



Organização de Computadores

(Aula 3)

Componentes de um Computador
Unidade Central de Processamento (CPU)

Componentes de um Computador ⁽¹⁾

- Computador Eletrônico Digital
 - É um sistema composto por:
 - Memória – Principal e Secundária
 - Unidade Central de Processamento (*CPU – Central Processing Unit*) ou Processador
 - Unidades de Entrada/Saída (*I/O – Input/Output Unit*)
 - Barramentos (vias de interligação)

Componentes de um Computador (2)

■ Memória

- Função: Armazenar dados e programas (conjunto de instruções), permitindo o processamento automático dos dados.
- Para serem executados pela CPU, os programas devem estar armazenados na Memória (como veremos, Memória principal).
- Unidade básica é o Bit
- Diferentes maneiras de se organizar a memória
 - Hoje em dia o padrão é o byte
 - Cada byte representa um endereço de memória

Componentes de um Computador ⁽³⁾

Tipos de Memória

■ Memória Principal (MP)

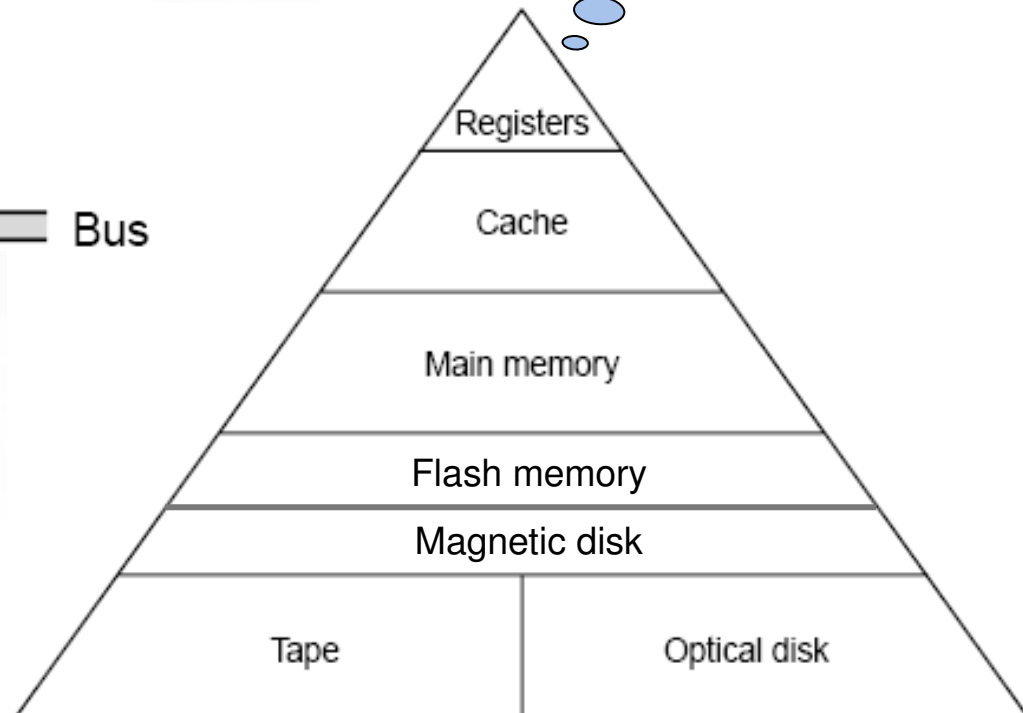
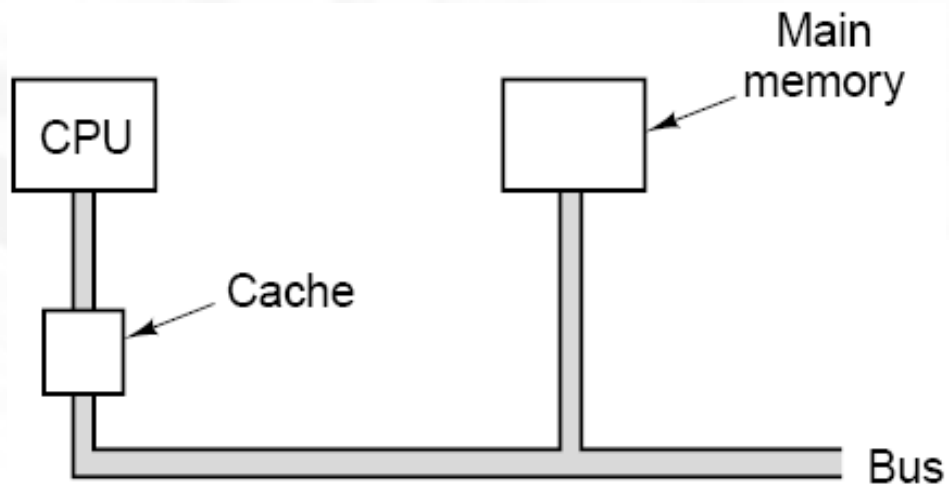
- Limitada (armazena somente alguns programas e dados).
- É a memória de trabalho, nela são carregados todos os programas e dados usados pela CPU no momento.
- Geralmente o seu conteúdo será perdido uma vez que o computador seja desligado.
- Ex.: RAM (*Random Access Memory*)

■ Memória Secundária (MS)

- Grande capacidade de armazenamento
- Mais lentas do que a MP
- Geralmente não-voláteis, permitindo guardar os dados permanentemente.
- Ex.: Discos rígidos (HD - *Hard Disk*), CDs, DVDs, disquetes, Pen drives, etc.

■ E a Memória Cache?

Componentes de um Computador (4)



Componentes de um Computador ⁽⁶⁾

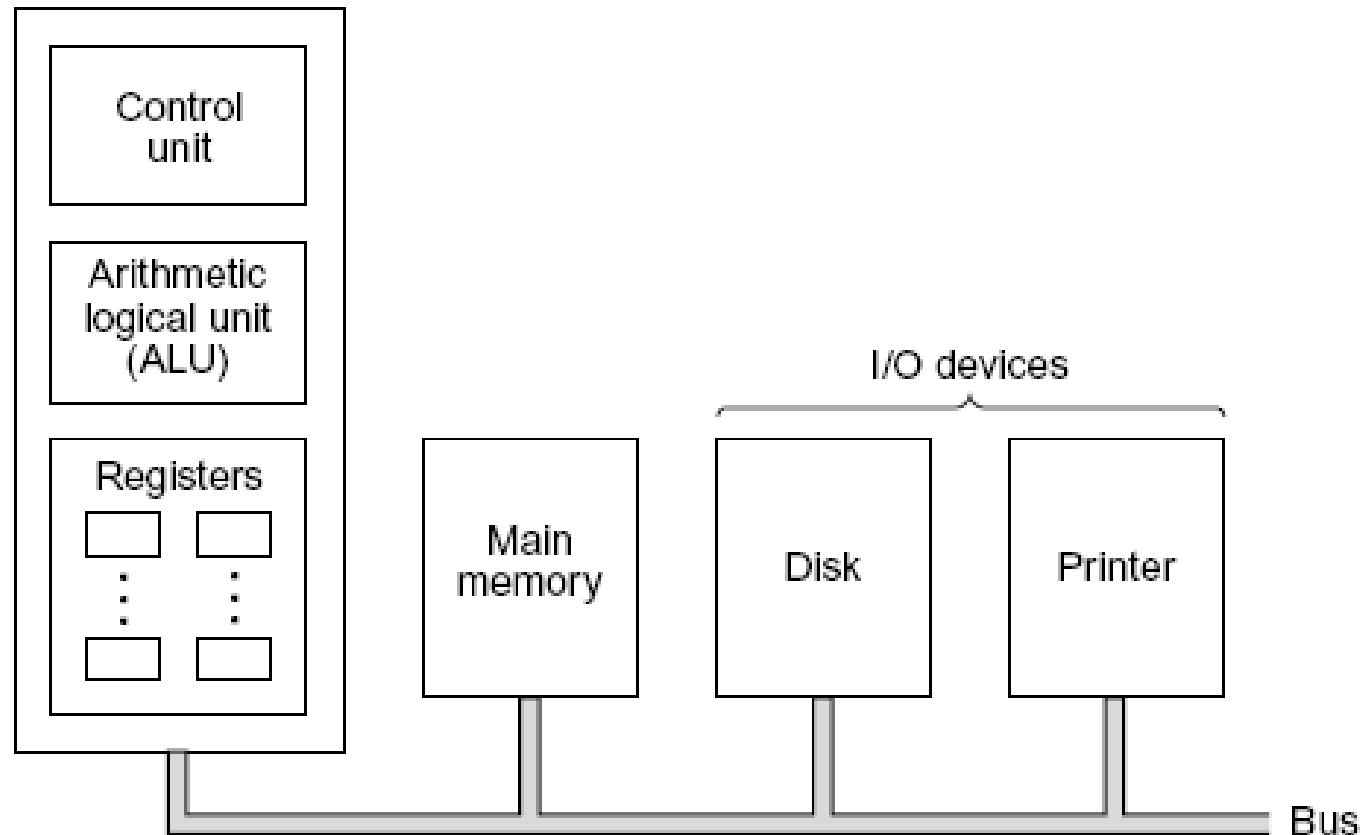
- **Unidade Central de Processamento – Processador**
 - Função: Executar os programas armazenados na MP, buscando cada instrução, interpretando-a e em seguida a executando.
 - **Unidade de Controle (UC):** busca instruções na MP e determina o tipo de cada instrução.
 - **Unidade Lógica e Aritmética (ULA):** realiza um conjunto de operações necessárias à execução das instruções.
 - **Conjunto (Banco) de Registradores:** pequena memória de alta velocidade, usada para armazenar resultados temporários e certas informações de controle.

Componentes de um Computador (7)

- Dispositivos de Entrada/Saída
 - Função: Servem para a comunicação do computador com o meio externo.
 - Muitos são usados para a interação com o usuário
 - Transformam e transferem da representação interna dos dados (sinais eletrônicos usados no computador) para a representação externa usada por pessoas ou vice-versa.
 - Ex.: impressoras, teclados, mouses, monitores, caixas de som, **discos**, etc.

Componentes de um Computador (8)

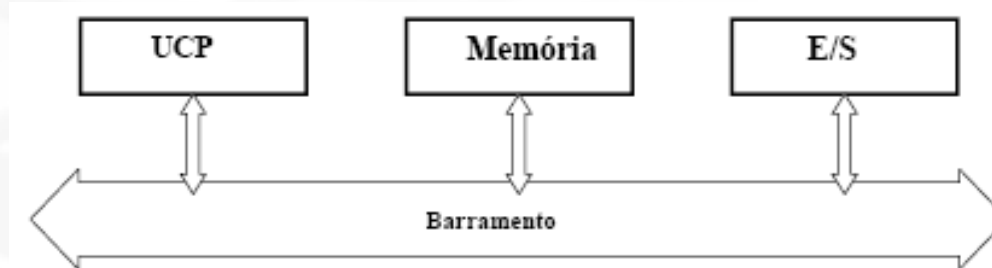
Central processing unit (CPU)



Componentes de um Computador ⁽⁹⁾

■ Barramentos (*Bus*)

- Função: interconectar os demais elementos básicos.



