

SSC510 – Arquitetura de Computadores

1ª AULA

REVISÃO DE ORGANIZAÇÃO DE COMPUTADORES

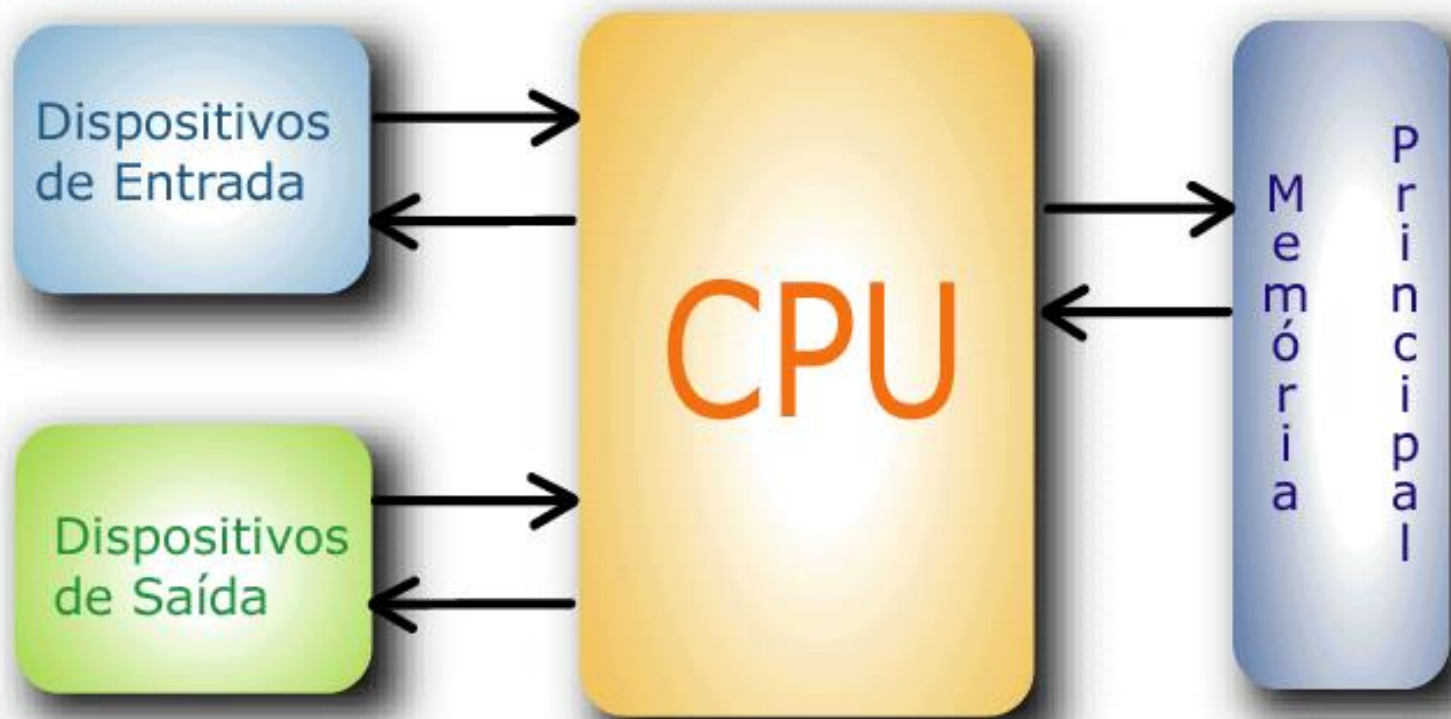
Arquitetura X Organização

Arquitetura - Atributos de um Sistema Computacional como visto pelo programador, isto é a estrutura e o comportamento funcional da máquina

Organização - organização dos fluxos de dados, projeto de controle lógico e a implementação física.

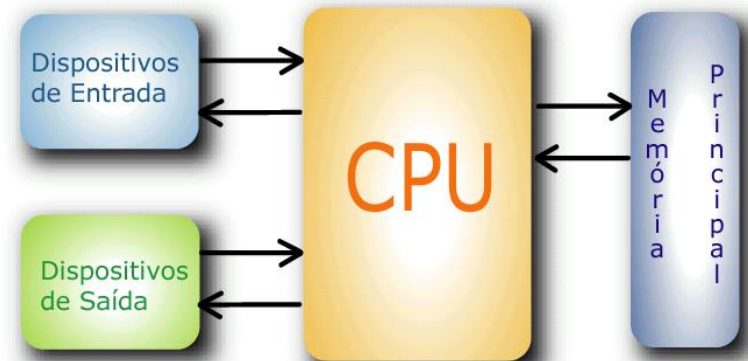
Amdahl, Blaaw, and Brooks, 1964

Estrutura da máquina de von Neumann



Estrutura da máquina de von Neumann

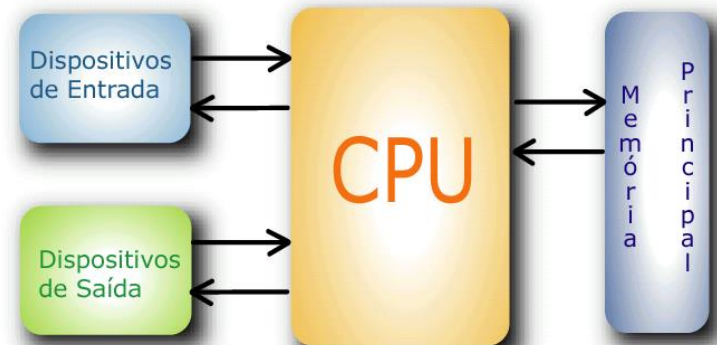
1. Memória Principal – Armazena dados e instruções
2. Dispositivos de Entrada e Saída – Comunicação com o mundo exterior
3. CPU – Processamento
4. Comunicação entre os blocos



