

Arquitetura e organização de computadores

Sistema de Computação

SIAC 202 - Arquitetura de Computadores

Prof.: Félix do Rêgo Barros

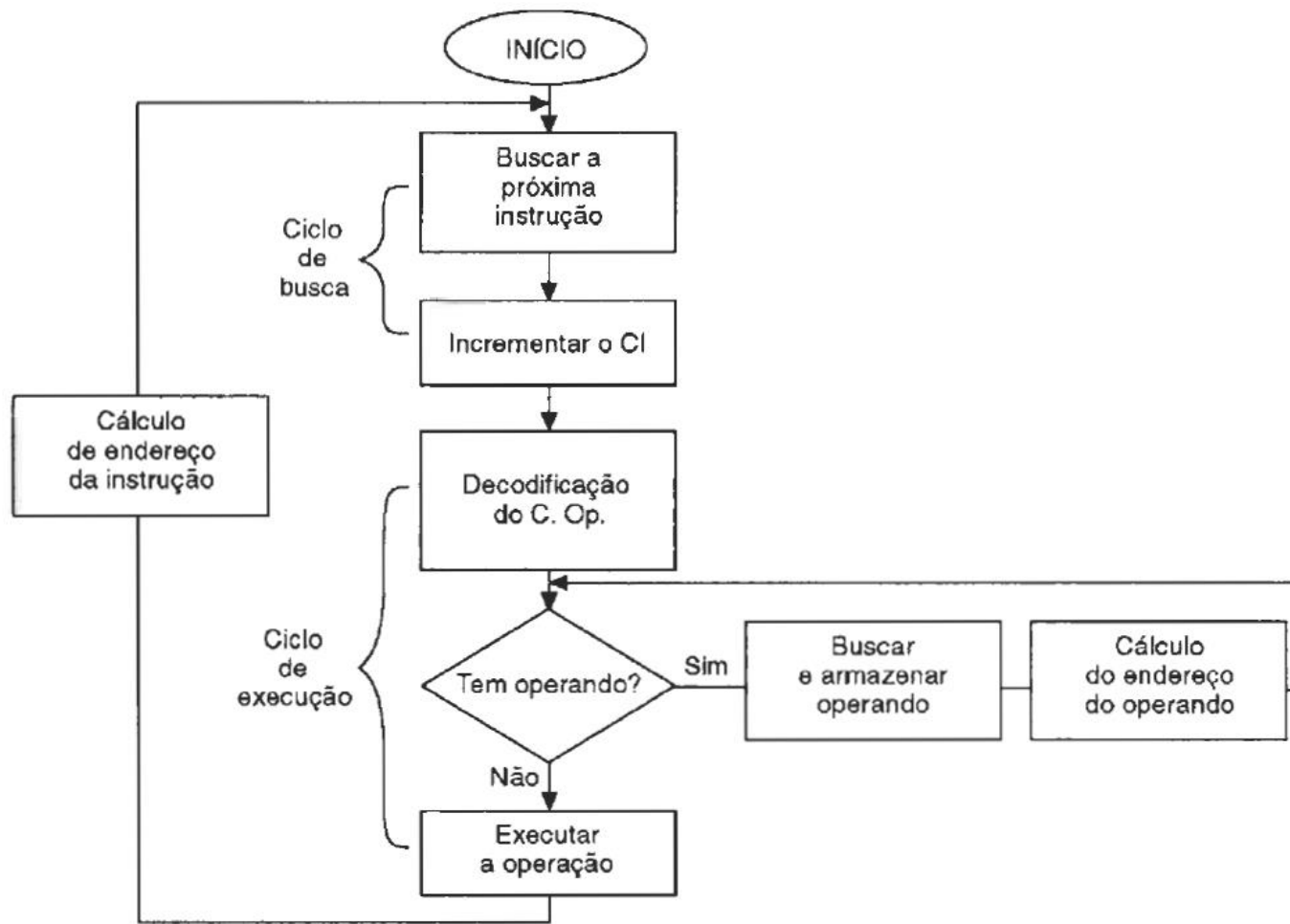
felixregobarros@gmail.com

Baseado em W. Stallings – Arquitetura e Organização de Computadores

Instruções de Máquina

Uma instrução de máquina é a formalização de uma operação básica (ou primitiva) que o hardware é capaz de realizar diretamente.

