

# Organização e Arquitetura de Computadores I

Conjunto de Instruções

# Sumário

- Características de Instruções de Máquina
- Tipos de Operandos
- Tipos de Operações
- Linguagem de Montagem

# Características de Instruções de Máquina

- A operação de uma CPU é determinada pelas instruções que ela executa, conhecidas como **instruções de máquina** ou **instruções do computador**.
- A coleção de diferentes instruções que a CPU é capaz de executar é conhecida como **conjunto de instruções** da CPU.

# Características de Instruções de Máquina

## ● Elementos de Instruções de máquina

- Cada instrução deve conter toda a informação necessária para que a CPU possa executá-la.
- Elementos de instrução de máquina:
  - **Código de operação**, especifica a operação a ser efetuada;
  - **Referência a operando fonte**, operandos que constituem dados de entrada para a operação;
  - **Referência a operando de destino**;
  - **Endereço da próxima instrução**, endereço onde a CPU deverá buscar a próxima instrução.

# Características de Instruções de Máquina

## ● Elementos de Instruções de máquina

- Na maioria dos casos, a próxima instrução é a que segue imediatamente instrução corrente. Neste caso, quando necessário, a instrução deve fornecer um endereço de memória principal ou virtual.

# Organização e Arquitetura de Computadores I

## Características de Instruções de Máquina

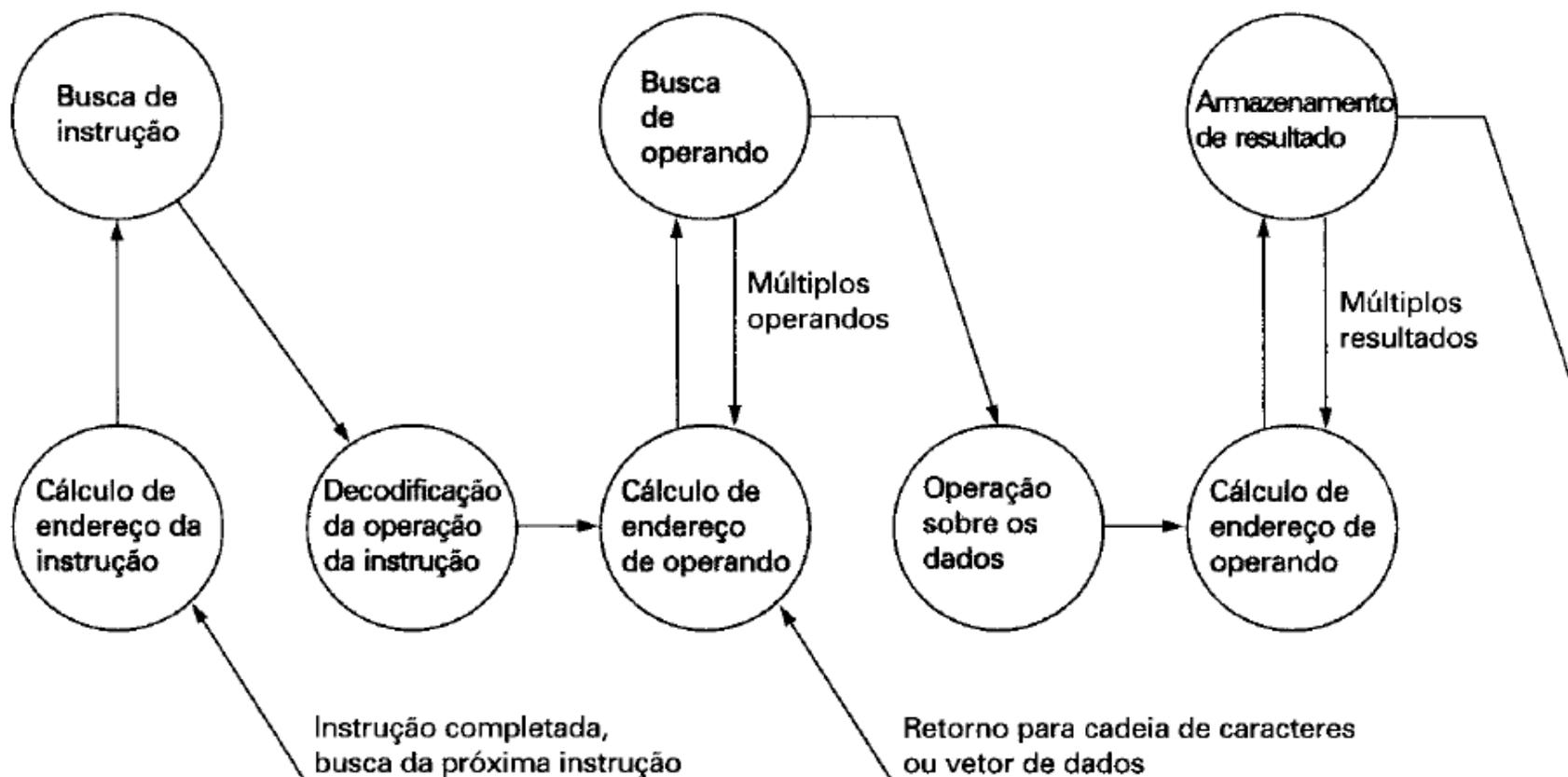


Diagrama de estados do ciclo de instruções

# Características de Instruções de Máquina

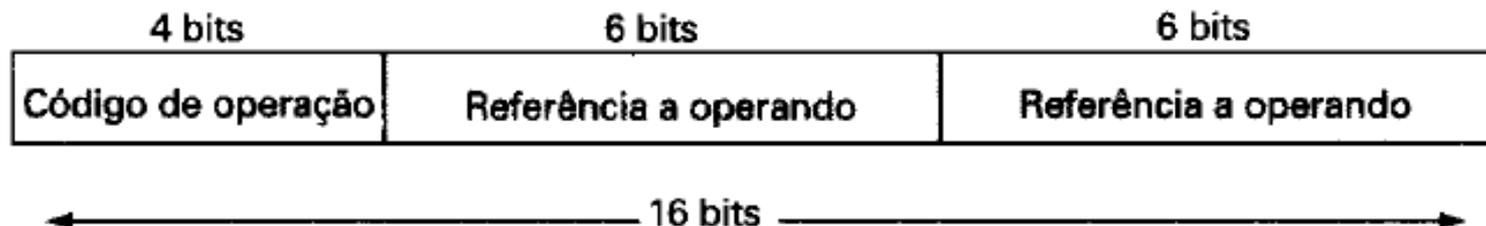
## ● Elementos de Instruções de máquina

- Os operandos fonte e destino podem estar localizados em uma das seguintes áreas:
  - **Memória principal ou virtual;**
  - **Registrador da CPU**, geralmente a CPU tem um ou mais registradores que podem ser referenciados pelas instruções de máquina;
  - **Dispositivo de E/S**, a instrução deve especificar um módulo de E/S e um dispositivo para operação.

# Características de Instruções de Máquina

## ● Representação de instruções

- Uma instrução é dividida em campos, correspondentes aos elementos da instrução.
- Geralmente é usado mais de um formato de instrução.
- Durante a execução, uma instrução é lida em um **registrador de instruções (IR)**.



# Características de Instruções de Máquina

## ● Representação de instruções

- A representação simbólica para instruções de máquina, é usada para facilitar a leitura e escrita das representações binárias das instruções de máquina pelo homem.
- Os códigos de operação são representados por abreviações, chamadas **mnemônicos**, que indicam a operação a ser efetuada.