- Nesse modelo, é possível transferência direta de dados entre os dispositivos de E/S e a memória
- Transferência feita por dispositivos especiais
 - Controlador de **DMA (Direct Memory Access)** ou processador E/S
 - Assumem momentaneamente o controle do barramento e "isolam" a comunicação CPU com a memória e dispositivos E/S.

Componentes de um Computador (10)

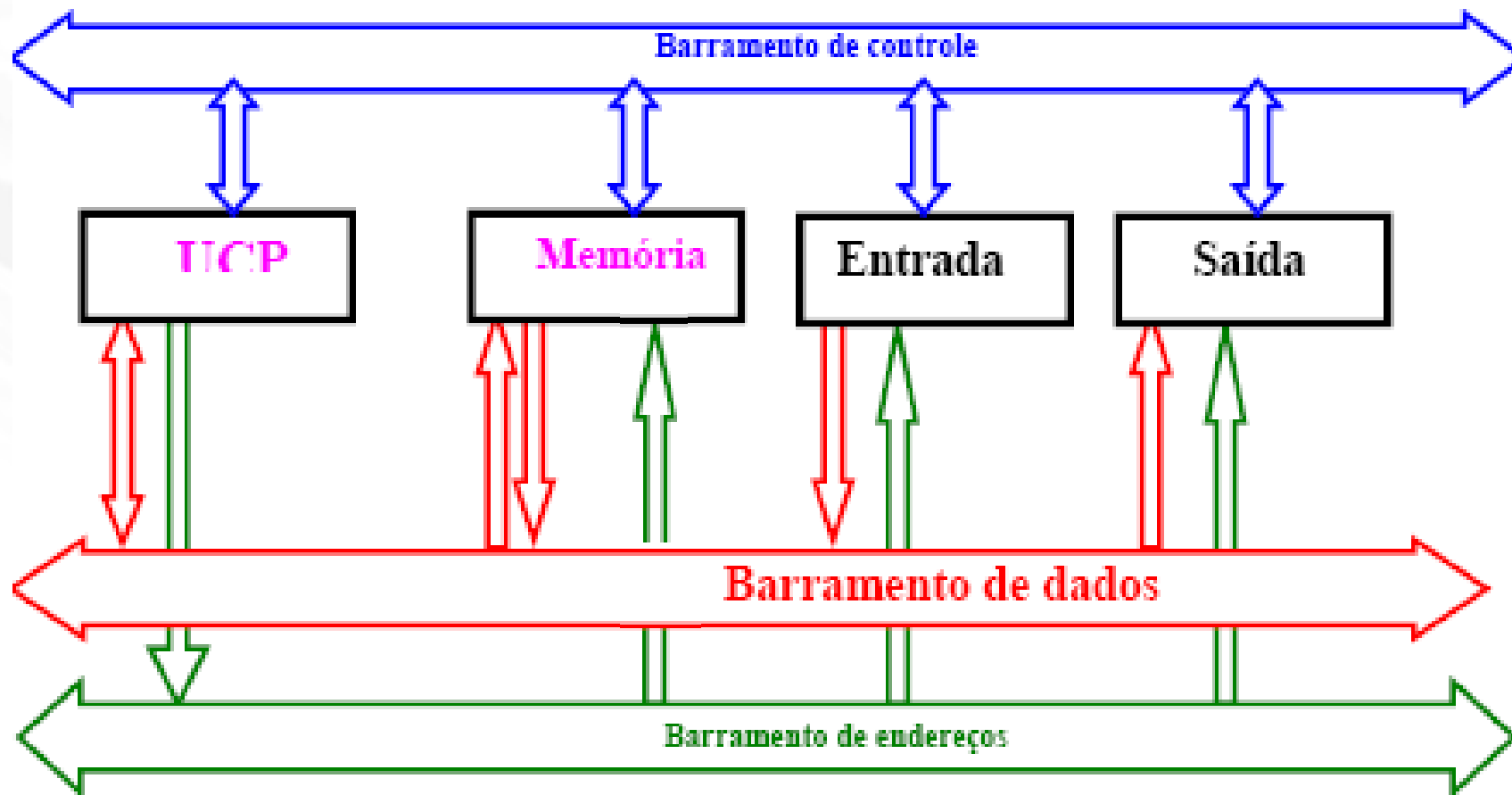
■ Barramentos

- Unidirecionais: conduzir os sinais em uma única direção
- Bidirecionais: conduzir ora em uma direção e ora em outra de acordo com comandos eletrônicos
- Podem ainda ser:
 - Três estados: bidirecionais acrescidos da possibilidade de se desconectarem eletricamente
 - Unidirecional com a possibilidade de desconexão elétrica
 - A possibilidade de desconexão elétrica controlada eletronicamente torna viável a comunicação seletiva entre os dispositivos.
 - Por exemplo, a CPU se comunica com a memória e se desconecta dos dispositivos E/S.

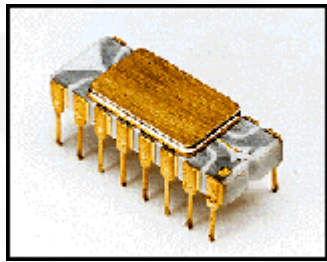
Componentes de um Computador (11)

- Tipos de barramentos:
 - **Barramento de Dados:** responsável pela transferência de dados e instrução entre os dispositivos.
 - É de três estados entre a memória e a CPU e unidirecional com a possibilidade de desconexão elétrica nos dispositivos E/S.
 - **Barramento de Endereços:** conduz o endereço a ser selecionado na memória ou dispositivos E/S.
 - É geralmente unidirecional entre a CPU (que define o endereço) e a memória e os dispositivos E/S.
 - **Barramento de Controle:** transfere os sinais de controle que ativam ou desativam os dispositivos, que selecionam determinado modo de operação ou sincronizam os circuitos.
 - Formam a **Arquitetura de Barramentos**

Componentes de um Computador (12)



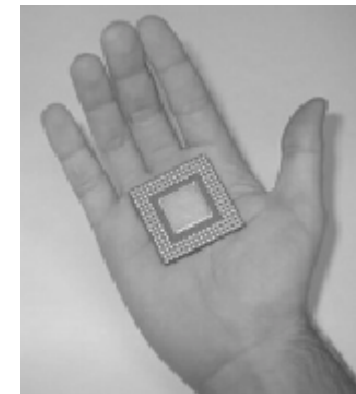
Unidade Central de Processamento - CPU (1)



Intel 4004



Pentium II – Slot 1



→ o processador ainda é o componente básico de qualquer computador

Unidade Central de Processamento - CPU (2)

- Conceitos básicos

- Instruções: comando que define integralmente uma operação a ser executada
 - Programas: instruções ordenadas logicamente
- “... Para que um programa possa ser executável ele precisa ser construído de uma série de instruções de máquina...”

- O **processador** é o “cérebro” do computador

- Função:

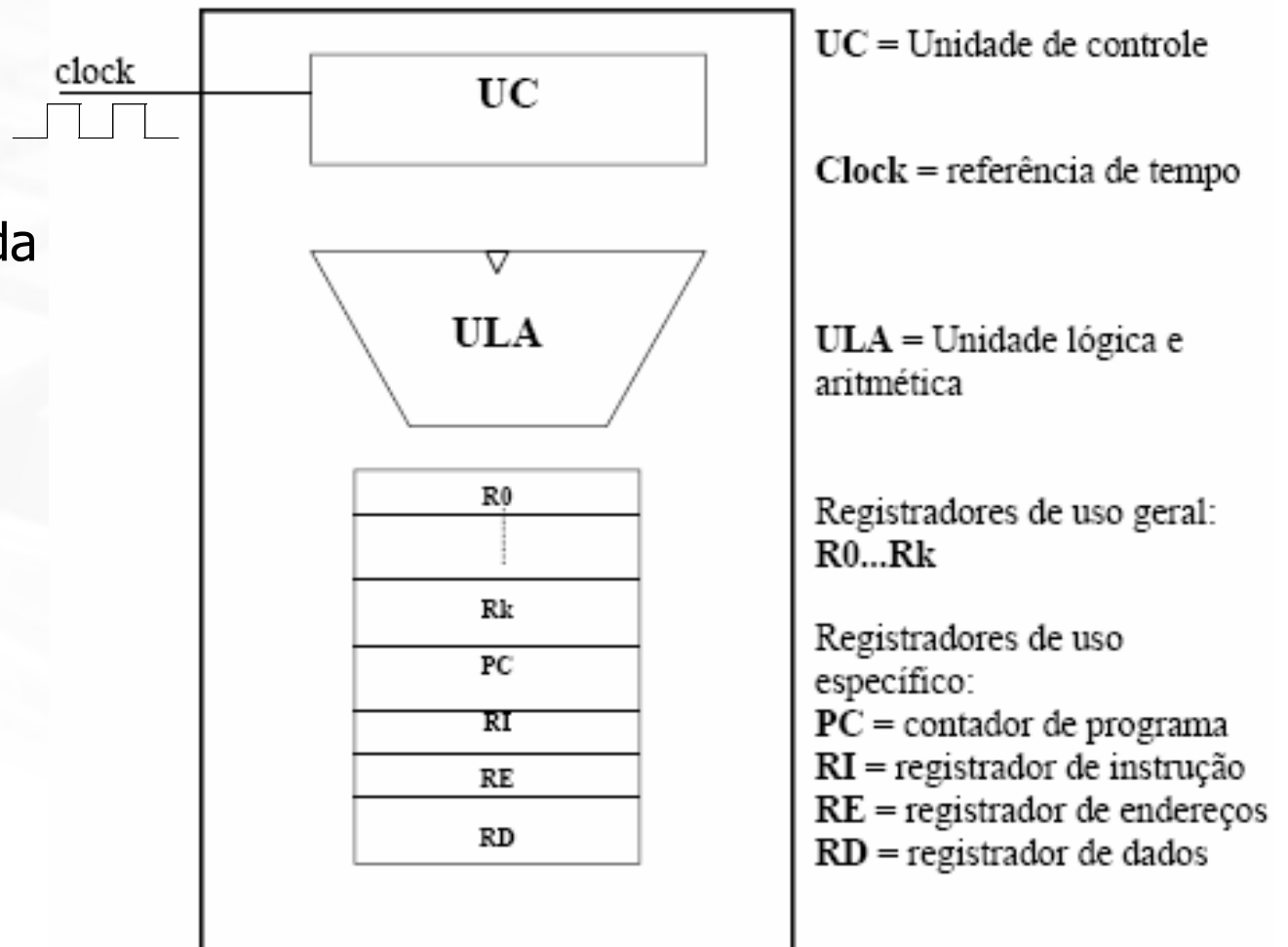
- Executar os **programas** armazenados na memória principal, buscando cada uma das **instruções** do programa, examinando-as, e executando-as uma após a outra.