Funcionamento da CPU. O ciclo da Instrução



Formato das Instruções



Código de operação = 8 bits . Então $2^8 = 256$ códigos de operações

Como cada código operação representa uma única instrução, então 256 C.O. indica 256 instruções de máquina.

Campo operando É(são) o(s) campo(s) da instrução cujo valor binário indica a localização do dado (oi dados) que será (ão) manipulado(s) durante a realização da operação.

Formato das Instruções



A instrução seria assim representada:

(operador 1) \leftarrow (operador 1) + (operador 2)

Ou

(operador 1) \leftarrow (operador 1) + (operador 2)

Podemos também utilizar o registrador Acumulador (**ACC**)

ACC \leftarrow ACC + (Operando)

Podemos ter duas maneiras de se criar um conjunto de instruções de um processador:

Instruções com Código de operação de tamanho fixo e variável

Quantidade de Operandos

Quatro Operandos

Código Operando	Operando 1	Operando 2	Operando 3	Endereço da próxima Instrução
-----------------	------------	------------	------------	-------------------------------

No caso de computador com memória de 2K células (endereços) e com instrução possuindo um código de operação de 6 bits (conjunto de 64 possíveis instruções), cada instrução teria um tamanho total de 50 bits.

Exemplo de uma instrução de soma (em Assembly):

ADD X,Y,Z,P, cuja descrição para execução seria:

$(Z) \leftarrow (X) + (Y)$, sendo $P \leftarrow$ endereço da próxima instrução

$C = A + B$

Podendo ser representada em linguagem Assembly como:

ADD A,B,C,P

Quantidade de Operandos

Três Operandos

Código Operando	Operando 1	Operando 2	Operando 3
-----------------	------------	------------	------------

As instruções de três operandos, empregadas em operações aritméticas, podem ser do tipo:

ADD A,B,X $(X) \leftarrow (A) + (B)$

SUB A,B,X $(X) \leftarrow (A) - (B)$

MPY A,B,X $(X) \leftarrow (A) (B)$

DIV A,B,X $(X) \leftarrow (A) (B)$

Quantidade de Operandos

Três Operandos

Código Operando	Operando 1	Operando 2	Operando 3
-----------------	------------	------------	------------

Exemplo:

$$X = A * (B + C * D - E / F)$$

Escrevendo esta expressão em Assembly:

MPY C, D,T1

DIV E,F,T2

ADD B,T1,Y

SUB Y,T2,Z

MPY A,Z,X

Quantidade de Operandos

Dois Operandos

Código Operando	Operando 1	Operando 2
-----------------	------------	------------

As instruções de dois operandos, empregadas em operações aritméticas, podem ser do tipo:

ADD op.1,op.2 $(op.1) \leftarrow (op.1) + (op.2)$

SUB op.1,op.2 $(op.1) \leftarrow (op.1) - (op.2)$

MPY op.1,op.2 $(op.1) \leftarrow (op.1) * (op.2)$

DIV op.1,op.2 $(op.1) \leftarrow (op.1) / (op.2)$

MOVE A,B $(A) \leftarrow (B)$

Quantidade de Operandos

Dois Operandos

Código Operando	Operando 1	Operando 2
-----------------	------------	------------

Exemplo:

$$X = A * (B + C * D - E / F)$$

Escrevendo esta expressão em Assembly:

MPY C, D

DIV E, F

ADD B, C

SUB B, E

MPY A, B

MOVE X, A

Quantidade de Operandos

Um Operando

Código Operando	Operando
-----------------	----------

Utilizamos o acumulador (ACC) é empregado como operando implícito (não é necessário especificar o seu endereço na instução, pois só há um ACC), guardando o valor de um dos dados e, posteriormente, o valor do resultado da operação.