# Características de Instruções de Máquina

## ● Representação de instruções

- Exemplos de mnemônicos:

ADD	Adição
SUB	Subtração
MPY	Multiplicação
DIV	Divisão
LOAD	Carregar dados da memória
STOR	Armazenar dados na memória

# Características de Instruções de Máquina

## ● Representação de instruções

- Os operandos também são representados de maneira simbólica:

ADD R, Y

Onde: R, refere-se a um registrador

Y, refere-se a um endereço de memória

# Características de Instruções de Máquina

## ● Tipos de instrução

- Uma única instrução em linguagem de alto nível pode requerer várias instruções de máquina.

### **alto nível**

$X = X + Y$  , sendo as posições de memória,  $X=513$  e  $Y=514$ .

### **instruções de máquina**

- 1.carregar um registrador com o conteúdo da posição de memória 513;
- 2.adicionar o conteúdo da posição de memória 534 ao registrador;
- 3.armazenar o conteúdo do registrador na posição de memória 513.

# Características de Instruções de Máquina

## ● Tipos de instrução

- Em uma linguagem de alto nível, as operações são expressas de uma maneira algébrica concisa, usando variáveis.
- Em uma linguagem de máquina, as operações são expressas de maneira mais básica, envolvendo a movimentação de dados de e para registradores.
- Um computador deve ter um conjunto de instruções que permita ao usuário formular qualquer tarefa de processamento de dados.

# Características de Instruções de Máquina

## ● Tipos de instrução

- Qualquer programa em uma linguagem de alto nível deve ser traduzido para uma linguagem de máquina, para que possa ser executado.
- O conjunto de instruções de máquina deve ser suficiente para expressar qualquer comando de uma linguagem de alto nível.

# Características de Instruções de Máquina

## ● Tipos de instrução

- **Processamento de dados**, instruções aritméticas e lógicas
  - **Armazenamento de dados**, instruções de memória
  - **Movimentação de dados**, instruções de E/S
  - **Controle**, instruções de teste e desvio
- **Instruções aritméticas** fornecem a capacidade computacional para processamento de dados numéricos.
  - **Instruções lógicas** (booleanas) operam sobre bits de uma palavra, como bits e não como números.
  - **Instruções de memória** move dados entre a memória e os registradores.
  - **Instruções de teste** são usadas para testar o valor de uma palavra de dados ou o estado de uma computação.
  - **Instruções de desvio** são utilizadas para desviar a execução do programa para uma nova instrução.

# Características de Instruções de Máquina

## ● Número de endereços

- Uma das maneiras tradicionais de descrever uma arquitetura é em termos do número de endereços contidos em cada instrução.
- A maioria das instruções tem um, dois ou três endereços de operando, **sendo implícito o endereço da próxima instrução.**

## Organização e Arquitetura de Computadores I

# Características de Instruções de Máquina

Instrução		Comentário
SUB	Y, A, B	$Y \leftarrow A - B$
MPY	T, D, E	$T \leftarrow D \times E$
ADD	T, T, C	$T \leftarrow T + C$
DIV	Y, Y, T	$Y \leftarrow Y \div T$

(a) Instruções com três endereços

Instrução		Comentário
MOVE	Y, A	$Y \leftarrow A$
SUB	Y, B	$Y \leftarrow Y - B$
MOVE	T, D	$T \leftarrow D$
MPY	T, E	$T \leftarrow T \times E$
ADD	T, C	$T \leftarrow T + C$
DIV	Y, T	$Y \leftarrow Y \div T$

(b) Instruções com dois endereços

Instrução		Comentário
LOAD	D	$AC \leftarrow D$
MPY	E	$AC \leftarrow AC \times E$
ADD	C	$AC \leftarrow AC + C$
STOR	Y	$Y \leftarrow AC$
LOAD	A	$AC \leftarrow A$
SUB	B	$AC \leftarrow AC - B$
DIV	Y	$AC \leftarrow AC \div Y$
STOR	Y	$Y \leftarrow AC$

(c) Instruções com um endereço

**Programas para executar o comando  $Y = (A-B) \div (C + D \times E)$**

# Características de Instruções de Máquina

## ● Número de endereços

- Formato de instruções com três endereços não são muito comuns, porque resultam em instruções de tamanho relativamente grande.
- No caso de instruções com dois endereços e das operações binárias, um dos endereços referencia tanto um operando quanto o resultado.
- Instruções de apenas um endereço são ainda mais simples. E nesse caso um segundo endereço deve ser implícito (acumulador – AC).

