

# 1. Introdução

- ◆ O processador é o componente vital do sistema de computação, responsável pela realização das operações de processamento (os cálculos matemáticos etc.) e de controle, durante a execução de um programa.
- ◆ Um programa, para ser efetivamente executado pelo processador, deve ser constituído de uma série de instruções de máquina. Para que a execução tenha início, as instruções devem ser armazenadas em células sucessivas, na memória principal.

## 2. A influência do tamanho da palavra

- ◆ A capacidade de processamento de uma CPU (a velocidade com que realiza o ciclo de uma instrução) é em grande parte determinada pelas facilidades embutidas no hardware da UAL para realizar as operações matemáticas projetadas.
- ◆ Um dos elementos fundamentais para isso é a definição do tamanho da palavra da CPU. O valor escolhido no projeto de fabricação da CPU determinará o tamanho dos elementos ligados à área de processamento, entre estes, a UAL.
- ◆ Um tamanho maior ou menor de palavra (e, por conseguinte, da UAL, dos registradores de dados, do barramento interno) acarreta, sem dúvida, diferenças fundamentais de desempenho da CPU.

◆ No projeto de uma CPU, a definição do tamanho da palavra tem enorme influência no desempenho global de toda a CPU e, por conseguinte, do sistema como um todo:

a) influência no desempenho devido ao maior ou menor tempo na execução de instruções com operações matemáticas na UAL.

b) influência no desempenho devido ao tamanho escolhido para o barramento interno e externo da CPU. Se a largura do barramento for, p.ex., igual a 16 bits em um sistema com palavra de 32 bits (UAL e registradores de 32 bits), então o movimento de 4 bytes de um dado tipo caractere requererá dois ciclos de tempo do barramento, ao passo que em barramento de 32 bits requereria apenas um ciclo de tempo;

c) influência também na implementação física do acesso à memória, porque o movimento de dados entre CPU e memória é normalmente medido em palavras (o barramento de dados que une o RDM à memória deve acompanhar em largura o valor da palavra). Para uma CPU de 32 bits de palavra, por exemplo, é desejável que a memória seja organizada de modo que sejam acessadas 4 células contíguas (4 bytes = 32 bits) em um único ciclo de memória. Se isto não ocorrer, a CPU deverá ficar em estado de espera ("wait state").

# 3. Funções da CPU

- A CPU é projetada e fabricada com o propósito único de executar sucessivamente pequenas operações matemáticas (ou outras manipulações simples com dados), na ordem e na sequência definidas pela organização do programa.
- As atividades realizadas pela CPU podem ser divididas em duas grandes categorias funcionais:
  - Função processamento;
  - Função controle.

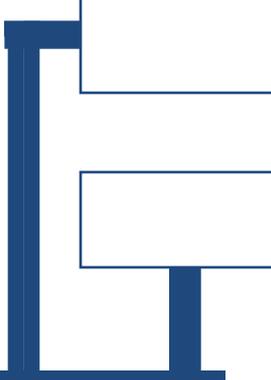
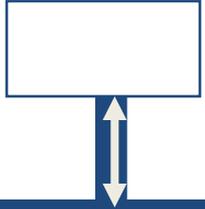
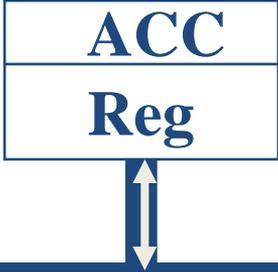
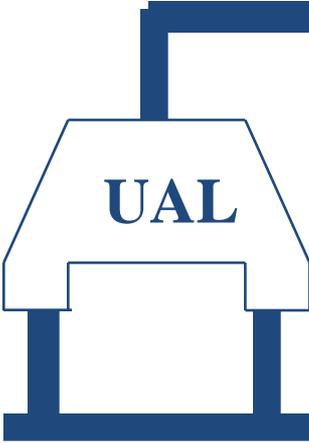
## 3.1 - Função de processamento

- A função processamento se encarrega de realizar as atividades relacionadas com a efetiva execução de uma operação, ou seja, processar.
- Processar o dado é executar com ele uma ação que produza algum tipo de resultado.

- Tarefas comuns a esta função são as que realizam:
  - Operações aritméticas (somar, subtrair, multiplicar, dividir);
  - Operações lógicas (and, or, xor, etc);
  - Movimentação de dados (memória - CPU, CPU - memória, registrador - registrador etc);
  - Desvios (alteração de sequência de execução de instruções);
  - Operações de entrada ou saída.

- Fazem parte dessa área funcional os seguintes dispositivos:
  - Unidade Aritmética e Lógica (UAL);
  - Acumulador (ACC);
  - Registradores.