ADD op. $ACC \leftarrow ACC + (op)$

SUB op. $ACC \leftarrow ACC - (op)$

MPY op. $ACC \leftarrow ACC (op)$

DIV op. $ACC \leftarrow ACC (op)$

Com propósito de permitir a transferência de dados entre a ACC e a MP, foram criadas duas novas instruções:

LDA OP. $ACC \leftarrow (OP.)$

STA OP. $(OP.) \leftarrow ACC$

Quantidade de Operandos

Um Operandos

Código Operando	Operando
-----------------	----------

Exemplo:

$$X = A * (B + C * D - E / F)$$

Escrevendo esta expressão em Assembly:

```
LDA C
MPY D
STA X
LDA E
DIV F
STA T1
LDA B
ADD X
SUB T1
MPY A
STA X
```

Modos de Endereçamento

- ✓ IMEDIATO;
- ✓ DIRETO;
- ✓ INDERETO;
- ✓ REGISTRO;
- ✓ REGISTRO INDERETO;
- ✓ DESLOCAMENTO.

Modo Imediato

Instrução



Modo Imediato

- ❑ Operando é parte da instrução: campo de endereço
- ❑ Nenhum outro acesso à memória é feito além da busca da instrução
- ❑ Maior rapidez
- ❑ Tamanho dos operandos é limitado ao tamanho do campo de endereços

Modo Imediato

Exemplos

C. Op.	Operando
--------	----------

4 bits

8 bits

JMP Op. $CI \leftarrow Op.$

C. Op. = 1010 = hexadecimal A

Instrução: 101000110101 ou A35 (C. Op. = A e Operando = 35) ——— Armazenar o valor 35 no CI.

C. Op.	R	Operando	MOV R, Op. $R \leftarrow Op.$ C. Op. = 0101 = hexadecimal 5
--------	---	----------	--

4 bits

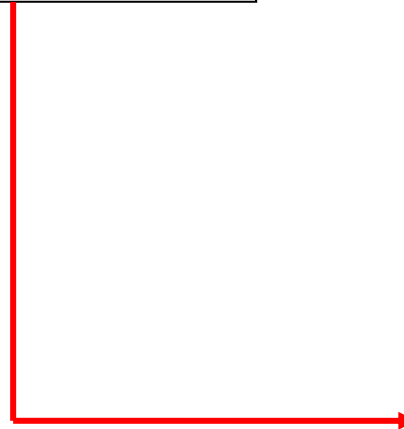
4 bits

8 bits

Instrução: 0101001100000111 ou 5307 (C. Op. = 5, R = 3 e Operando = 07) ——— Armazenar o valor 07 no registrador de endereço 3 (R3).

Modo Direto

Instrução



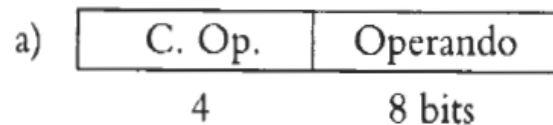
Memória

Modo Direto

- ❑ Campo de endereço contém o endereço efetivo do operando
- ❑ Só uma referência à memória é feita
- ❑ Espaço de endereçamento limitado

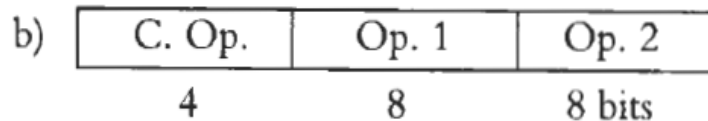
Modo Direto

Exemplos:



C. Op. = 7 LDA Op. $ACC \leftarrow (Op.)$

Após a execução da instrução, o ACC conterá o valor 05A.



