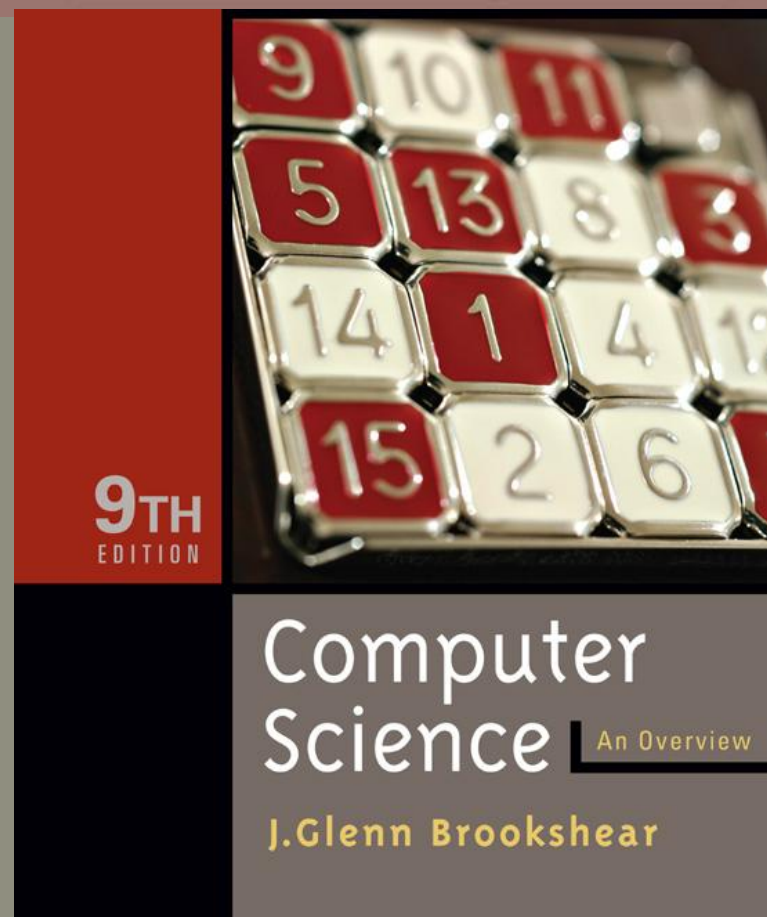




Capítulo 2

Manipulação De Dados



Slides originais: © 2007 Pearson Addison-Wesley. All rights reserved
Modificados para MIM-Introdução à Informática (2425) por Pedro Ribeiro, Miguel Areias e Nuno
Guimarães

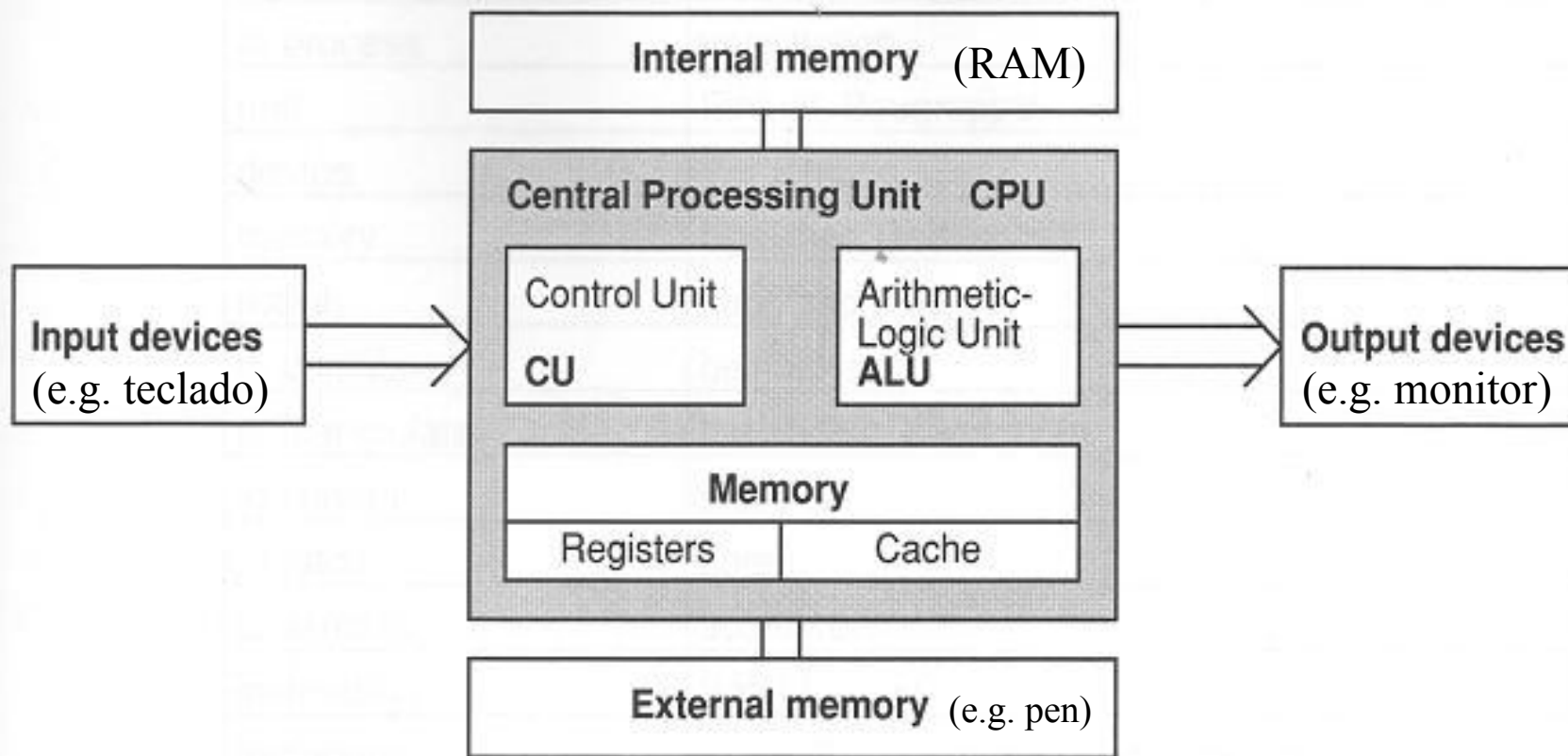


Capítulo 2: Manipulação de dados

- 2.1 Arquitetura de computadores
- 2.2 Linguagem máquina
- 2.3 Execução de programas
- 2.4 Instruções aritméticas/lógicas
- 2.5 Comunicação com outros dispositivos