U C P



Barramento de dados



Barramento de endereços



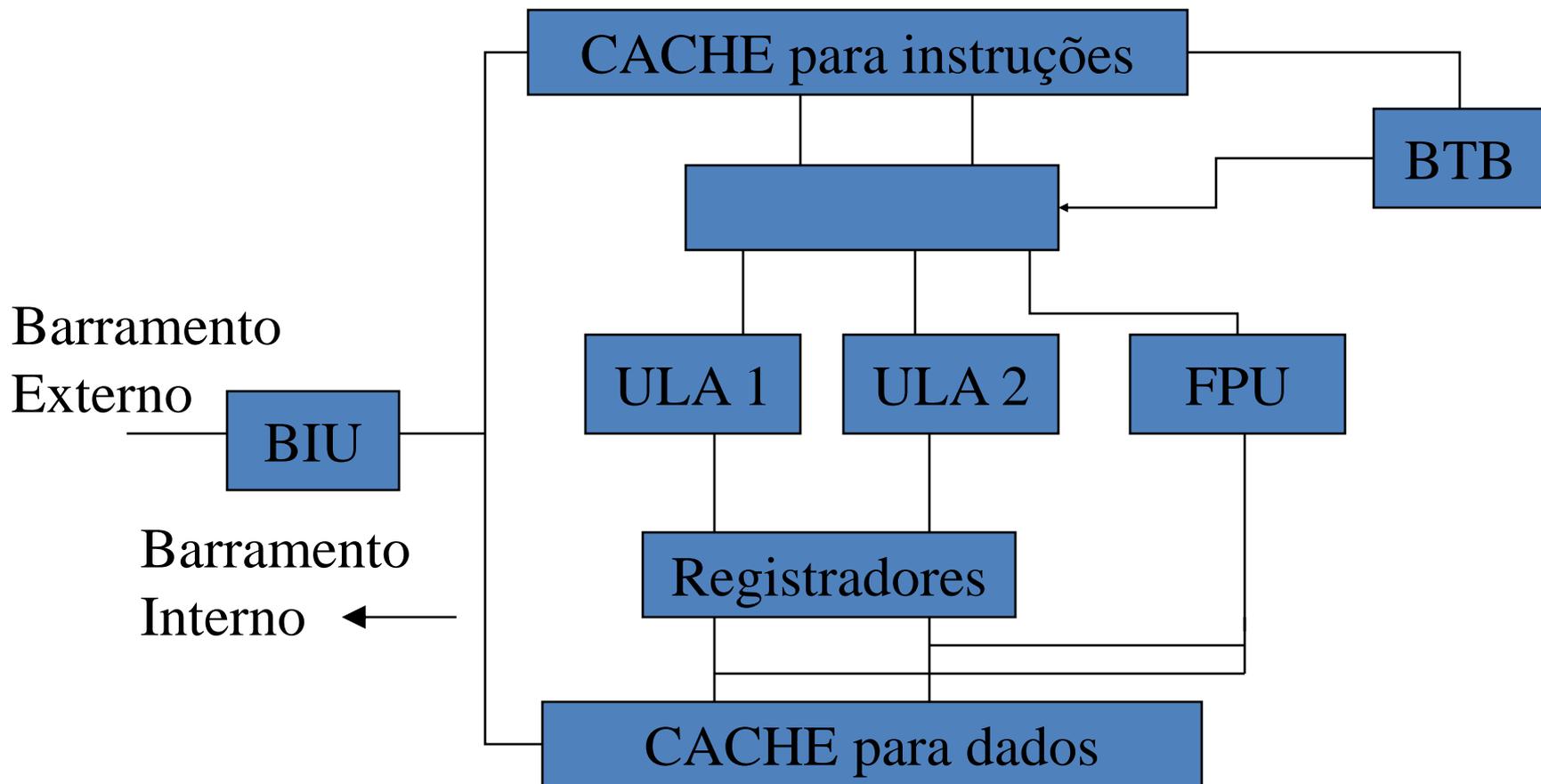
### 3.1.1 - Unidade Aritmética e Lógica – UAL (ou ULA)

- ◆ A UAL é o dispositivo da CPU que executa realmente as operações matemáticas com os dados. Tais operações podem ser:
  - soma, multiplicação, subtração e divisão;
  - op lógica AND, OR e XOR;
  - deslocamentos à direita e à esquerda;
  - incremento e decremento;
  - op complemento.

- ◆ Qualquer UAL é um aglomerado de circuitos lógicos e componentes eletrônicos simples que, integrados, realizam as operações já mencionadas.
- ◆ Essas operações são sempre realizadas sobre duas palavras da máquina.
- ◆ Em geral, esse circuito que trabalha com palavras de  $n$  bits é construído a partir de  $n$  circuitos idênticos, cada qual responsável por uma posição individual de bits.

- ◆ A seqüência a seguir, mostra a construção de uma ULA que pode calcular uma de quatro funções a saber:
- ◆ A and B;
- ◆ A or B;
- ◆  $\overline{B}$ ;
- ◆  $A + B$ .

Os processadores mais modernos utilizam em sua arquitetura mais de uma UAL, de modo a tornar a execução das instruções mais rápida.



BIU = Interface do barramento

Cache para instruções e dados = 8 KB

UAL 1 e 2 = n° inteiros

FPU = n° fracionários

## 3.1.2 – Registradores

- ◆ Para que um dado possa ser transferido para a UAL, é necessário que ele permaneça, mesmo que por um breve instante, armazenado em um registrador (a memória da CPU).
- ◆ O resultado de uma operação aritmética ou lógica realizada na UAL é armazenado temporariamente, para que possa ser reutilizado (por outra instrução) ou apenas para ser transferido para a memória.
- ◆ Para atender a estes propósitos, a CPU é fabricada com uma certa quantidade de registradores, destinados ao armazenamento de dados.
- ◆ O número de registradores e suas funções varia de processador para processador e é um dos maiores problemas no projeto de uma CPU.

- ◆ Há sistemas nos quais um desses registradores, denominado acumulador (abrevia-se, em inglês, ACC), além de armazenar dados, serve de elemento de ligação da ULA com os outros dispositivos da CPU.
- ◆ Os registradores da CPU são divididos em 2 tipos:
- ◆ Registradores visíveis para o usuário
  - Permite ao usuário minimizar referências à memória através do uso deles.
- ◆ Registradores de controle e de estado
  - Utilizados pela UC para controlar operações da CPU e programas privilegiados do sistema operacional para controlar a execução de programas.

### 3.1.2.1. Registradores visíveis para o usuário

Categorias:

- a. Registradores de propósito geral
- b. Registradores de dados
- c. Registradores de endereço (segmento, apontador de pilha e de índice)
- d. Registradores de código de condição (*flags*)

## 3.1.2.2. Registradores de Controle e de Estado

### Categorias:

#### a. Contador de Programa

- Contém o endereço da instrução a ser buscada.

#### b. Registrador de Instrução

- Contém a última instrução a ser buscada.

#### c. Registradores de endereçamento à memória

- Contém o endereço de uma posição da memória.