C. Op. = B ADD Op.1, Op.2 $(Op.1) \leftarrow (Op.1) + (Op.2)$

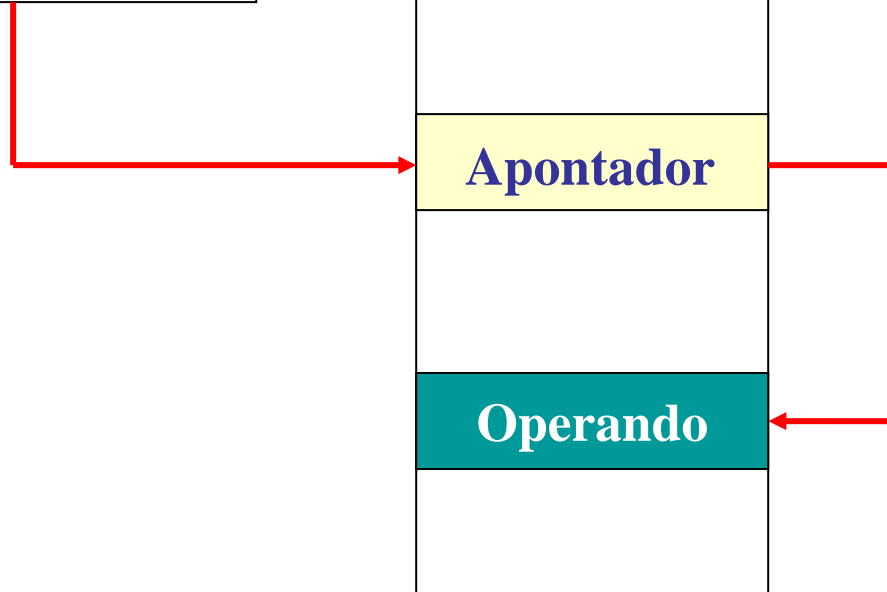
Instrução: B55C3B

	MP
	"
	"
	"
3B	05A
	"
	"
	"
5C	103 15D
	"
	"

Somar o dado de valor binário 000100000011 ou hexadecimal 103 armazenado na célula de endereço hexadecimal 5C (Op. 1) com o dado de valor binário 000001011010 ou hexadecimal 05A armazenado na célula de endereço 3B (Op. 2) e armazenar o resultado (15D) na célula de endereço hexadecimal 5C (Op. 1).

Modo Indireto

Instrução



Memória

Modo Indireto

- Campo de endereço referencia um apontador em memória, que referencia o operando
- Para uma palavra de tamanho N , um espaço de endereçamento de 2^N é acessível
- Duas referências à memória são feitas: mais lento

Modo Indireto

Exemplo:

C.Op.	Operando
-------	----------

4

8 bits

C.Op. = 4 LDA Op. $ACC \leftarrow ((Op.))$

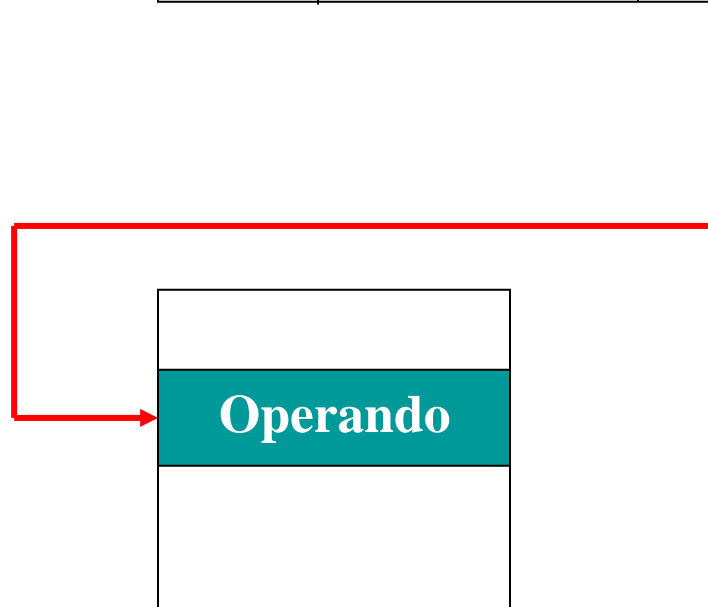
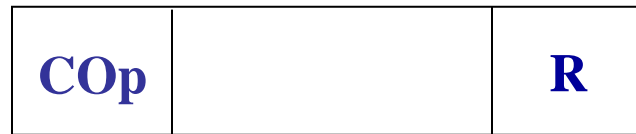
Instrução: 474

74 é o endereço da célula cujo conteúdo (5D) é o endereço do dado (1A4). Após a execução, o valor 1A4 estará armazenado no ACC.