Arquitetura do computador



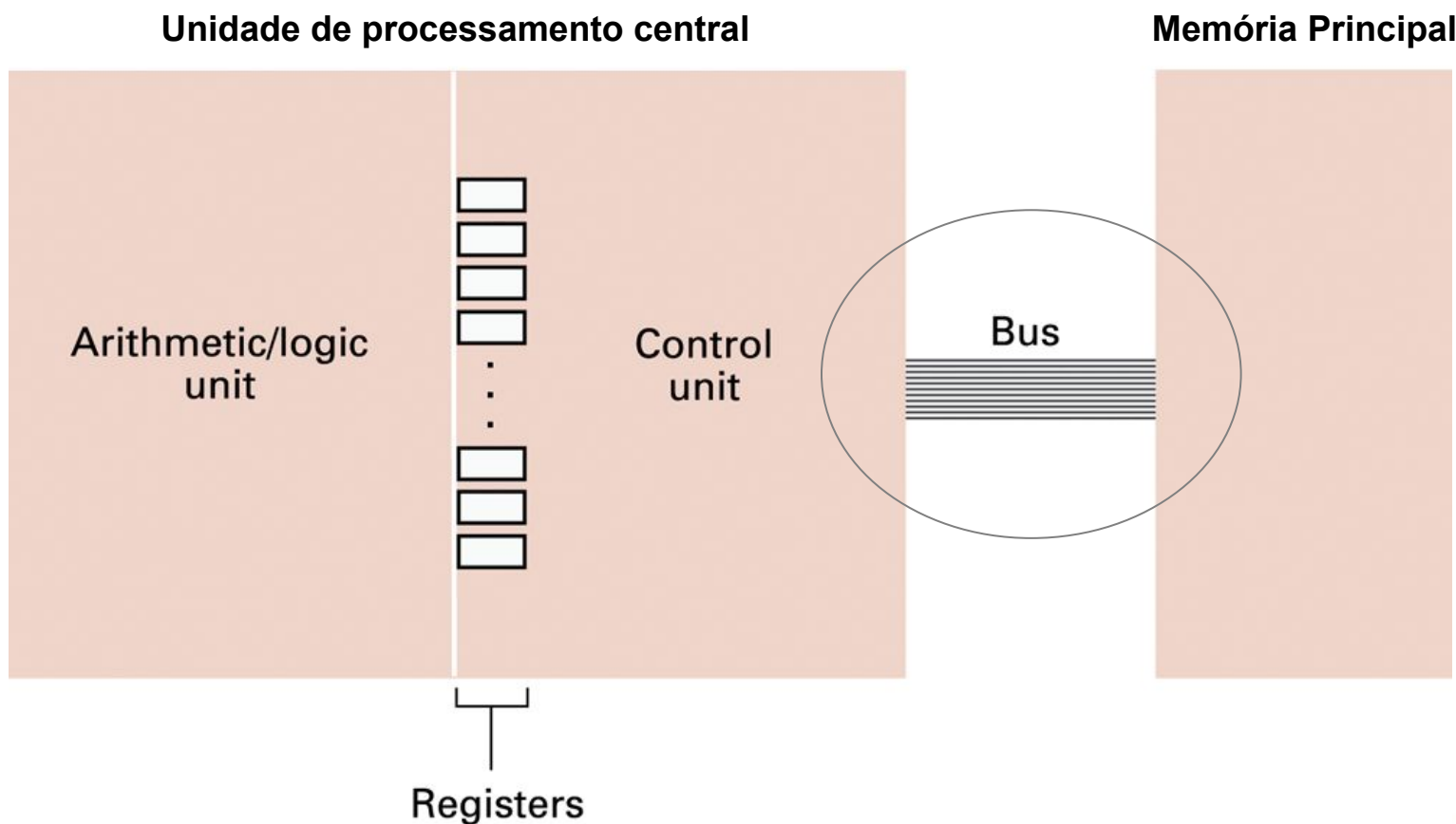


Arquitetura do computador

- Unidade de processamento central (CPU)
 - Unidade Aritmética/Lógica: *executa operações aritméticas/lógicas*
 - Unidade de controlo: *controla a execução das instruções e das operações de input/output*
 - Registos: *células de memória “dentro” do CPU onde as instruções são executadas*
 - Memória cache: *memória de acesso rápido usada em dados que são frequentemente acedidos*



CPU e memória principal ligados via um *bus*





Conceito de programa armazenado

- Um programa é uma sequência de instruções que implementam um algoritmo
- Um programa é “apenas” um tipo especial de dados.
 - Pode ser armazenado na memória



O que são instruções?

- As linguagens humanas são demasiado “pesadas” e ambíguas para um computador **descodificar e executar**
- As linguagens máquina devem ser:
 - concisas: fáceis de descodificar
 - precisas: executáveis de apenas uma maneira



Linguagem máquina: definições

- **Instrução máquina:** uma instrução codificada como uma sequência de bits que é diretamente reconhecida pelo CPU
- **Linguagem máquina:** o conjunto de todas as instruções reconhecidas por um CPU
 - Cada CPU têm a sua própria linguagem máquina, chamada de Instruction Set Architecture (ISA)



Linguagem máquina: abordagens

- *Reduced Instruction Set Computing (RISC)*
 - Instruções: poucas, simples, eficientes e rápidas.
 - Exemplos: PowerPC da Apple/IBM/Motorola, ARM
- *Complex Instruction Set Computing (CISC)*
 - Instruções: várias, convenientes e poderosas.
 - Exemplo: Pentium da Intel



Tipos de instruções máquina

- *Data Transfer*: copiar os dados entre o CPU e a memória principal
- *Arithmetic/Logic*: usar valores de dados existentes para calcular novos valores
- *Control*: dirigir a execução de um programa



Qual é o tipo de instruções?