#### d. Registradores de armazenamento temporário de dados.

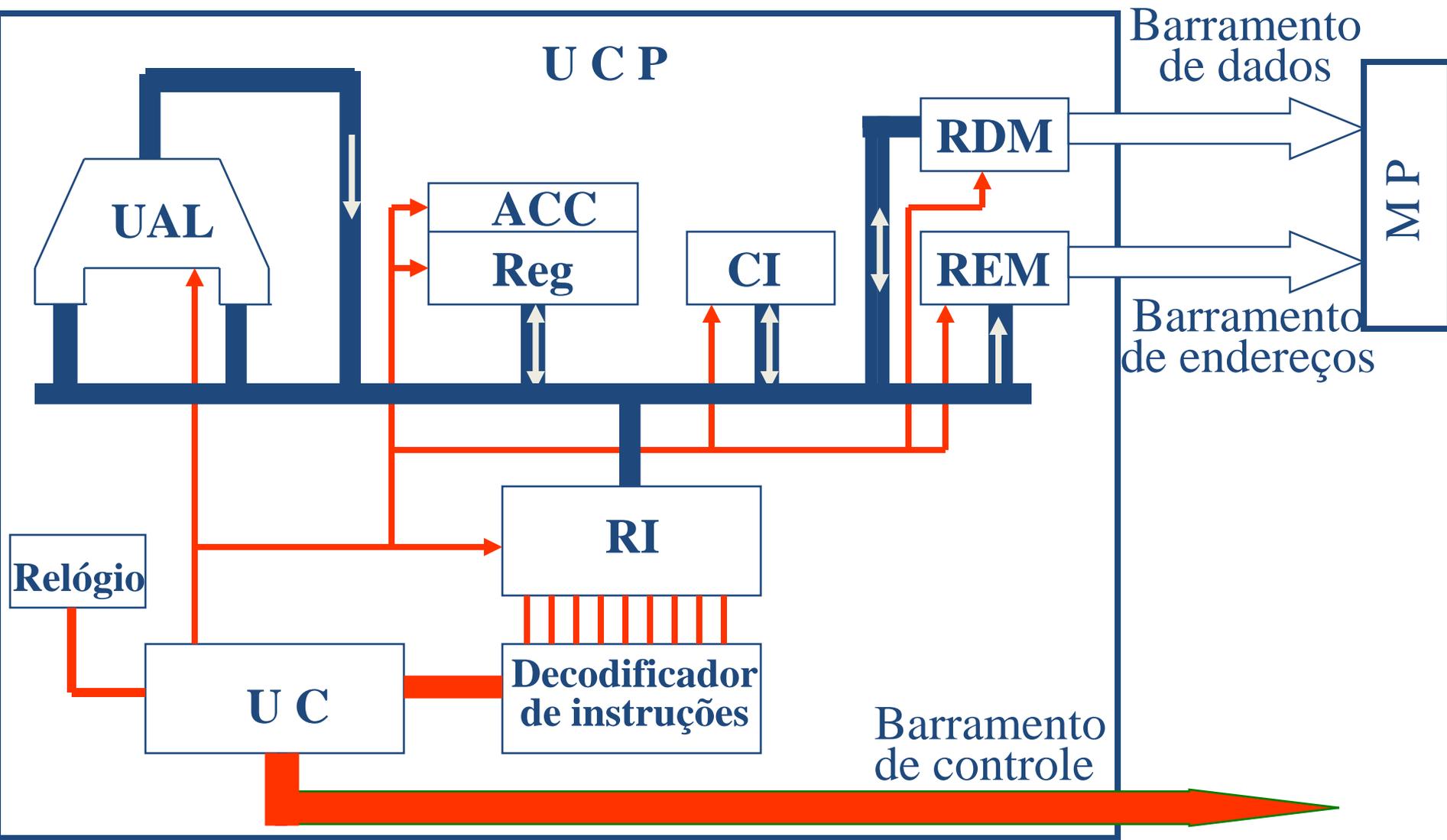
- Contém uma palavra de dados a ser escrita na memória ou a palavra lida mais recentemente.