Unidade Central de Processamento - CPU ⁽³⁾

- Gera sinais de controle para os demais componentes do computador, além de reconhecer e responder sinais externos (**interrupções**)
- A CPU é responsável pela **supervisão e controle** de todo o **funcionamento** da máquina e pela realização das operações lógicas e aritméticas sobre os dados
- **Clock** (ou Relógio) é a referência de tempo necessária a CPU.
 - Circuito eletrônico oscilador que gera uma forma de onda quadrada, essencial para o seqüenciamento das operações eletrônicas realizadas pela CPU.
 - Está relacionado com a frequência (taxa) de operação do processador.

Unidade Central de Processamento - CPU (4)

- A UCP pode ser dividida em :
 - Unidade de controle (UC)
 - Unidade lógica e aritmética (ULA)
 - Registradores



Unidade Central de Processamento - CPU (5)

■ **Unidade de Controle (UC)**

- Função: dirigir e coordenar as atividades das demais unidades do sistema
- É a parte mais complexa da UCP (o centro nervoso)
- Controla todos os componentes e dispositivos de um computador, emitindo sinais elétricos (pulsos de controle)
 - Sincronizados através do clock
- Responsável pela busca das instruções na memória principal, sua decodificação e execução.
 - Controla a ação da ULA e a movimentação de dados entre a UCP e a memória
 - Solicita a busca da instrução que será executada, armazena-a em um **registrador**
 - Dispara ações necessárias para execução da instrução

Unidade Central de Processamento - CPU ⁽⁶⁾

■ **Unidade de Controle (UC)** (cont.)

■ **Contador de Instrução (*Program Counter – PC*):**

- Armazena o **endereço** da (aponta para a) próxima instrução a ser executada.
- Determina a seqüência (fluxo) de execução de um programa, endereçando instrução por instrução

■ **Registrador de Instruções (Instruction Register – *RI*)**

- Armazena (contém) a instrução que está sendo executada.

Unidade Central de Processamento - CPU (7)

■ **Unidade de Lógica e Aritmética (ULA)**

- Possui os circuitos necessários para executar operações lógicas e aritméticas.
 - Exemplos: Somas, subtrações, determinação de sinal, comparações algébricas de números, verificar se um número é maior ou menor que zero, etc
- ... os operandos utilizados pela ULA são armazenados em registradores internos da UCP
- ... o resultado das operações da ULA também é armazenado em um registrador
 - **Acumulador (ACC)**

Unidade Central de Processamento - CPU ⁽⁸⁾

Instrução

- Comece a ler entradas a partir da unidade de fitas 2
- Se X é maior que Y, vá para o passo 46 do programa
- Multiplique A por B e armazene o resultado em C

Local de execução

- Unidade de controle
- Unidade lógica e aritmética (instrução lógica)
- Unidade lógica e aritmética (instrução aritmética)

Unidade Central de Processamento - CPU ⁽⁹⁾

- Conjunto (Banco) de Registradores
 - Memória interna de pequena capacidade de armazenamento, mas de alta velocidade, usada para armazenar resultados temporários e certas informações de controle
 - Local onde os dados ficam armazenados enquanto são processados ou enquanto sujeitos a operações lógicas.
 - Formada por:
 - **Registradores de uso geral:** acessíveis ao usuário no nível convencional de máquina
 - **Registradores de uso específico:** acessíveis diretamente apenas pela UC durante a execução de um programa.
 - O número de registradores e seus tamanhos variam entre arquiteturas de computadores

Unidade Central de Processamento - CPU (10)

- Conjunto (Banco) de Registradores (cont.)
 - Contador de programa (PC)
 - Aponta para as instruções a serem processadas
 - Incrementado ao fim do processamento da instrução
 - Registrador de instrução (RI)
 - Armazena o código binário da instrução propriamente dita
... modificado toda vez que uma nova instrução deve ser executada
 - Registrador de endereço de memória (REM)
 - Mantém o endereço da posição de memória usada como operando
... modificado quando é necessário apontar para um operando em outra posição de memória

Unidade Central de Processamento - CPU ⁽¹¹⁾

■ Instruções

- São comandos atômicos fornecidos ao computador para que ele execute uma tarefa específica (soma, teste, desvio, etc)
- Uma série dessas instruções constituem um programa.
- Cada computador (leia-se processador) tem um repertório particular de instruções, cuja rapidez de execução é um fator determinante na “potência de computação” (performance) da máquina
- Conforme proposto por Von Neumann, as instruções, tal qual os dados, **devem residir na MP**, e dispostas **sequencialmente**.
- A seqüência de execução pode ser alterada por **Instruções de Desvio**

Unidade Central de Processamento - CPU ⁽¹²⁾

■ Formato das Instruções

Código da Instrução (<i>OpCode</i>)	Operando - <i>Op</i> (Informação Complementar)
--	---

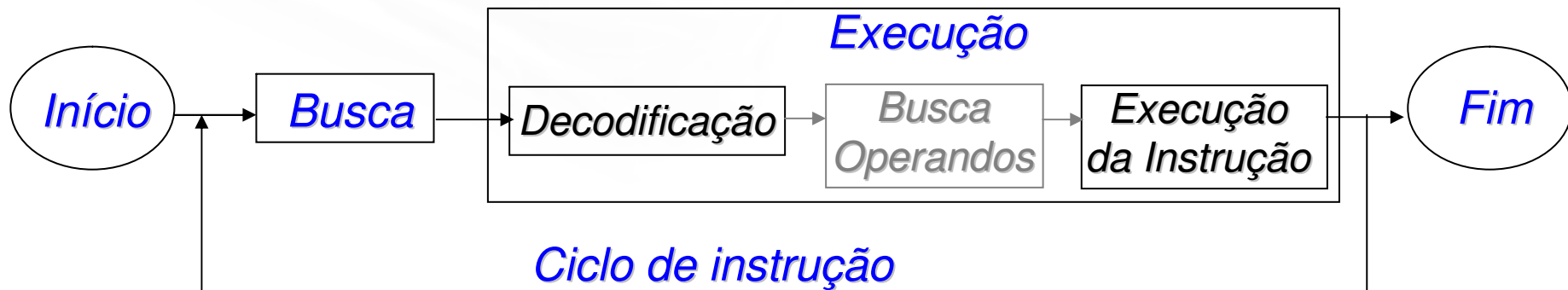
- *OpCode*: identifica a instrução a ser efetuada
- *Op*: identifica sobre o que (quem) a instrução manipulará
- Existem instruções de 1, 2 ou 3 operandos
- O tamanho do *OpCode* indica o número máximo de instruções da máquina.
 - $n \text{ bits} \Rightarrow \text{até } 2^n \text{ instruções}$
- **Ciclo de Instrução**: conjunto de operações realizadas pelo computador para completar a execução de uma instrução

Unidade Central de Processamento - CPU (13)

- **Palavras** são as unidades de dados movidas entre a memória e os registradores.
- **Instruções Registrador-Memória**
 - Permitem que o conteúdo de palavras de memória seja armazenado nos registradores (**load**) ou conteúdos de registradores armazenados na memória (**store**)
 - Referências a uma palavra deve ser feita através de **Endereços**.
- **Instruções Registrador-Registrador**
 - Típico:
 - busca dois operandos em registradores
 - coloca-os nas entradas da ULA
 - realiza alguma operação sobre eles
 - ... e armazena o resultado em um dos registradores
 - Determina um caminho de dados

Unidade Central de Processamento - CPU (14)

- Ciclo de Instrução
 - Um programa em execução é uma série de instruções de máquina que estão armazenadas na memória principal e serão interpretadas e executadas pela UCP
 - Conjunto de ações (operações) realizadas pelo computador (leia-se CPU) para completar a execução de uma instrução
- Diagrama do Ciclo de Instrução
 - Ciclo de Busca (Fetch Cycle) e Ciclo de Execução (Execution Cycle).



Unidade Central de Processamento - CPU (15)

■ **Ciclo de Busca**

1. A UC lê (busca) a instrução na memória e carrega (armazena) a instrução em RI para ser decodificada e executada.
2. Atualização do valor de PC (incremento), fazendo-o apontar para a instrução seguinte

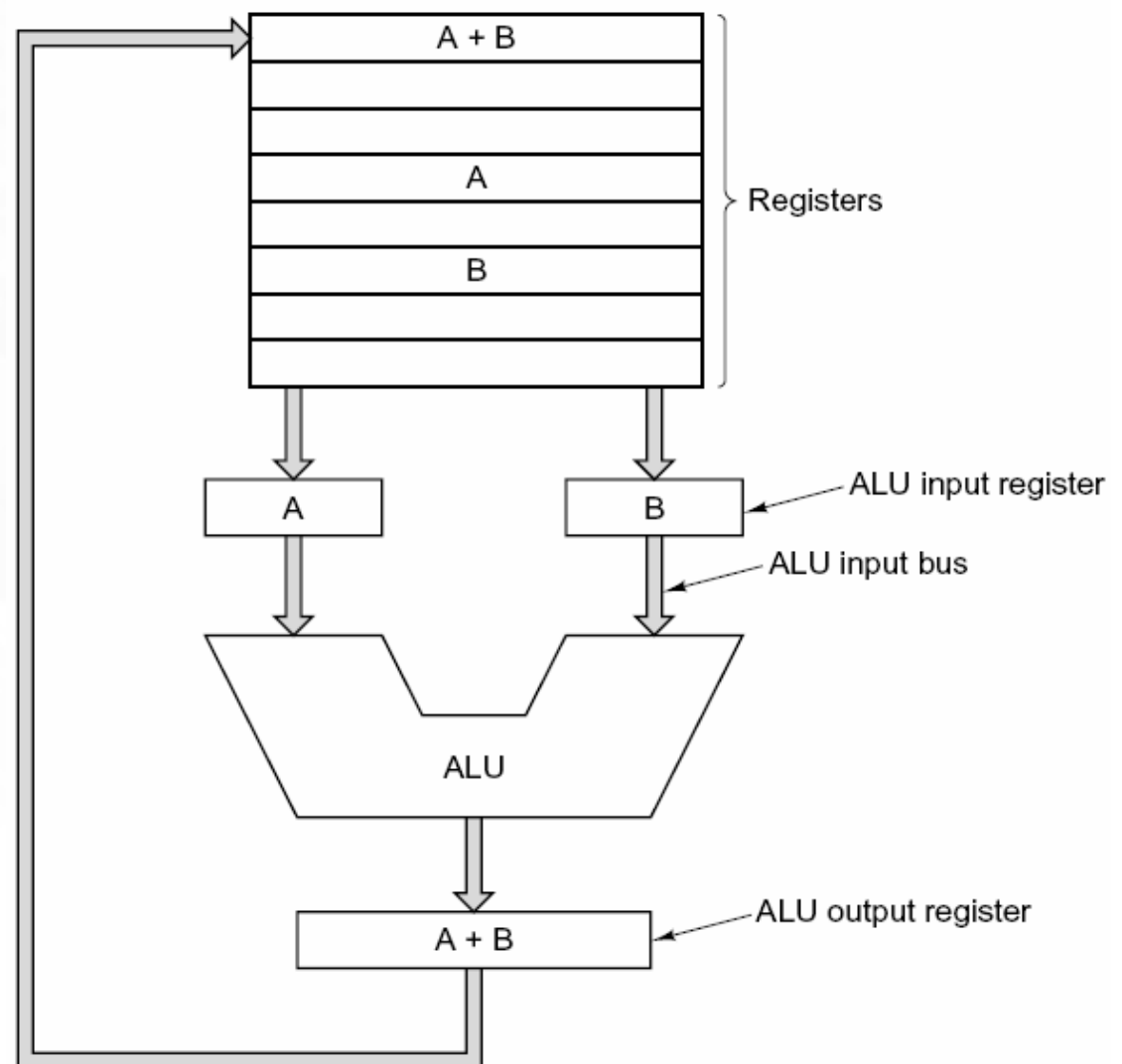
■ **Ciclo de Execução**

3. Determinação do **tipo de instrução** que está armazenada em RI (decodificação)
4. Determinação de **onde (endereço)** uma palavra de dado **ESTÁ** armazenada, caso necessário
5. **Busca da palavra**, se necessário, e armazenamento em um dos registradores do processador
6. **Execução** da instrução
7. **Retorno ao passo 1** para iniciar a execução da instrução seguinte

Unidade Central de Processamento - CPU (14)

■ Caminho de Dados

- Registradores
 - ULA
 - Controlado pela UC
-
- A velocidade do ciclo do caminho de dados (Ciclo de Instrução) determina, em última análise, a velocidade do processador.



