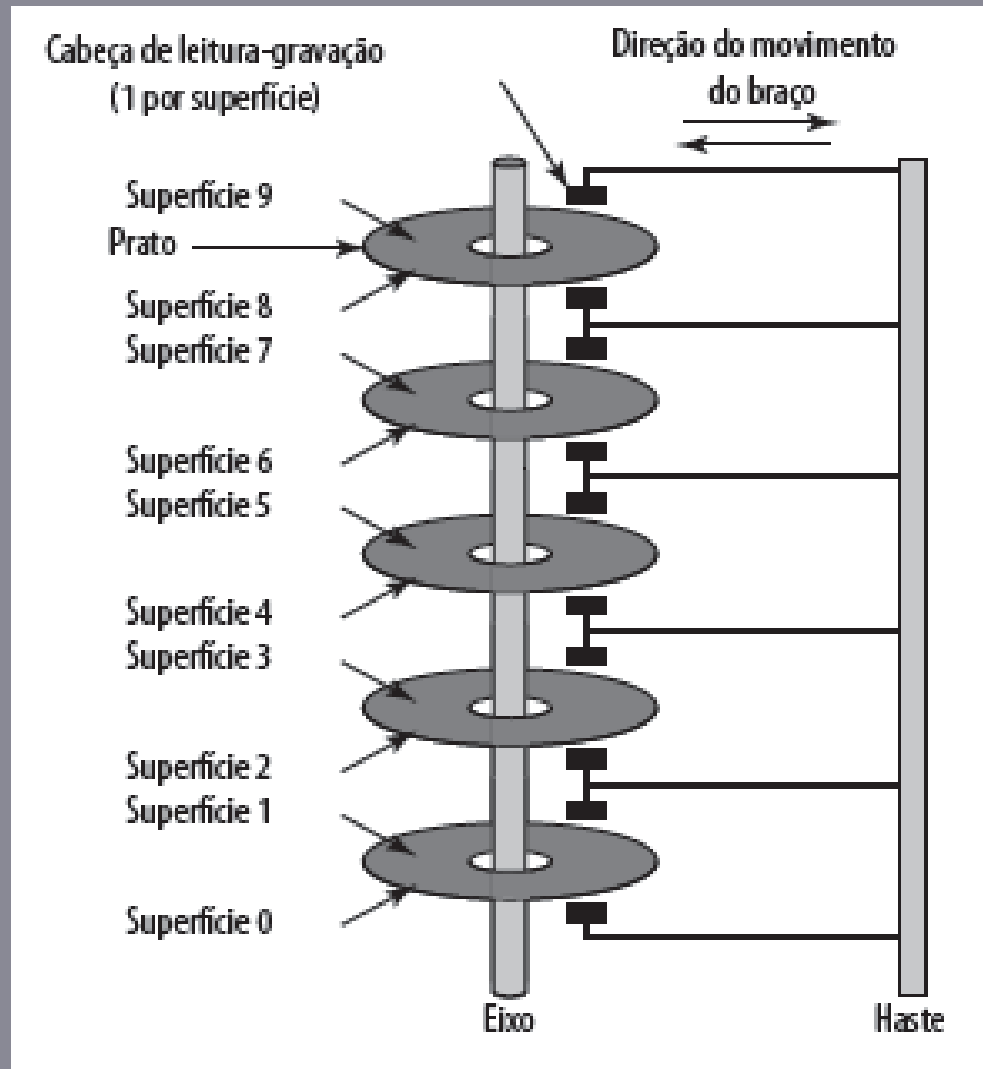
MEMÓRIA EXTERNA

DISCOS, FITAS, CD'S, etc...