Passo 1: **Carregar** um registo com um valor da memória

Passo 2: **Carregar** outro registo com outro valor da memória

Passo 3: Se este segundo valor é 0, **saltar** para o passo 6

Passo 4: **Dividir** os conteúdos do primeiro registo pelo segundo registo e colocar o resultado num terceiro registo

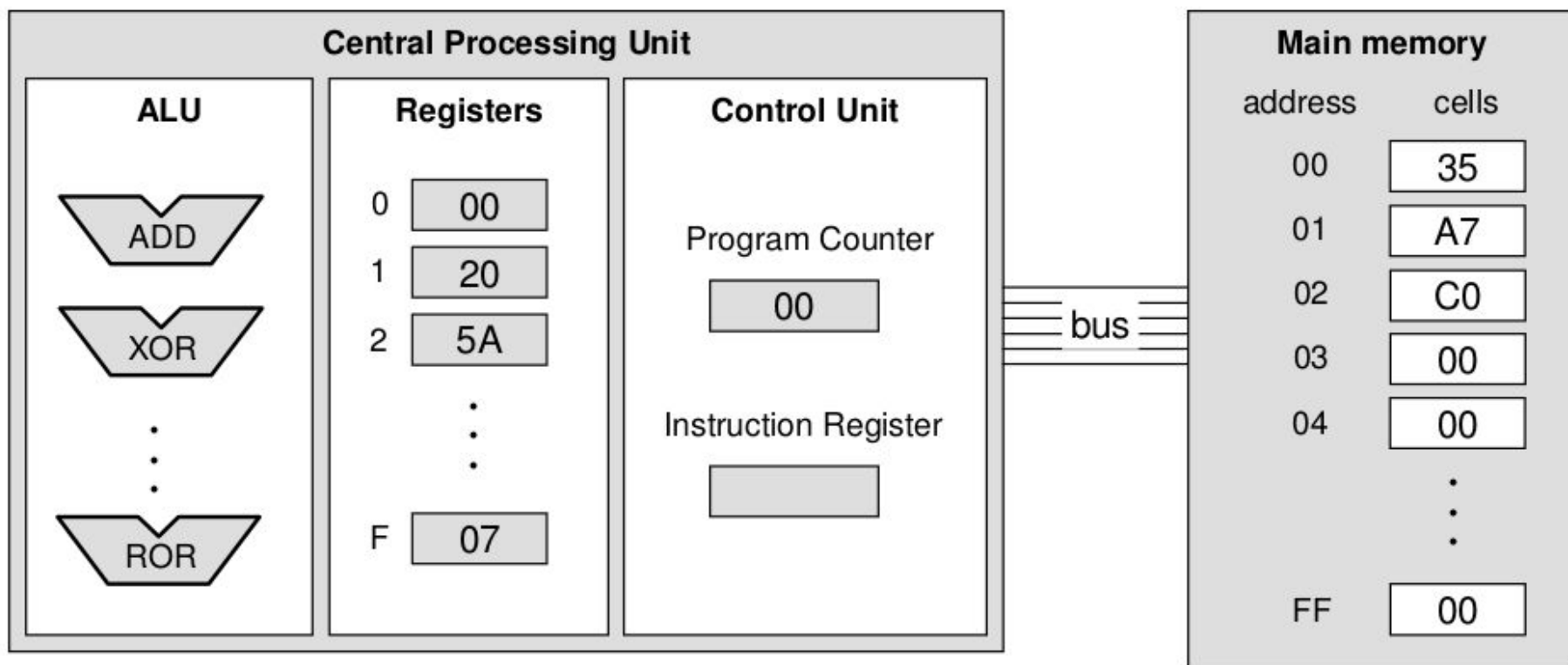
Passo 5: **Guardar** o conteúdo do terceiro registo na memória

Passo 6: STOP



Exemplo de arquitetura simples

- O livro da UC descreve um exemplo de arquitetura de CPU





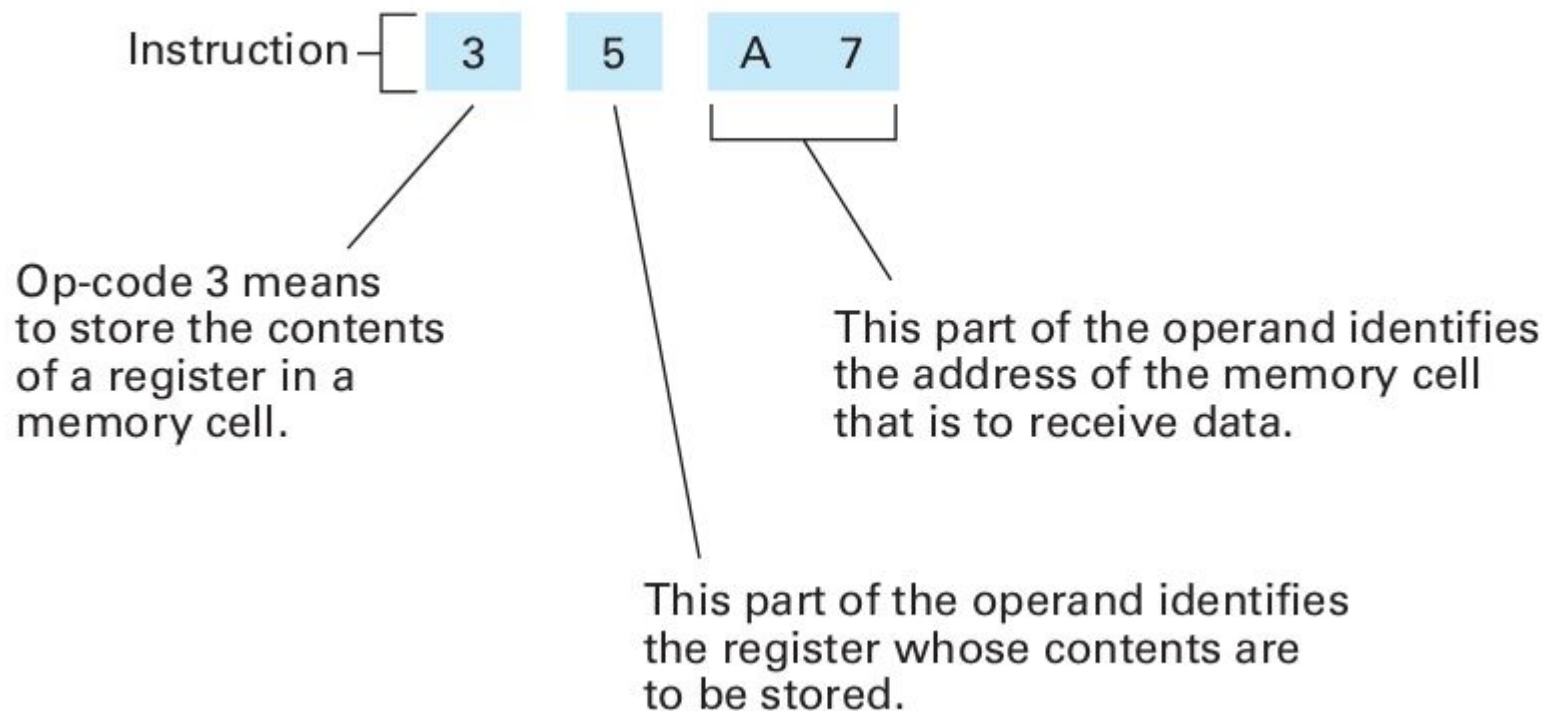
Partes de uma instrução máquina

- *Op-code*: especifica qual a operação a executar
 - Um por cada instrução
- *Operand*: informação mais detalhada sobre a operação
 - Número de operandos depende do *op-code*
- Exemplo: esta instrução pede ao CPU para adicionar data7 e data2 e colocar resultado em data6