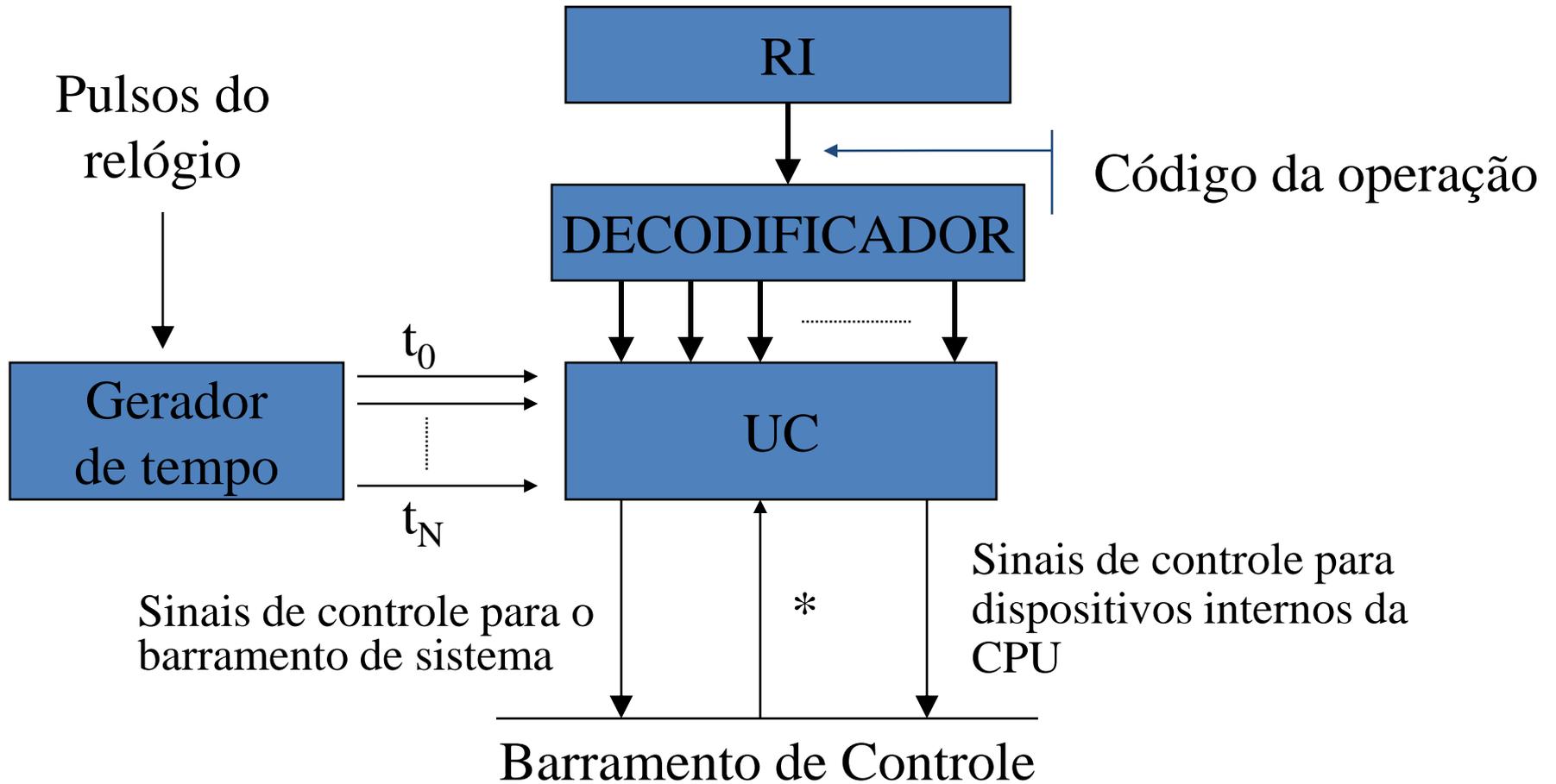
## 3.2 - Função de controle:

### Atividades:

- Busca da instrução a ser executada, armazenando-a em um registrador especialmente projetado para esta finalidade;
- Interpretação das ações a serem desencadeadas com a execução da instrução (se é soma, subtração, etc e como realizá-la);
- Geração de sinais de controle apropriados para ativação das atividades requeridas para a execução propriamente dita da instrução identificada. Estes sinais são enviados tanto para componentes internos (ULA, por ex.) quanto externos (MEM, E/S).

- Em resumo, a área de controle é projetada para entender o que fazer, como fazer e comandar quem vai fazer, no momento adequado.
- Os dispositivos básicos que devem fazer parte dessa área funcional são:
  - Unidade de controle (UC);
  - Decodificador;
  - Registrador de instrução (RI) ou IR - "instruction register";
  - Contador de instrução (CI) ou PC "program counter";
  - Relógio ou "clock";
  - Registradores de endereço de memória (REM) e de dados da memória (RDM).





\* Sinais de controle vindos do barramento do sistema

## 3.2.1 - Registrador de Instrução - RI

- ◆ É o registrador que tem a função específica de armazenar a instrução a ser executada pela CPU.
- ◆ Ao se iniciar um ciclo de instrução, a UC emite o sinal de controle que acarretará a realização de um ciclo de leitura para buscar a instrução na memória, e que, via barramento de dados e RDM, será armazenada no RI.

## 3.2.2 - Contador de Instrução – CI

- ◆ É o registrador cuja função específica é armazenar o endereço da próxima instrução a ser executada.
- ◆ Tão logo a instrução que vai ser executada seja buscada (lida) da memória para a UCP (início do ciclo de instrução), o sistema providencia a modificação do conteúdo do CI de modo que ele passe a armazenar o endereço da próxima instrução na seqüência.
- ◆ Por isso, é comum definir a função do CI como sendo a de "armazenar o endereço da próxima instrução", que é o que realmente ele faz durante a maior parte da realização de um ciclo de instrução.

### **3.2.3 - REM e RDM**

- ◆ Já vistos em memória.

## 3.2.4 - Unidade de Controle

- ◆ É o dispositivo mais complexo da CPU.
- ◆ Possui a lógica necessária para realizar a movimentação de dados e instruções de e para a CPU, através dos sinais de controle que emite em instantes de tempo programados.
- ◆ Exemplo: busca de 1 instrução:

$T_0$  : REM  $\leftarrow$  (CI)

$T_1$  : CI  $\leftarrow$  CI + N

RDM  $\leftarrow$  M(op)

$T_2$  : RI  $\leftarrow$  RDM

M = Célula

M(op) = Conteúdo da célula

As operações menores são conhecidas como **microoperações**

## 4. Instruções de máquina

- ◆ Uma instrução de máquina é a especificação de uma operação básica (ou primitiva) que o hardware é capaz de realizar diretamente.
- ◆ O conjunto de instruções fornece os requisitos funcionais para a CPU.
- ◆ Quando se escreve "conjunto de instruções", estamos nos referindo a todas as possíveis instruções que podem ser interpretadas e executadas por um processador.

## 4.1 - Tipos de instruções:

- ◆ Operações matemáticas (aritméticas, lógicas, de complemento, de deslocamento);
- ◆ Movimentação de dados (memória - UCP e vice-versa);
- ◆ Entrada e saída (leitura e escrita em dispositivo de E/S);
- ◆ Controle (desvio da sequência de execução, parar etc.).

### Observação:

- ◆ Quando se escreve "conjunto de instruções", estamos nos referindo a todas as possíveis instruções que podem ser interpretadas e executadas por um processador.