	MP
74	05D
	"
5D	1A4
	"

Registrador

Instrução

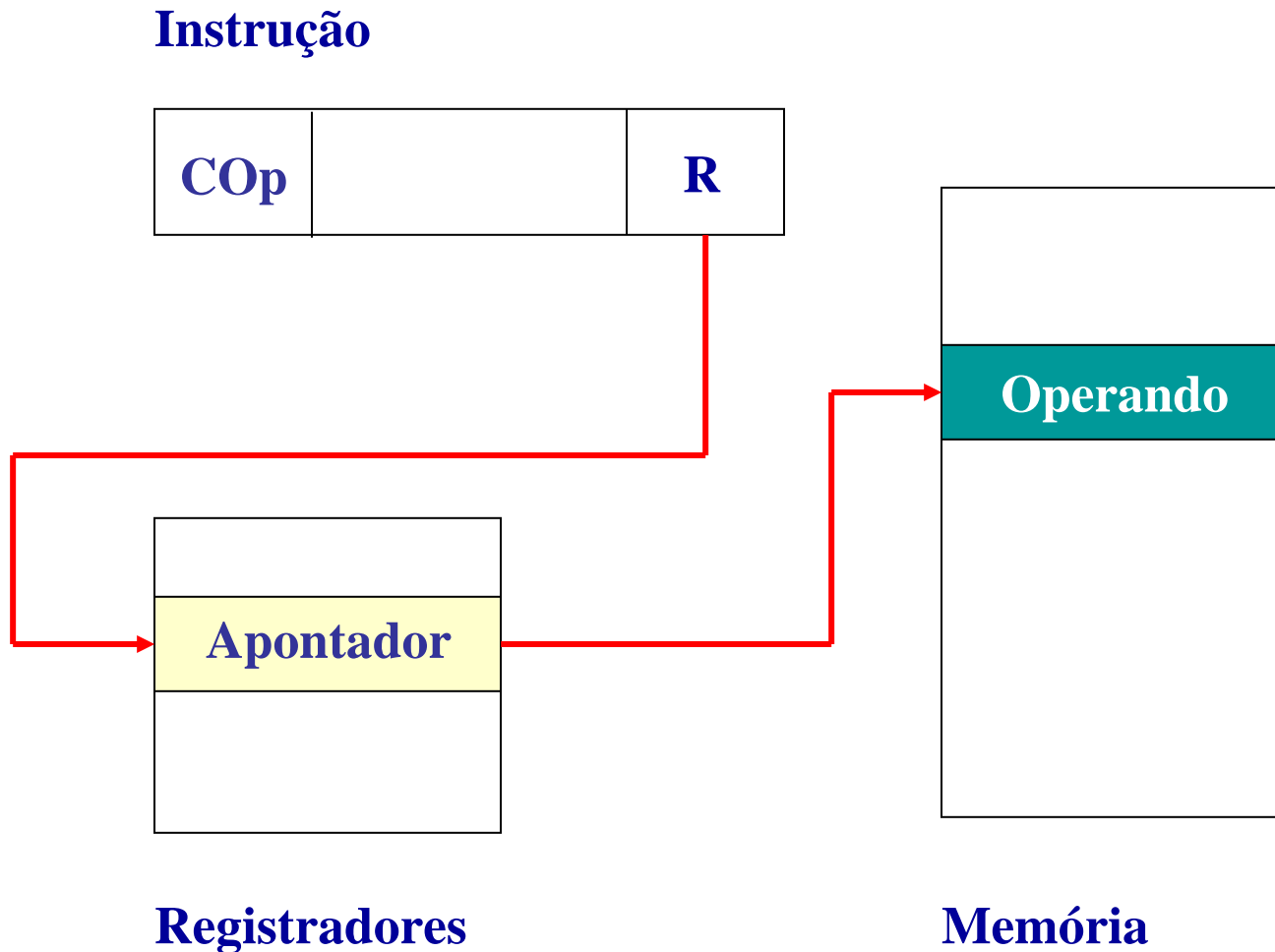


Registradores

Registrador

- Similar ao endereçamento direto, contudo o campo de endereço refere-se a um registrador em vez de memória
- Como há número limitado de registradores, campo de endereço (por registrador) é pequeno
- Não há necessidade de referências à memória \Rightarrow maior rapidez