Partes de uma instrução máquina



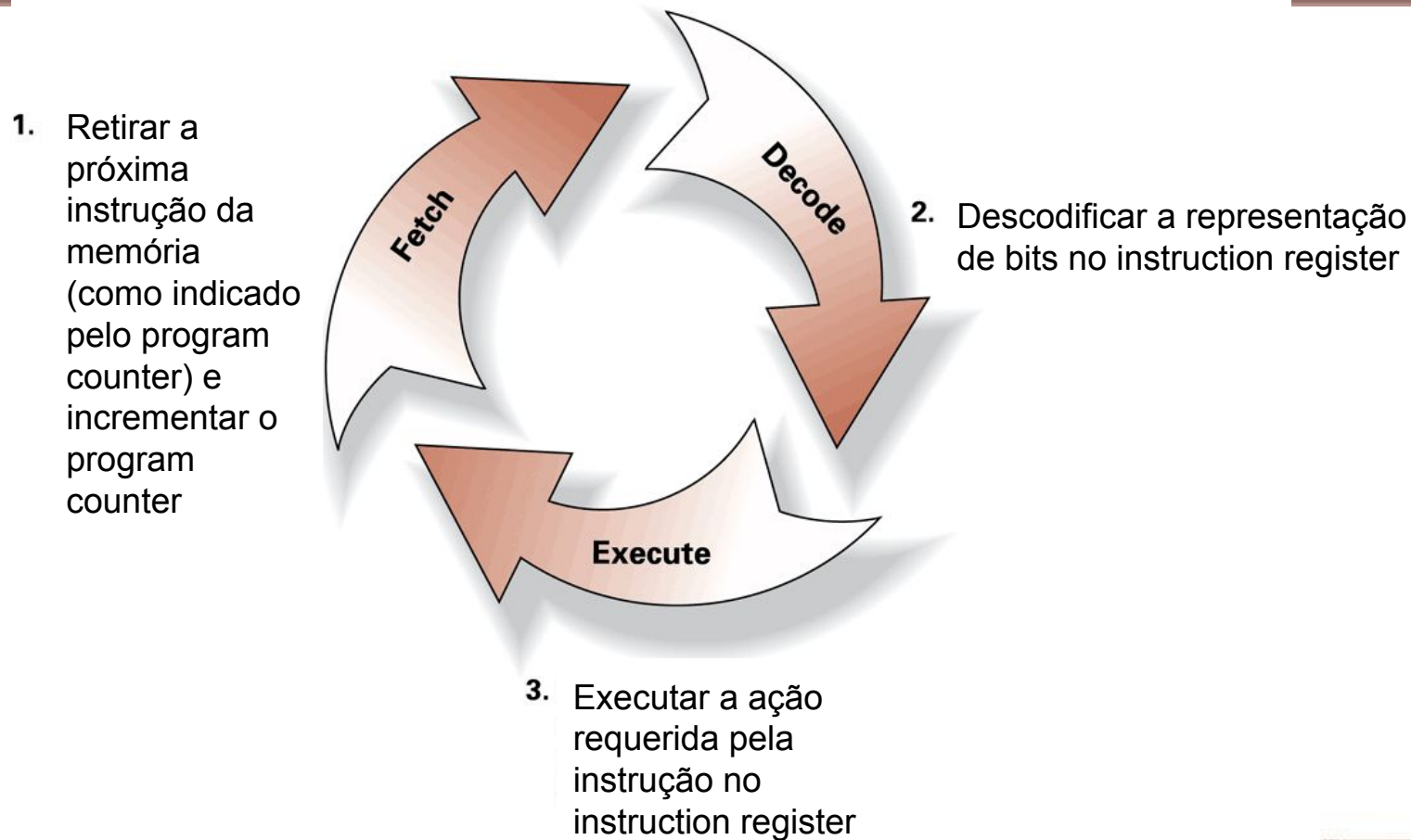


Execução do programa

- Controlada por dois tipos especiais de registos
 - *Program counter*: endereço da próxima instrução
 - *Instruction register*: instrução corrente
- Tarefas efetuadas pela unidade de controlo
 - *Fetch*
 - *Decode*
 - *Execute*