## 4.2 - Formato das instruções:

<b>C.Op.</b>	<b>Don't care</b>
--------------	-------------------

<b>C.Op.</b>	<b>Operando 1</b>	<b>Operando 2</b>
--------------	-------------------	-------------------

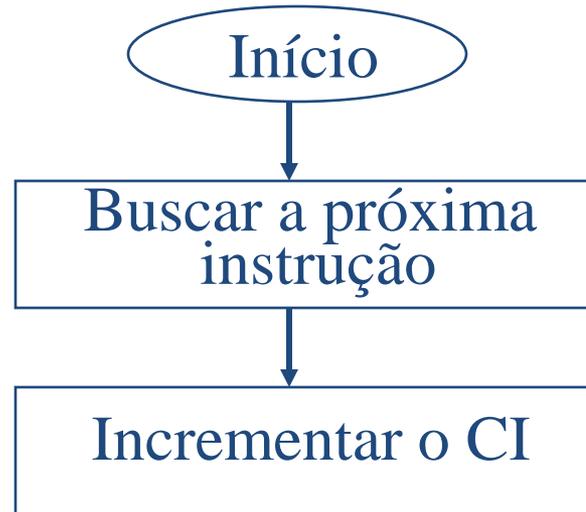
<b>C.Op.</b>	<b>Operando 1</b>
--------------	-------------------

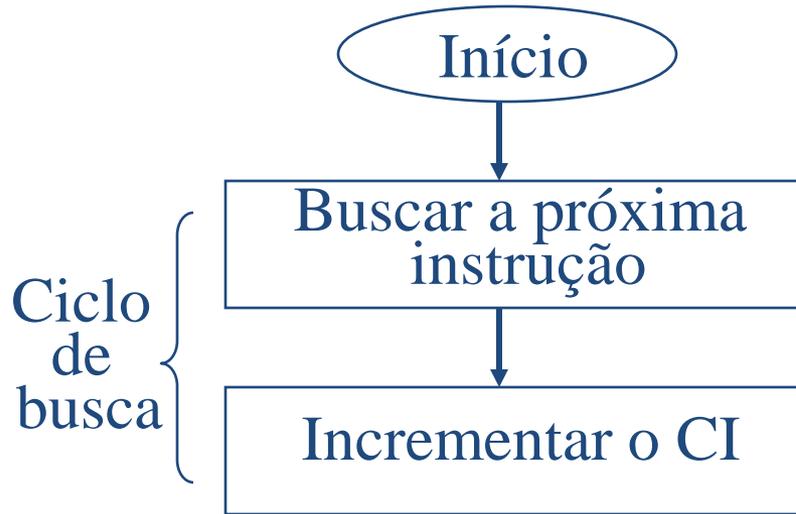
- ◆ De modo geral, podemos separar o grupo de bits que constitui a instrução em dois campos:
  - Um campo (um subgrupo de bits) chama-se código de operação;
  - O restante grupo de bits (se houver) denomina-se campo do(s) operando(s) ou, simplesmente, operando(s).

