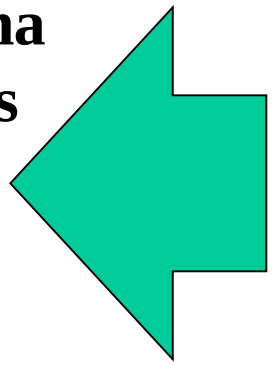
Conjunto de instruções: Características e funções e endereçamento

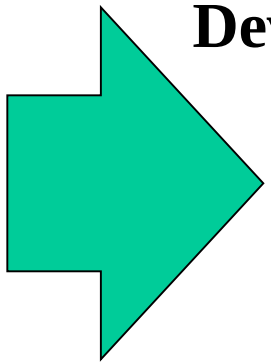
Programador

Designer

INSTRUCTION SET

Ao programar em assembly, toma conhecimento dos registradores e estrutura de memória.





Deve implementar o conjunto de instruções.

Característica das Instruções de Máquina

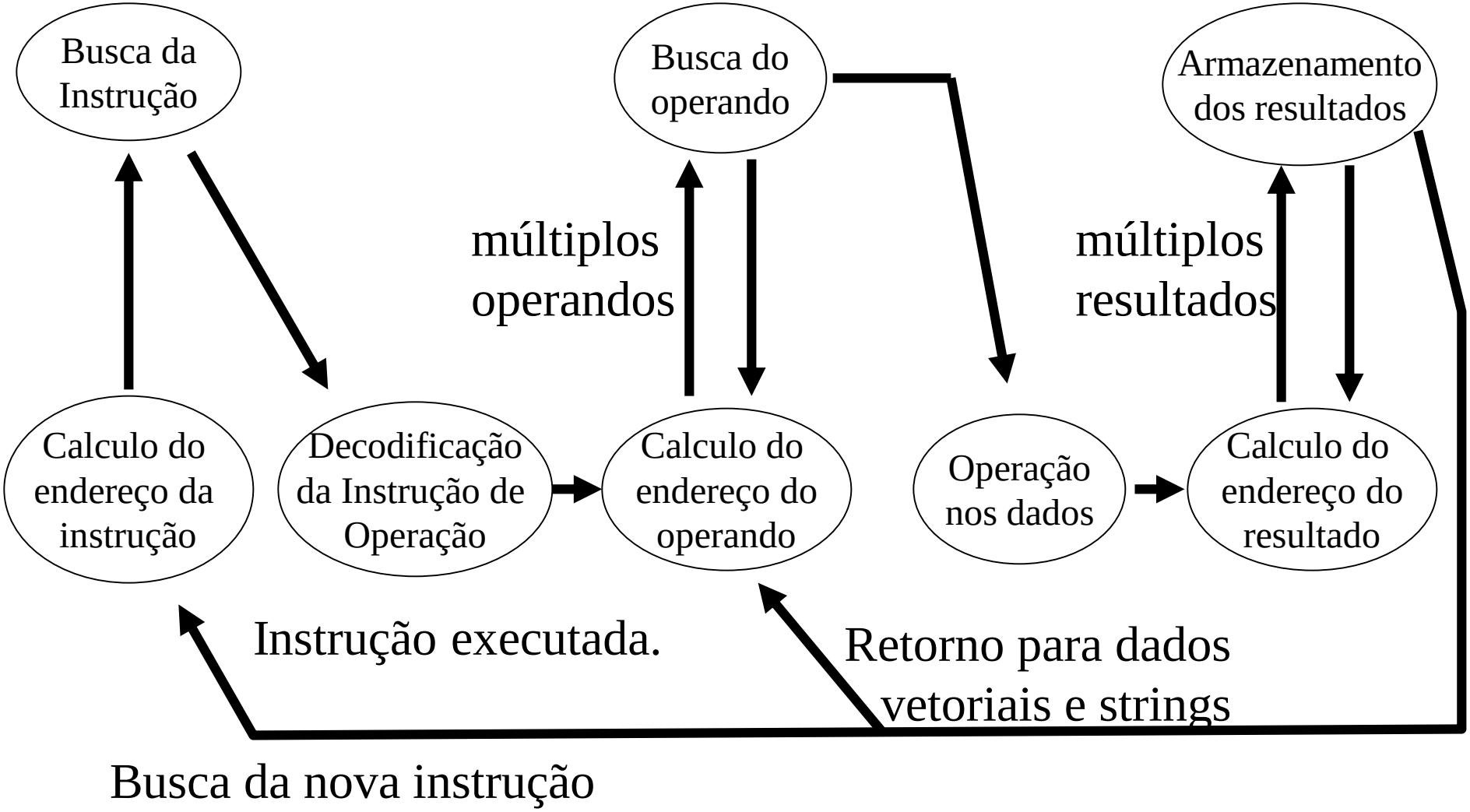
A operação de uma CPU é determinada pelas instruções executadas.