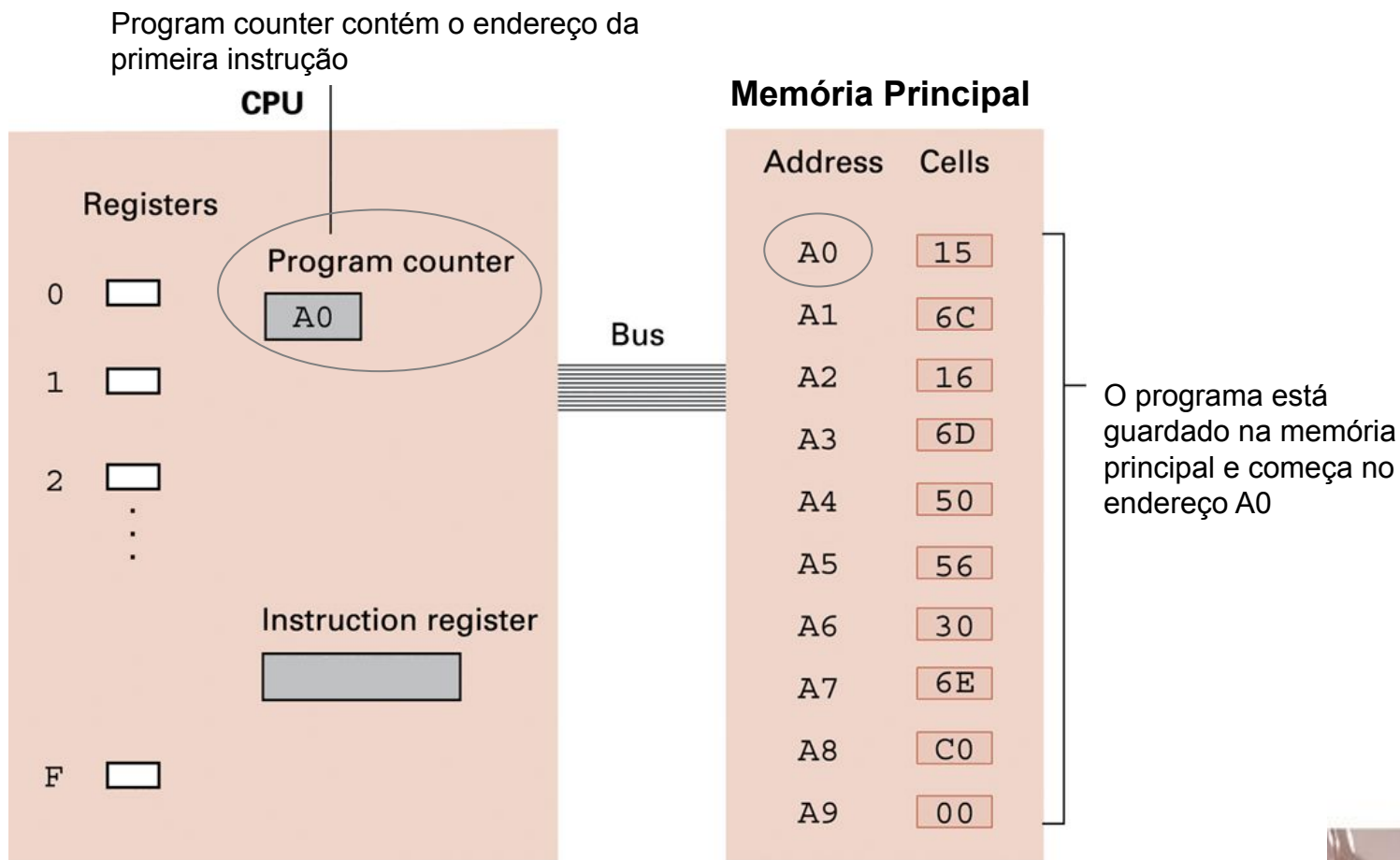


O ciclo da máquina



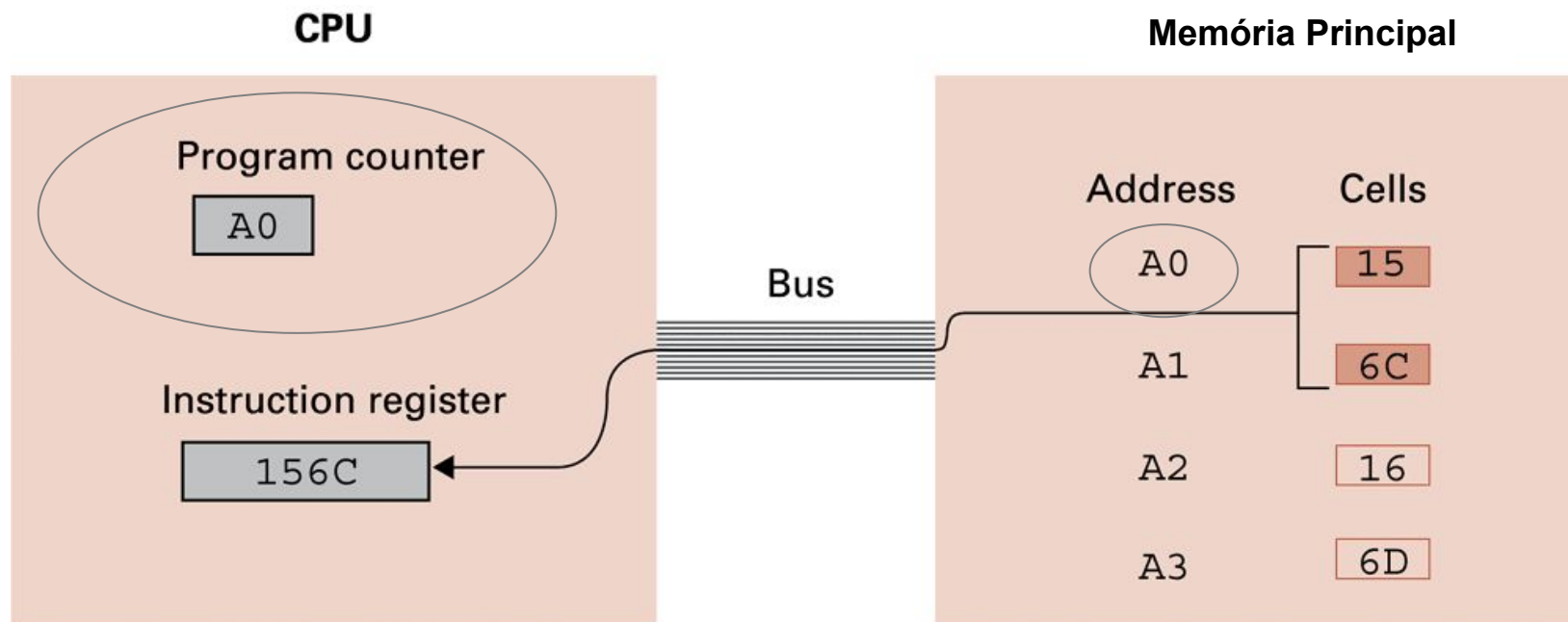


Um programa guardado na memória principal pronto para execução





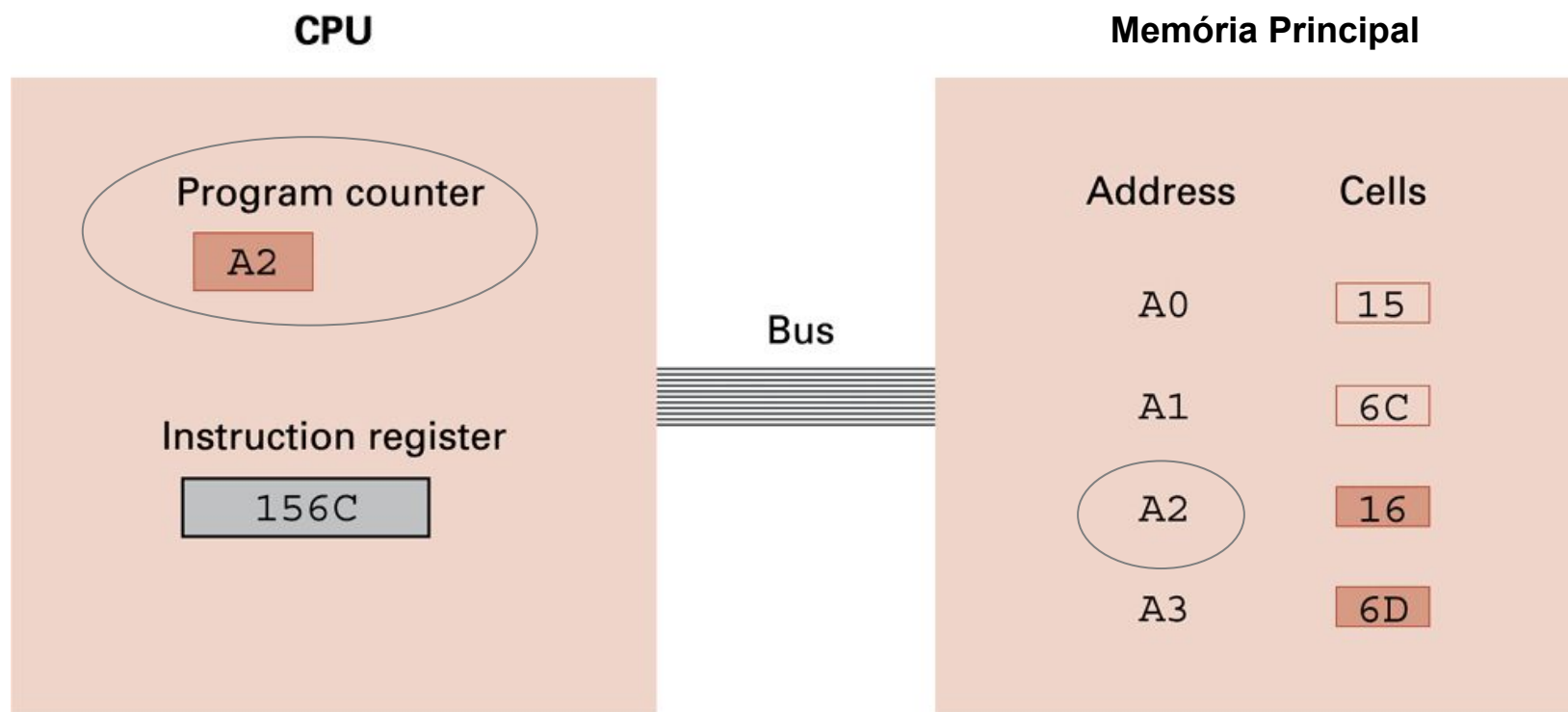
Executando o passo de *fetch*



- a. No início do passo de fetch, a instrução guardada no endereço A0 é retirada da memória e colocada no instruction register



Executando o passo de *fetch* (cont.)



