



UNIVERSIDADE  
FEDERAL DO CEARÁ

# **Instruções**

## **Modos e Formatos de endereçamento**

---

ARQUITETURA DE COMPUTADORES

PROF. Alex Lima

# Introdução

---

- Endereçamento
  - Conceito
  - Modos de endereçamento
  - Formatos de instrução
  - Arquiteturas CISC
  - Arquiteturas RISC

# Endereçamento

---

- Conceito
- Endereçamento é o meio pelo qual a CPU localiza uma instrução ou operando.

# Endereçamento

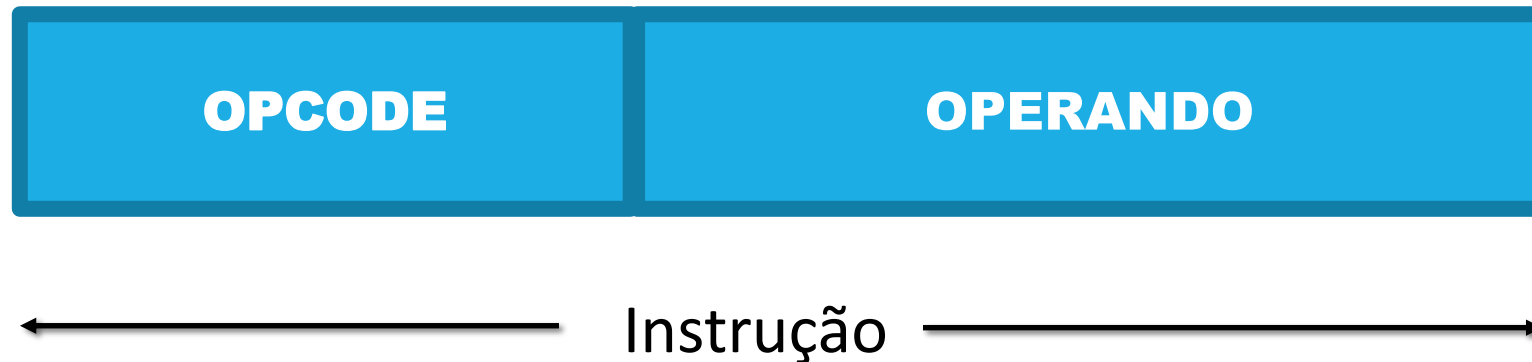
---

- Modos de endereçamento
  - Imediato
  - Direto
  - Indireto
  - Registradores
  - Indireto por registradores
  - Deslocamento
  - Pilha

# Endereçamento imediato

---

- O operando está contido na instrução.



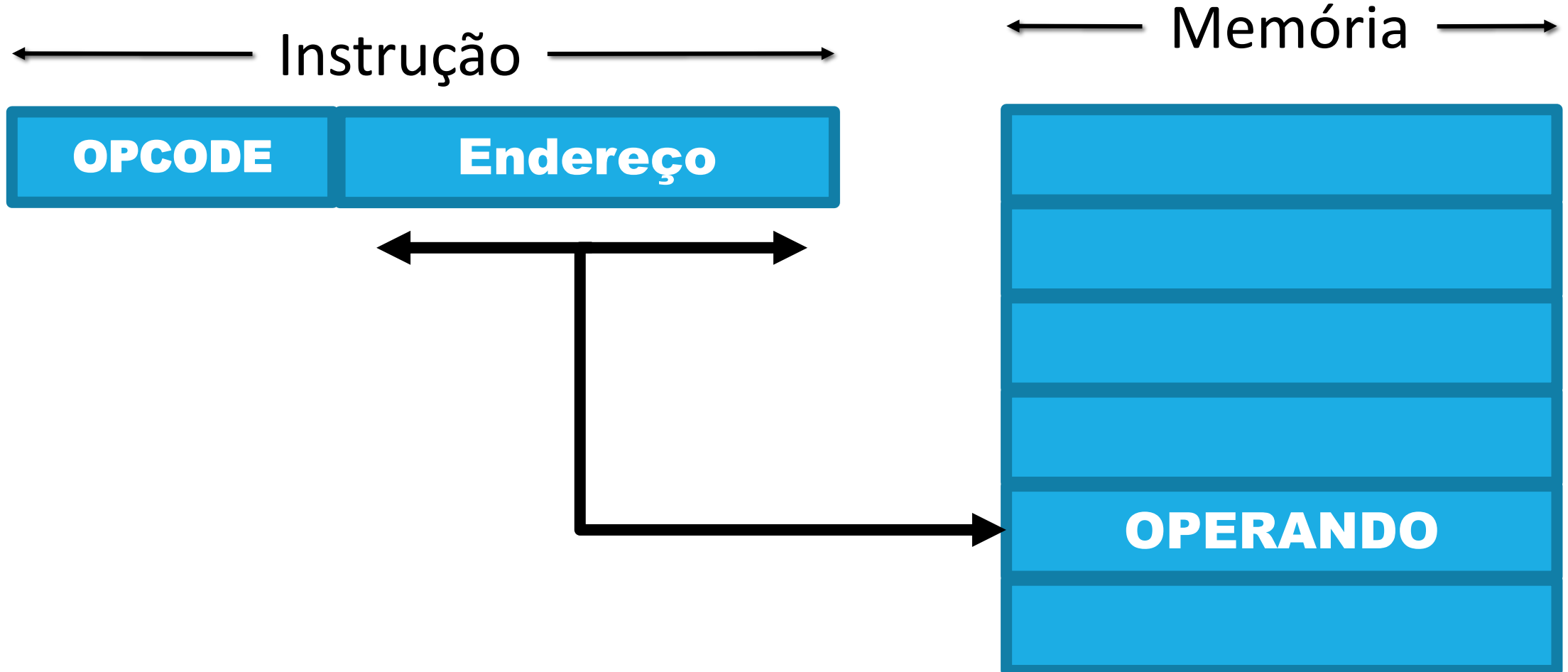
# Endereçamento

---

## Endereçamento imediato

- O campo de operando contém o operando.
- Usado na declaração de constantes.
- Usado na inicialização de variáveis.
- **Vantagens**
  - Nenhuma busca a memória é feita para se obter o valor.
- **Desvantagens**
  - O tamanho do número é limitado ao tamanho do campo de endereço.