Instruções de máquinas.

O conjunto de diferentes instruções que uma CPU pode executar é chamado de *Conjunto de Instruções*.

Instruction Set

Arquitetura de computadores



Fluxo de execução de uma instrução

Elementos das instruções:

- Código da operação: especifica a operação (opcode);
- Referência aos operandos;
- Referência aos resultados;
- Referência à próxima instrução.

Possíveis localizações dos operandos:

- Memória principal ou virtual;
- Registradores da CPU;
- Dispositivos de I/O.

Representação da Instrução:

Cada instrução é representada por uma seqüência de bits.

Exemplo:

Opcode	referência ao operando	referência ao operando

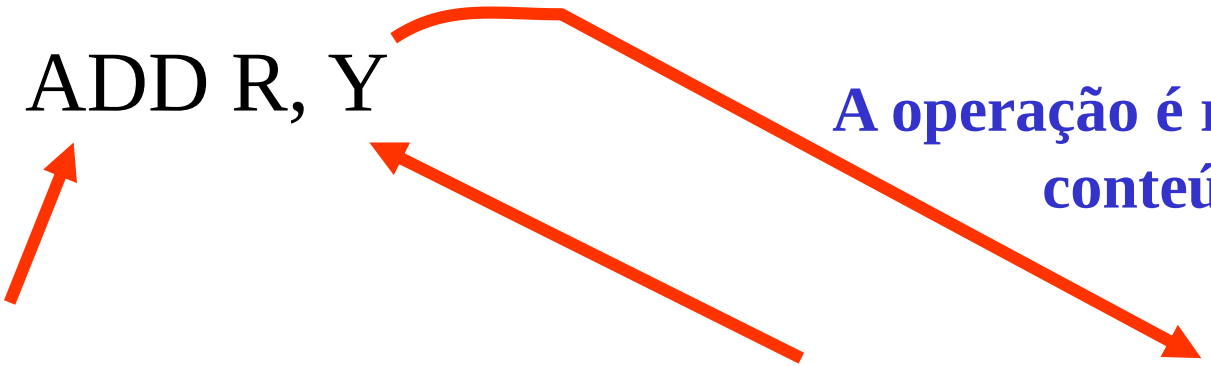
Durante a execução da instrução, a CPU deve extrair o conteúdo de cada campo.

Representação simbólica:

Para facilitar a interpretação das instruções, são usados comandos mnemônicos:

Exemplo:

ADD R, Y



A operação é realizada no conteúdo e não no endereço.

adicionar o valor armazenado em Y no registrador R

Tipos de Instruções:

Exemplo:

$x = x + y$

O computador deve somar o valor armazenado em y ao valor armazenado em x e colocar o resultado em x.

Instruções:

1. Carregar um registrador com o conteúdo de y;
2. Somar o conteúdo de x ao registrador;
3. Armazenar o conteúdo do registrador na posição de memória x.

Arquitetura de computadores

Um computador deve ter um conjunto de instruções que permita a manipulação e operação de dados.

Qualquer programa de alto nível deve ser traduzido para uma linguagem de máquina(por um compilador), assim o conjunto de instruções determina as instruções de uma linguagem de alto nível.

Tipos de Instruções:

- processamento de dados: instruções lógicas e aritméticas;
- armazenamento de dados: instruções de memória;
- movimento de dados: instruções de I/O;
- controle: instruções de teste e de redirecionamento;

Número de endereços:

Uma operação aritmética pode ser unária (um operando) ou binária (dois operandos), e gera um resultado. Como a próxima instrução deve ser buscada, pode-se pensar em usar até 4 endereços para cada instrução.

Considerando-se o endereço da próxima instrução obtido pelo PC (*program counter*), pode-se diminuir para 3 endereços.