b. De seguida, o program counter é incrementado para apontar para a próxima instrução

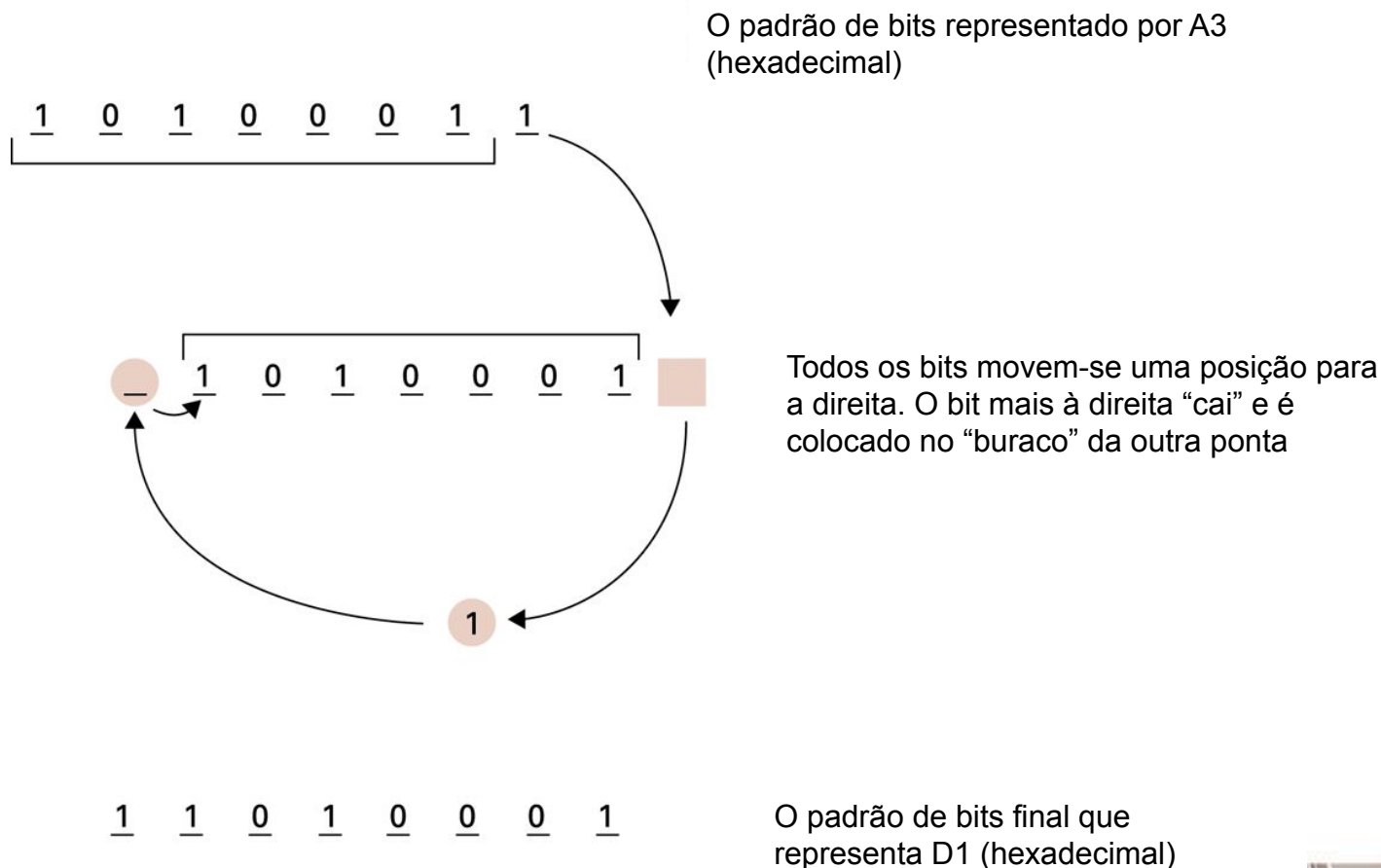


Operações Aritméticas/Lógicas

- A Unidade Aritmética/Lógica (ALU) é então responsável pela manipulação de dados e tem 3 tipos de instruções:
 - Lógicas: AND, OR, XOR, NOT, ...
 - *Rotate* e *Shift*: *shift* circular, *shift* lógico, *shift* aritmético
 - Aritméticas: *add*, *subtract*, *multiply*, *divide*