# Endereçamento direto



# Endereçamento

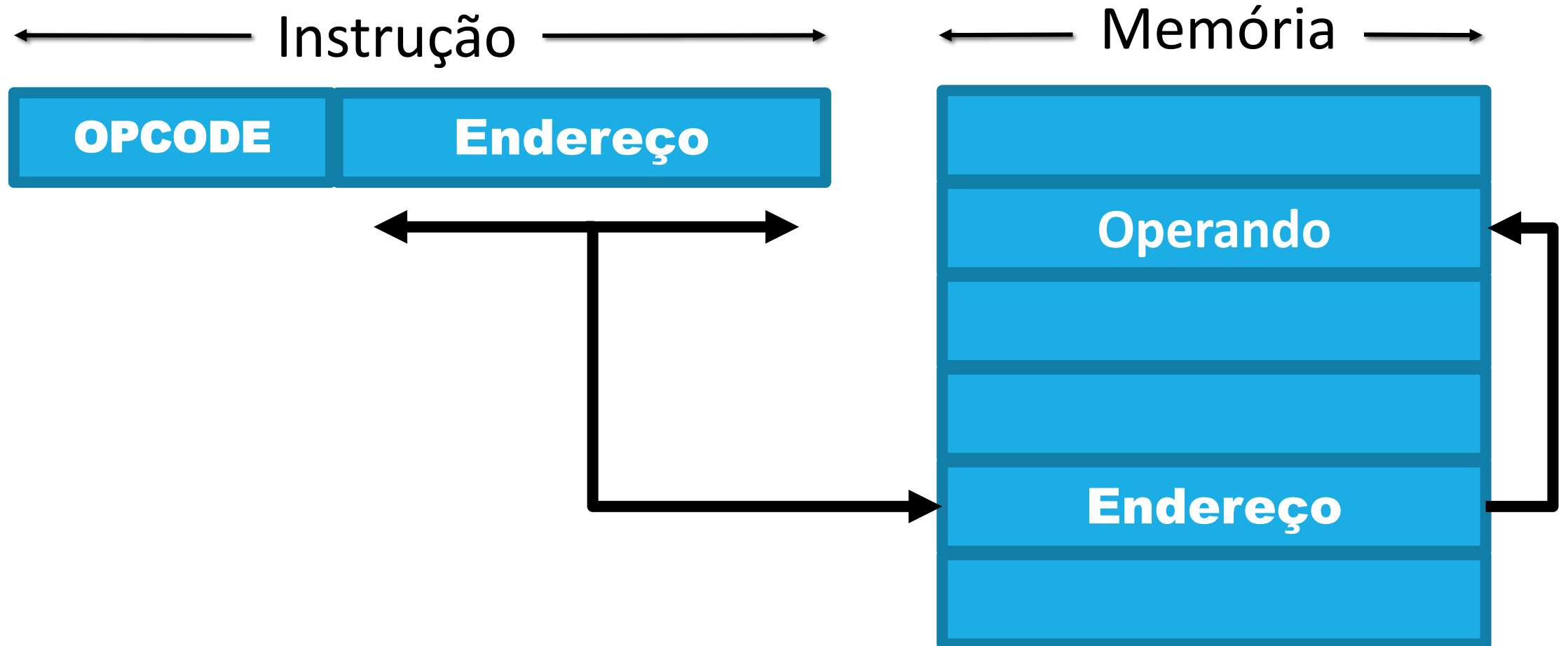
---

## Endereçamento direto

- O campo de operando contém o endereço que aponta para o operando.
- **Vantagens**
  - Simplicidade.
- **Desvantagens**
  - Espaço de endereços limitado pelo tamanho da palavra.



# Endereçamento indireto



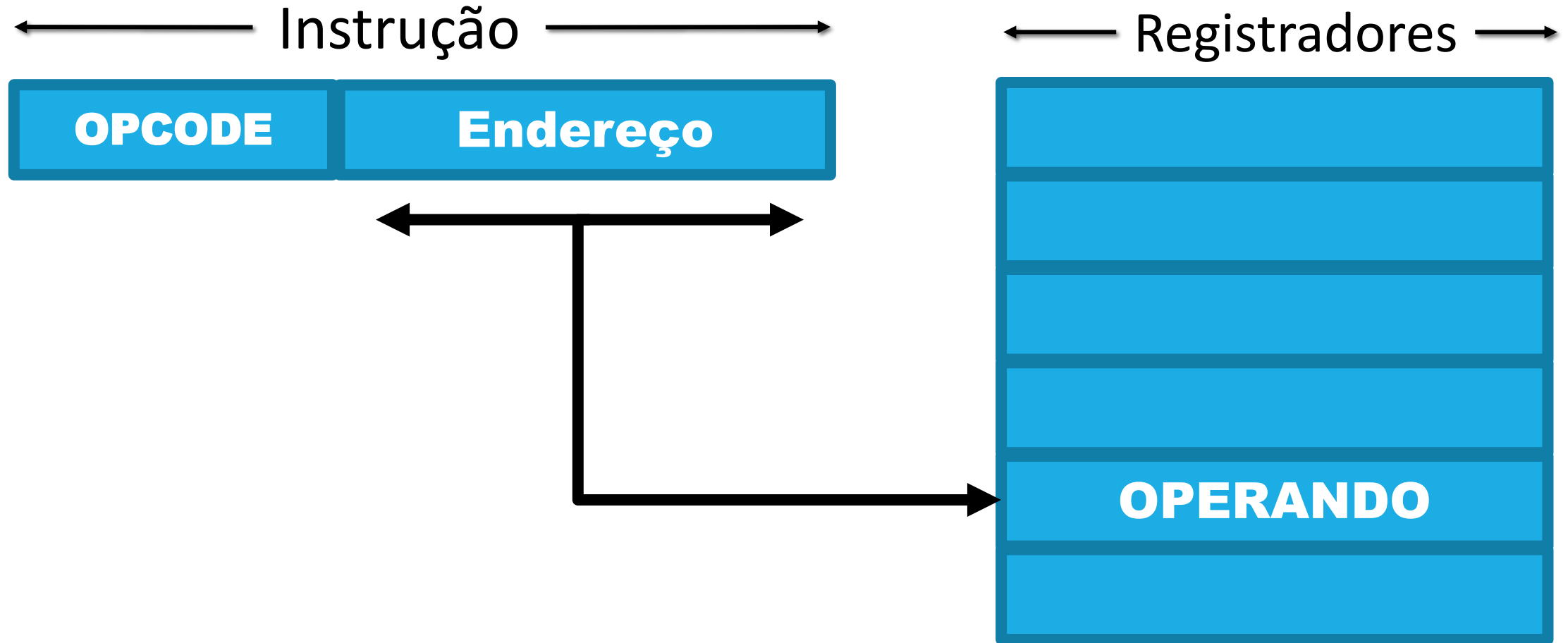
# Endereçamento

---

## Endereçamento indireto

- O campo de operando contém o endereço que aponta para o endereço do operando.
- **Vantagens**
  - Espaço de endereçamento de tamanho  $2^n$ .
- **Desvantagens**
  - São necessárias duas buscas a memória para carregar o operando.