Unidade Central de Processamento - CPU (13)

- Programa Interpretador
 - Equivalência em software do Ciclo de Instrução realizado em hardware
 - Emular um processador: “interpretando instruções através de software”
 - Exemplo em C

Questão

- De acordo com o que foi discutido no texto, na sua opinião, quais os fatores que afetam a velocidade de processamento de um computador?
 - Tamanho da memória RAM e cache
 - Largura do barramento
 - Arquitetura interna do processador
 - Tamanho e a quantidade de registradores
 - Busca de paralelismo no nível das instruções!

Paralelismo no Nível das Instruções ⁽¹⁾

- Existem limites tecnológicos para desenvolvimento / desempenho / performance do hardware do chip de processamento
- Solução para aumentar a velocidade do processador:

Uso de Paralelismo

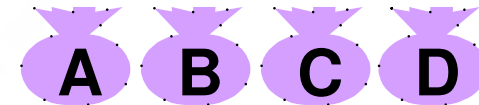
- No **Nível das Instruções**: um único processador deve executar mais instruções por segundo
- No **Nível do Processador**: vários processadores trabalhando juntos na solução do mesmo problema

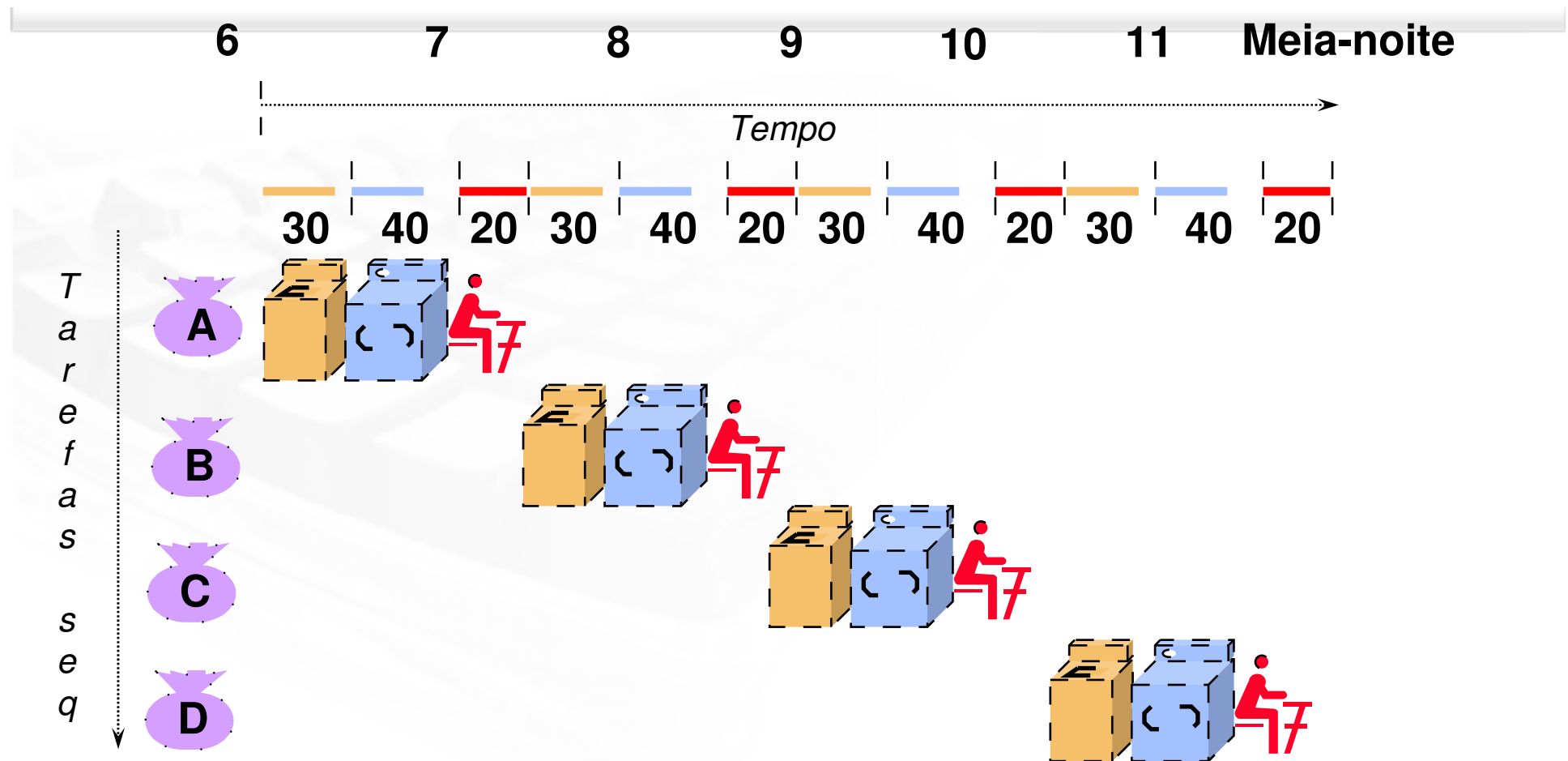
Paralelismo no Nível das Instruções (2)

- Maior gargalo para a velocidade de execução de instruções é o acesso à memória
- Execução em **Pipeline**
 - O processamento em pipeline divide a execução de instruções em várias “partes”
 - Cada “parte” é tratada por um hardware dedicado exclusivamente a ela.

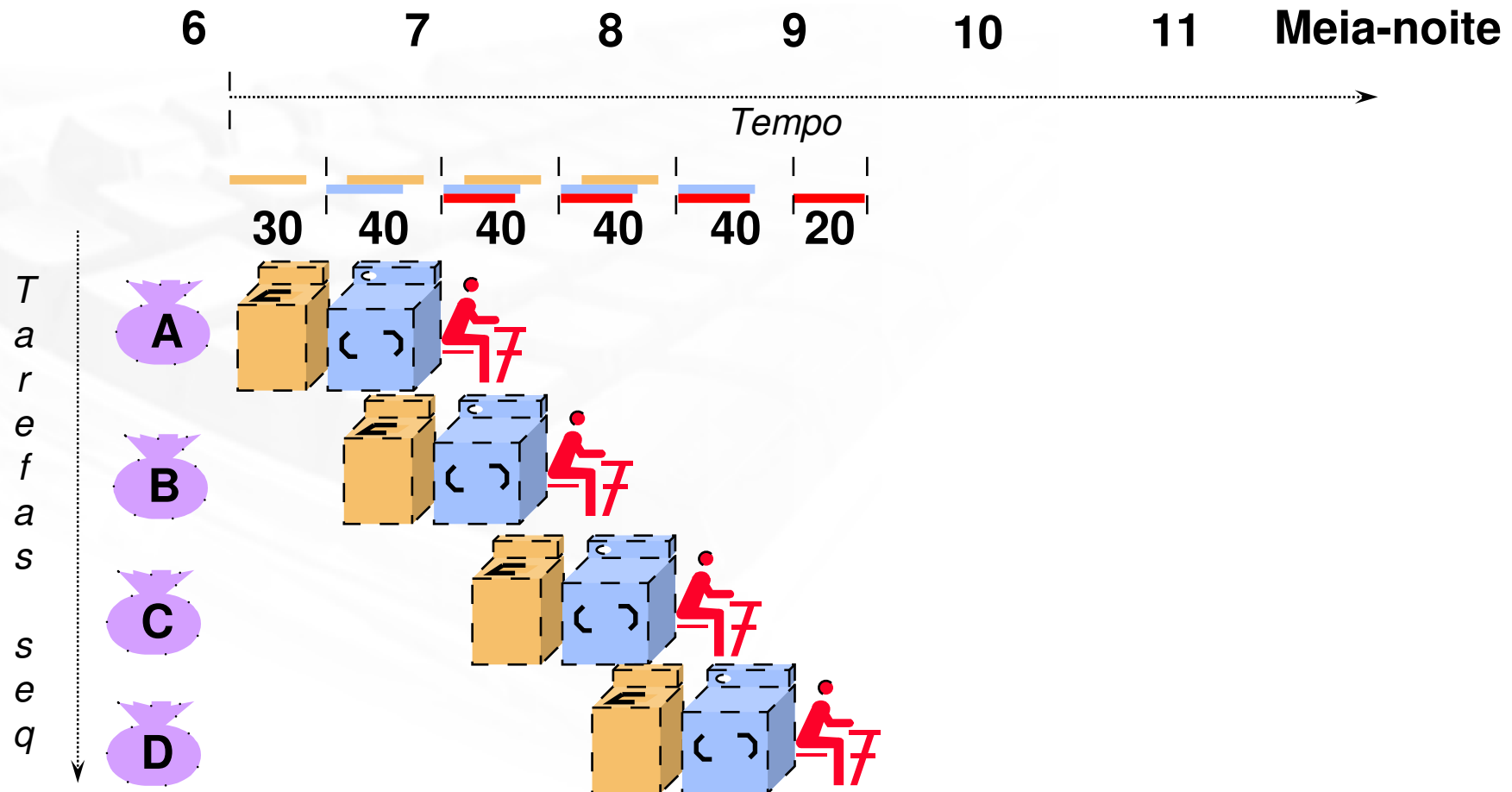
Pipelining em uma Lavanderia !?!

- 4 pessoas A, B, C e D possuem
 - 4 sacolas de roupa para lavar, secar e dobrar
- Lavar leva 30 minutos
- Secar leva 40 minutos
- Passar e dobrar leva 20 minutos





- Lavanderia sequencial leva 6 horas para terminar
- Se eles conhecessem computação, quanto tempo levaria?



- Lavanderia com *pipelining* leva 3hs e 30min !!!