Rodando a sequência de bits de A3 um bit à direita



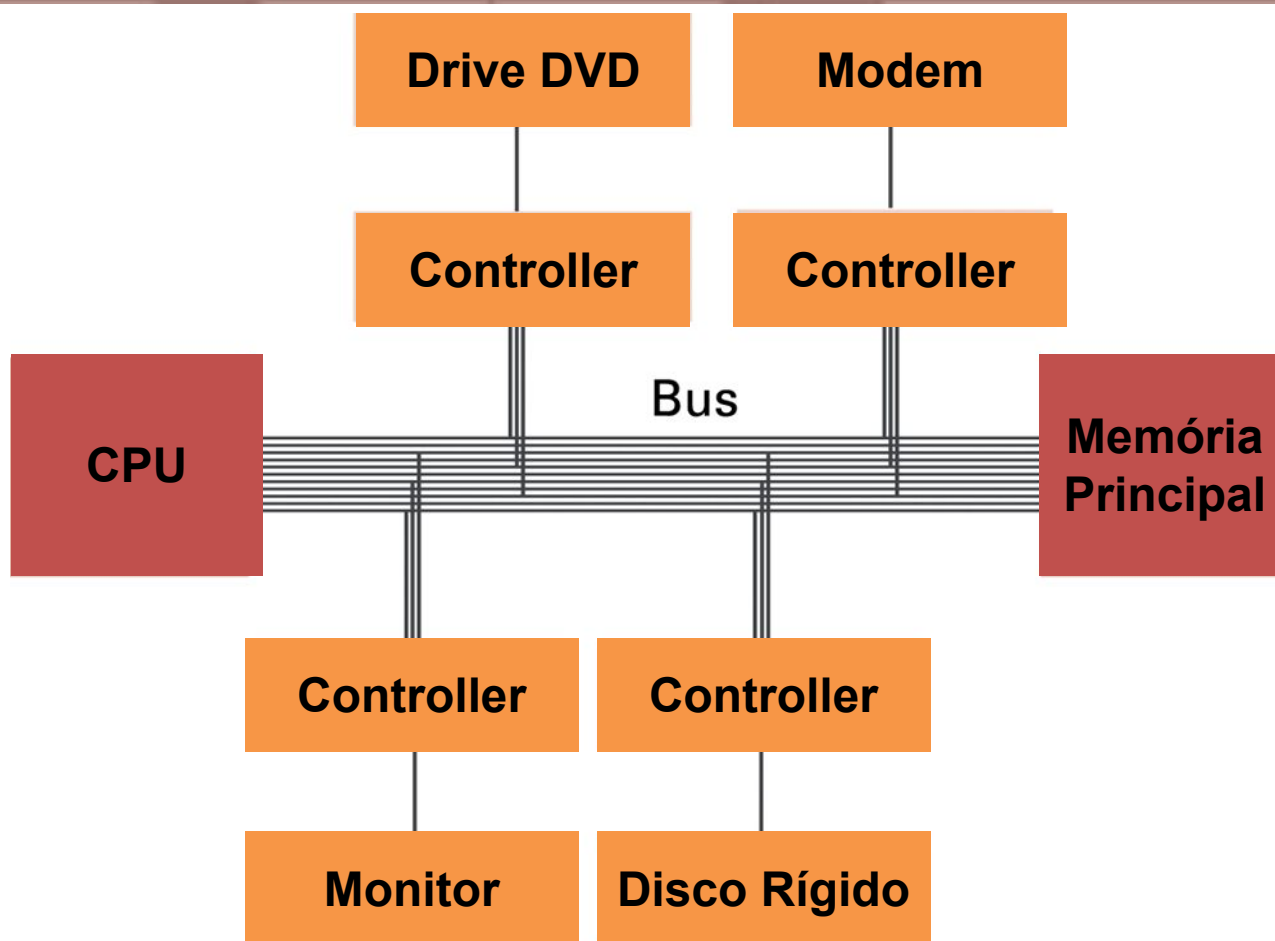


Comunicando com outros dispositivos

- *Controller*: aparelho intermediário que trata a comunicação entre o computador e o dispositivo
 - Especializado no tipo de dispositivo
 - De uso geral (ex: USB ou FireWire)
- *Port*: conjunto de endereços atribuídos a um aparelho.
 - *Memory-mapped I/O*: CPU lê ou escreve no endereço da porta do controlador

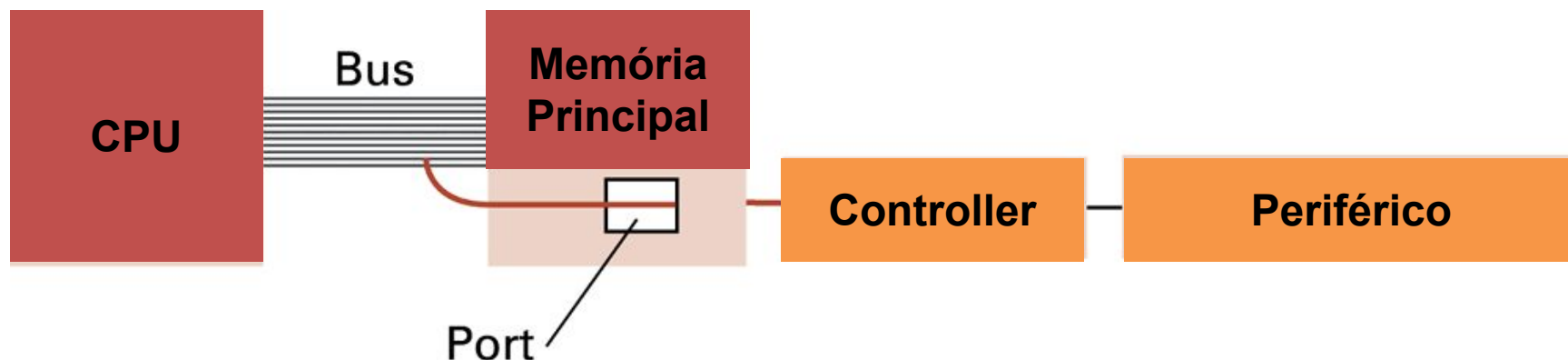


Controladores ligados ao bus da máquina





Representação conceptual de *memory-mapped I/O*





Comunicando com periféricos

- *Direct memory access (DMA)*:
 - acesso a memória principal por um controlador usando o *bus*
- *Von Neumann Bottleneck*:
 - coordenação das atividades entre CPU, controller e memória afeta a velocidade baixa do bus e degrada a performance
- *Handshaking*: o processo que controla a transferência de dados entre componentes ligados através do *bus*
 - *exemplo: impressora*