# Endereçamento direto de registrador



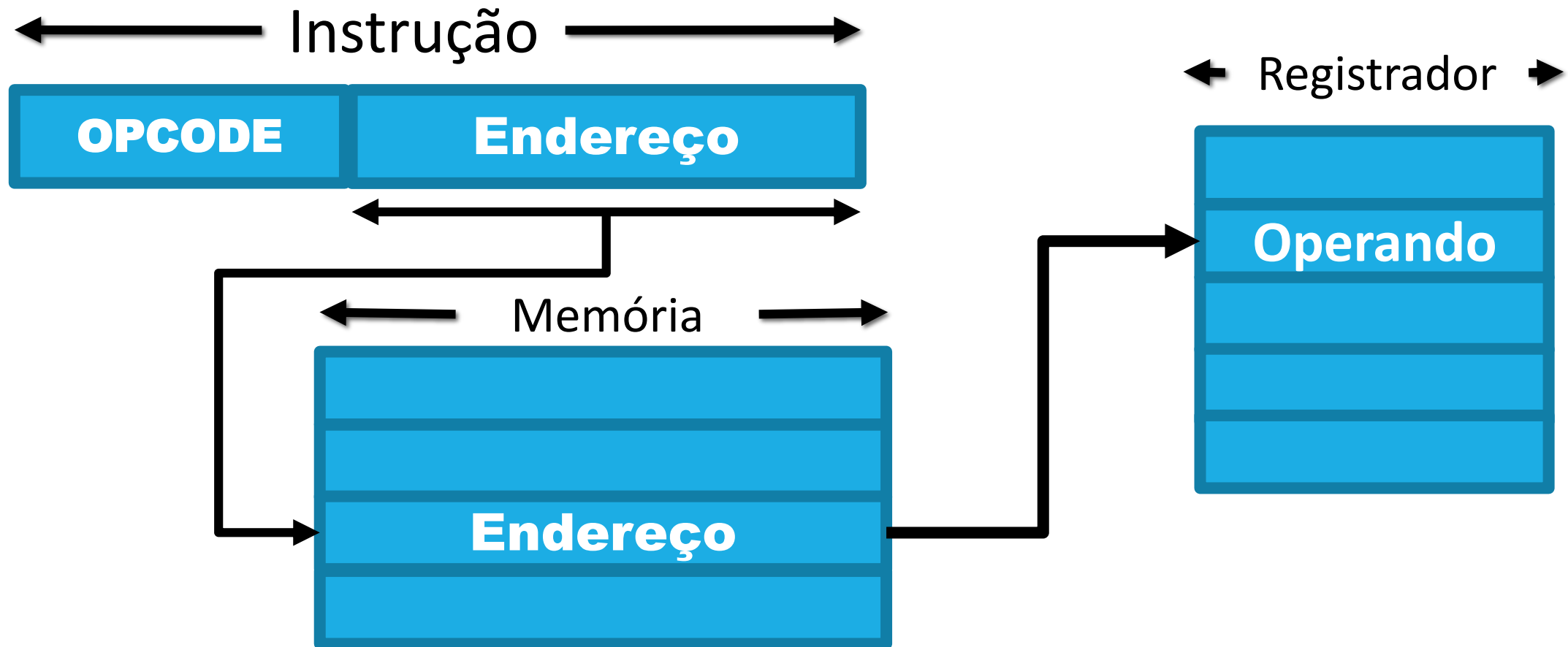
# Endereçamento

---

## Endereçamento direto de registrador

- Similar ao endereçamento direto.
- O campo de operando uma referência ao registrador que armazena o valor.
- **Vantagens**
  - Apenas um pequeno campo de referência ao registrador.
  - Não há consumo significativo de tempo.
- **Desvantagens**
  - Espaço de endereçamento limitado.

# Endereçamento indireto de registrador



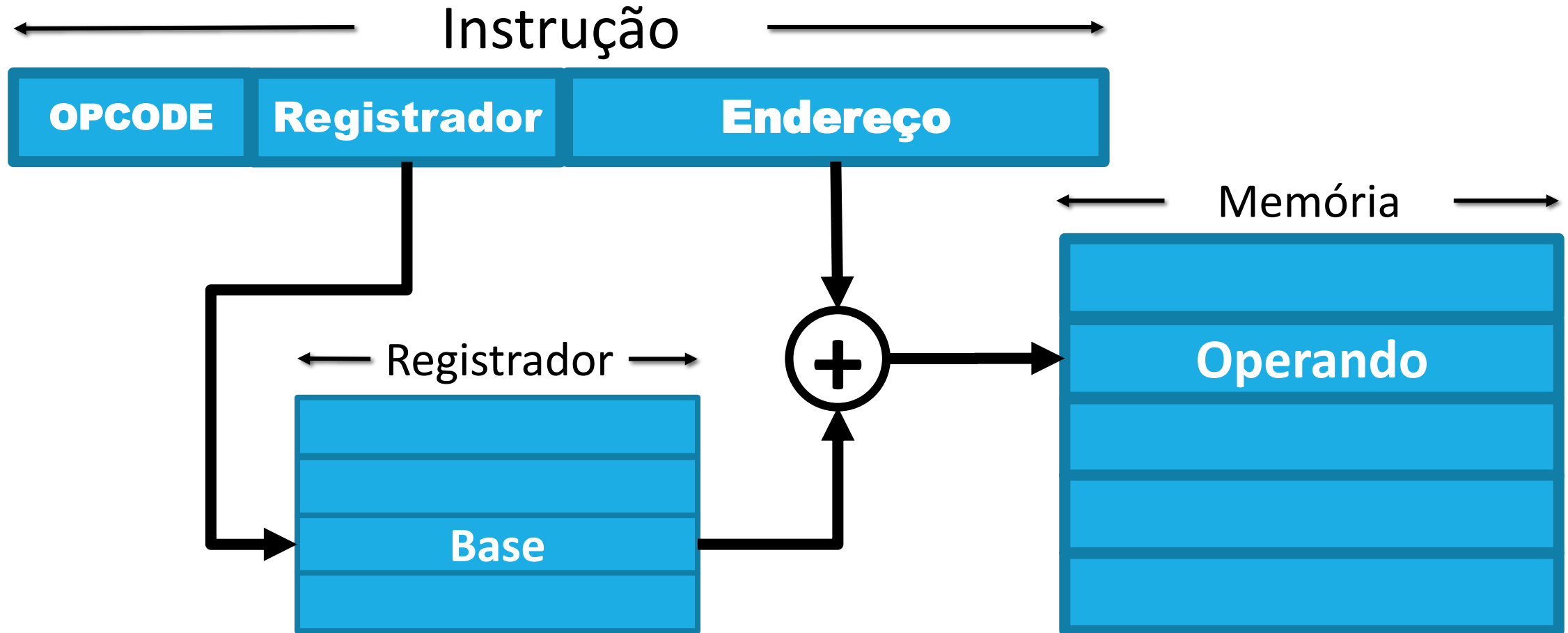
# Endereçamento

---

## Endereçamento indireto de registrador

- Similar ao endereçamento indireto.
- O campo de operando armazena um endereço de memória que contém uma referência ao registrador que armazena o valor.
- **Vantagens**
  - Espaço de endereçamento de tamanho  $2^n$ .
  - Uma busca a menos na memória.
- **Desvantagens**
  - É necessário buscar o operando na memória.