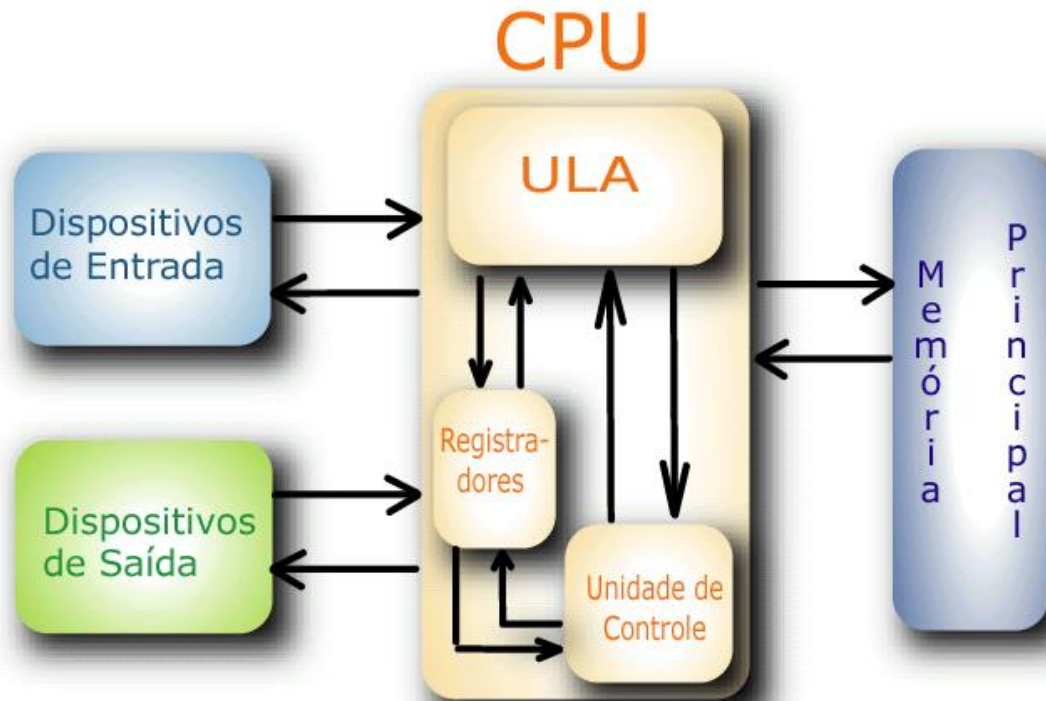
Estrutura da máquina de von Neumann

Execução de um programa:

1. Ler um Programa e Armazenar na Memória
2. Transformar programa de Linguagem de Alto Nível em Linguagem de Máquina – conjunto de instruções
3. Executar
4. Enviar Resultados para o mundo exterior



Estrutura da máquina de von Neumann



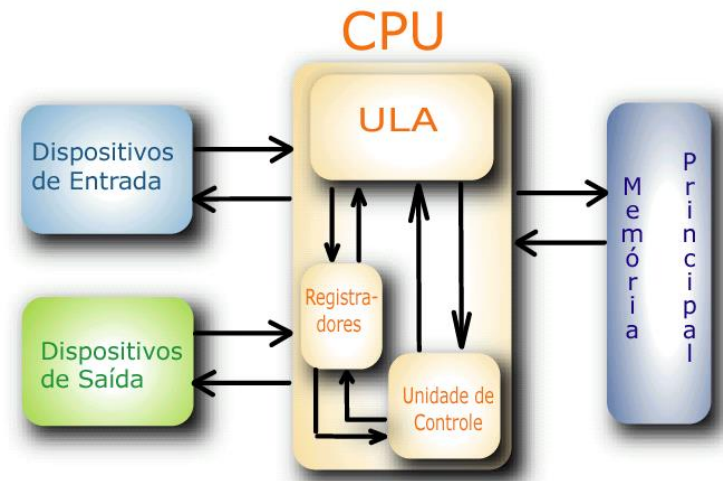
Estrutura da máquina de von Neumann

Passos mais detalhados:

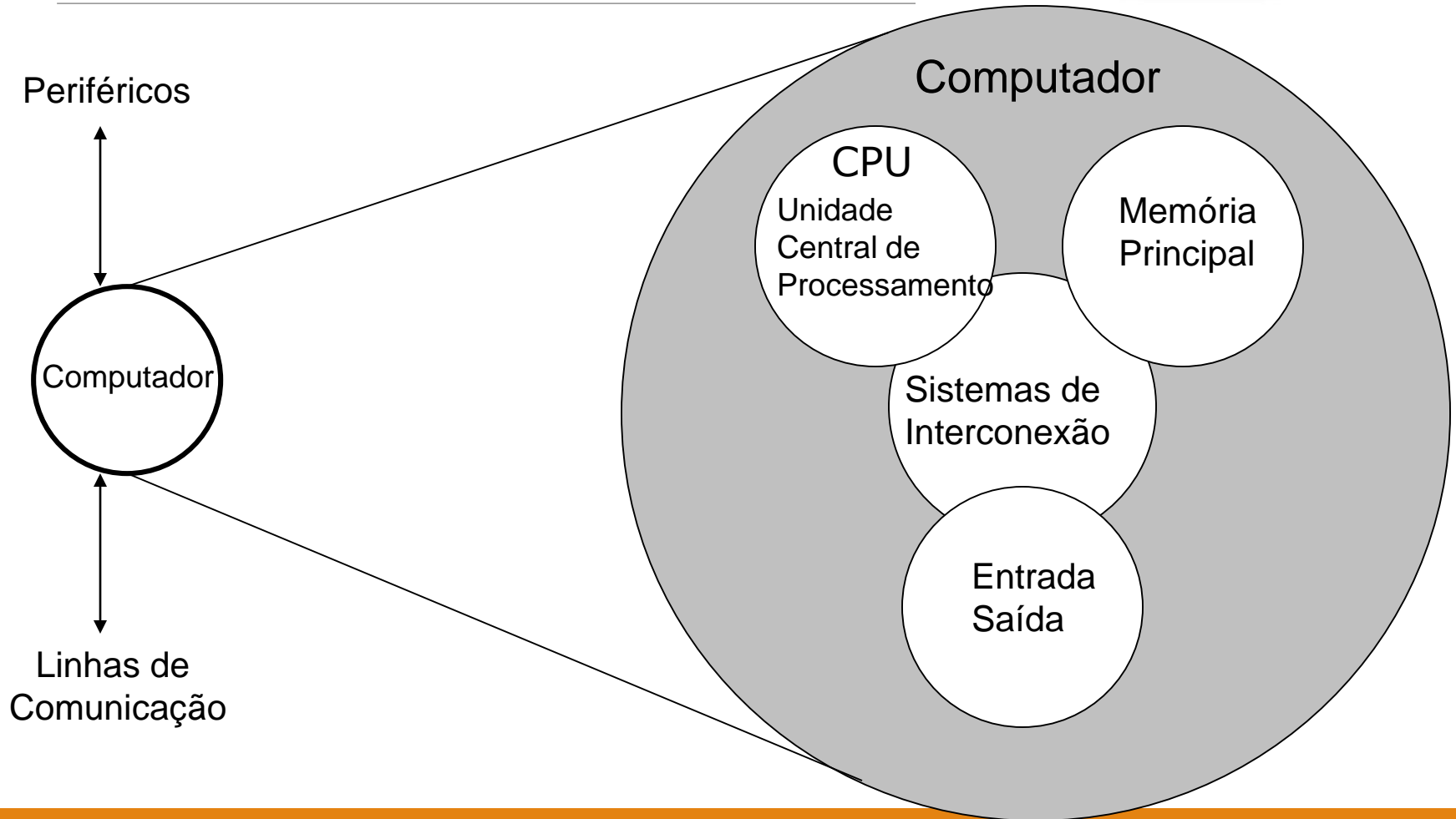
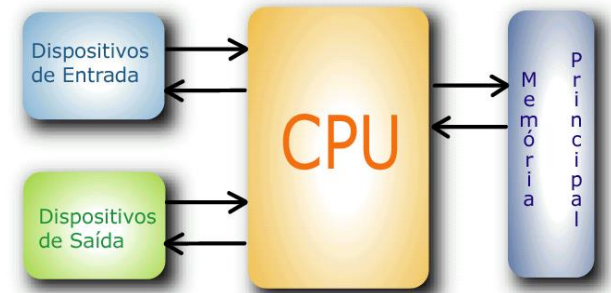
- Ler instrução da memória
- Decodificar a instrução
- Unidade de Controle gera sinais elétricos para execução
- ULA executa operações
- Registradores – armazenam os valores intermediários