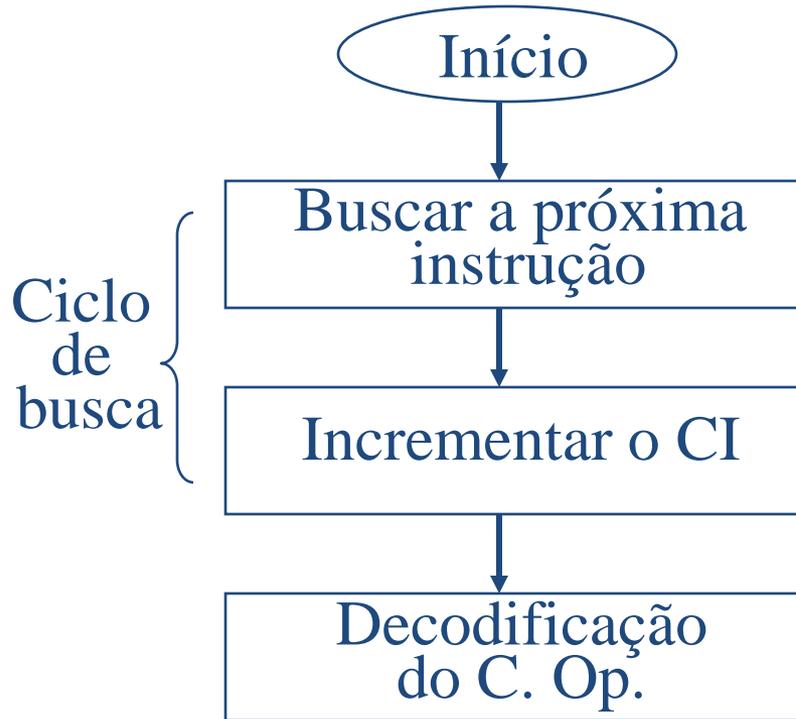
# 5. Funcionamento da UCP

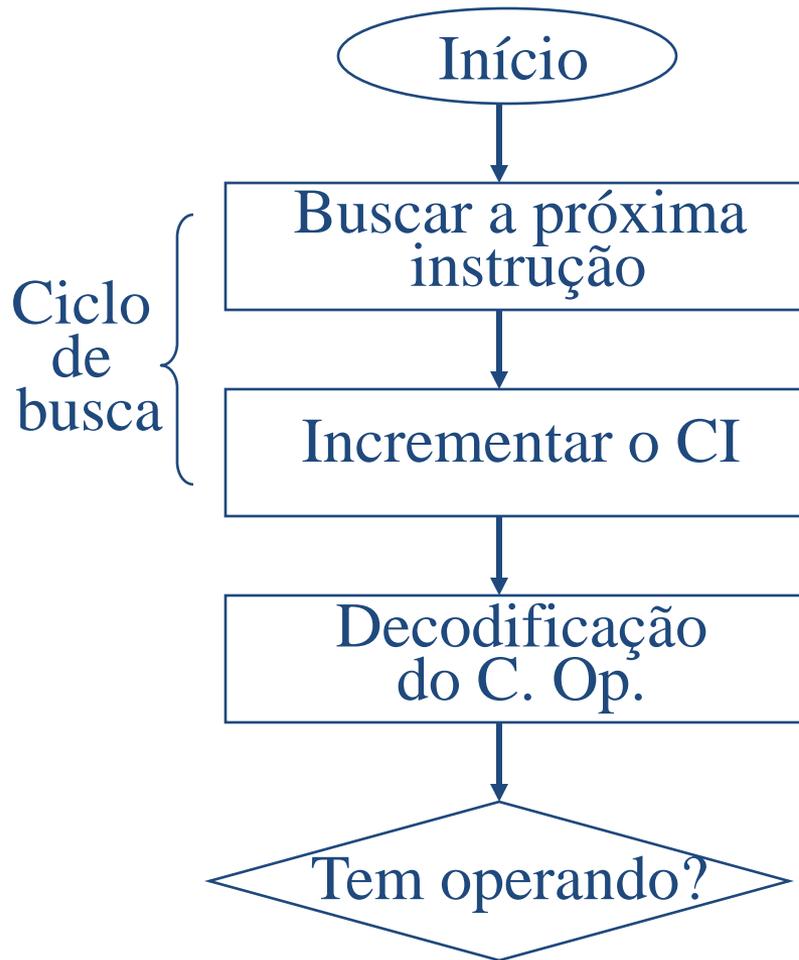
# Ciclo de Instrução:

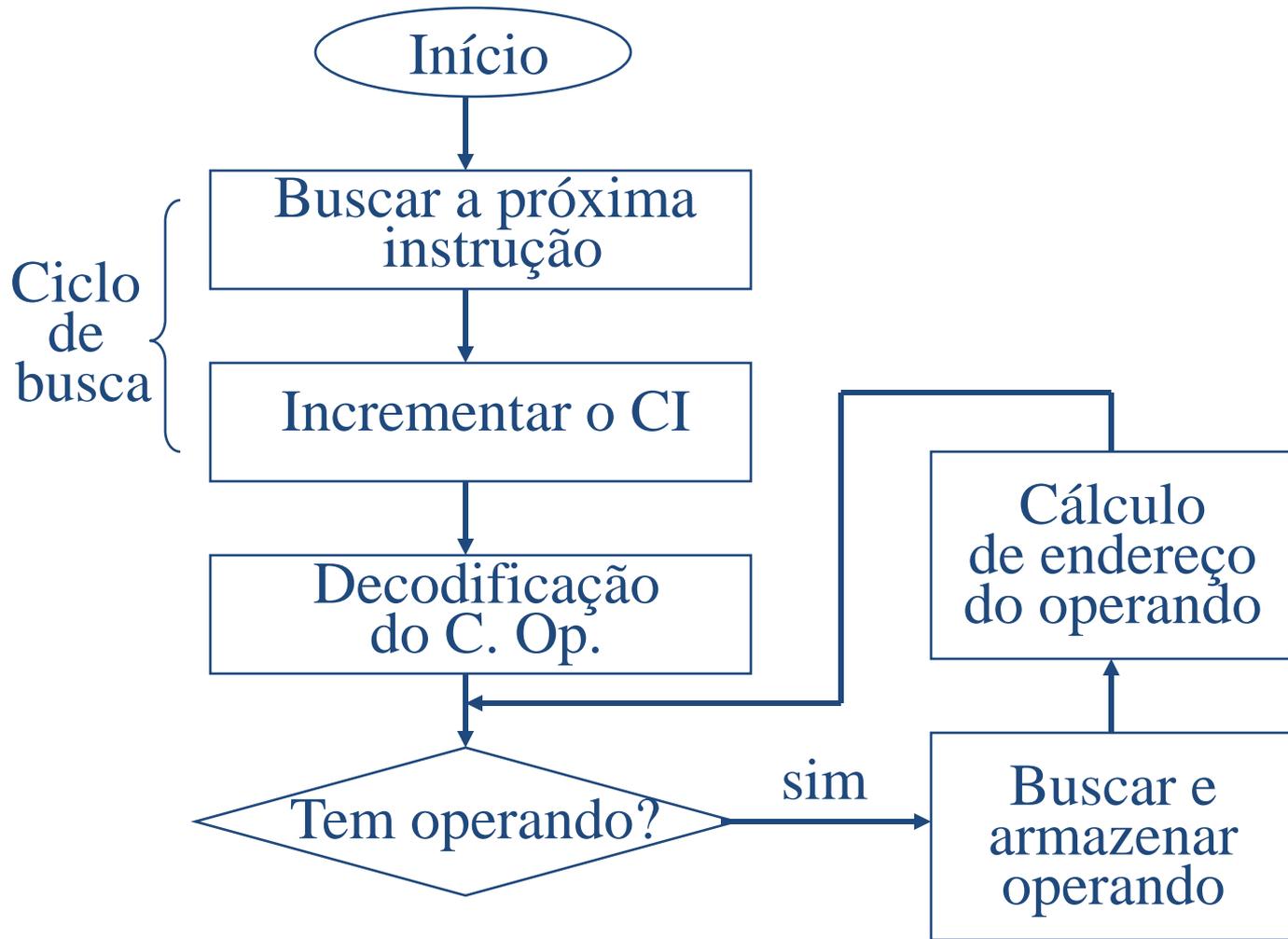


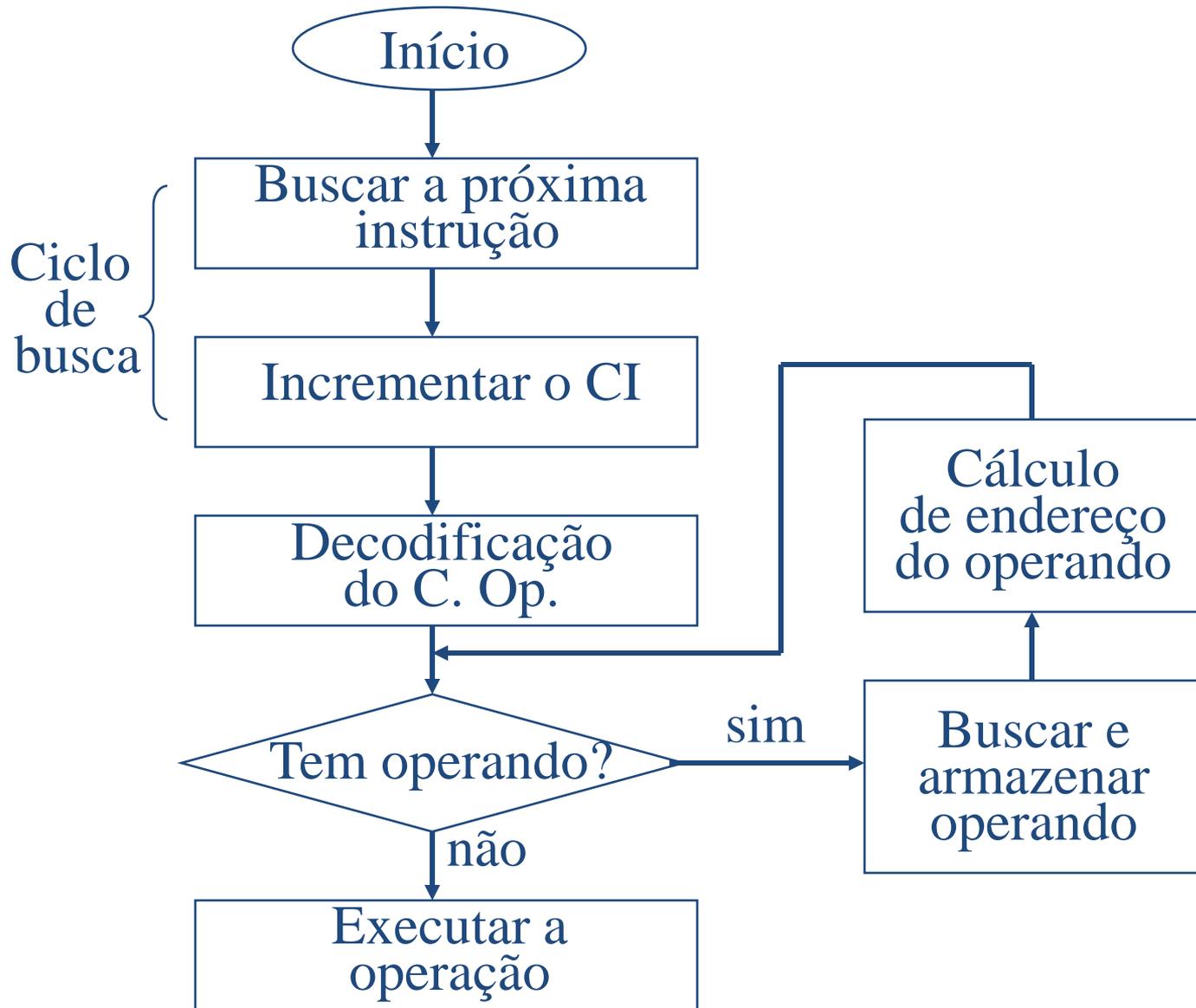


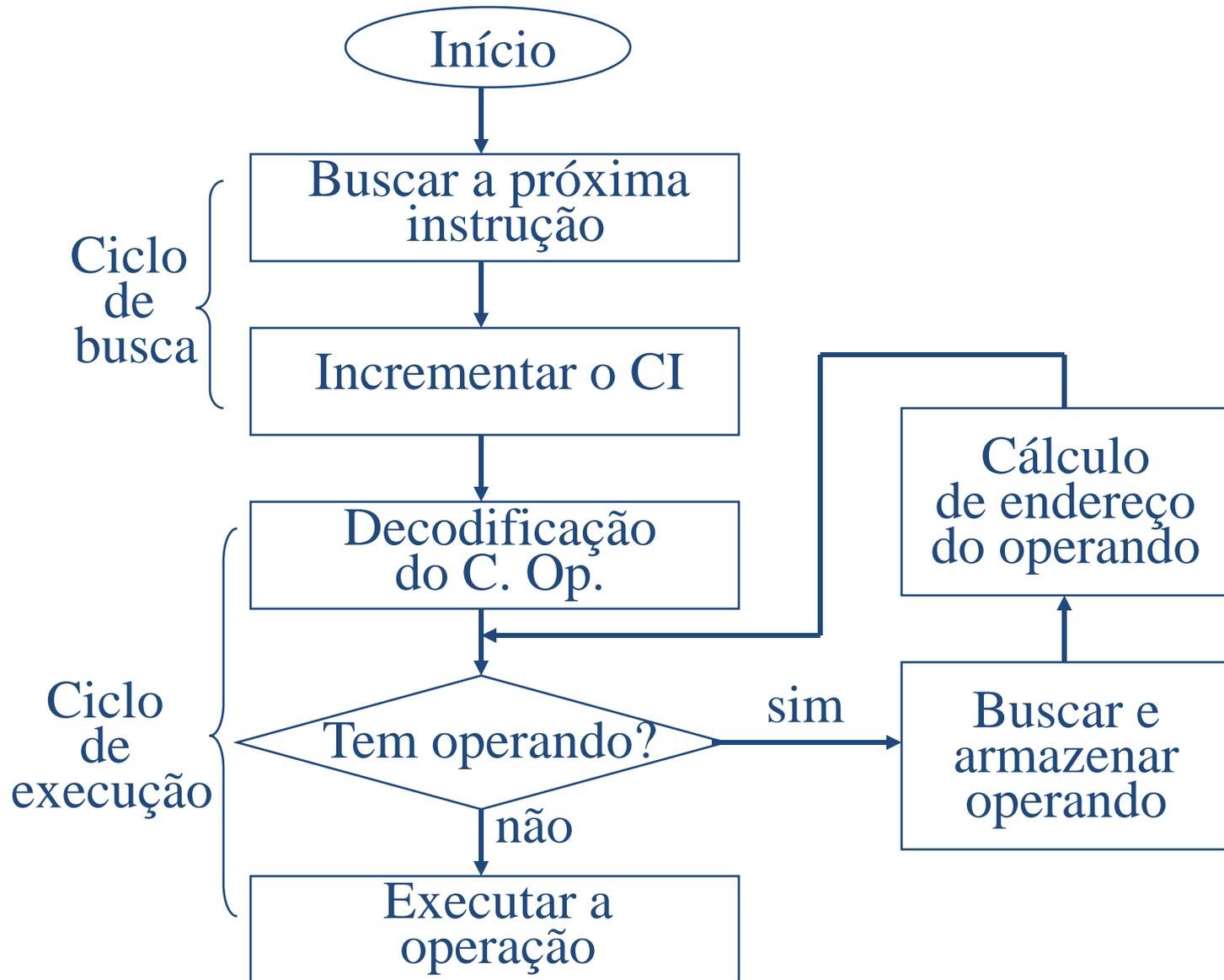


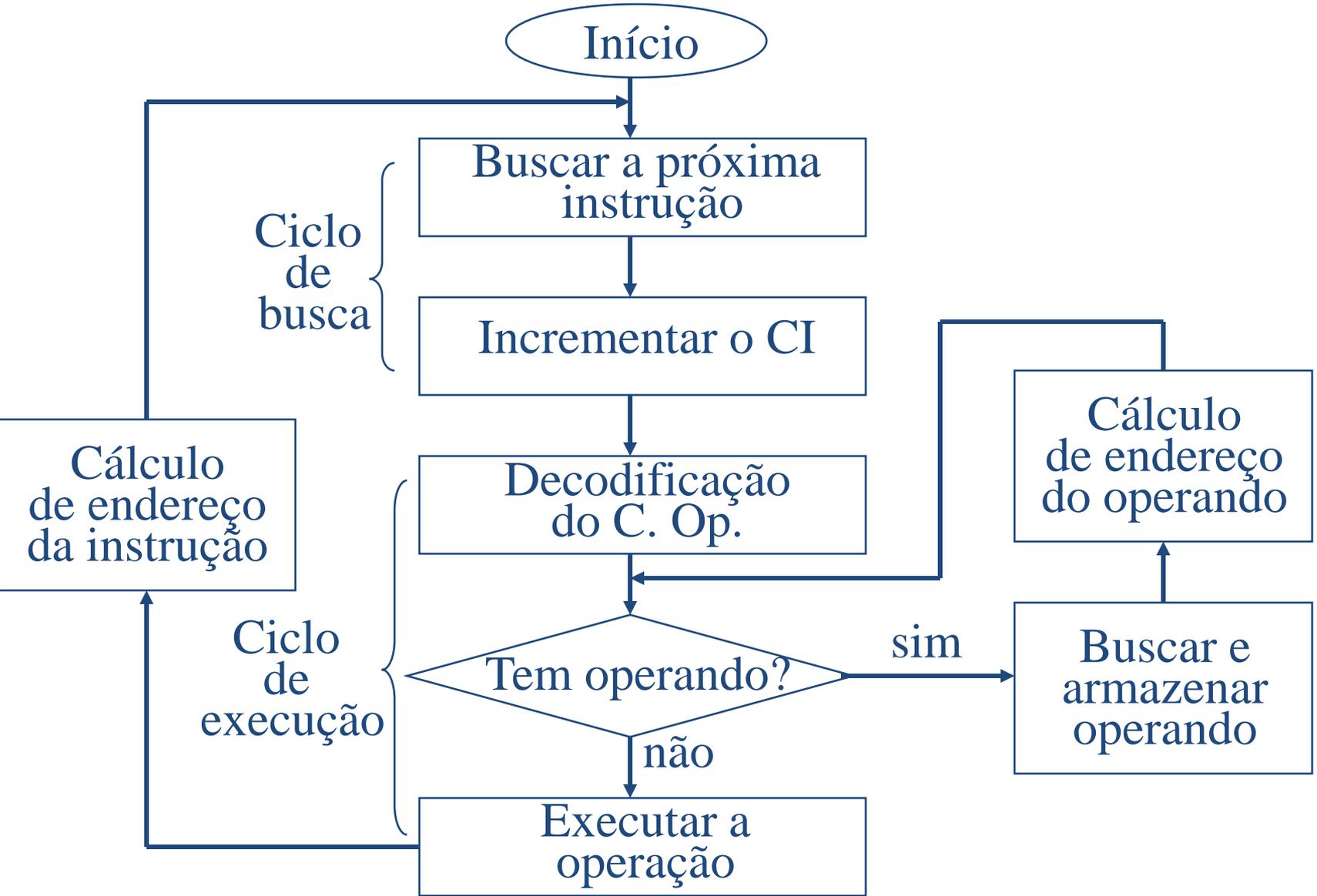












# Ciclo de Instrução:

- ◆ Buscar a próxima instrução:

- $RI \longleftarrow (CI)$

- ◆ Incrementar o CI:

- $(CI) \longleftarrow (CI) + 1$

- ◆ Decodificação do C.Op.:

- o decodificador recebe os bits do C.Op. e gera uma saída para a UC.

- ◆ Buscar Op se houver:

- $RDM \longleftarrow (Op)$

- ◆ Executar a instrução