Exemplo:

$Y = (A - B) / (C + D \times E)$

Instruções com 3 operandos:

SUB Y,A,B	Y ← A - B
MPY T,D,E	T ← D x E
ADD T,T,C	T ← T + C
DIV Y,Y,T	Y ← Y / T

Instruções com 2 operandos:

MOVE Y,A	Y ← A
SUB Y,B	Y ← Y - B
MOVE T,D	T ← D
MPY T,E	T ← T x E
ADD T,C	T ← T + C
DIV Y,T	Y ← Y / T

Instruções com 1 operando:

LOAD D	AC ← D
MPY E	AC ← AC x E
ADD C	AC ← AC + C
STOR Y	Y ← AC
LOAD A	AC ← A
SUB B	AC ← AC - B
DIV Y	AC ← AC / Y
STOR Y	Y ← AC

rário

Quatro instruções e três operandos . Formato de instruções longos.

Seis instruções. Para evitar a perda de informação, deve-se usar um espaço temporário.

Oito instruções. Deve-se usar um registrador implícito, denominado acumulador.

Exemplo:

$Y = (A - B) / (C + D \times E)$

Instruções com 3 operandos:

SUB Y,A,B	Y ← A - B
MPY T,D,E	T ← D x E
ADD T,T,C	T ← T + C
DIV Y,Y,T	Y ← Y / T

Instruções com 2 operandos:

MOVE Y,A	Y ← A
SUB Y,B	Y ← Y - B
MOVE T,D	T ← D
MPY T,E	T ← T x E
ADD T,C	T ← T + C
DIV Y,T	Y ← Y / T

Instruções com 1 operando:

LOAD D	AC ← D
MPY E	AC ← AC x E
ADD C	AC ← AC + C
STOR Y	Y ← AC
LOAD A	AC ← A
SUB B	AC ← AC - B
DIV Y	AC ← AC / Y
STOR Y	Y ← AC

Com instruções de múltiplos operandos, deve-se utilizar múltiplos registradores possibilitando operações somente entre os registradores

Oit… Deve-se usar um registrador implícito, denominado acumulador.

Arquitetura de computadores

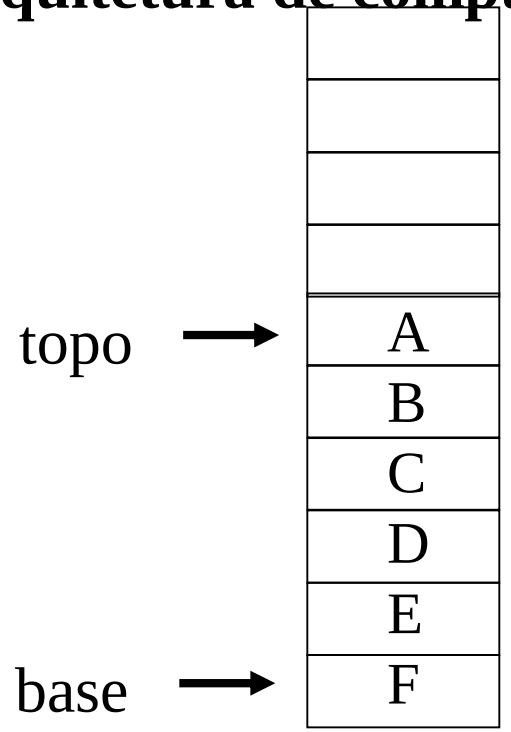
Exercícios:

1) Escreva um programa que calcule a equação Delta da equação de bascará para os quatro modos de uso de operadores (três, dois, um e zero).
   (B x B – (4 x A x C))/ 2 x A

MPY T,B,B	T ← B x B
MPY Y,A,C	Y ← A x C
ADD Y,Y,Y	Y ← Y +Y
ADD Y,Y,Y	Y ← Y +Y
SUB Y,T,Y	Y ← T - Y
ADD T,A,A	T ← A + A
DIV Y,Y,T	Y ← Y / T

Pilhas:

- Uma pilha é um conjunto ordenado de elementos tal que apenas um elemento de cada vez possa ser acessado.
- O ponto de acesso é chamado de topo da pilha.
- Uma pilha é uma lista do tipo LIFO (*least-in-first-ou*t).
- A operação PUSH coloca um novo elemento no topo da pilha.
- A operação POP remove o elemento do topo da pilha.
- Operações binárias (SOMA) usam os dois elementos do topo da pilha e colocam o resultado de volta no topo.
- Operações unárias (NOT) usam o elemento do topo e colocam o resultado de volta no topo.



Como as operações usam os elementos do topo da pilha, os endereços dos operandos podem ser considerados implícitos, não sendo necessário a inclusão nas instruções (instruções com zero operandos).