Necessidade de interconexões

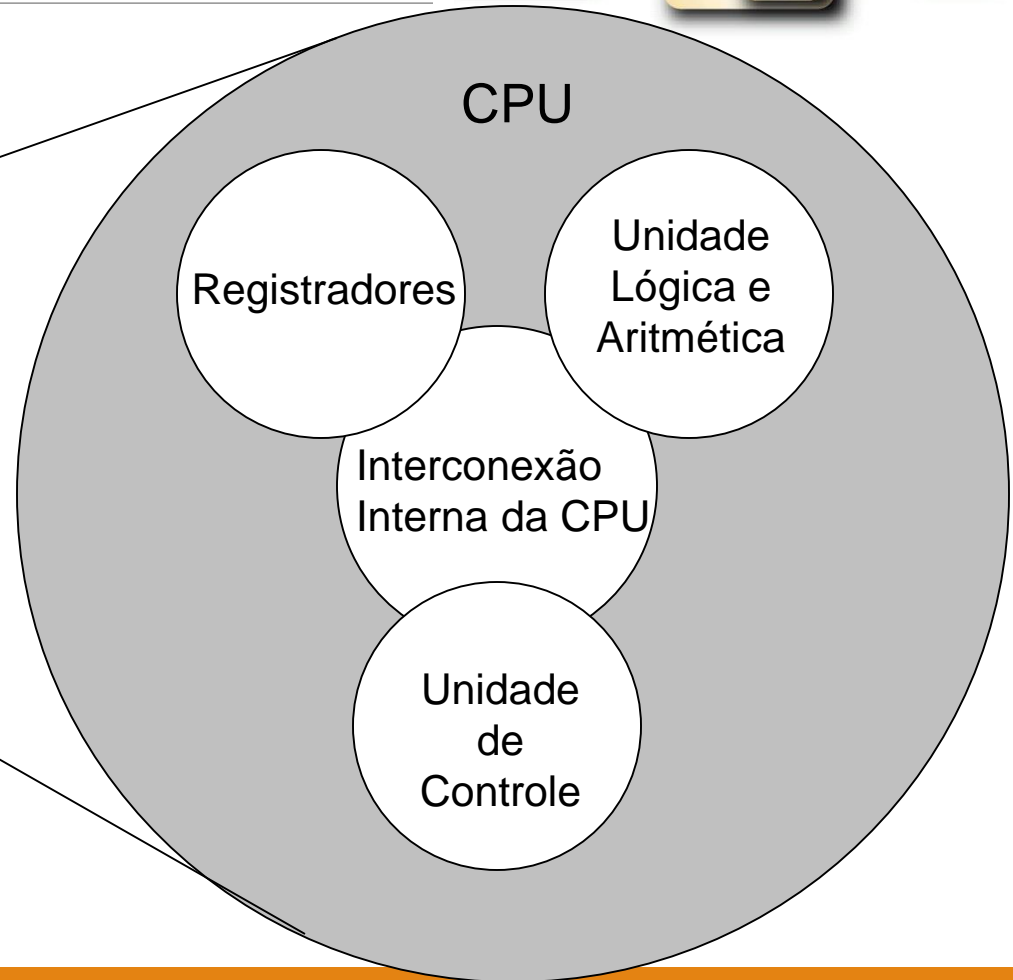
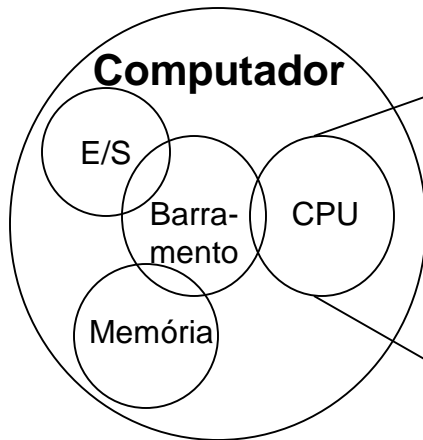
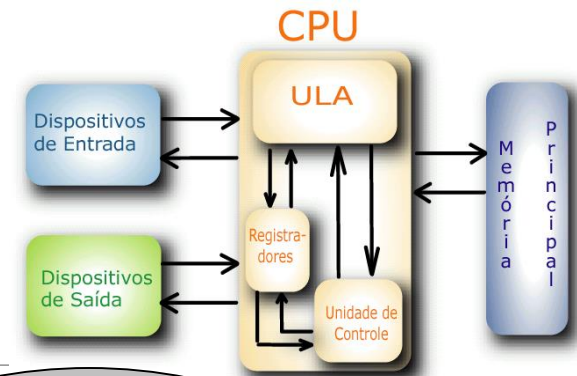
Exemplo: $c = a + b$