# Endereçamento por deslocamento



# Endereçamento

---

## Endereçamento por deslocamento

- Combina endereçamento direto e indireto de registradores.
- A instrução contém 2 campos de operando:  
**referência + deslocamento**
- **Vantagens**
  - Espaço de endereçamento de tamanho  $2^n$ .
  - Uma busca a menos na memória.
- **Desvantagens**
  - É necessário buscar o operando na memória.



# Endereçamento

---

## Endereçamento por deslocamento

- Endereçamento relativo
- Endereçamento por base
- Endereçamento por indexação

# Endereçamento

## Endereçamento por deslocamento

- Endereçamento relativo

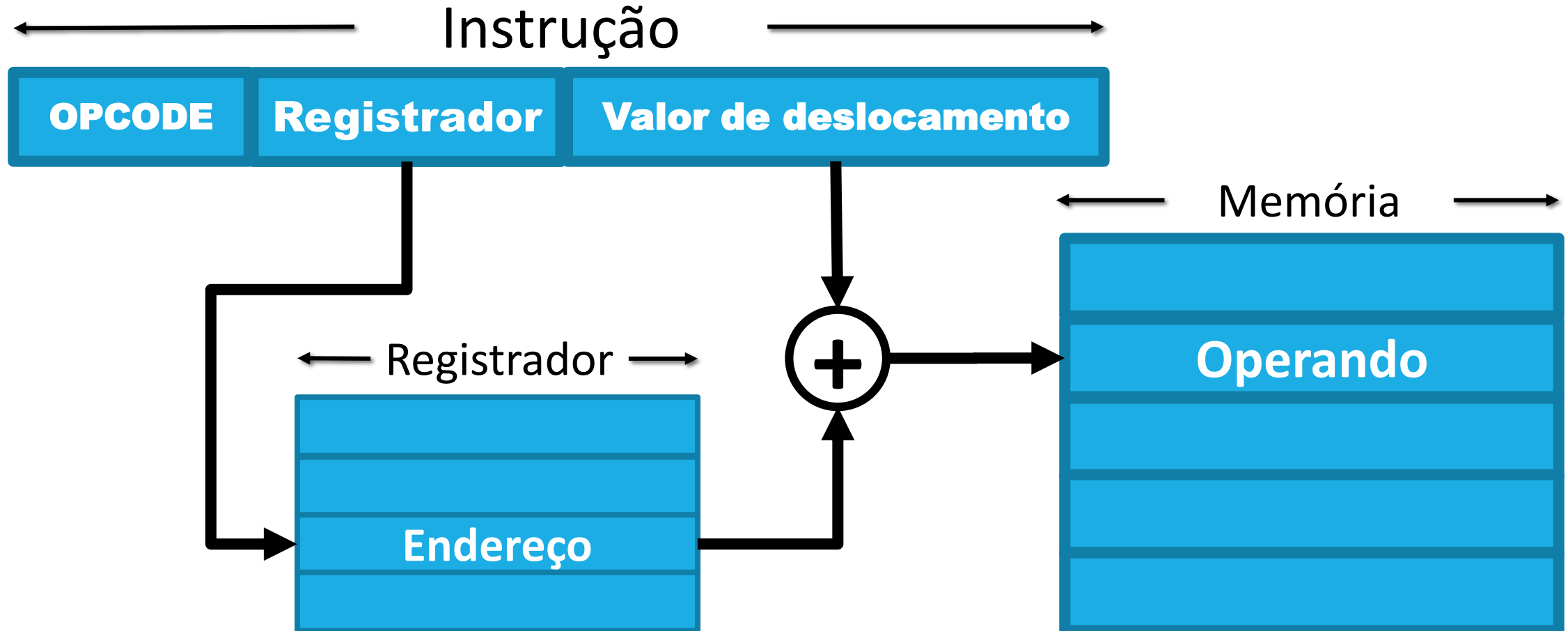


## Exemplo

- Jump incondicional

# Endereçamento por deslocamento

## Endereçamento por registrador base



# Endereçamento por deslocamento

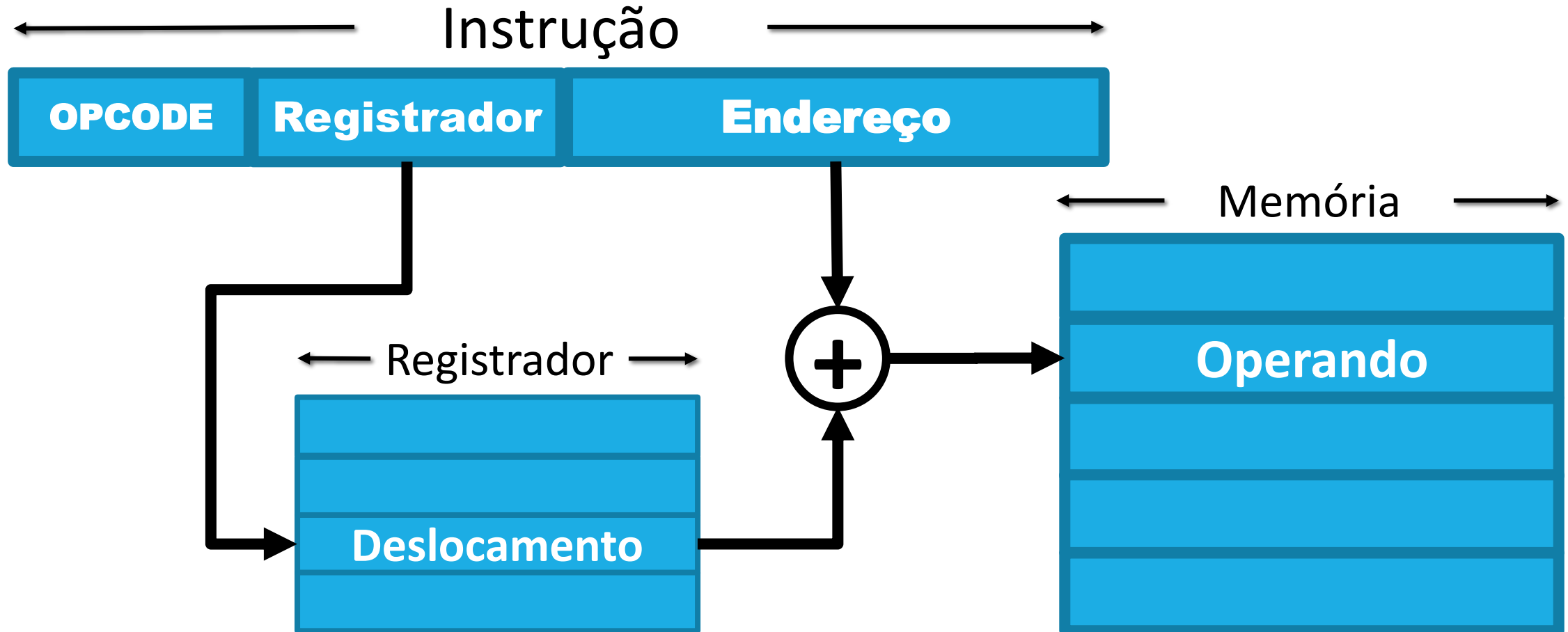
---

## Endereçamento por registrador base

- Segmentação de processos na memória.
- O registrador base por ser utilizado para acessar diferentes segmentos, com base no endereço armazenado.

# Endereçamento por deslocamento

## Endereçamento por indexação



# Endereçamento por deslocamento

---