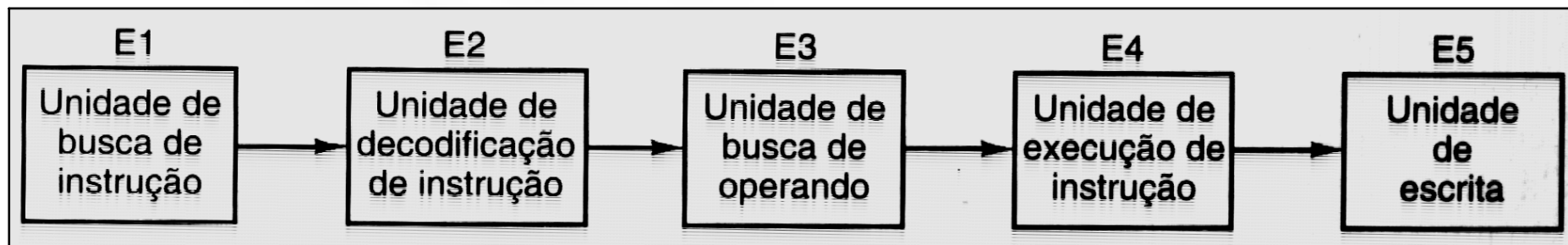
Considerações sobre Pipeline (Texto comp.)

- *Pipelining* **não** melhora a **latência** de uma única instrução, mas melhora o **throughput** do trabalho todo (nº de instruções por unidade de tempo)
 - $$\text{Tempo de cada instrução} = \frac{\text{Tempo de execução sem pipeline}}{\text{num. de estágios}}$$
- Taxa de inserção de instruções é limitada pelo estágio mais lento
- Existem múltiplas instruções sendo executadas em um dado instante
 - SpeedUp potencial => **número de estágios**
- Tempo para encher o pipeline e tempo de dreno **reduzem** o speedup

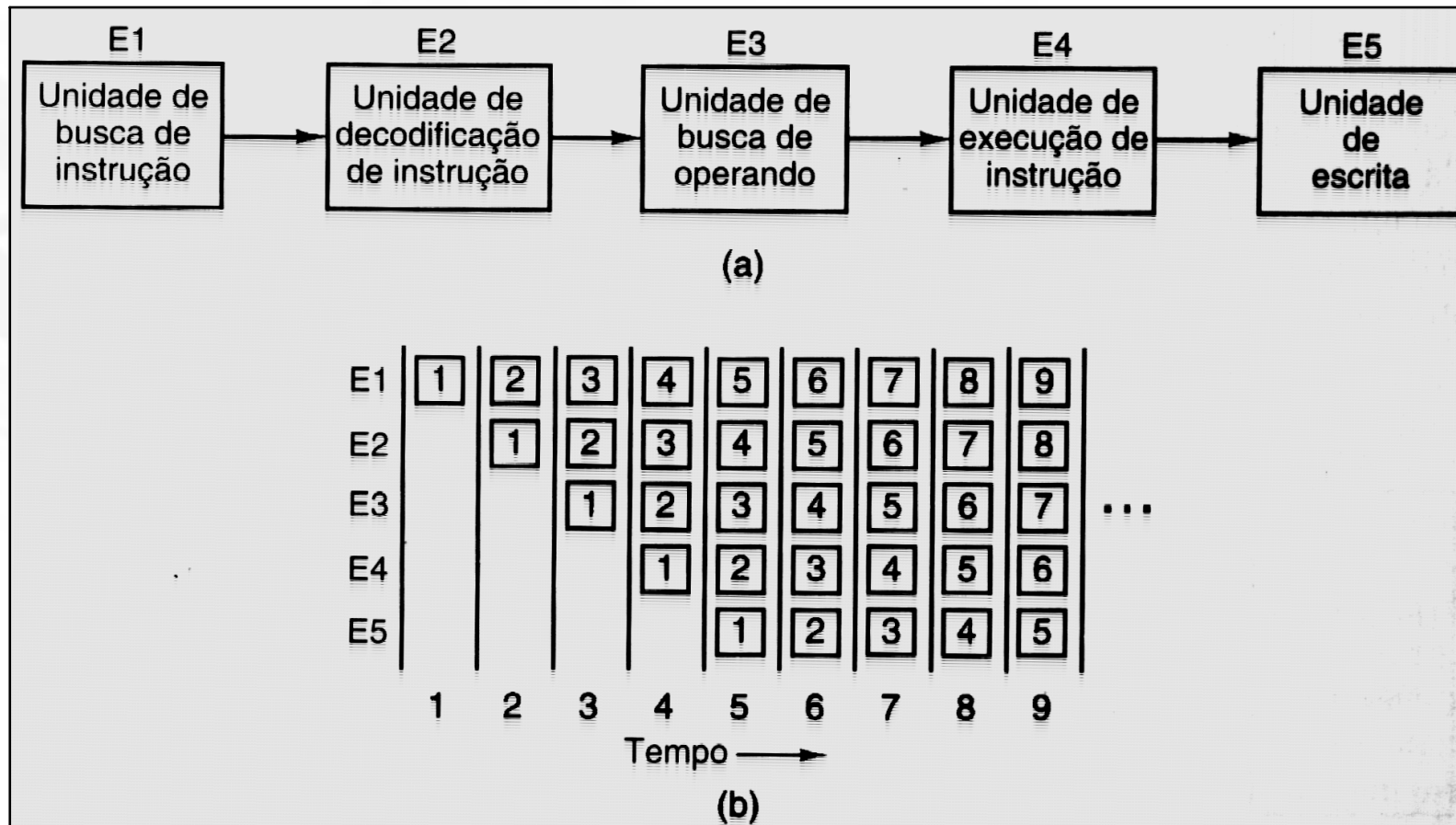
Paralelismo no Nível das Instruções ⁽³⁾

■ Pipeline de 5 estágios

1. Busca a instrução da memória e armazena em uma buffer até chegar a hora de executá-la
2. Decodificação da instrução, determinando tipo e operandos
3. Busca dos operandos na memória ou nos registradores
4. Execução - passagem pelo caminho de dados
5. Resultado do processamento é escrito em uma registrador



Paralelismo no Nível das Instruções (4)

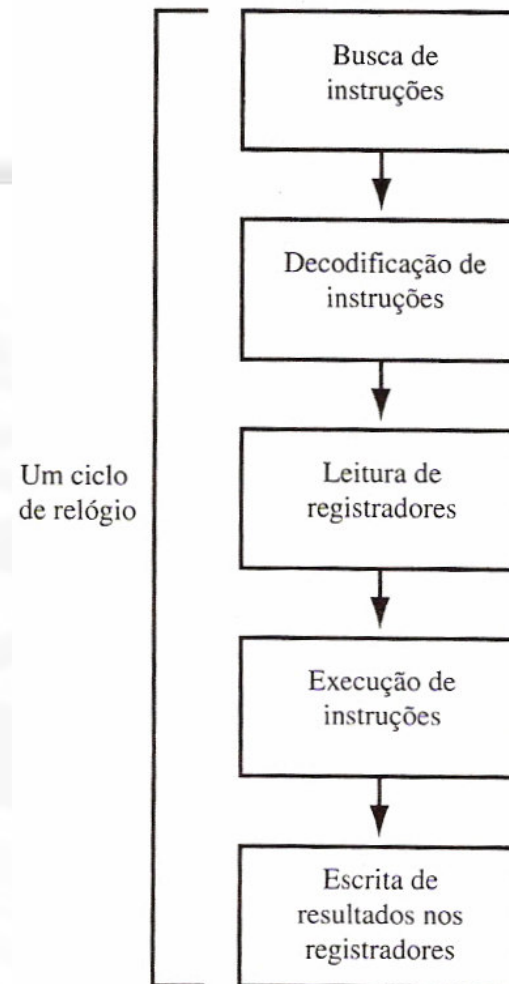


(a) Pipeline de 5 estágios. (b) Estado de cada um dos estágios em função do tempo (estão ilustrados 9 períodos do clock).

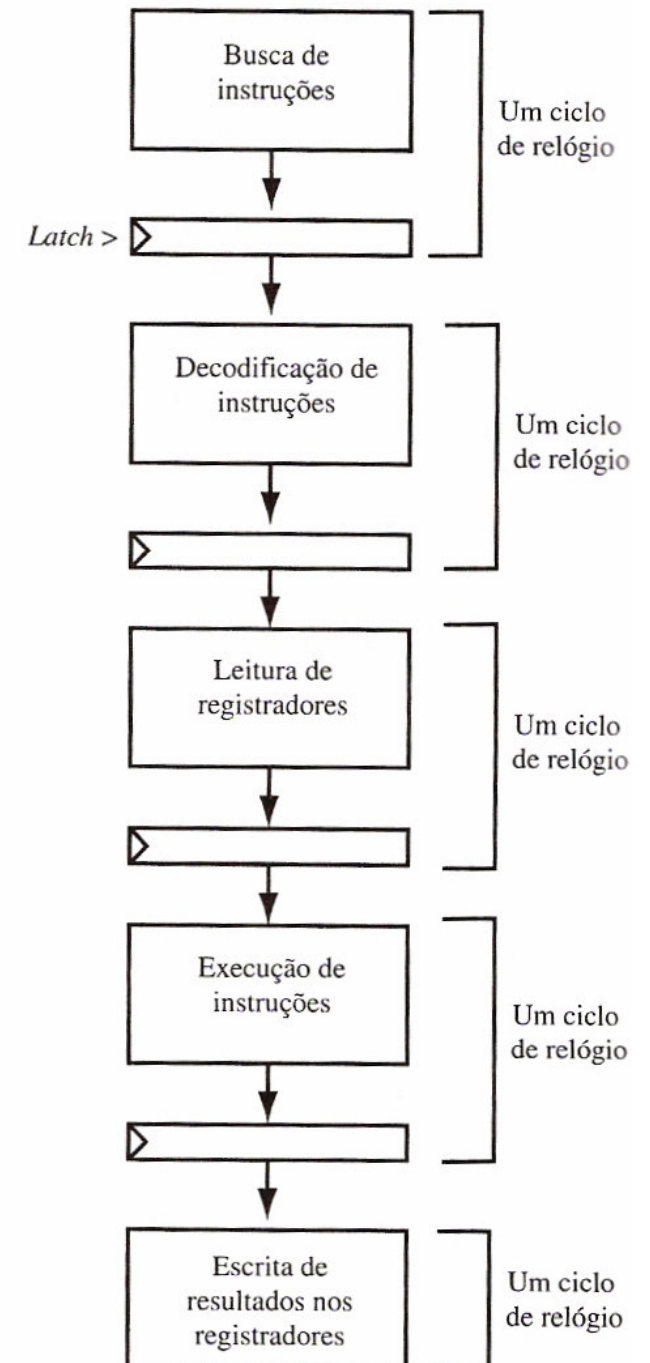
Paralelismo no Nível das Instruções (5)

- A idéia básica do pipeline é a mesma de uma linha de produção em série.
- Vários processamentos estão sendo executados ao mesmo tempo.
- Os estágios de cada processamento são aplicados a várias instruções ao mesmo tempo.
 - Exemplo:
 - **No tempo 1** a instrução 1 está sendo lida
 - **No tempo 2** a instrução 1 está sendo decodificada enquanto que a instrução 2 está sendo lida
 - **No tempo 3** a instrução 1 está buscando dados, a instrução 2 está sendo decodificada e a instrução 3 está sendo lida
 - ... e assim por diante.