# Características de Instruções de Máquina

## ● Número de endereços

- É possível utilizar um formato de instrução com nenhum endereço, para alguns tipos de instrução. **Esse formato se aplica a uma organização de memória especial, denominada pilha.**
- Uma pilha consiste em um conjunto de posições de memória, que são manipuladas de maneira que cada leitura efetuada sobre a pilha recupere o último dado nela armazenado (topo da pilha), retirando-o da pilha.

# Características de Instruções de Máquina

## ● Número de endereços

- Poucos endereços por instrução resultam em instruções de menor extensão e mais primitivas, que requerem uma CPU menos complexa. Por outro lado, o número de instruções por programa é maior, o que, em geral, resulta em maior tempo de execução e em programas mais complexos.
- Com instruções de múltiplos endereços, é comum haver múltiplos registradores de propósito geral.

# Características de Instruções de Máquina

## ● Número de endereços

- Como referências a registradores são mais rápidas do que referências à memória, a execução dessas instruções é mais rápida.
- **Em razão da flexibilidade e da capacidade para usar múltiplos registradores, a maioria das máquinas modernas emprega instruções de dois ou de três endereços.**

# Características de Instruções de Máquina

## ● Número de endereços

- Existem vários fatores que influem na escolha do número de endereços, além dos já vistos:
  - Decidir se um endereço se refere a uma posição de memória ou a um registrador;
  - Uma máquina pode oferecer uma variedade de modos de endereçamento.

# Características de Instruções de Máquina

## ● Projeto do conjunto de instruções

- O projeto do conjunto de instruções é muito complexo, pois afeta diversos aspectos do sistema.
- Define muitas das funções desempenhadas pela CPU e, portanto, tem efeito significativo sobre a implementação da CPU.

# Características de Instruções de Máquina

## ● Projeto do conjunto de instruções

- Algumas das questões mais importantes são:
  - **Repertório de operações**, quantas e quais operações devem ser fornecidas e o quão complexas elas serão;
  - **Tipos de dados**, aos quais as operações são efetuadas;
  - **Formatos de instrução**;
  - **Registradores**, a quantidade e qual o propósito de cada um;
  - **Endereçamento**, como o endereço de um operando pode ser especificado.

# Tipos de Operandos

- As classes de dados mais importantes são:
  - Endereços
  - Números
  - Caracteres
  - Dados lógicos

# Tipos de Operandos

## ● Números

- Os números armazenados no computador são limitados em dois sentidos:
  - Magnitude de números representáveis;
  - Precisão (números de ponto flutuante).
- Três tipos de dados numéricos são comuns:
  - Número inteiro ou de ponto fixo;
  - Número de ponto flutuante;
  - Número decimal (utiliza código de 4 bits conhecido como BCD – *binary coded decimal*).

# Tipos de Operandos

## ● Caracteres

- O código de caracteres mais usado é o **Alfabeto de Referência Internacional** (*International Reference Alphabet – IRA*), conhecido como **ASCII** (*American Standard Code for Information Interchange*).
- No ASCII, cada caractere é representado por um conjunto de 7 bits. Sendo o 8º 0 ou um bit de paridade.
- Outro código usado para codificar caracteres é o **EBCDIC** (*Extended Binary Coded Decimal Interchange Code*).