A implementação de uma pilha é obtida alocando-se um bloco de memória e armazenando em registradores o endereço do topo da pilha, o endereço da base da pilha e o endereço do limite superior da pilha.

Se for tentado realizar um POP quando a pilha está vazia, é retornada uma mensagem de erro.

Se for tentado realizar um PUSH quando a pilha está cheia, é retornada uma mensagem de erro.

Exemplo 1: Y = (A - B) / C

(A - B) / C — notação *in-fixa*

AB-C/ — notação *pós-fixa* ou Polonesa reversa

```
PUSH A
PUSH B
SUBTRACT
PUSH C
DIVIDE
POP Y
```

Exemplo 2: Y = (A - B) / (C + D x E)

(A - B) / (C + D x E) notação *in-fixa*

AB-C DEx+/ notação *pós-fixa*
ou Polonesa reversa

PUSH A
PUSH B
SUBTRACT
PUSH C
PUSH D
PUSH E
MULTIPLY
ADD
DIVIDE
POP Y

Projeto de um conjunto de instruções:

- **Repertório de operações:** quantas, quais e quão complexas devem ser as operações;
- **Tipos de dados:** definições dos operandos;
- **Formato das instruções:** tamanho das instruções, número de operandos, tamanho dos campos;
- **Registradores:** número de registradores que podem ser endereçados;
- **Endereçamento:** modos de endereçamento dos operandos.

Tipos de operandos:

- Endereços
- Números
- Caracteres
- Dados lógicos

Em alguns casos deve-se realizar algumas operações para se obter o endereço na memória principal

Mesmo em processamentos não numéricos existe a necessidade de

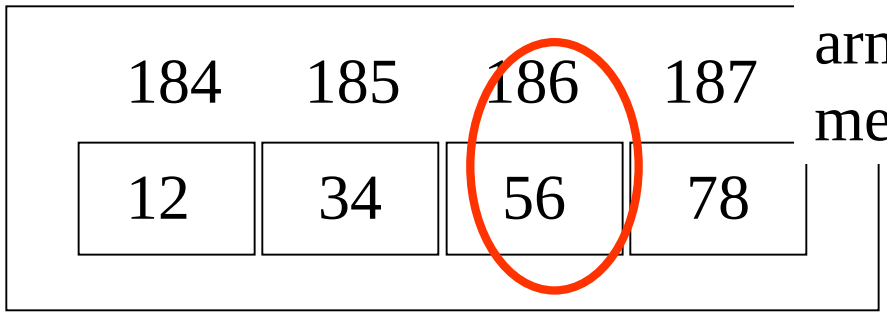
Foram criados códigos com seqüências de bits que representam caracteres. O mais usado é o IRA (International

Em geral, cada palavra é considerada como sendo uma única unidade de dados. No entanto, é possível que uma palavra de n bits seja considerada como formada de n itens individuais.

Ordenamento de Bytes:

Exemplo: Como guardar uma palavra de 32 bits com o valor hexadecimal 12345678 em uma memória com endereçamento de 1 byte iniciando-se no endereço 184?

os bytes mais significativos são armazenados no endereço de menor valor numérico



Big-endian

os bytes mais significativos são armazenados no endereço de maior valor numérico

Little-endian

O problema de armazenamento surge quando é necessário tratar palavras formadas por unidades menores, como uma única informação.

Exemplos de máquinas little-endian:
Pentium II, VAX e Alpha.

Exemplos de máquinas big-endian:
IBM System 370/390, Motorola 680 e Sun SPARC.

Arquitetura de computadores

Exemplo:

```
struct{
  int a;        //0x1112_1314
  int pad;      //
  double b;     //0x2122_2324_2526_27_28
  char* c;      //0x3132_3334
  char d[7];    //'A', 'B', 'C', 'D', 'E', 'F', 'G'
  short e;      //0x5152
  int f;        //0x6162_6364
}s;
```

Big-endian

11	12	13	14				
00	01	02	03	04	05	06	07
21	22	23	24	25	26	27	28
08	09	0A	0B	0C	0D	0E	0F
31	32	33	34	A	B	C	D
10	11	12	13	14	15	16	17
E	F	G		51	52		
18	19	1A	1B	1C	1D	1E	1F
61	62	63	64				
20	21	22	23	24	25	26	27