Formato de disco Winchester (Seagate ST506)

Discos Magnéticos



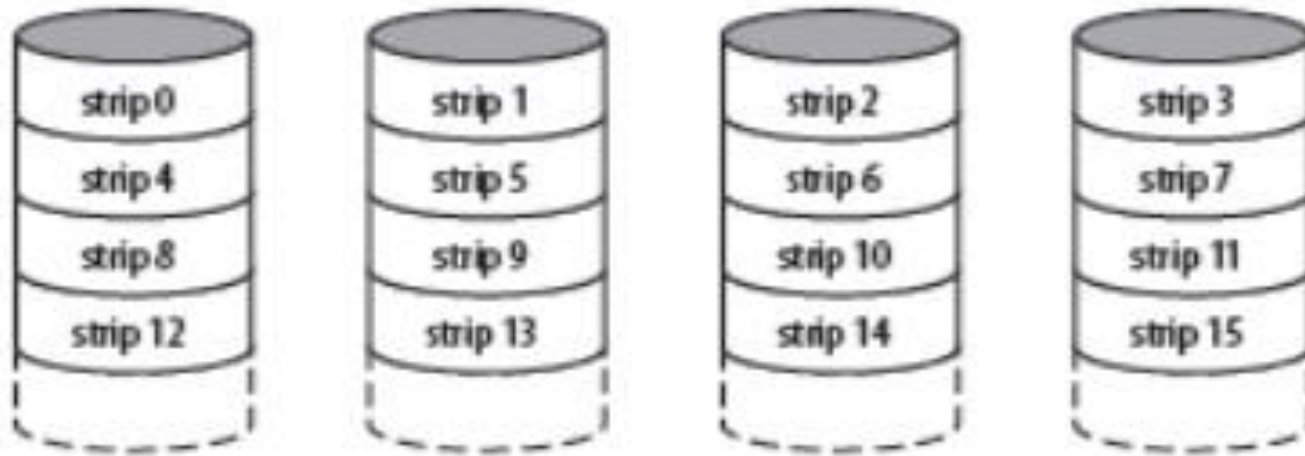
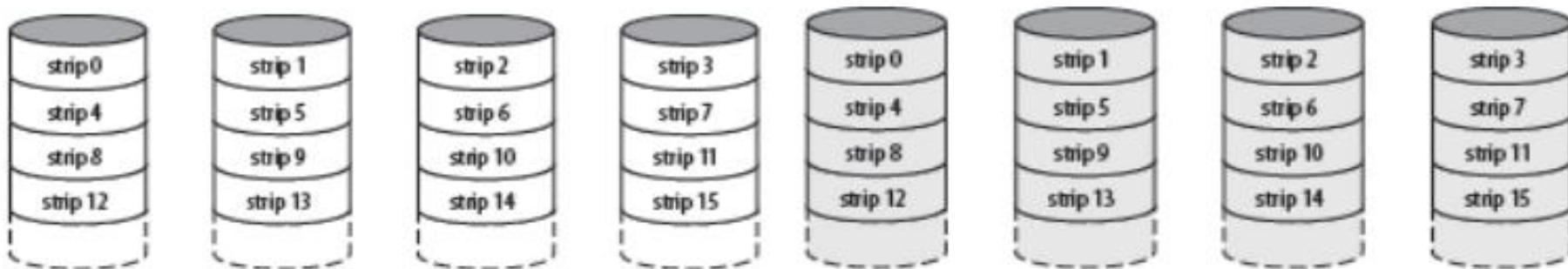
**ARQUITETURA E ORGANIZAÇÃO
DE COMPUTADORES**

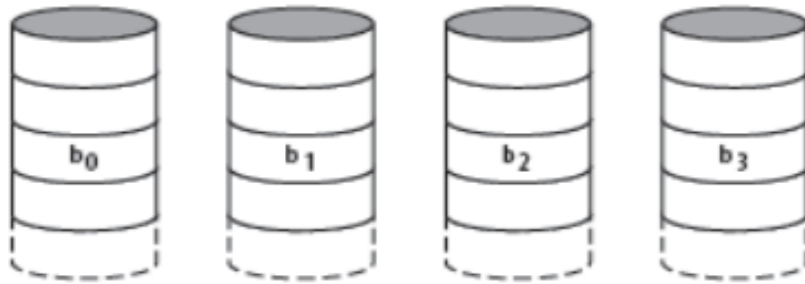
Velocidade

- Tempo de busca:
 - Movendo cabeça para trilha correta.
- Latência (rotacional):
 - Esperando dados passarem sob a cabeça.
- Tempo de acesso = Busca + Latência.
- Taxa de transferência.