## Endereçamento por indexação

- Processo oposto ao de endereçamento por registrador base.
- O endereço da memória encontra-se no segundo campo de operando e o deslocamento encontra-se no registrador referenciado.

# Endereçamento por deslocamento

---

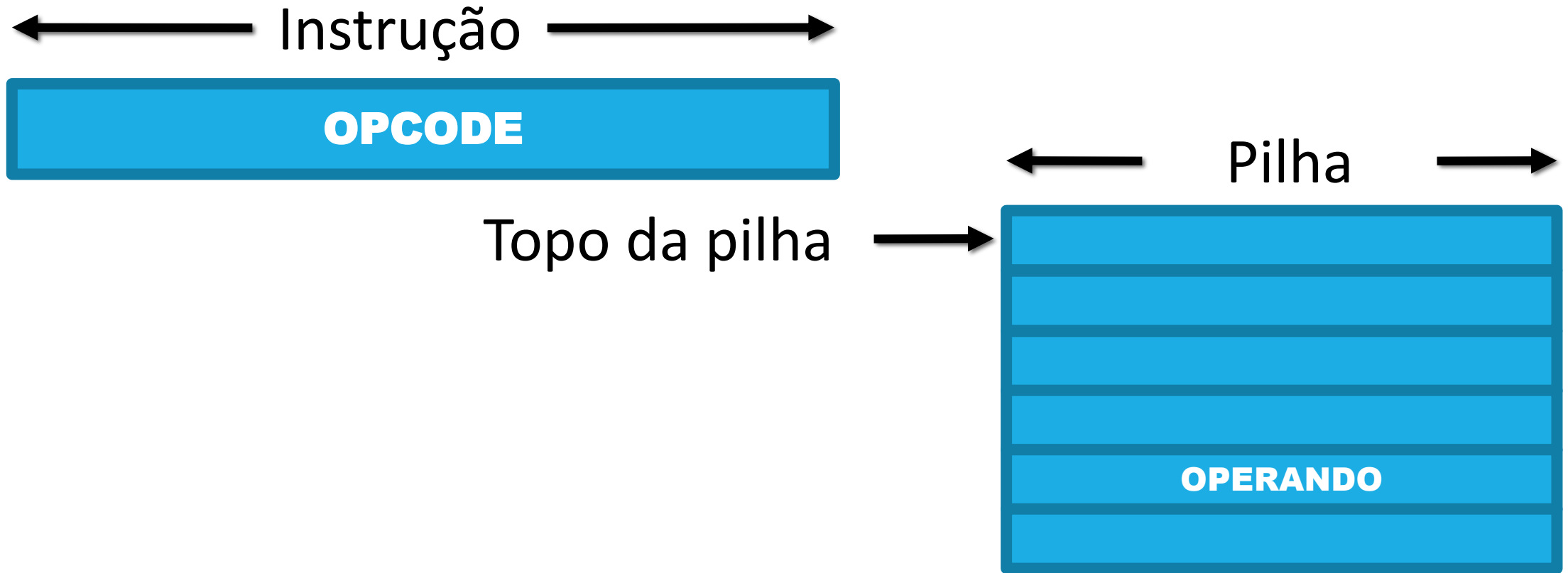
## Endereçamento por indexação

### Exemplo

- A alocação de vetores é feita por meio de um endereço base.
- O primeiro valor do vetor ocupa um endereço e os demais valores são alocados com base no endereço do primeiro valor.
- No MIPS:
  - `LW $t0, 10($s2)`

# Endereçamento de pilha

---





# Endereçamento

---

## Pilha

- Vetor de acesso linear.
- As operações (inserção/remoção) são realizadas a partir do seu topo.
- **Endereçamento de pilha**
  - As expressões precisam ser reescritas em notação polonesa inversa.
  - As operações são realizadas na ordem da expressão, sempre fazendo referência ao topo da pilha.

# Endereçamento de pilha

---

## Notação Polonesa Inversa

- Na notação polonesa invertida, o operador vem após os operandos.
- Exemplo:
- $A + B$  torna-se  $AB+$
- $A + (B \times C)$  torna-se  $ABC \times +$
- $(A + B) \times C$  torna-se  $A B + C \times$

