

# SSC0902

# Organização e Arquitetura de Computadores

## **3ª Aula – Visão Geral e Conceitos Básicos**

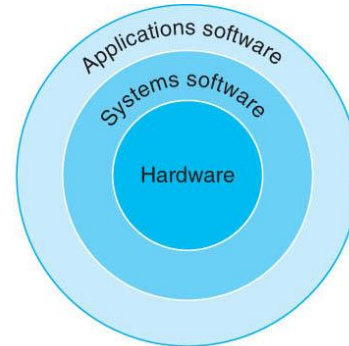
Profa. Sarita Mazzini Bruschi  
[sarita@icmc.usp.br](mailto:sarita@icmc.usp.br)

*Copyright William Stallings & Adrian J Pullin*

*Tradução, revisão e adaptação por Paulo S. L. de Souza*

# Arquitetura de von Neumann

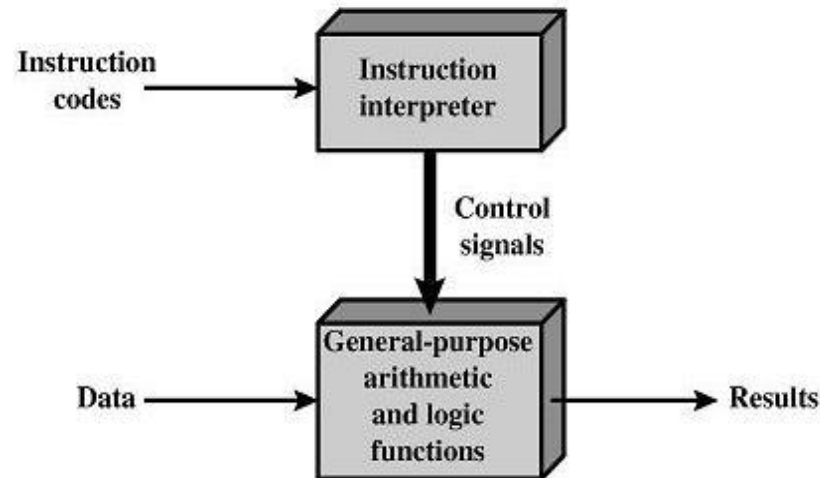
- A arquitetura proposta por von Neumann é composta por:
  - Conjunto de componentes lógicos básicos + programação
- Visão hierárquica do hardware e das camadas de software



- Três pontos importantes:
  - Dados e instruções são representados na memória;
  - A memória é endereçada pela posição e não pelo conteúdo;
  - A execução das instruções é considerada sequencial

# Hardware de propósito geral

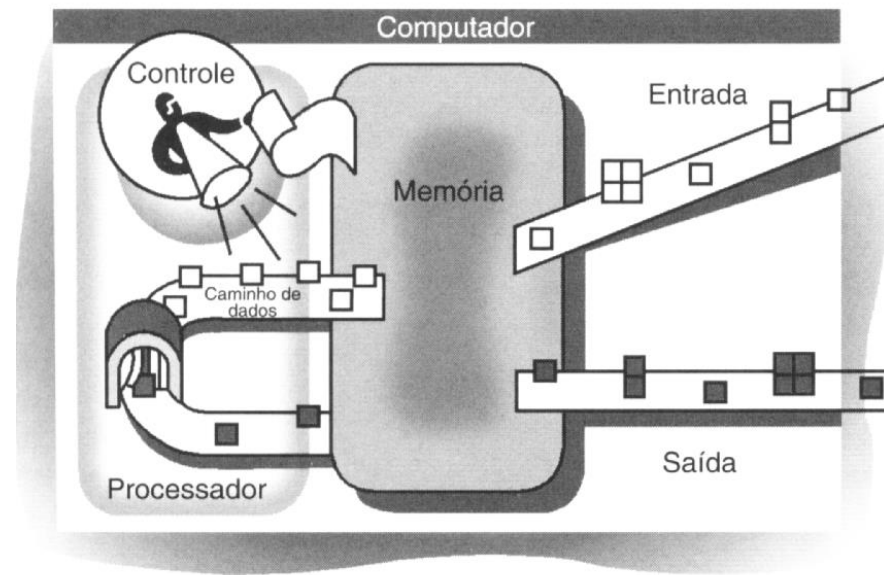
- Programa determina uma sequência de passos
  - cada passo faz uma operação aritmética ou lógica
  - cada operação requer sinais de controle diferentes