RAID

- Redundant Array of Independent Disks.
- Redundant Array of Inexpensive Disks.
- 6 níveis de uso comum.
- Não é uma hierarquia.
- Conjunto dos principais discos vistos como uma única unidade lógica pelo SO.
- Dados distribuídos pelas unidades físicas.
- Pode usar capacidade redundante.
- Pode usar capacidade redundante para armazenar informação de paridade.

**ARQUITETURA E ORGANIZAÇÃO
DE COMPUTADORES****(a) RAID 0 (não redundante)****(b) RAID 1 (espelhado)**

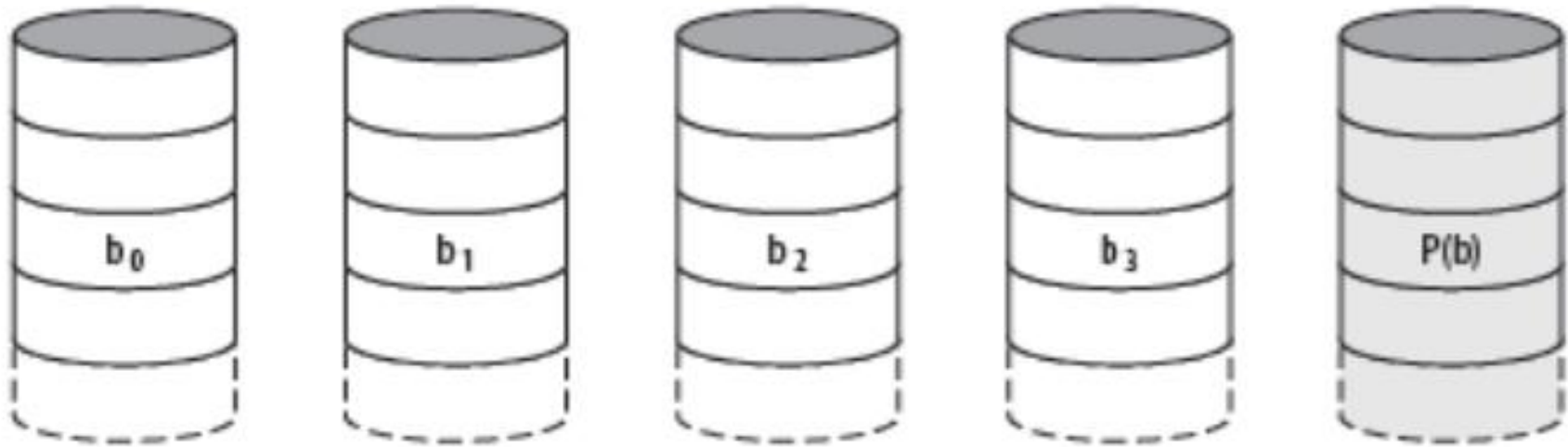
**ARQUITETURA E ORGANIZAÇÃO
DE COMPUTADORES**

(c) RAID 2 (redundância por código de Hamming)



Não utilizado

Muito caro



(d) RAID 3 (paridade de bit intercalada)

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES



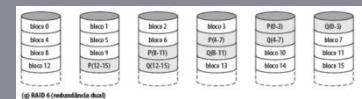
(e) RAID 4 (paridade em nível de bloco)