Processador sem *pipeline*



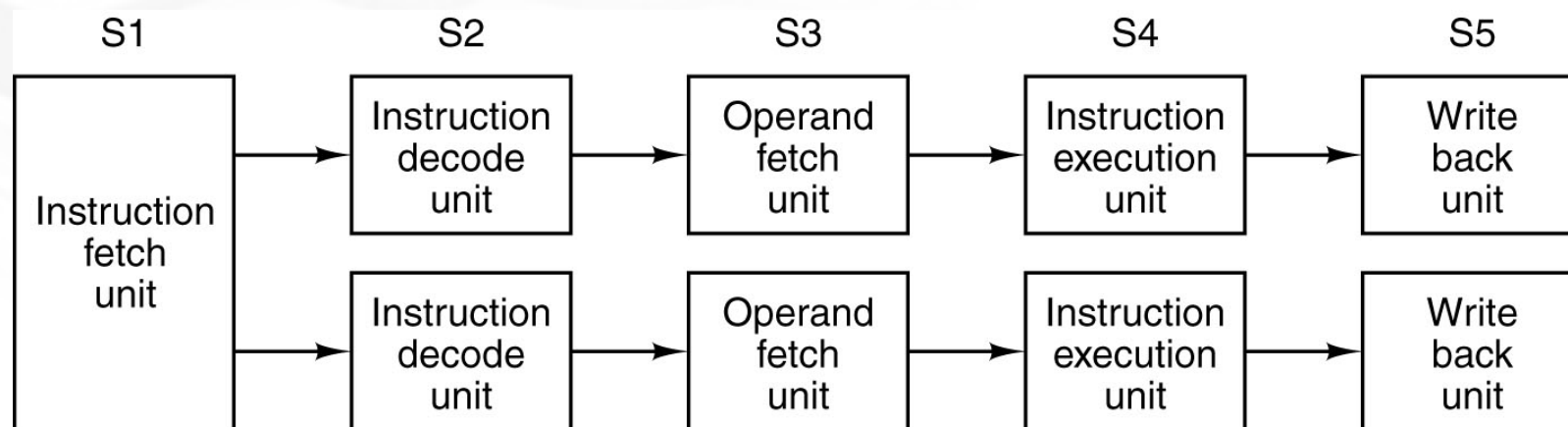
Processador com *pipeline*



Paralelismo no Nível das Instruções (6)

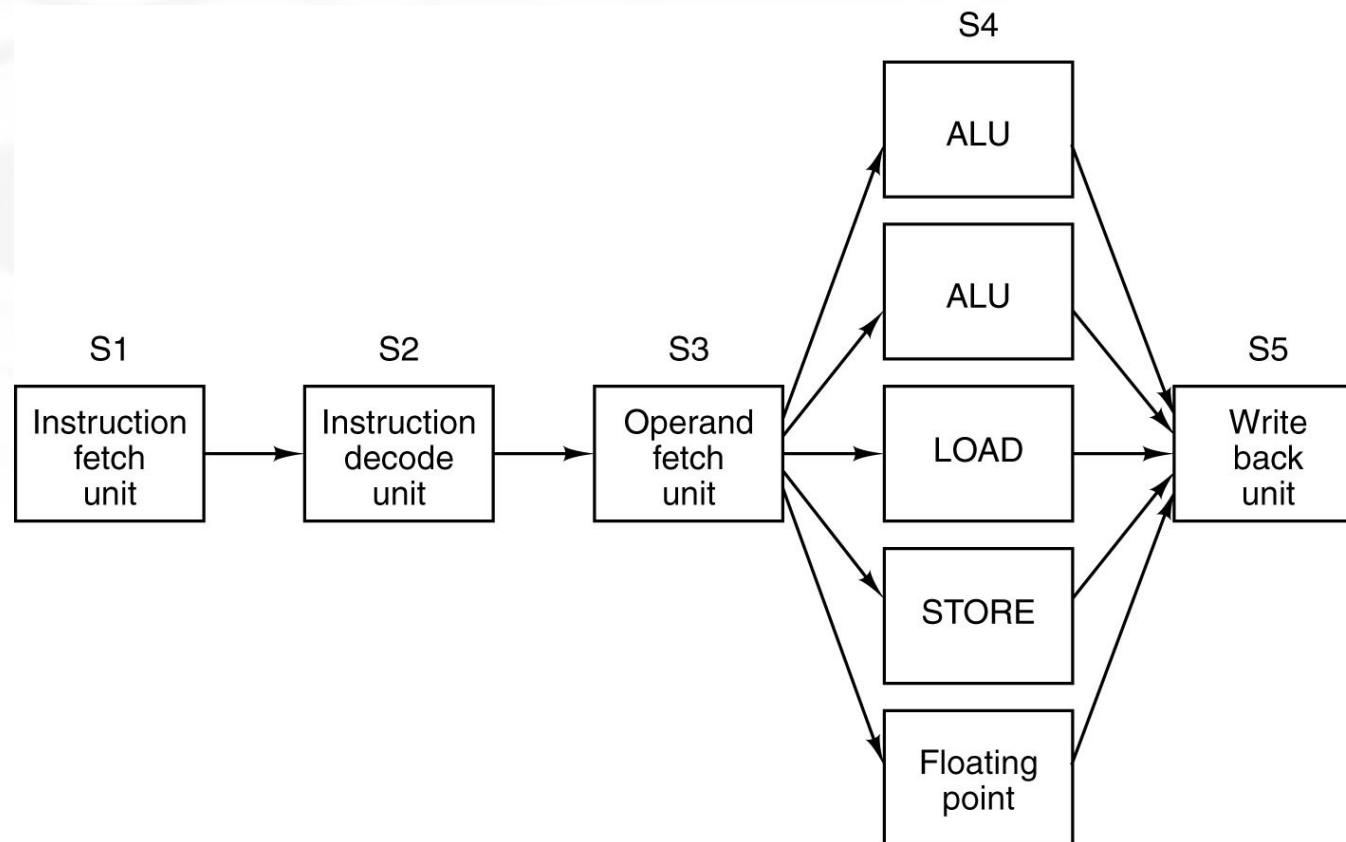
Paralelismo no Nível das Instruções (6)

- Arquiteturas superescalares



Paralelismo no Nível das Instruções (7)

- Arquiteturas superescalares (cont.)

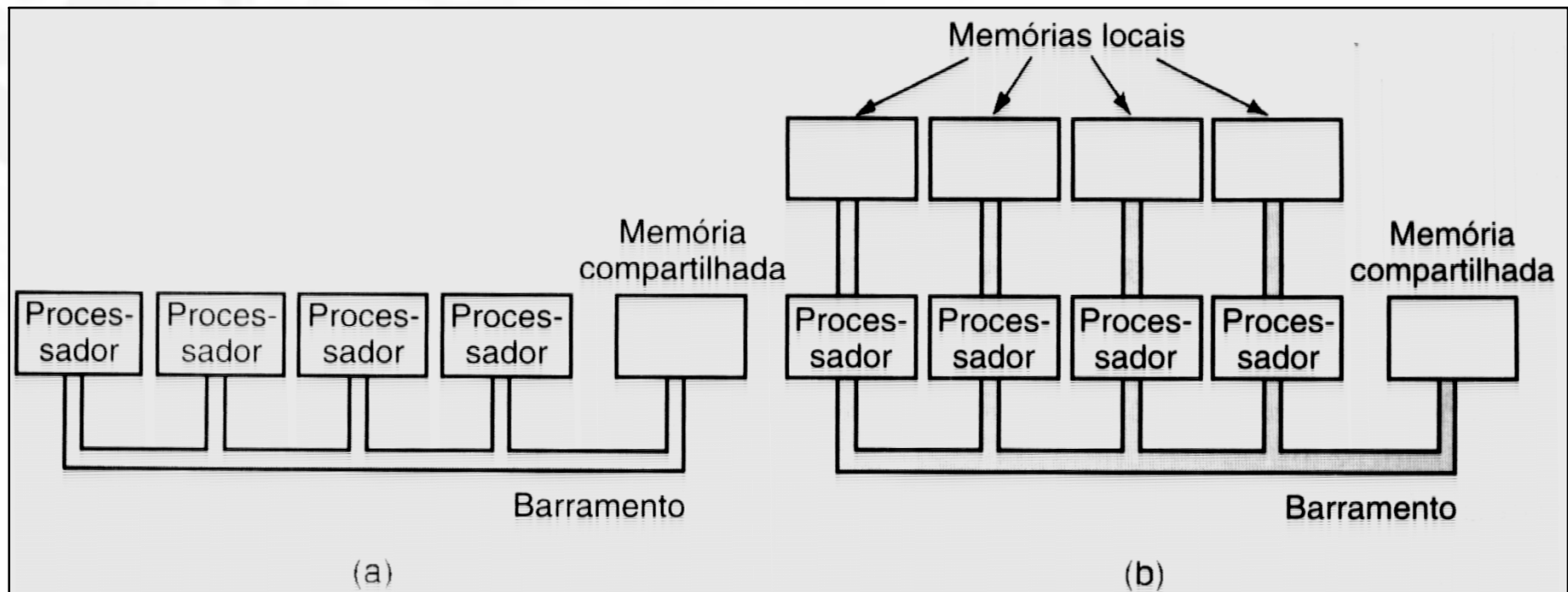


Paralelismo no Nível do Processador ⁽¹⁾

■ Multiprocessadores

- é composto de vários processadores independentes
- compartilham uma mesma memória por um barramento principal
 - OU compartilham uma memória e têm memórias locais
- executam processamentos locais
- liberam tráfego do barramento principal
- é necessário gerenciar conflitos
 - Ex: acesso ao mesmo end. de memória, dispositivo de E/S

Paralelismo no Nível do Processador (2)



- (a) Multiprocessador organizado em torno de um único barramento.
- (b) Multiprocessador com memórias locais.

Paralelismo no Nível do Processador ⁽³⁾

- Multicomputadores

- Sistemas (distribuídos) com um grande número de computadores interconectados
- Não existe memória comum sendo compartilhada
- Comunicação entre computadores é feita através de troca de mensagens a uma velocidade bem alta
 - Protocolo de comunicação
- Computador não precisa estar ligado diretamente com todos os outros (uso de topologias em árvore, anéis, etc..)

- Existem também Sistemas Híbridos

Desempenho de Computadores ⁽¹⁾

- Comparando o Desempenho de Dois Computadores
 - Sejam X e Y dois computadores. X é mais rápido que Y se:
Tempo de execução em X < 1
Tempo de execução em Y
 - O desempenho de um computador X pode ser definido como:
1
Tempo de execução em X
 - A unidade de medida de desempenho computacional mais importante é o **tempo**.