# Tipos de Operandos

## ● Dados lógicos

- Algumas vezes é útil considerar uma unidade de  $n$  bits como composta de  $n$  itens de dado, de 1 bit cada, com valor 0 ou 1. Quando um dado é visto dessa maneira, ele é considerado um dado lógico.
- Economiza espaço na memória quando queremos armazenar um vetor de dados booleano.
- Possibilita a manipulação de bits de um item de dado, o que pode ser requerido em determinadas situações.

# Tipos de Operandos

## ● Dados lógicos

- Um mesmo dado pode ser tratado algumas vezes como um dado lógico e outras vezes como um dado numérico ou textual.
- **O 'tipo' de um dado é determinado pela operação efetuada sobre ele (comum em linguagens de máquina).**

# Tipos de Operações

- O número de códigos de operações varia muito de máquina para máquina. Entretanto, o mesmo conjunto de classes de operações é encontrado em todas as máquinas. São elas:
  - Operações de transferência de dados
  - Operações aritméticas
  - Operações lógicas
  - Operações de conversão
  - Operações de E/S
  - Operações de controle de sistema
  - Operações de transferência de controle

# Tipos de Operações

## ● Operações de transferência de dados

- O tipo mais fundamental de instrução de máquina.
- Informações que este tipo de operação deve especificar:
  1. Especificar os endereços dos operandos fonte e de destino da operação
  2. Indicar o tamanho dos dados a serem transferidos
  3. Especificar o modo de endereçamento de cada operando
- A escolha das instruções de transferência de dados que devem ser incluídas em um conjunto de instruções mostra vários tipos de decisão que um projetista deve tomar.

# Tipos de Operações

## ● Operações aritméticas

- A maioria das máquinas fornece operações aritméticas básicas para soma, subtração, multiplicação e divisão. Oferecidas, invariavelmente, para números inteiros com sinal. Podendo ser oferecidas também para números da representação decimal empacotada (BCD) e de ponto flutuante.
- A execução de uma instrução aritmética pode envolver transferência de dados, para fornecer os valores dos operandos como entrada para a ULA e para armazenar na memória o valor obtido como saída.

# Tipos de Operações

## ● Operações lógicas

- Existe uma variedade de operações comuns para manipulação de bits individuais de uma palavra ou de qualquer unidade endereçável.

P	Q	NOT P	P AND Q	P OR Q	P XOR Q	P=Q
0	0	1	0	0	0	1
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	1	0	1	1	0	1

# Tipos de Operações

## ● Operações lógicas

- Essas operações lógicas podem se aplicadas bit a bit a unidades de dados lógicos de n bits, então:

$$(R1) = 10100101$$

$$(R2) = 00001111$$

$$(R1) \text{ AND } (R2) = 00000101$$

- A maioria das máquinas fornece uma variedade de funções de deslocamento e rotação de bits.

# Tipos de Operações

## ● Operações de conversão

- São aquelas que mudam ou operam sobre o formato de dados, como a conversão de um número decimal para binário.

## ● Operações de entrada/saída

- Muitas implementações fornecem apenas algumas instruções de E/S, com ações específicas determinadas por meio de parâmetros, códigos ou palavras de comando.

# Tipos de Operações

## ● Operações de transferência de controle

- Uma fração significativa das instruções de qualquer programa tem como função alterar a sequência de execução de instruções.
- Nessas instruções, a CPU atualiza o contador de programa com o endereço de alguma outra instrução armazenada na memória.

# Tipos de Operações

## ● Operações de transferência de controle

- Existem diversas razões para a utilização de operações de transferência de controle:
  - A repetida execução (milhares de vezes) de uma mesma instrução ou instruções.
  - Tomada de decisões.
  - Implementação de programas grandes e/ou complicados.

# Tipos de Operações

## ● Instruções de desvio

- Uma instrução de desvio tem como um de seus operandos o endereço da próxima instrução a ser executada.
- Com frequência é um desvio condicional, ou seja, o contador do programa é atualizado com o endereço especificado no operando apenas se uma condição é satisfeita. Caso contrário, o contador de programa é incrementado.