# Endereçamento de pilha

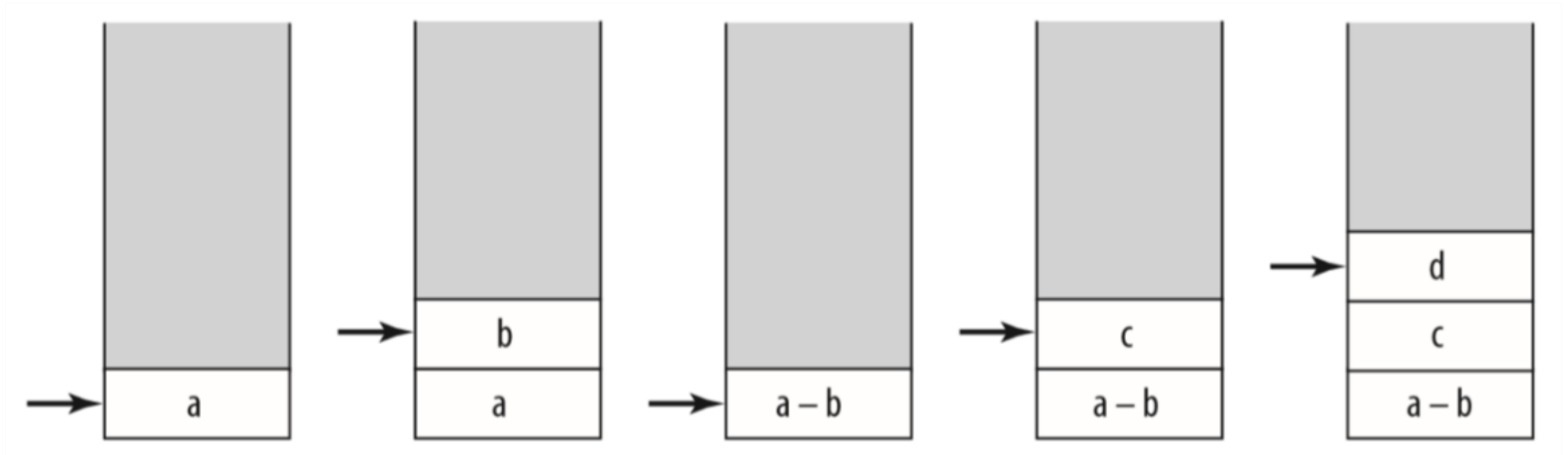
---

## Exemplo

- $F = \frac{A - B}{C + (D \times E)}$
- Na notação polonesa inversa:
- $A - B$  torna-se  $AB-$
- $C + (D \times E)$  torna-se  $CDE \times +$
- $(A - B) / (C + (D \times E))$  torna-se  $AB - CDE \times + /$

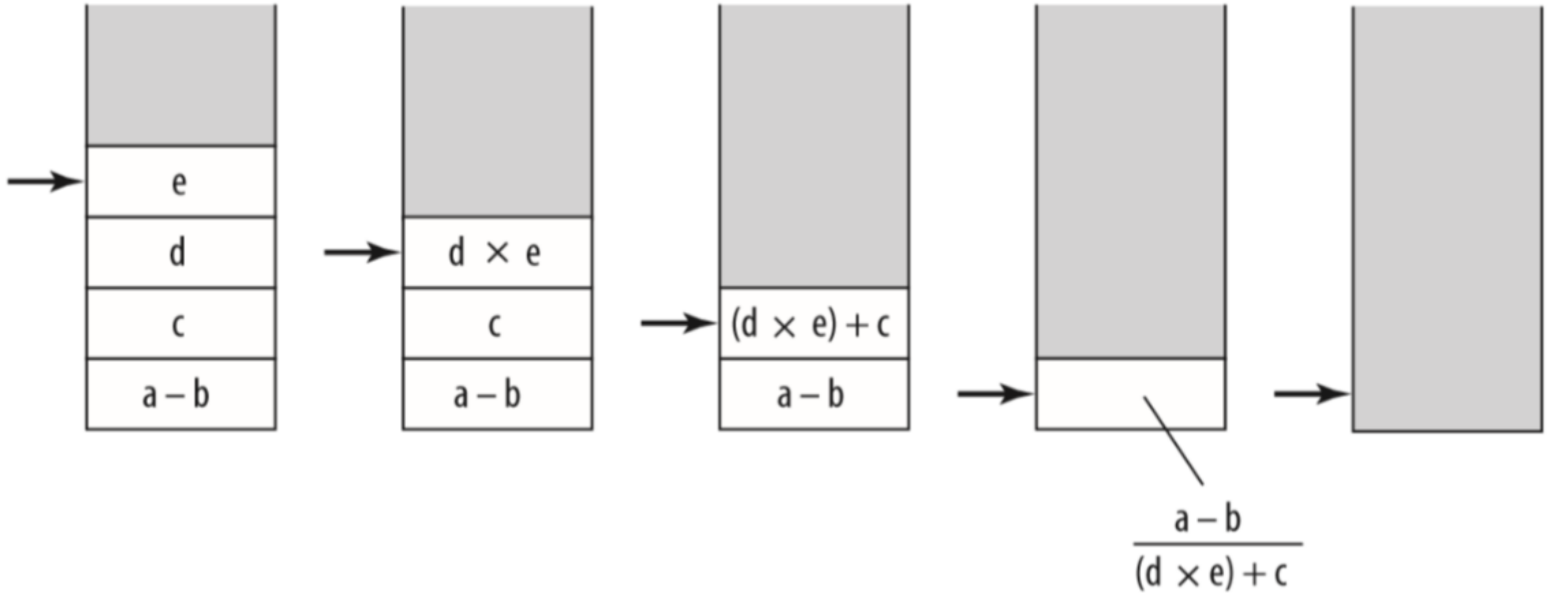
# Endereçamento de pilha

## Exemplo



# Endereçamento de pilha

## Exemplo



# Endereçamento de Pilha

---

## Exemplo

INSTRUÇÃO	DESCRIÇÃO
PUSH A	Empilha A
PUSH B	Empilha B
SUB	Subtrai A e B e armazena o resultado em A
PUSH C	Empilha C
PUSH D	Empilha D
PUSH E	Empilha E
MULT	Multiplica E e D e armazena o resultado em E
ADD	Soma C e E e armazena o resultado em C
DIV	Divide A por B e armazena o resultado em A

# MIPS

---

## Instruções de controle

- Instruções de controle estão relacionadas a controle do fluxo de execução do programa.
- No fluxo padrão de execução as instruções são executadas de forma sequencial, uma após a outra.
- Alguns instruções podem alterar esse fluxo, gerando desvios ou saltos de instrução.

# MIPS

---

## Instruções de controle

- **Desvio incondicional**
  - Salto de instruções
- **Desvio condicional**
  - Teste + Salto de instruções



# MIPS

---

## Instruções de Controle

- Instruções de teste
- Instruções de teste e desvio
- Instruções de desvio

# MIPS

---

## Instruções de teste (Comparação)

- **slt** (Set on Less Than – Atribui se for menor que)
  - `slt $t0, $t1, $2`
- **slti**
  - `slt $t0, $t1, 2`

# MIPS

---

## Instruções de teste e desvio (Desvio condicional)

- **beq** (Branch on equal – Desvie se for igual)
  - beq \$t0, \$t1, l1
- **bne** (Branch on not equal – Desvie se não for igual)
  - bne \$t0, \$t1, l1

# MIPS

---

## Instruções de desvio incondicional

- **j** – (Jump – Salto)
  - j L1
- **jr** – (Jump Register – Salto para registrador)
  - j \$s0

# MIPS

---

- **Array**
- **Testes**
- **Desvios**

# PRÁTICA

---

1. Descreva o funcionamento de uma pilha e como são executadas as instruções.
2. Descreva o endereçamento por indexação.
3. Escreva um código em MIPS que realize a soma de dois números.
4. Escreva um código em MIPS que leia seu nome e exiba a mensagem “Bem vindo a UFC, ...”.