(b) Programming in software

# Função da Unidade de Controle

- Para cada operação, um código único é fornecido:
  - Exemplo: ADD, MOVE, etc
- Função da Unidade de Controle:
  - interpretar o código e gerar os sinais de controle que executarão a instrução requerida



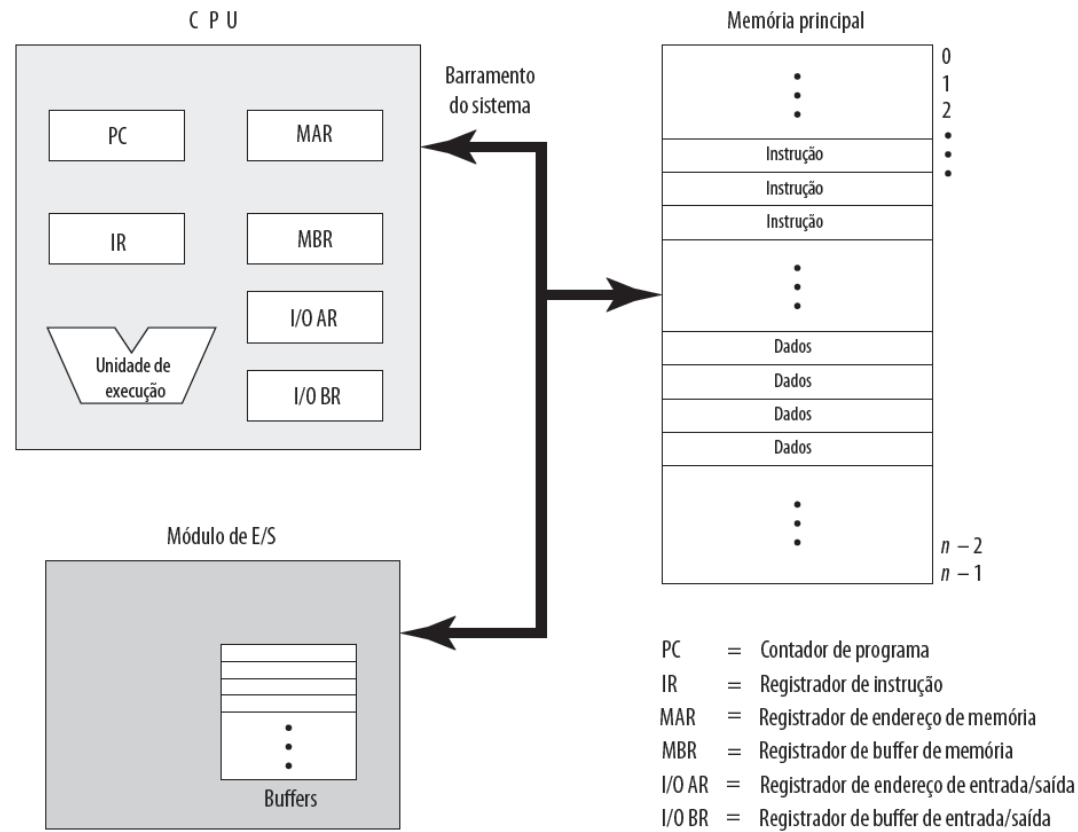
- **Com isso tem-se um computador!!**

# Componentes

- **CPU** (Unidade de Controle + Unidade Lógica e Aritmética)
  - Patterson & Hennessy => UC + Caminho de Dados (barramento interno + unidades funcionais)
- **E/S** – comunicação com o mundo exterior
  - permite inserir dados e instruções no computador
  - permite enviar resultados para fora do computador
- **Memória principal** - responsável por:
  - fornecer instruções e operandos para execução
  - armazenar resultados fora da CPU