Little-endian

14	13	12	11				
00	01	02	03	04	05	06	07
28	27	26	25	24	23	22	21
08	09	0A	0B	0C	0D	0E	0F
34	33	32	31	A	B	C	D
10	11	12	13	14	15	16	17
E	F	G		52	51		
18	19	1A	1B	1C	1D	1E	1F
64	63	62	61				
20	21	22	23	24	25	26	27

o endereço de
cada item permanece o mesmo

Arquitetura de computadores

Exercícios:

1) Indique como ficaria os valores, na memória, na representação big-endian e little-endian :
   a) $30_{(16)}$ em números de 16 bits/2 bytes
   b) $303_{(16)}$ em números de 16 bits/2 bytes
   c) $0F3301_{(16)}$ em números de 32 bits/4 bytes

2) Escreva uma função em linguagem C que converta números de quatro bytes de big-endian à little-endian.

Instruction Sets

Endereçamento

Tipos de endereçamento

- Imediato
- Direto
- Indireto
- Registrador
- Registrador Indireto
- Deslocamento
- Pilha

A escolha envolve:

- tamanho do campo de endereçamento;
- flexibilidade;
- número de referências à memória;
- complexidade no cálculo do endereço;

Endereçamento Imediato

O operando é colocado como parte da instrução.

Vantagens:

- não há referências à memória;

Desvantagem:

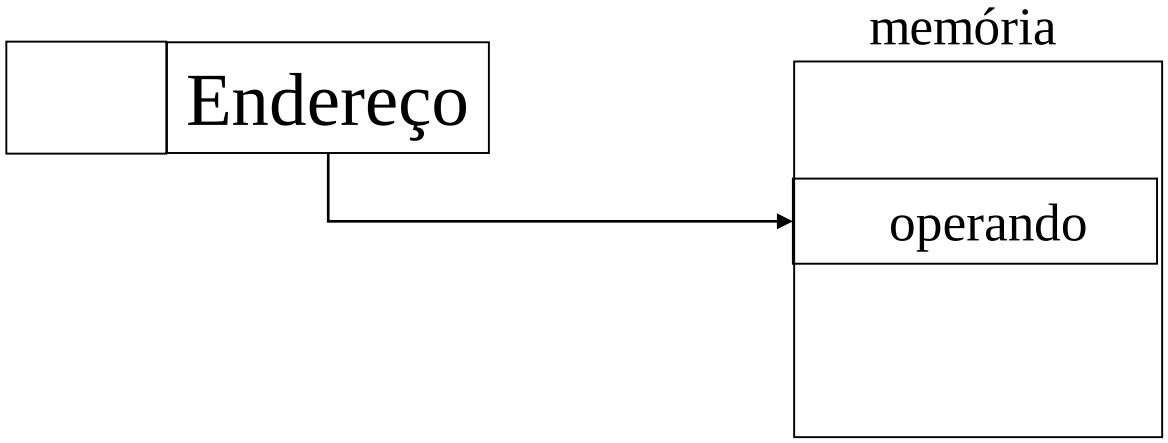
- o tamanho do operando é restrito ao tamanho do campo;

Uso recomendando:

- inicialização de variáveis;
- constantes;

Endereçamento Direto

O endereço do operando é colocado como parte da instrução.



Vantagens:

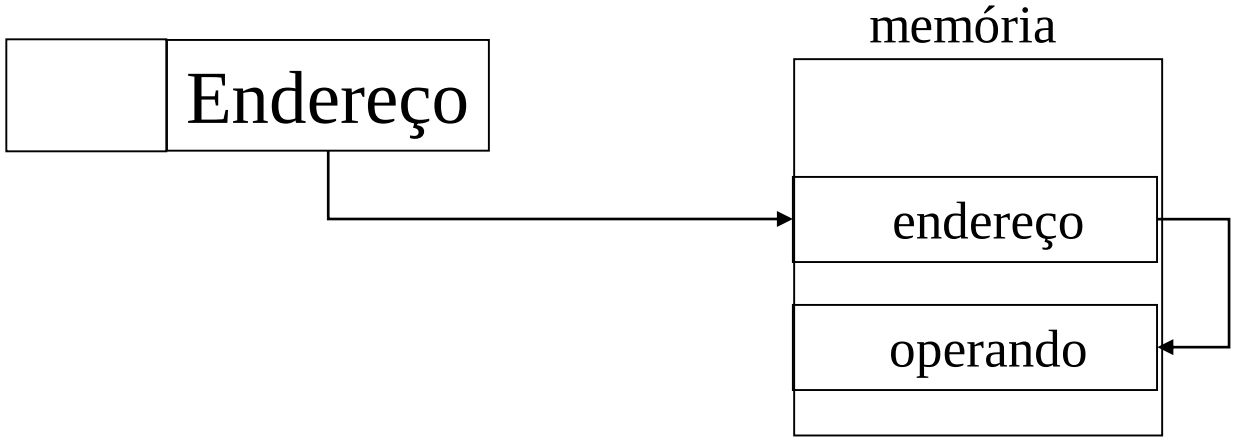
- não requer nenhum tipo de cálculo de endereço;