Desempenho de Computadores (2)

■ Medindo o Desempenho

- Em muitas situações, o tempo de execução de um programa não é fácil de se medir
- Sistema Multiprogramado
 - Sistema em que vários usuários ou programas compartilham a mesma CPU
- Tempo de execução de um programa pode ser descrito por:

User CPU time + System CPU time + I/O (Input/Output)

- Tempo de CPU de um programa** {
- *User CPU time*: tempo de CPU gasto pelo usuário
 - *System CPU time*: tempo de CPU gasto pelo sistema
 - *I/O (Input/Output) time*: tempo gasto com entrada/saída

Desempenho de Computadores (3)

Quais programas irão de fato testar o desempenho de um computador?
Como escolher esses programas?

■ Programas para Medir Desempenho

- **Programas Reais:** aqueles usados no dia-a-dia (ex: compiladores)
- **Kernels:** programas contendo apenas um trecho de código representativo (núcleo) de uma parte freqüentemente executada por programas típicos (ex: multiplicação de matrizes)
- **Toy Benchmarks:** programas reais bem simples
- **Synthetic Benchmarks:** são programas artificiais. São criados apenas para gerar medidas de desempenho.

Os programas reais são os mais adequados para se medir desempenho.

Desempenho de Computadores (4)

■ **Benchmark Suites**

- *Benchmark Suite* é um conjunto de programas de avaliação.
- *Standard Performance Evaluation Corporation* (SPEC) tem lançado vários *benchmark suites*: SPEC89, SPEC92, SPEC95 e SPEC2000.
- Esses *benchmark suites* são compostos por programas reais, escolhidos para serem representativos de programas que tipicamente demandam muita CPU e pouco I/O.

Exemplos de
programas inteiros
do SPEC2000:

gzip	Programa para compactar arquivos
gcc	Compilador C
vpr	FPGA Circuit Placement and Routing
craft	Game Playing: Chess
parser	Word Processing
vortex	Sistema de Banco de Dados
perlbnk	PERL Programming Language

Desempenho de Computadores (5)

- Comparando e Sumarizando Desempenho
 - Dúvidas ao comparar/sumarizar desempenho

	Computador A	Computador B	Computador C
Programa P1 (seg.)	1	10	20
Programa P2 (seg.)	1000	100	20
Tempo Total	1001	110	40

- Diferentes definições de “mais rápido”:
 - A é 10 vezes mais rápido que B para o programa P1.
 - B é 10 vezes mais rápido que A para o programa P2.
 - Se tomarmos o tempo total para executar P1 e P2 como medida de desempenho, C é o computador mais rápido; contudo, ele é 20 vezes mais lento que A na execução do programa P1.
- Melhor forma de sumarizar desempenho: **tempo total de execução.**

Desempenho de Computadores (6)

- Comparando o desempenho de um *benchmark suite*:

- **Média Aritmética dos tempos de execução**

$$\text{Média aritmética de } Tempo(i,n) = \frac{1}{n} \sum_{i=1}^n Tempo_i$$

- **Média Harmônica**

$$\text{Média harmônica de } Tempo(i,n) = \frac{n}{\sum_{i=1}^n \frac{1}{Tempo_i}}$$

	Computador A	Computador B	Computador C
Programa P1 (seg.)	1	10	20
Programa P2 (seg.)	1000	100	20
Tempo Total	1001	110	40
Média Aritmética	500,5	55,0	20,0
Média Harmônica	1,998	18,181	20,000

Desempenho de Computadores (7)

- **Princípios Quantitativos para o Projeto de Computadores**

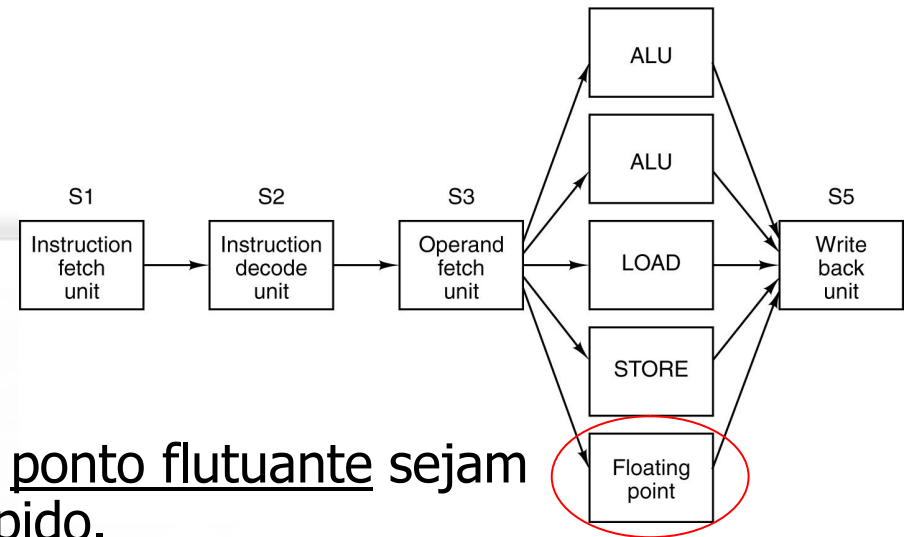
- O principal princípio a ser seguido é: **Torne mais rápidas as tarefas mais freqüentes**

- **Lei de Amdahl**

- *A quantidade de melhoria de desempenho provida por um atributo da arquitetura é limitada pela quantidade de tempo que este atributo é utilizado.*
- Relaciona o **speedup** total de um sistema com o **speedup** de uma porção (*atributo da arquitetura*) do sistema

- $Speedup = \frac{\text{Tempo de execução da tarefa inteira sem usar o novo atributo}}{\text{Tempo de execução da tarefa inteira usando o novo atributo quando possível}}$

Lei de Amdahl - Exemplo



- Suponha que todas as instruções em ponto flutuante sejam melhoradas rodando 5 vezes mais rápido.
- O tempo de execução de alguns benchmark antes da melhoria no ponto flutuante é de 10 segundos.
- Qual será o aumento da velocidade (speedUp) se, antes da melhoria, metade dos 10 segundos eram gastos executando instruções em ponto flutuante?

$$T = T_f + T_o \quad \left\{ \begin{array}{l} T_f: \text{Tempo instruções em ponto flutuante} \\ T_o: \text{outras instruções} \end{array} \right.$$

$$\text{speedUp} = \frac{T_1 \text{ (T sem melhoria)}}{T_2 \text{ (T com melhoria)}} = \frac{T_{f1} + T_{o1}}{T_{f2} + T_{o2}} = \frac{5 + 5}{5/5 + 5} = \frac{10}{5/5 + 5} = 1,67$$

Desempenho de Computadores (9)

■ **Tempo de Execução de um Programa**

- Tempo de CPU + I/O
- PS: por simplicidade chamaremos Tempo de execução o Tempo de CPU (desconsideraremos I/O):

Tempo de CPU de um Programa =

Número de Ciclos de Clock despendidos no Programa \times Tempo de um Ciclo

Tempo de CPU de um Programa =

Número de Ciclos de Clock despendidos no Programa / Frequência de Clock

Desempenho de Computadores ⁽¹⁰⁾

- Exercício:

Um programa roda em 10 segundos em um computador A, que tem um clock de 100MHz.

Um computador B gasta 1.2 vezes mais ciclos de clock para rodar o mesmo programa mas roda este programa em 6 segundos.

Qual é a frequência de clock do computador B?

$$\text{Comp. A: } 10\text{s} = n_{\text{ciclos_A}} / 100\text{MHz}$$

$$n_{\text{ciclos_A}} = 10\text{s} \times 100\text{MHz}$$

$$\text{Comp. B: } 6\text{s} = 1.2 \times n_{\text{ciclos_A}} / \text{freq_B}$$

$$\text{freq_B} = 1.2 \times 10\text{s} \times 100\text{MHz} / 6\text{s}$$

$$\text{freq_B} = 200\text{MHz}$$

Desempenho de Computadores (11)

- ***Clocks per Instruction (CPI)***: é a média do número de ciclos por instrução
- É comum calcularmos o desempenho da CPU em função do CPI
 - O tempo de execução de um programa, T , em uma determinada CPU:

$$T \text{ (segundos)} = N \text{ (instrução)} \times C \text{ (ciclo/instrução)} \times S \text{ (segundos/ciclo)}$$

- N é o número total de instruções executadas
 - C (ou clocks por ciclo de máquina – ***Clocks per Instruction (CPI)***): é a média do número de ciclos por instrução
 - S é o número de segundos por ciclo
-
- Quanto menor o tempo, maior o desempenho.
 - N , C e S são afetados primariamente:
 - pela capacidade de otimização do compilador
 - pela arquitetura do processador e de seu conjunto de instruções
 - pela tecnologia empregada na implementação da máquina.

Desempenho de Computadores (12)

- Exercício: Um projetista de compiladores está tentando decidir sobre qual seqüência de instruções ele deve utilizar para implementar um comando de uma linguagem de alto nível. Ele dispõe das seguintes informações:
 - Existem duas seqüências de instruções possíveis para implementar o comando
 - Seqüência 1: 2 instruções A, uma B e duas C;
 - Seqüência 2: quatro A, uma B e uma C.
 - Na máquina alvo, a instrução A tem o CPI de 1, a instrução B tem CPI de 2 e a C, de 3.
 - **1** - Que seqüência de código executa mais instruções? **2** - Qual seqüência executa mais rápido? **3** - Qual é o CPI médio para cada seqüência?
- Respostas:
 - 1) $\text{seq1} = 2+1+2 = 5$ instruções; **seq2** = $4+1+1 = 6$ instruções
 - 2) $\text{seq1} = 2 \times 1 + 1 \times 2 + 2 \times 3 = 10$ ciclos; **seq2** = $4 \times 1 + 1 \times 2 + 1 \times 3 = 9$ ciclos
 - 3) $\text{CPI}(\text{seq1}) = 10 \text{ ciclos} / 5 \text{ instruções} = 2 \text{ CPI}$; $\text{CPI}(\text{seq2}) = 9 / 6 = 1.5 \text{ CPI}$

Unidades de Medida de Performance ⁽¹⁾

- MIPS – *Million Instructions per Second*

- A unidade MIPS pode ser definida como:

$$\text{MIPS} = \text{Frequência de Clock} / \text{CPI} \times 10^6$$

$$= \text{nº de instruções} / \text{tempo de execução} \times 10^6$$

- O Tempo de Execução de um programa pode ser definido em função da unidade MIPS:

$$\text{Tempo de execução} = \text{No. de Instruções} / \text{MIPS} \times 10^6$$

- MIPS é uma unidade de performance ruim ... Porquê ???

1. Não há como comparar computadores com conjuntos de instruções diferentes. Pois o número de instruções necessário para executar um determinado programa nestes dois computadores vai quase certamente diferir, e se a frequência de clock for a mesma, o computador que demora mais para executar o programa aparentemente terá o mesmo desempenho do que demora menos.
2. A quantidade de MIPS varia entre programas em um mesmo computador, assim, uma máquina não pode ter uma performance única em termos de MIPS.