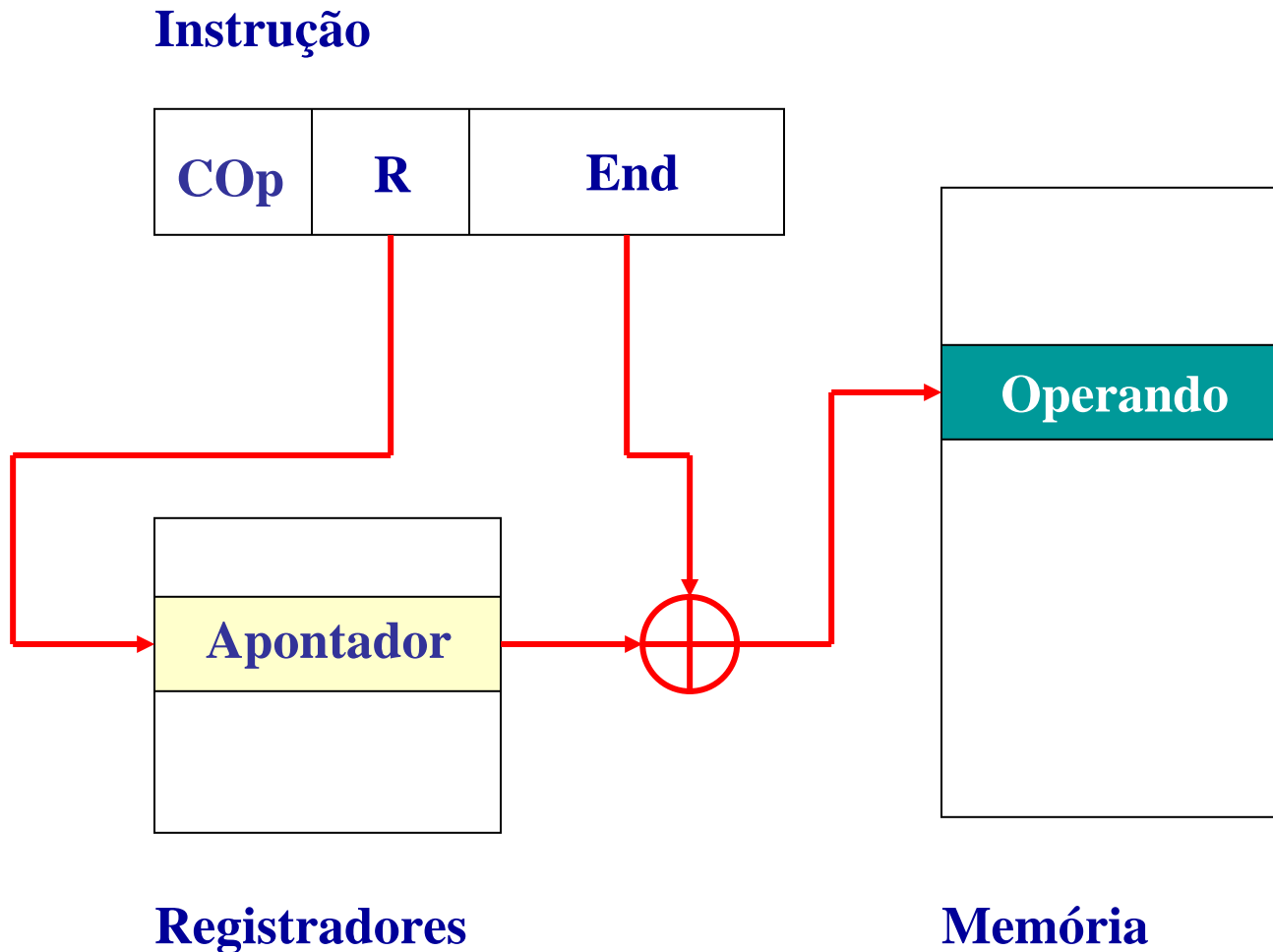
Registrador



Registrador

- Similar ao endereçamento indireto
- Operando é obtido por referência de um apontador mantido em registrador
- Maior espaço de endereçamento (2^N) que o modo registrador
- Uma referência de memória a menos que o modo indireto