Desvantagem:

- o número de endereços é limitado pelo tamanho do campo;

Endereçamento Indireto

O campo de endereço se refere a um local na memória com o endereço do operando.



Vantagens:

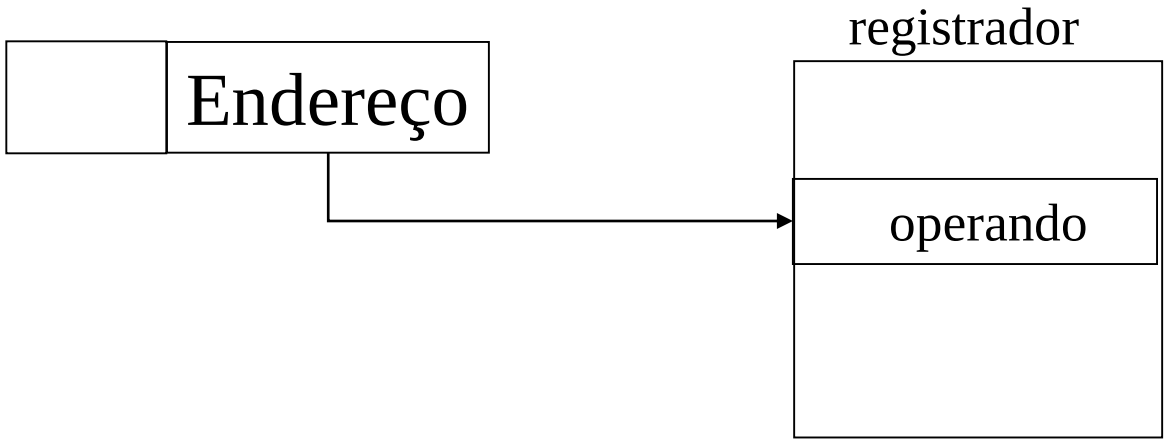
- o número de endereços passa a ser limitado pelo tamanho da palavra;

Desvantagem:

- a execução de uma instrução requer duas referencias à memória;

Endereçamento a Registrador

O campo de endereço se refere a um registrador que contenha o operando.



Vantagens:

- são usados poucos registradores sendo então necessários poucos dígitos para referencia-lo;
- não são necessárias referências a memória;

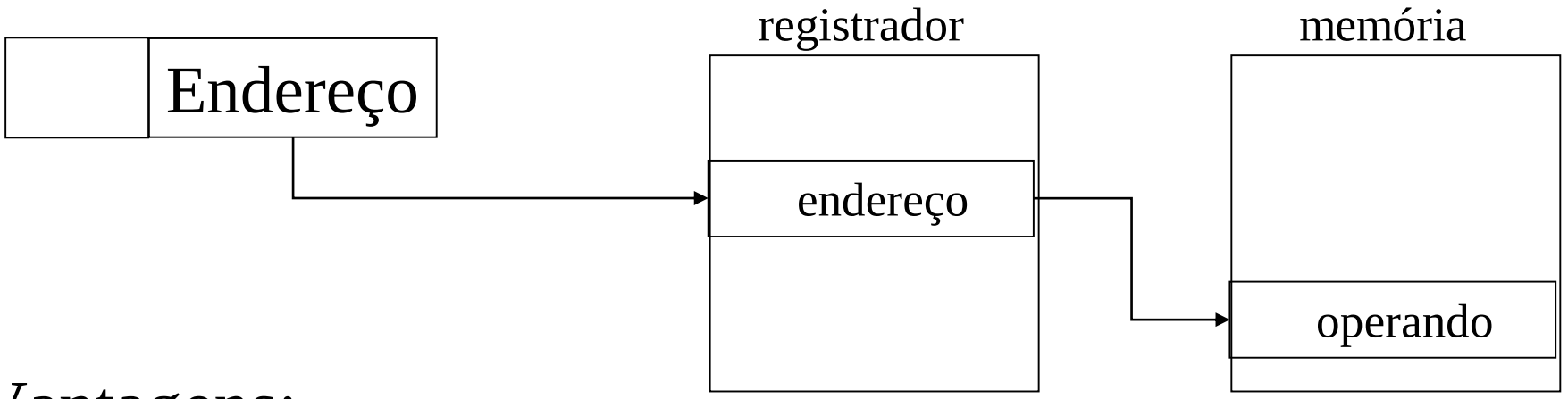
Desvantagem:

- o espaço disponível em registradores é bastante limitado;

Para se usar o endereçamento a registrador de forma eficiente, os operandos devem ser mantidos nos registradores, evitando-se movimentação de operandos entre a memória principal e os registradores.

Endereçamento Indireto a Registrador

O campo de endereço se refere a um registrador que contenha o endereço do operando.

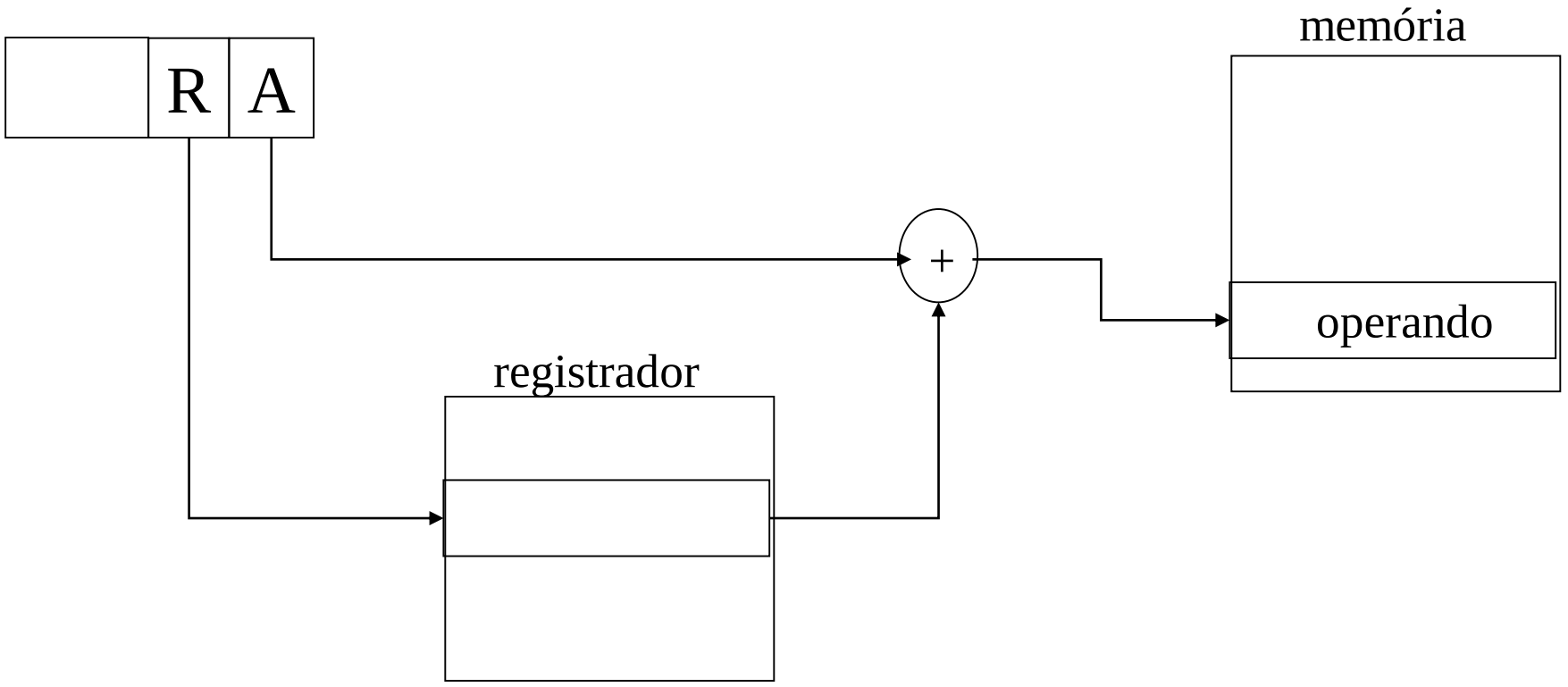


Vantagens:

- o número de endereços é limitado pelo tamanho do registrador;
- a execução de uma instrução requer uma referencia à memória (endereçamento indireto duas);

Endereçamento de Deslocamento

A instrução tem dois campos de endereço, sendo pelo menos um explícito.