# Componentes do computador: visão glogal

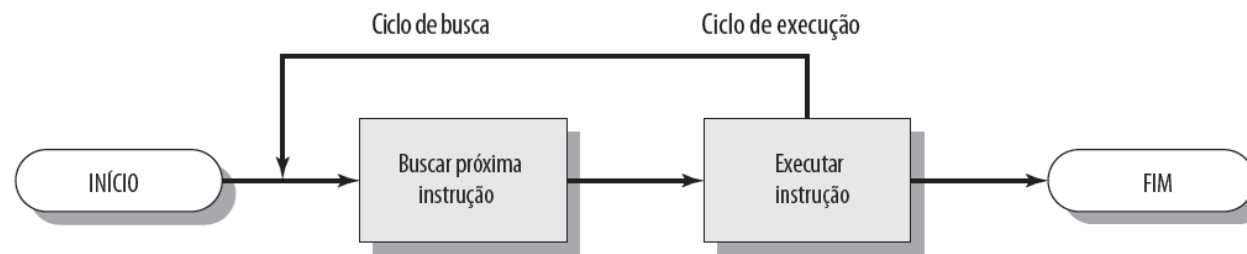
**Figura 3.2** Componentes do computador: visão de alto nível



# Ciclo de Instrução

- Dois passos:
  - Ciclo de Busca (*Fetch*)
  - Ciclo de Execução (*Execute*)

**Figura 3.3** Ciclo de instrução básico



# Ciclo de busca

- No contador de programas (PC) estará o endereço da próxima instrução a ser executada
  - Não conta programa algum, aponta para a próxima instrução
- O processador faz a busca da instrução na posição de memória que está armazenada no PC
  - $MAR = PC$
  - $MBR = \text{memória}(MAR)$
- O PC é incrementado
  - $PC = PC + 1$ 
    - incrementa 1 se palavra; incrementa N se N Bytes formam a palavra E o endereçamento é a Byte
  - A não ser que a próxima instrução não esteja armazenada na posição seguinte (instruções de desvio)
- A instrução é armazenada no registrador de instrução (IR)
  - $IR = MBR$



# Ciclo de execução

- UC decodifica a instrução e determina quais ações são necessárias, ou seja, a UC configura os sinais de controle de acordo com a instrução
- A execução da instrução se resume a uma das seguintes possibilidades:
  - **Processador-Memória:** transferência de dados do processador para a memória ou da memória para o processador
  - **Processador-E/S:** transferência de dados entre o processador e um dispositivo de E/S
  - **Processamento de dados:** execução de operações aritméticas ou lógicas sobre os dados
  - **Controle:** especifica que a sequência de execução de instruções seja alterada
  - **Combinação dessas 4 possibilidades**