# Formatos de instruções

---

**Formato de instruções** define a estrutura de uma instrução.

- Quantidade de campos
- Ordem dos campos
- Tamanho de cada campo

# Formatos de instruções

---

## Formatos de instrução





# Formatos de instruções

---

## Tamanho da instrução

- Opcodes
- Operandos
- Modos de endereçamento
- Faixa de endereçamento

# Formatos de instruções

---

## Tamanho da instrução

### Vantagens

- **Operações e Operandos** - Programas mais curtos
- **Modos de endereçamento** - Acesso a locais de memória específicos
- **Faixa de endereçamento** - Maior uso da memória

### Desvantagens

- Desperdício
- Uma instrução de 64 bit não será duas vezes mais útil que uma instrução de 32 bits!

# Formatos de instruções

---

## Tamanho da instrução

- Tamanho do barramento
- Velocidade da memória (Gargalo)
  - Memória cache
  - Instruções menores
- Tamanho da palavra

# Formatos de instruções

---

## Tamanho da instrução

### Tamanho do opcode

- Opcodes de tamanhos variáveis

### Tamanho dos campos de endereçamento

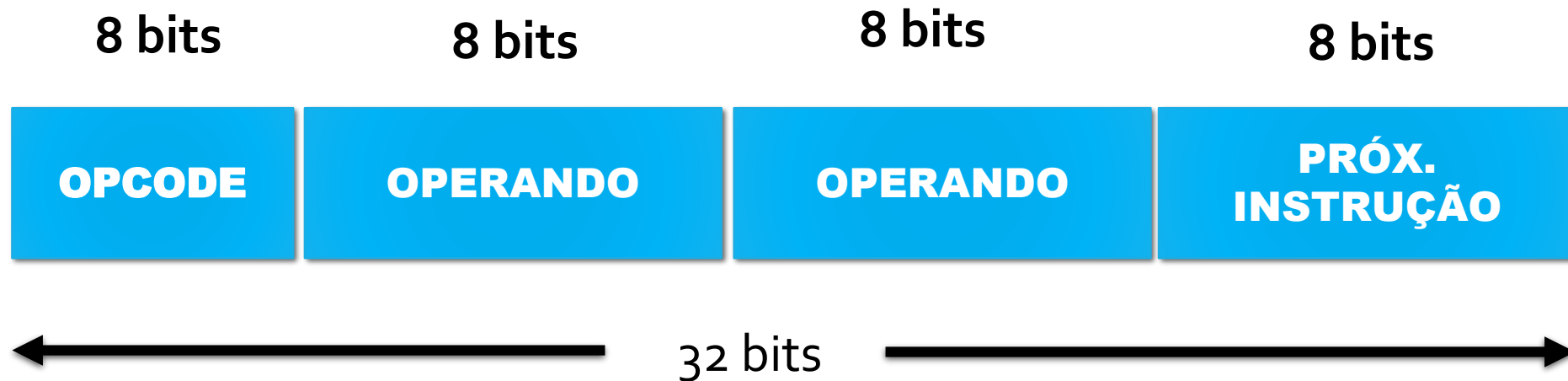
- Número de modos de endereçamento
- Número de operandos
- Registrados vs Memória
- Intervalo de endereços
- Granularidade do endereço

# Formatos de instruções

---

## Tamanho dos campos de endereçamento

- Número de modos de endereçamento / Intervalo de endereços



# Formatos de instruções

---

## Tamanho dos campos de endereçamento

- Número de modos de endereçamento / Intervalo de endereços
- Campos de endereçamento de 8 bits podem endereçar até  $2^8$  endereços
- Apenas 2 MB!
- Essa limitação é contornada por modos de endereçamento indireto.
  - Endereçamento indireto
  - Endereçamento indireto de registrado

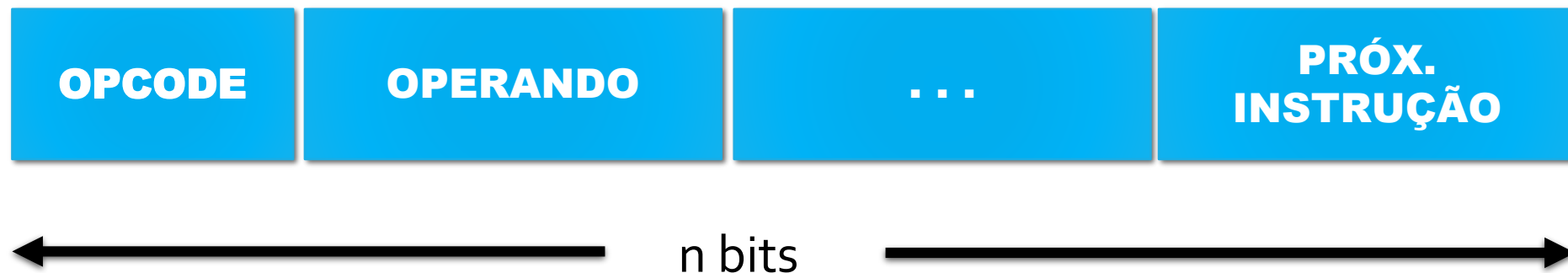
# Formatos de instruções

---

## Tamanho da instrução

Tamanho dos campos de endereçamento

- Número de operandos



# Formatos de instruções

---

## Tamanho dos campos de endereçamento

- Registrados vs Memória
  - Número de registradores  $\Leftrightarrow$  Número de operandos
  - Número de registradores  $\Leftrightarrow$  Número de operações
  - Número de registradores  $\Leftrightarrow$  Tamanho do campo de endereçamento



# Formatos de instruções

---

## Tamanho da instrução

- **Granularidade do endereço** (Divisão do espaço de memória)
  - Endereçamento por byte
  - Endereçamento por palavra

# Formatos de instruções

---

## Instruções de tamanho variável

### Vantagens

- Flexibilidade
- Campos de tamanhos diferentes

### Desvantagens

- Aumento na complexidade do processador  $\Rightarrow$  Aumento no preço do processador

# MIPS

---

O MIPS possui três classes de instrução:

- Instruções do tipo R
- Instruções do Tipo I
- Instruções do tipo J

# MIPS

---

O Simulador MARS trabalha como um montador da linguagem assembly para a arquitetura MIPS.