Endereçamento de Deslocamento

Endereçamento Relativo

O registrador de referência implícita é o PC. Assim, o endereço efetivo é obtido a partir de um deslocamento relativo ao endereço da instrução.

É mais eficiente, quanto mais agrupados estiverem as instruções e os operandos (conceito de localidade).

Endereçamento de Deslocamento

Endereçamento Baseado em Registradores

O registrador de referência implícita contém um endereço de memória. O endereço efetivo é obtido a partir de um deslocamento relativo a este endereço.

Se baseia também no conceito de localidade.

Endereçamento de Deslocamento

Endereço Indexado

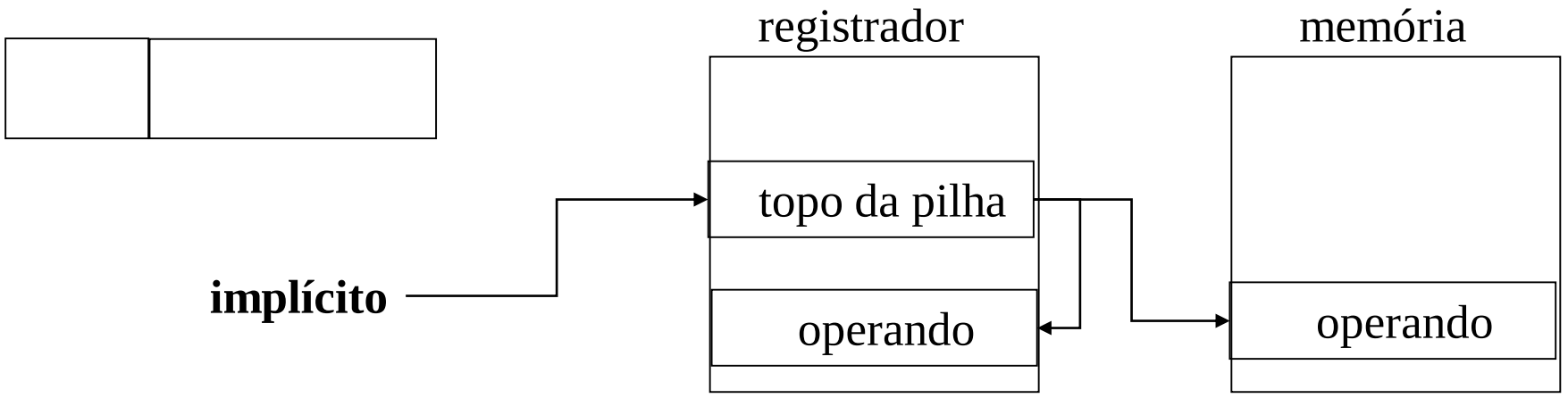
O campo endereço da instrução contém um endereço de memória e o registrador de referência contém um deslocamento relativo a este endereço (inverso ao endereço baseado em registradores) .

Este tipo de endereçamento torna mais eficiente a operação em elementos agrupados de forma seqüencial na memória.

Utilizado em matrizes.

Endereçamento por Pilha

O ponteiro para o topo da pilha é mantido em um registrador.



Formatos de Intrução

Define o arranjo dos bits na instrução.

Deve incluir:

- um opcode;
- o endereço ou o próprio operando ou operandos;
- o modo de endereçamento;

Tamanho da instrução

A sua definição deve levar em conta:

- tamanho da memória;
- organização da memória;
- estrutura de barramento;
- complexidade da CPU;
- velocidade da CPU.

Arquitetura de computadores

Os programadores querem:

- mais opcodes;
- mais operandos;
- mais modos de endereçamento;
- maior número de endereços.

Os programas ficam menores e mais fáceis de serem criados.

Tamanho de instruções cada vez maiores!

para implementar algumas funções.

Maior liberdade na confecção de um programa.

Condições de contorno:

- O tamanho da instrução deve ser um múltiplo inteiro da largura de transferência para/e da memória, para se otimizar o ciclo de busca.

- Taxa de transferência entre a memória e a CPU. Instruções de 16 bits podem ser buscadas na metade do tempo de instruções de 32 bits.

- Um caracter tem 8 bits.

Arquitetura de computadores

Exercícios:

1) Dos modos de endereçamento apresentados melhor trabalharia com as estruturas: matriz e ponteiro conhecidas da linguagem C