# Tipos de Operações

## ● Instruções de salto

- Outra forma comum de instrução de transferência de controle é a instrução de salto. Ela inclui um endereço de desvio implícito.
- Um salto indica, geralmente, que a execução de uma instrução da sequência de instruções deve ser omitida.

# Tipos de Operações

## ● Instruções de chamada de procedimento

- Um procedimento é um subprograma autocontido, que é incorporado em um programa maior. **Um procedimento pode ser chamado em qualquer ponto do programa.**
- Uma chamada a um procedimento instrui o processador a executar todo o procedimento, e então, retornar ao ponto em que ocorreu a chamada.
- Permite economia da memória e modularidade.

# Tipos de Operações

## ● Instruções de chamada de procedimento

- O mecanismo de controle de procedimentos envolve duas instruções básicas:
  - Uma instrução de chamada, que desvia a execução da instrução corrente para o início do procedimento;
  - Uma instrução de retorno, que provoca o retorno da execução do procedimento para o endereço em que ocorreu a chamada.

# Linguagem de Montagem

- A CPU é capaz de entender e executar instruções de máquina, constituídas de números binários armazenados no computador.
- Se formos montar um programa com uma função simples,  $N = I + J + K$ , é reservada uma área de memória a partir do endereço 201 para armazenar as variáveis ( $I=201$ ,  $J=202$ ,  $K=203$ ,  $N=204$ ), o programa deverá ter os seguintes passos:
  1. Carrega o conteúdo do endereço 201 em AC
  2. Adiciona o conteúdo do endereço 202 a AC
  3. Adiciona o conteúdo do endereço 203 a AC
  4. Armazena o conteúdo de AC no endereço 204

# Linguagem de Montagem

- O programa escrito em linguagem binária, é tedioso de se implementar e propenso a erros.

Endereço	Conteúdo			
101	0010	0010	0000	0001
102	0001	0010	0000	0010
103	0001	0010	0000	0011
104	0011	0010	0000	0100
201	0000	0000	0000	0010
202	0000	0000	0000	0011
203	0000	0000	0000	0100
204	0000	0000	0000	0000

# Linguagem de Montagem

● Uma forma melhor é	Endereço	Conteúdo
usar a notação	101	2201
hexadecimal para	102	1202
programar. Precisando	103	1203
de um programa que	104	3204
“traduza” do número	201	0002
hexadecimal para	202	0003
binário, antes de	203	0004
carregar na memória.	204	0000

# Linguagem de Montagem

- E para facilitar mais ainda, podemos usar um nome simbólico, ou mnemônico, para cada instrução. Cada linha do programa consiste em três campos, separados por espaços: endereço, símbolo, endereço ou valor (dependendo da instrução).

101	LDA	201
102	ADD	202
103	ADD	203
104	STA	204
201	DAT	2
202	DAT	3
203	DAT	4
204	DAT	0

# Linguagem de Montagem

- O uso de um programa simbólico torna a tarefa de programação mais fácil, mas ainda é ineficaz. Pois temos de fornecer um endereço absoluto para cada palavra. O que gera vários problemas.
- A utilização de endereços simbólicos é a solução. Ao contrário de se utilizar um endereço absoluto, usamos um endereço simbólico.

# Linguagem de Montagem

- Utilizando, mnemônicos para os comandos e endereços simbólicos, temos uma **linguagem de montagem**, a qual é traduzida através de um montador.
- O montador, além de traduzir os mnemônicos, deve associar os endereços simbólicos com endereços absolutos.
- A linguagem de montagem foi o primeiro passo para o desenvolvimento das linguagens de alto nível.

# Linguagem de Montagem

<u>Rótulo</u>	<u>Operação</u>	<u>Operando</u>
FORMUL	LDA	I
	ADD	J
	ADD	K
	STA	N
I	DATA	2
J	DATA	3
K	DATA	4
N	DATA	0