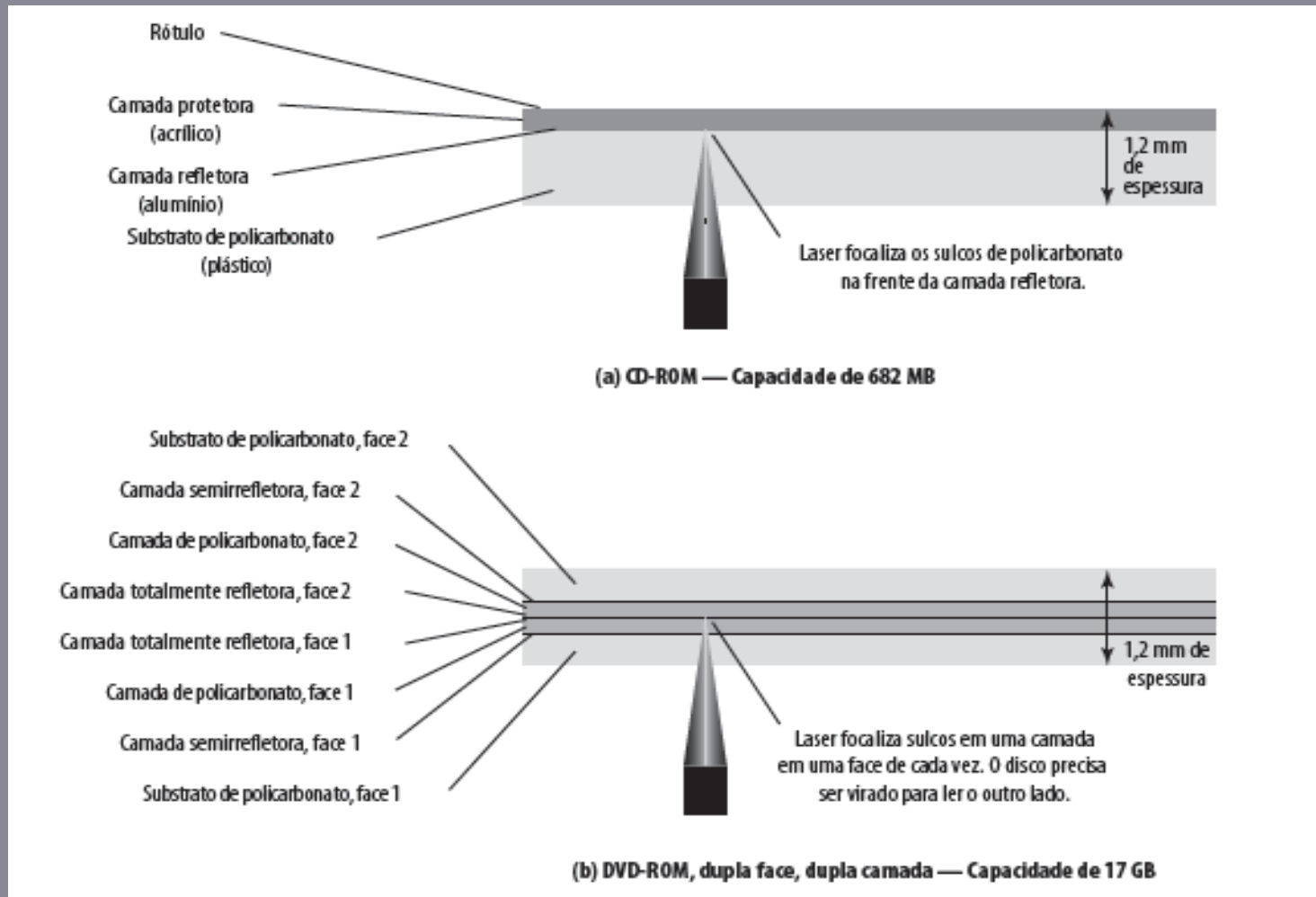
(f) RAID 5 (paridade em nível de bloco distribuída)

Raid 6.

Similar ao 5 !!!



CD e DVD



Fita magnética

- Acesso serial.
- Lenta.
- Muito barata.
- Backup e arquivamento.
- Unidades de fita Linear Tape Open (LTO).
 - Desenvolvida no final da década de 1990.
 - Alternativa de fonte aberto para os diversos sistemas de fita patenteados.

ENTRADA E SAÍDA

Na outra retrospectiva....