Terminologia sobre comunicação de dados

- **Comunicação em série:**
 - bits são transmitidos em sequência (1 de cada vez)
- **Comunicação em paralelo:**
 - várias linhas para transmitir bits em paralelo
- **Multiplexing:**
 - consiste em intervalar bits de dados, de forma a que diferentes dados possam ser transmitidos sobre um único meio de comunicação



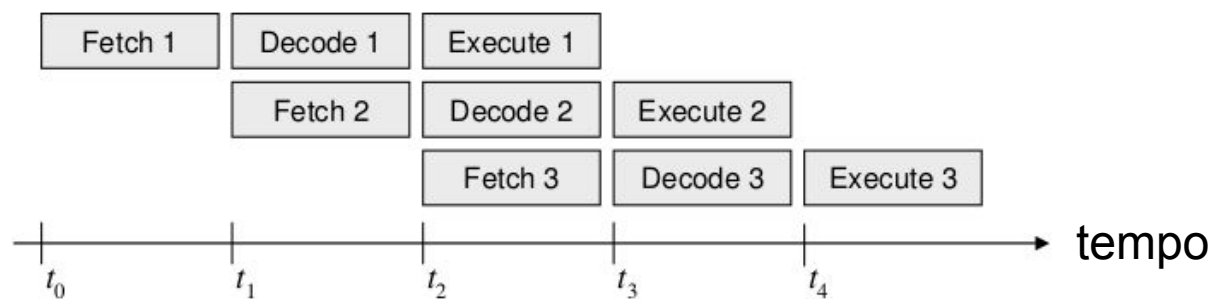
Débitos na comunicação de dados

- Unidades de medida
 - Bps = bits por segundo
 - kbps = kilo-bps, ou 1,000 bps
 - Mbps = mega-bps, ou 1,000,000 bps
 - Gbps = giga-bps, ou 1,000,000,000 bps
- Largura de banda (*Bandwidth*): débito máximo permitido



Outras arquiteturas

- Tecnologias para aumentar o “*throughput*” :
 - **Pipelining:** Sobrepor passos do ciclo da máquina



- **Processamento paralelo:** Usar múltiplos processadores simultaneamente
 - MIMD: Instruções diferentes em diferentes dados
 - SIMD: Instruções iguais em dados diferentes



Algumas perguntas

- O que significa **overclocking**?
 - Colocar o processador a funcionar a uma frequência superior à que é indicada pelo fabricante
 - Aumenta consumo de energia
 - Pode causar sobreaquecimento



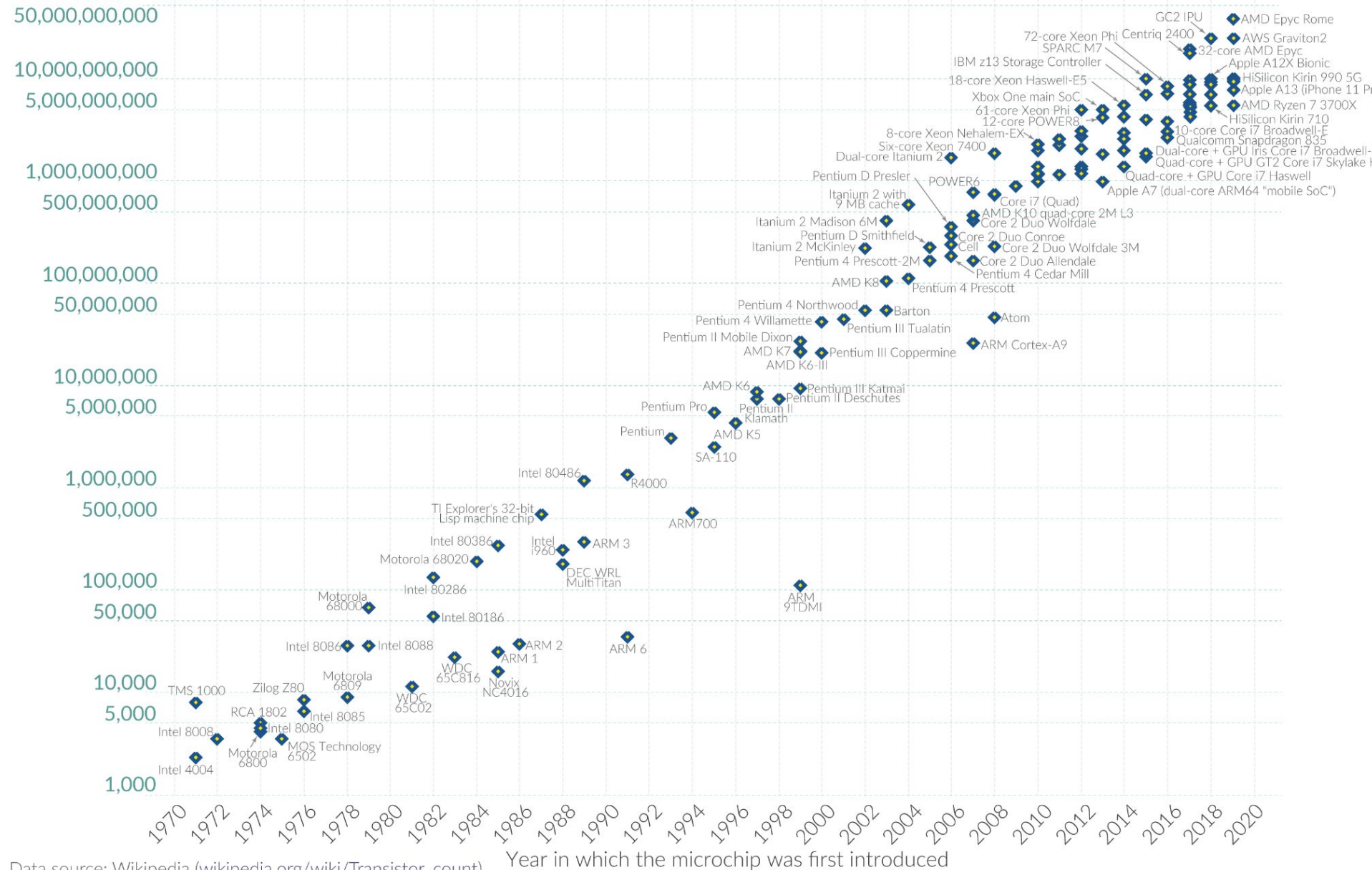
Algumas perguntas

- Quão rapidamente evolui a **velocidade de processamento** dos computadores?
 - O que é a “**lei de Moore**”?
 - Em 1975 Moore estimou que o número de transistors em circuitos integrados iria duplicar a cada 2 anos
 - Aumento na capacidade de processamento
 - A lei poderá verificar-se eternamente ?
 - Limites de miniaturização (nível atómico)

Our World
in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Transistor count



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Rose