Tradução do programa em assembly para linguagem de máquina (binário).

# MIPS

---

## Instruções do tipo R

- Instruções aritméticas ou lógicas



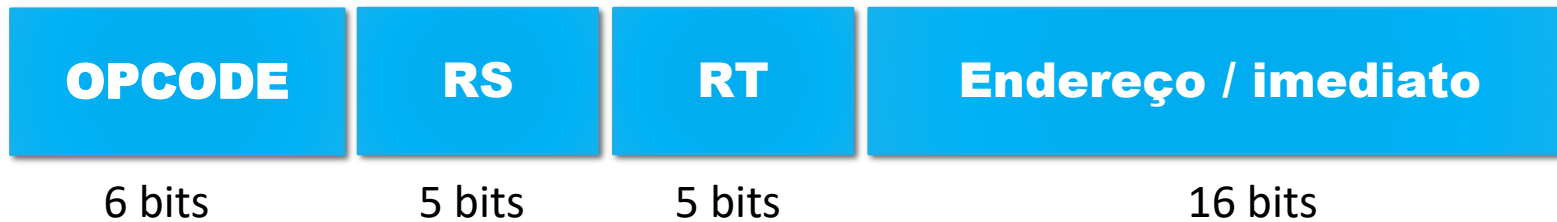
- **opcode** – Código de operação
- **rs e rt** – Registradores fonte
- **rd** – Registrador destino
- **shamt** – Shift amount (usado em funções de deslocamento)
- **funct** – Código de função (Classe de operação)

# MIPS

---

## Instruções do tipo I

- Instruções para transferências e desvios



- **opcode** – Código de operação
- **rs e rt** – Registradores fonte
- **Endereço/ Imediato** – Endereço do operando / Valor do operando

# MIPS

---

## Instruções do tipo I

- Instruções para salto



- **opcode** – Código de operação
- **Endereço de destino** – Local para onde o correrá o desvio

# MIPS

---

## Modos de endereçamento do MIPS

- Endereçamento imediato
- Endereçamento direto ao registrador
- Endereçamento por deslocamento
- Endereçamento relativo ao PC



# MIPS

---

## Modos de endereçamento do MIPS

- Endereçamento imediato

### 1. Endereçamento imediato



# MIPS

## Modos de endereçamento do MIPS

- Endereçamento direto ao registrador

### 2. Endereçamento em registrador

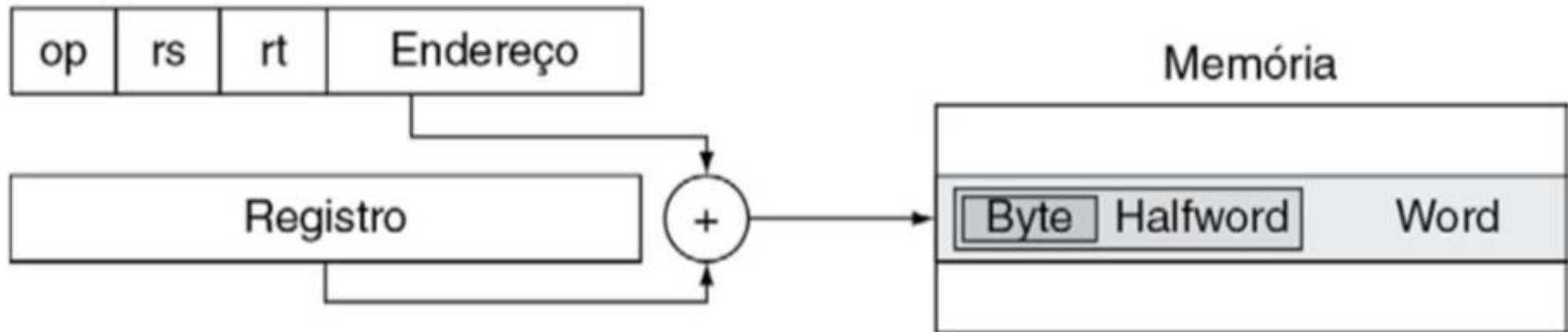


# MIPS

## Modos de endereçamento do MIPS

- Endereçamento por deslocamento

### 3. Endereçamento de base

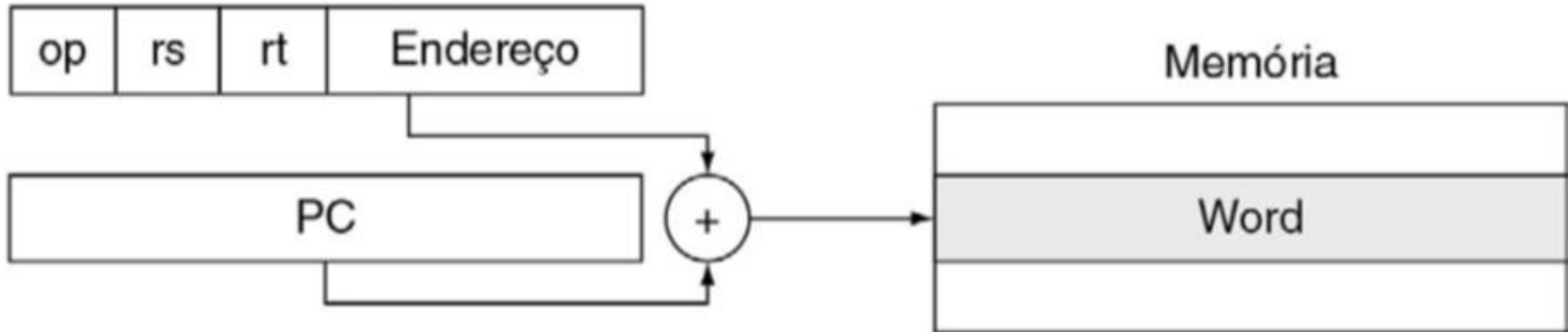


# MIPS

## Modos de endereçamento do MIPS

- Endereçamento relativo ao PC

### 4. Endereçamento relativo ao PC



# MIPS

---

## Exemplo

- Instruções do Tipo R (lógicas ou aritméticas)
- ADD \$s1, \$s2, \$s3.



# PRÁTICA

---

1. Explique porque não é um bom investimento adicionar mais do que 4 GB de memória RAM em um computador cujo processador seja de 32 bits.
2. O que é um montador (assembler)?
3. Quais os formatos de instrução adotados pelo MIPS? Explique a função de cada campo.