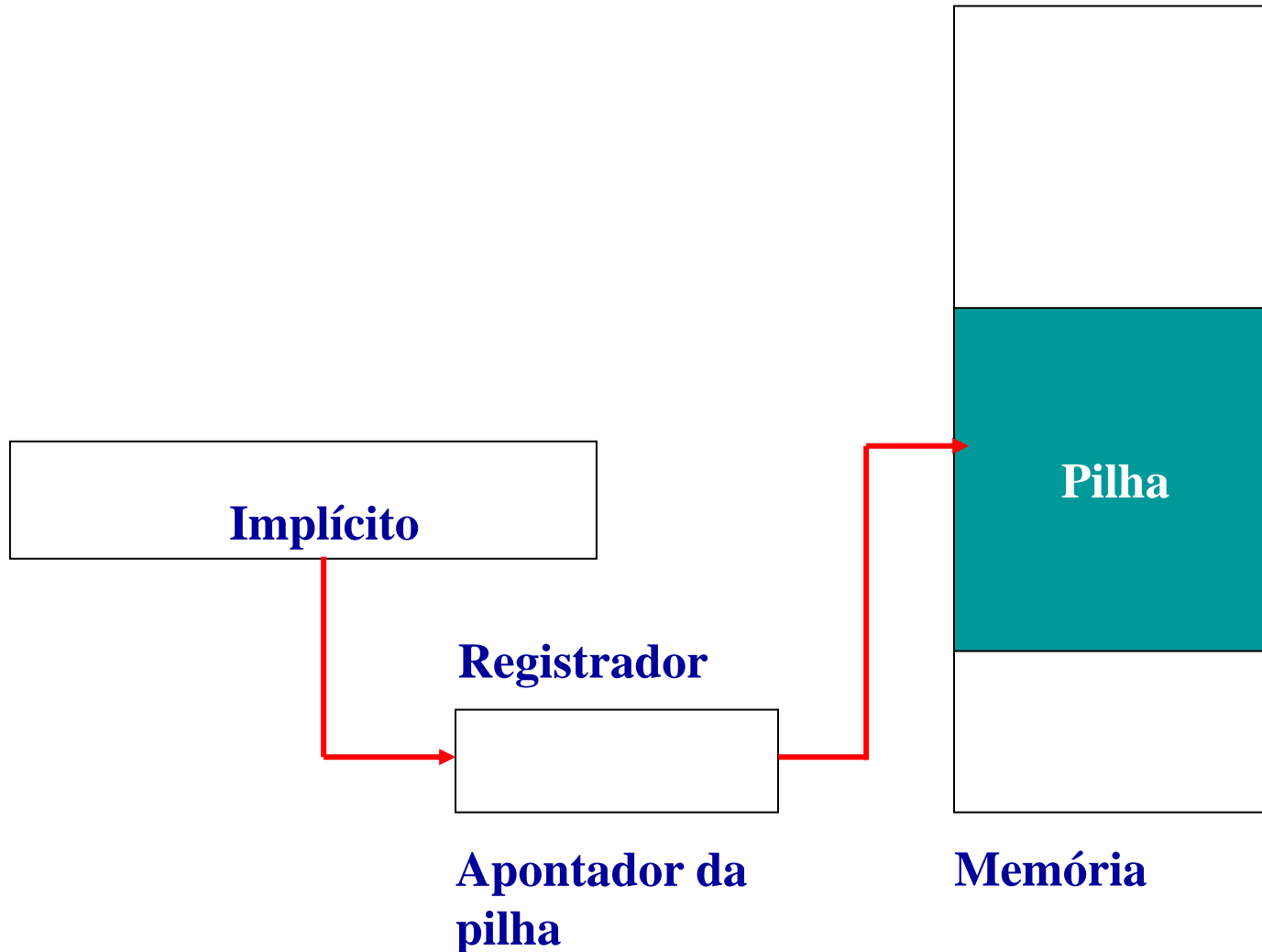
Deslocamento



Deslocamento

- O endereço efetivo é obtido como:
 - $\text{EndEf} = \text{End} + (\text{R})$
- Os três modos mais comuns são:
 - endereçamento relativo
 - endereçamento registrador base
 - endereçamento indexado

Pilha



Pilha

- É uma forma de endereçamento implícito
- As instruções de máquina sempre atuam no(s) operando(s) do topo da pilha

Exercícios

Considere a seguinte instrução em linguagem C:
Resolva matematicamente

$$a = b - (c - d) + e$$

Exercícios

- Considere a seguinte instrução em linguagem C:
- Escreva em linguagem Assembly

$$a = b - (c - d) + e$$

Resolução

$$a = b - (c - d) + e$$