Unidades de Medida de Performance (2)

- MIPS – *Million Instructions per Second*
 - Exercício: Considere um computador com clock de 100MHz. Sabendo que para um programa específico o compilador 1 gera
 - 5 milhões de instruções do tipo A
 - 1 milhão do tipo B
 - 1 milhão do tipo C,e que o compilador 2 gera
 - 10 milhões do tipo A,
 - 1 milhão do tipo B
 - 1 milhão do tipo Ce que o CPI de cada instrução é igual ao do exercício anterior (a instrução A tem o CPI de 1, a instrução B tem CPI de 2 e a C, de 3),
mostre que a medida de performance em termos de MIPS pode variar inversamente com a performance verdadeira. Suponha que cada instrução gerada é executada apenas uma vez.

Performance = 1 / tempo de execução

MIPS = Frequência de Clock / **CPI** x10⁶

no de instruções / tempo de execução x10⁶

Unidades de Medida de Performance ⁽³⁾

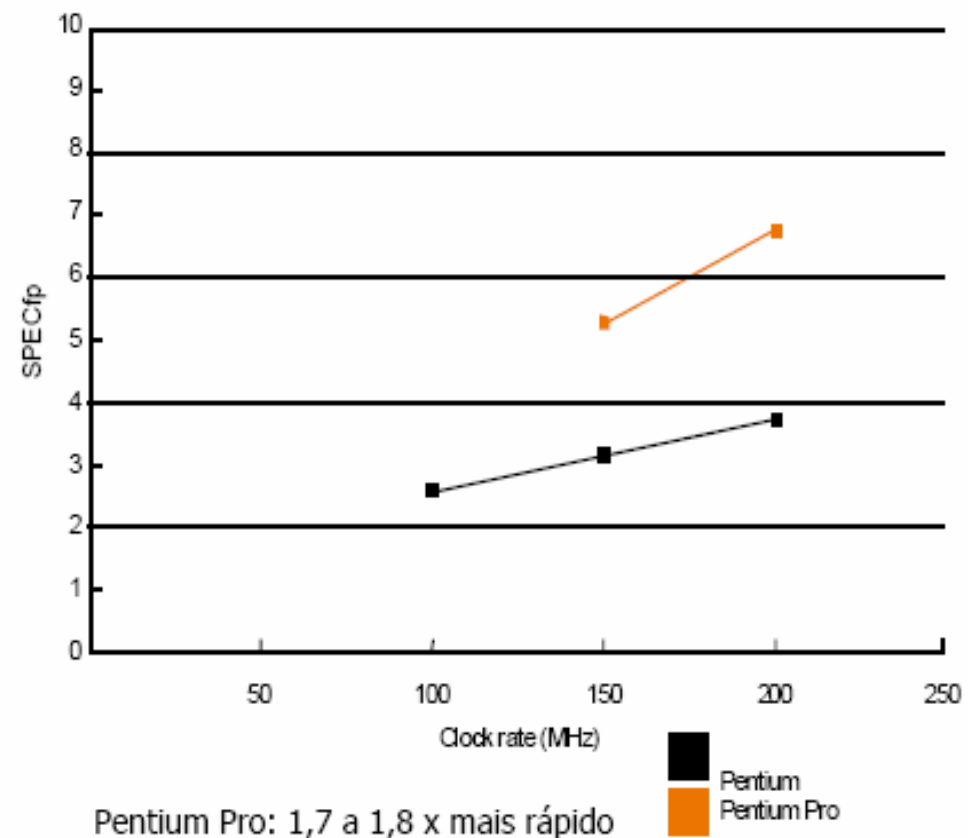
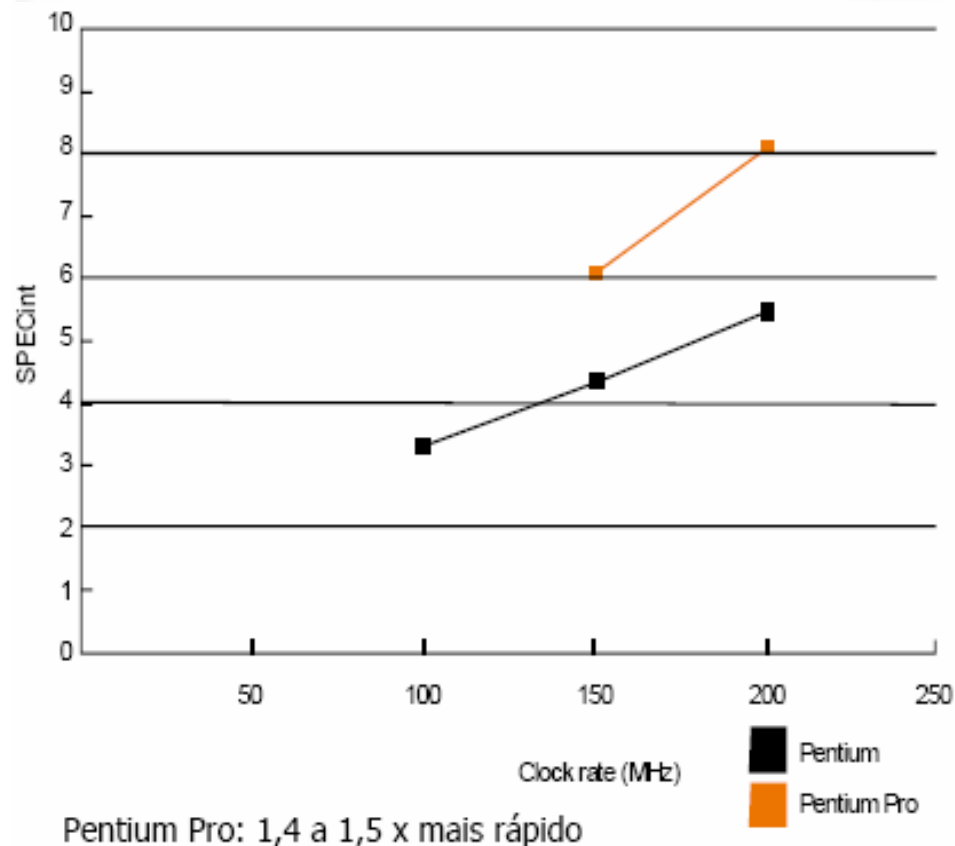
■ *SPEC*

- Medir o tempo de execução usando um conjunto de programas de uso generalizado, com entradas conhecidas, e sob circunstâncias conhecidas e especificadas é a melhor maneira de se medir o desempenho de um computador.
- As unidades de medidas (*score*) **SPECint** e **SPECfp** são consideradas, atualmente, boas unidades de medida de desempenho.

Unidades de Medida de Performance (4)

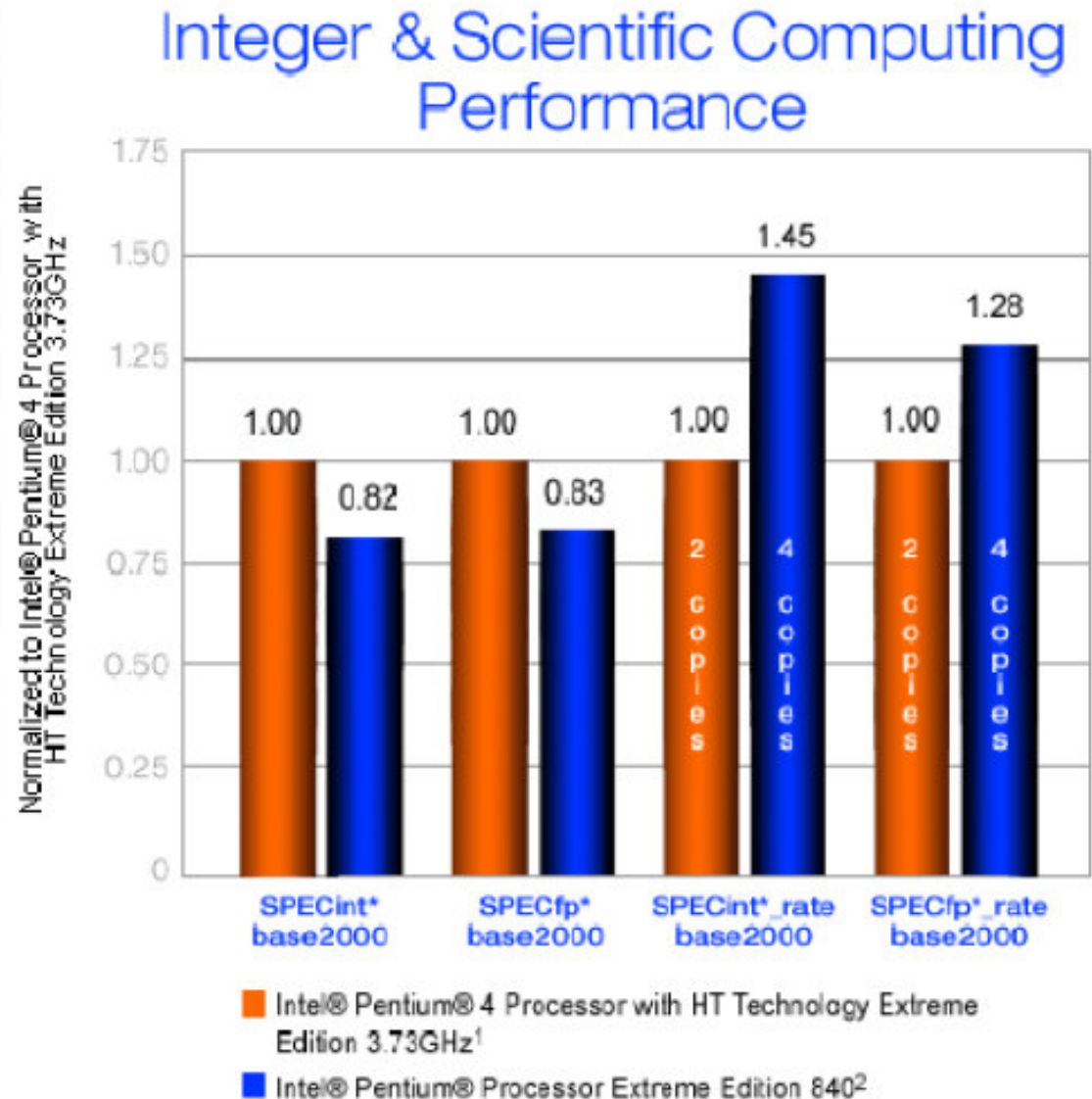
- SPEC'95 (8 programas inteiros, 10 de ponto flutuante)

Pentium e Pentium Pro



Unidades de Medida de Performance ⁽⁵⁾

- Spec2000
 - Pentium 4 3.73 GHz
 - Pentium dual-core 840 3.20 GHz



Referências

- Andrew S. Tanenbaum, ***Organização Estruturada de Computadores***, 5ª edição, Prentice-Hall do Brasil, 2007.
- John L. Hennessy and David A. Patterson, **Arquitetura de Computadores: Uma Abordagem Quantitativa**. 3ª edição. Editora Campus, 2003.
- Lúcia Helena M. Pacheco, **Visão Geral de Organização Estruturada de Computadores e Linguagem de Montagem**. Universidade Federal de Santa Catarina. Centro Tecnológico, Departamento de Informática e de Estatística.
- <http://www.wikipedia.org>