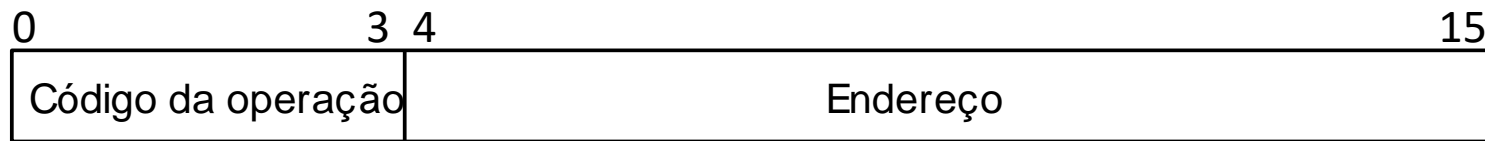
# Visão de uma Máquina Hipotética

# Características de uma máquina hipotética

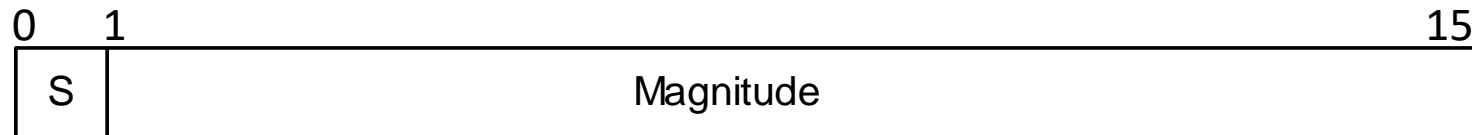
- Registradores:
  - Contador de programa (PC) = endereço da próxima instrução
  - Registrador de instrução (IR) = instrução que está sendo executada
  - Acumulador (AC) = armazenamento temporário de dados
- Códigos de operações:
  - 0001 = carregar AC a partir do endereço de memória especificado
    - $(AC) \leftarrow (mem)$
  - 0010 = armazenar o valor contido em AC no endereço de memória especificado
    - $(mem) \leftarrow (AC)$
  - 0101 = acrescentar ao valor contido em AC o valor contido no endereço de memória especificado
    - $(AC) \leftarrow (AC) + (mem)$

# Características de uma máquina hipotética

- Formato de instruções



- Formato de números inteiros



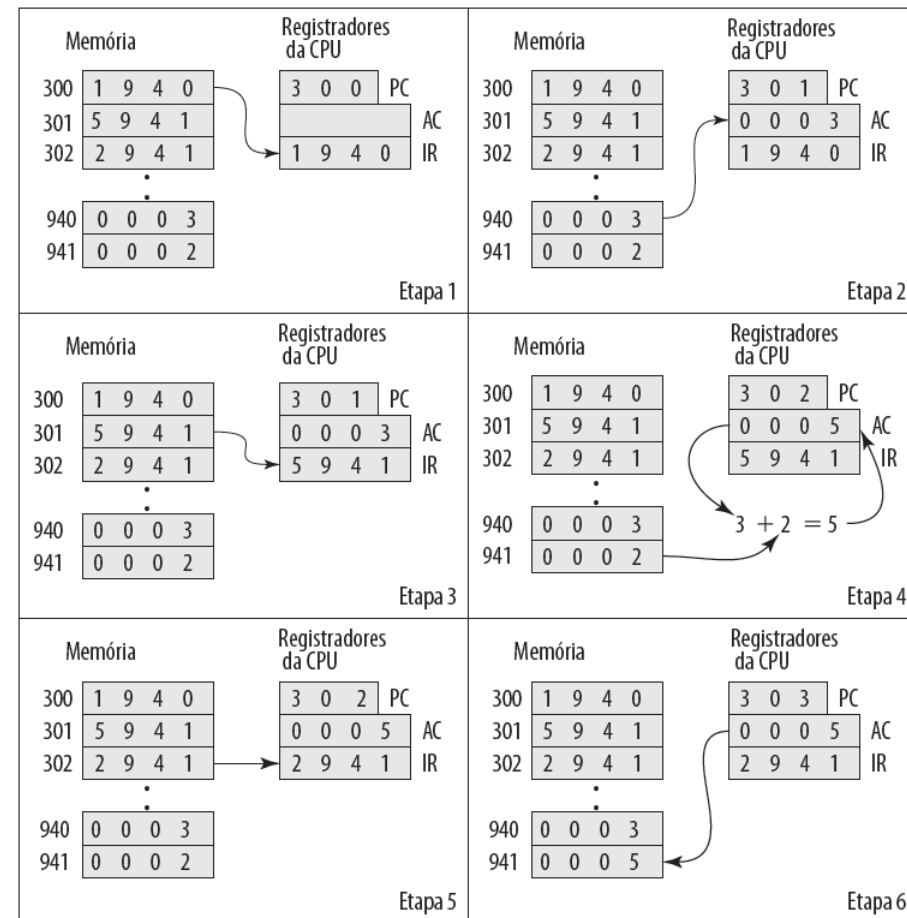
# Exemplo da execução parcial de um programa

**Figura 3.5** Exemplo de execução de programa (conteúdo da memória e dos registradores em hexadecimal)

Soma de dois números  
representados com  
sinal-magnitude

*3 ciclos de  
busca e  
execução*

**0001 - (AC) ← (mem)**  
**0010 - (mem) ← (AC)**  
**0101 - (AC) ← (AC) + (mem)**



# Ciclo de Instrução

## Diagrama de estados

- Estados e transições
  - parte superior: transferência de valores
  - parte inferior: atividade feita na CPU

**Figura 3.6** Diagrama de estado do ciclo de instrução