Estrutura

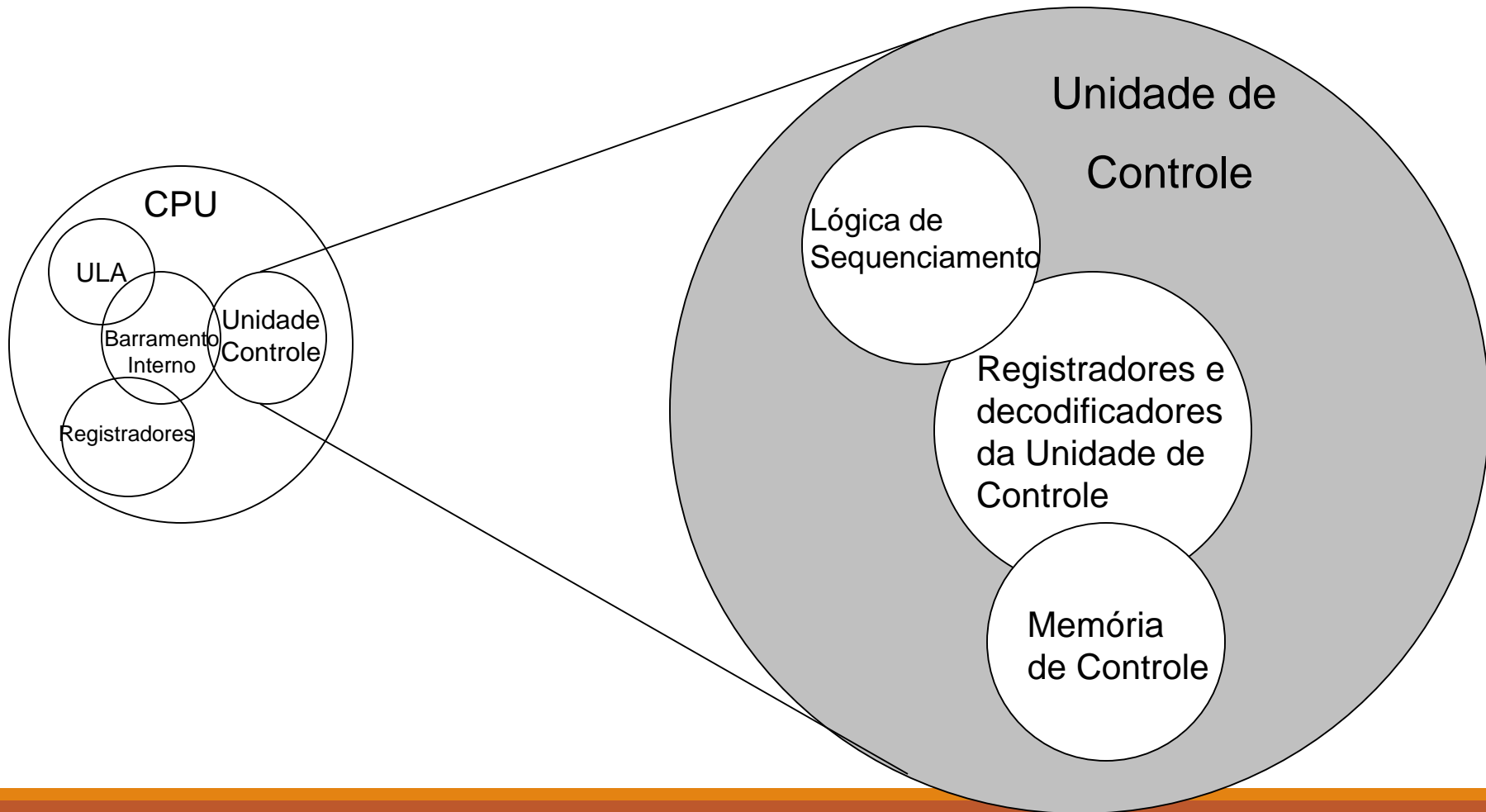


Estrutura CPU



Estrutura

Unidade de Controle



Conceitos de Programa

Sistemas em que o hardware é dedicado para uma aplicação particular não são flexíveis

Sistemas de propósito geral podem executar diferentes tarefas através dos sinais de controle

Ao invés de se reprogramar o hardware, muda-se o conjunto de sinais de controle

Programa

Seqüência de passos

Para cada passo

- uma operação lógica ou aritmética é realizada
- um conjunto diferente de sinais deve ser fornecido

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

Instruções

A arquitetura de Von Neumann é uma arquitetura que a memória pode ser usada para armazenar tanto os dados como as instruções

Palavras (= conjunto de bits) → podem ter diferentes significados agrupados em: dados e instruções

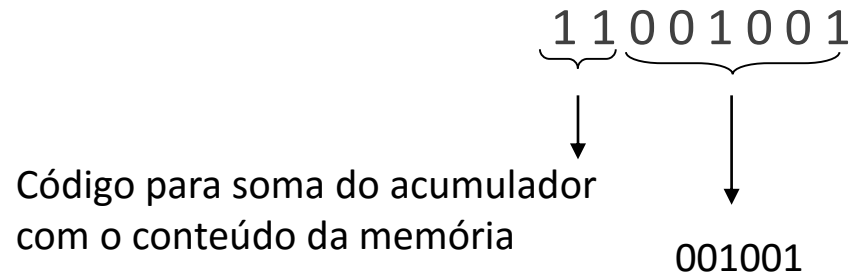
Exemplo

Palavra de 8 bits:

11001001

Qual o significado?

- Pode ser um dado binário $\rightarrow (201)_{10}$
- Pode ser um dado hexadecimal $\rightarrow (C9)_H$
- Pode ser um dado ASCII $\rightarrow 'I'$
- Pode ser uma instrução do tipo:



Instruções

Contêm as informações que o computador necessita para executar as várias operações

Composta de:

- Código: o que deve ser feito
- Referência aos operandos: sobre que dados
- Referência aos operandos para os resultados: onde colocar o resultado

Instruções

Operandos:

- Memória Principal
- Registradores
- Dispositivos de E/S

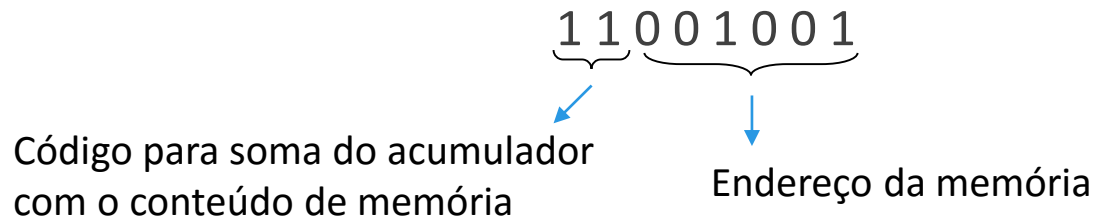
Representação

- Código de Máquina – número binário
- Usuário – Símbolo

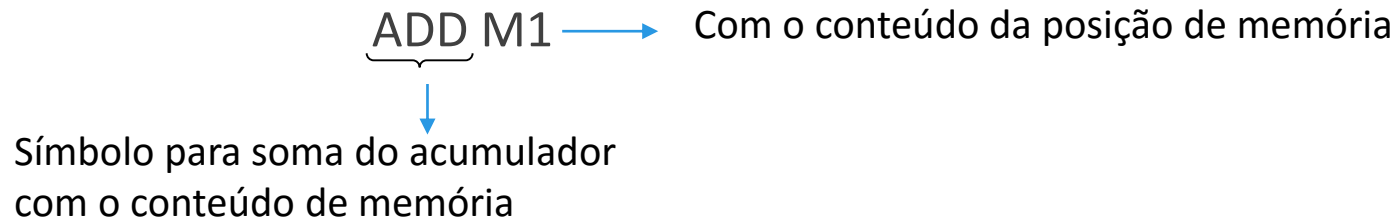
Exemplo

Qual o significado?

- Pode ser uma instrução do tipo:



- Pode ser representado como:



Instruções

Cada máquina possui um conjunto de instruções

- Coleção completa de instruções que será entendida pela CPU
- Código da Máquina

O que define o significado do conteúdo de uma dada palavra é o contexto em que ela é empregada.

**Isto caracteriza o fundamento da
Arquitetura de von Neumann.**

Componentes

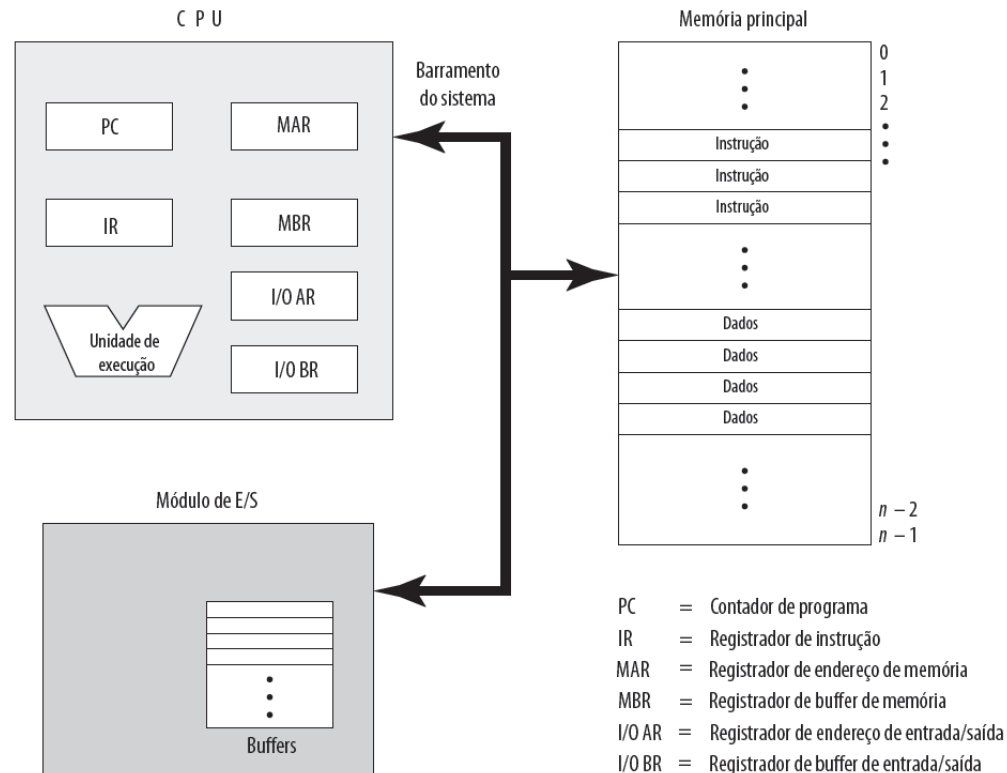
Como agregar dispositivos básicos para se obter, no final, uma CPU funcionando adequadamente, segundo um conjunto de instruções

Dispositivos Básicos:

- ULA – Unidade Lógica-Aritmética
- Registradores
- Unidade de Controle
- *Conjunto de Instruções*

Componentes

Figura 3.2 Componentes do computador: visão de alto nível

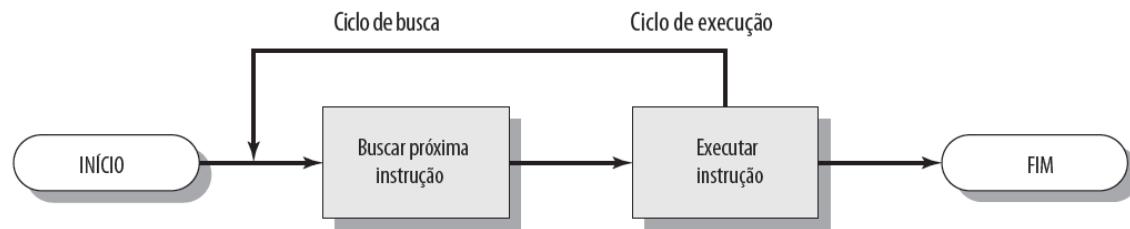


Ciclo de Instrução

Dois passos:

- Busca (Fetch)
- 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

O processador faz a busca da instrução na posição de memória que está armazenada no PC

O PC é incrementado

- 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)

O processador interpreta a instrução

Ciclo de Execução

Com o código da instrução armazenado no IR, 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 seqüência 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

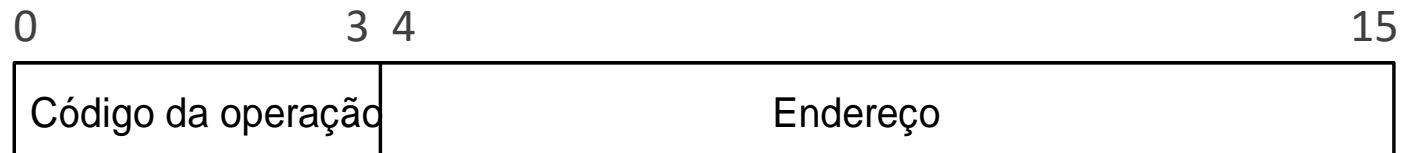
Características de uma máquina hipotética

Códigos de operações:

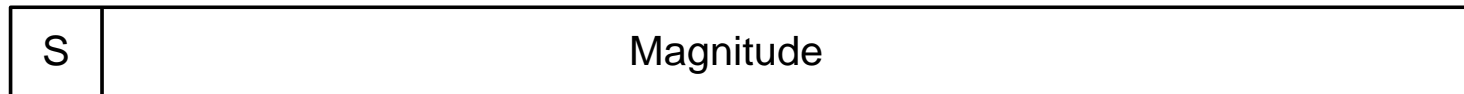
- 0001 = carregar AC a partir do endereço de memória especificado – $(AC) \leftarrow (mem)$
- 0010 = armazenar o valor contido em AC no endereço de memória especificado $(mem) \leftarrow (AC)$
- 0101 = acrescentar ao valor contido em AC o valor contido no endereço de memória especificado $(AC) \leftarrow (AC) + (mem)$

Características de uma máquina hipotética

Formato de instruções



Formato de números inteiros



Exemplo da execução parcial de um programa

Soma de

dois números

3 ciclos de

busca e

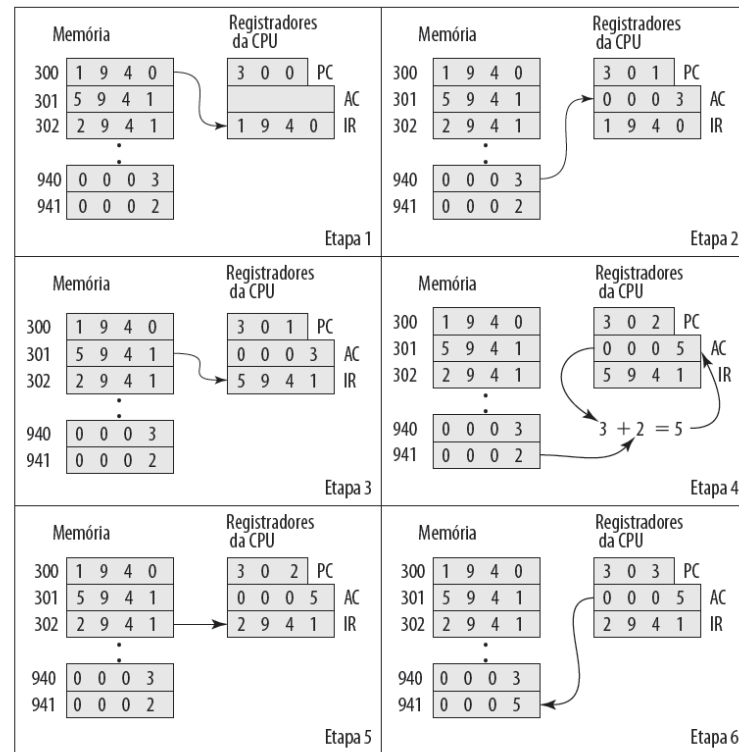
execução

0001 - (AC) \leftarrow (mem)

0010 - (mem) \leftarrow (AC)

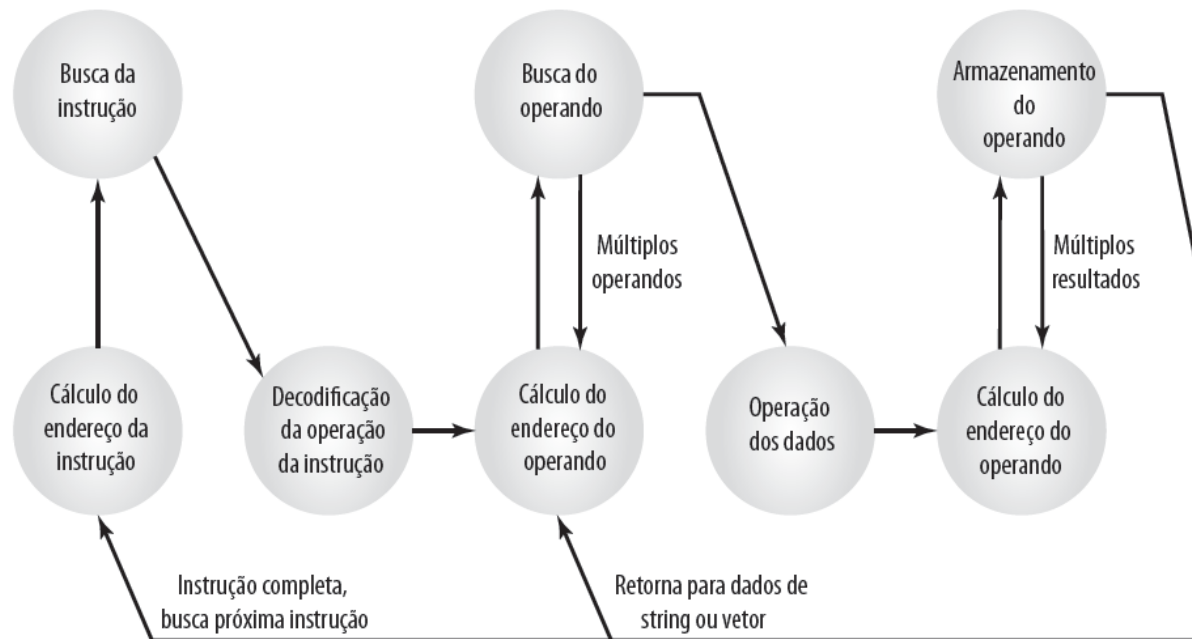
0101 - (AC) \leftarrow (AC) + (mem)

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



Ciclo de Instrução – Diagrama de estados

Figura 3.6 Diagrama de estado do ciclo de instrução



Interrupções

Mecanismo pelo qual outros módulos (por exemplo, E/S) podem interromper a sequência normal de processamento

Podem ser:

- Software: resultado da execução de alguma instrução, por exemplo, divisão por zero, overflow, etc
- Clock: gerada pelo relógio interno do processador, permitindo operações regulares
- E/S: gerada por controladores de E/S para indicar a conclusão de uma operação ou a ocorrência de uma situação de erro
- Falha de hardware: exemplo: queda de energia e erro de paridade da memória

Ciclo de Interrupção

Este ciclo é adicionado ao ciclo de instrução

O processador verifica constantemente a chegada de uma interrupção, indicada por um *signal de interrupção*

Se não existe interrupção, busca a próxima instrução

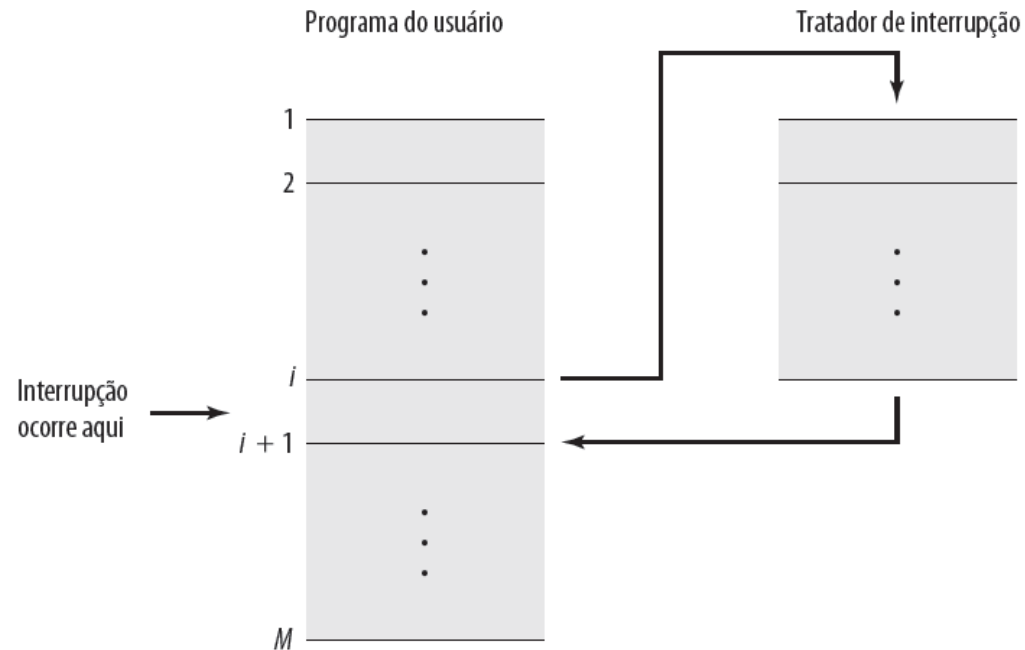
Ciclo de Interrupção

Se existe uma interrupção:

- A execução do programa corrente é suspensa
- O contexto é salvo
- O PC é carregado com o endereço da rotina que trata a interrupção
- A interrupção é processada
- O contexto é restaurado e o programa continua sua execução

Ciclo de Interrupção

Figura 3.8 Transferência de controle via interrupções



Ciclo de instrução (com interrupções)

Figura 3.12 Diagrama de estado do ciclo de instruções, com interrupções

