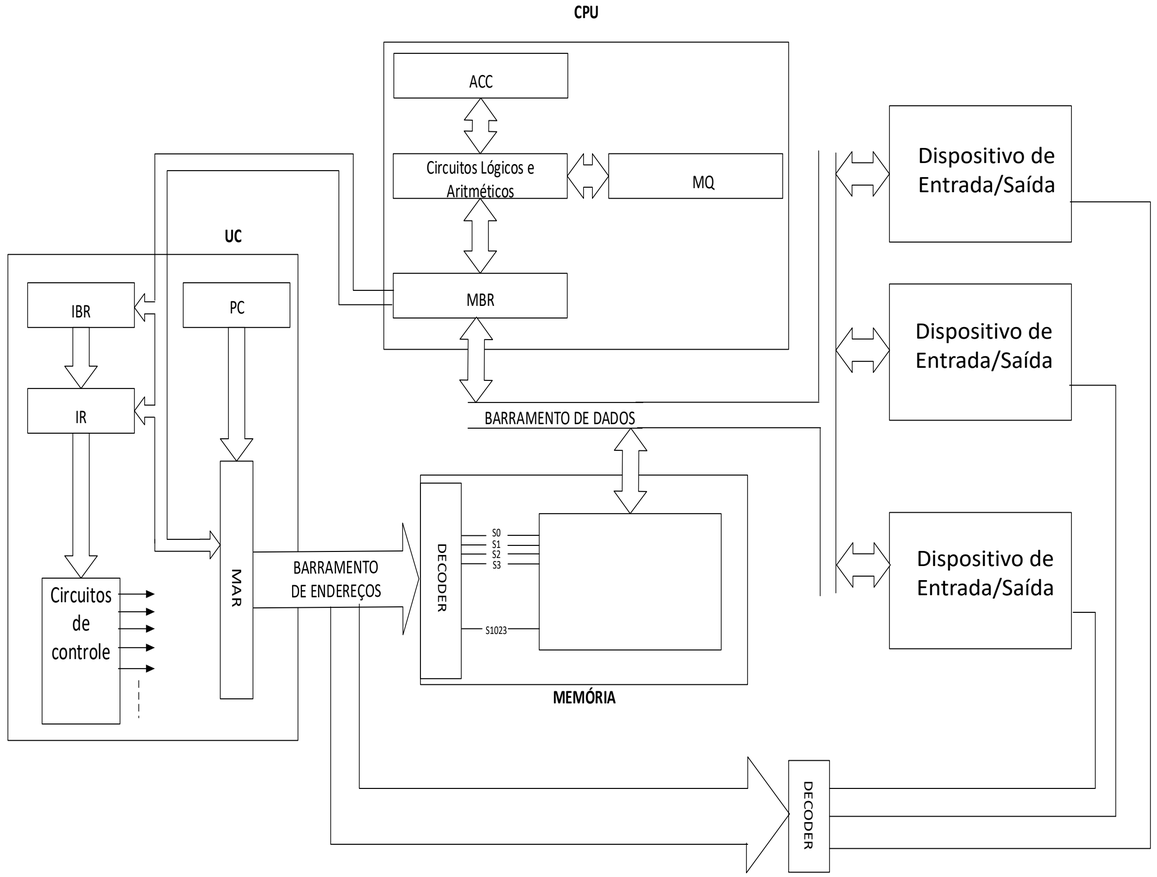
Curso: Engenharia de Computação

Arquitetura de Computadores

Prof. Clayton J A Silva, MSc
clayton.silva@professores.ibmec.edu.br

ibmec

IAS - a máquina de von Neumman
Referência para as arquiteturas modernas



Gargalos: limitações de performance

Quais são os elementos das arquiteturas modernas?

Elementos do sistema de computação

- Memória
- Barramento
- Processadores
- Dispositivos de Entrada/Saída

Elementos do sistema de computação

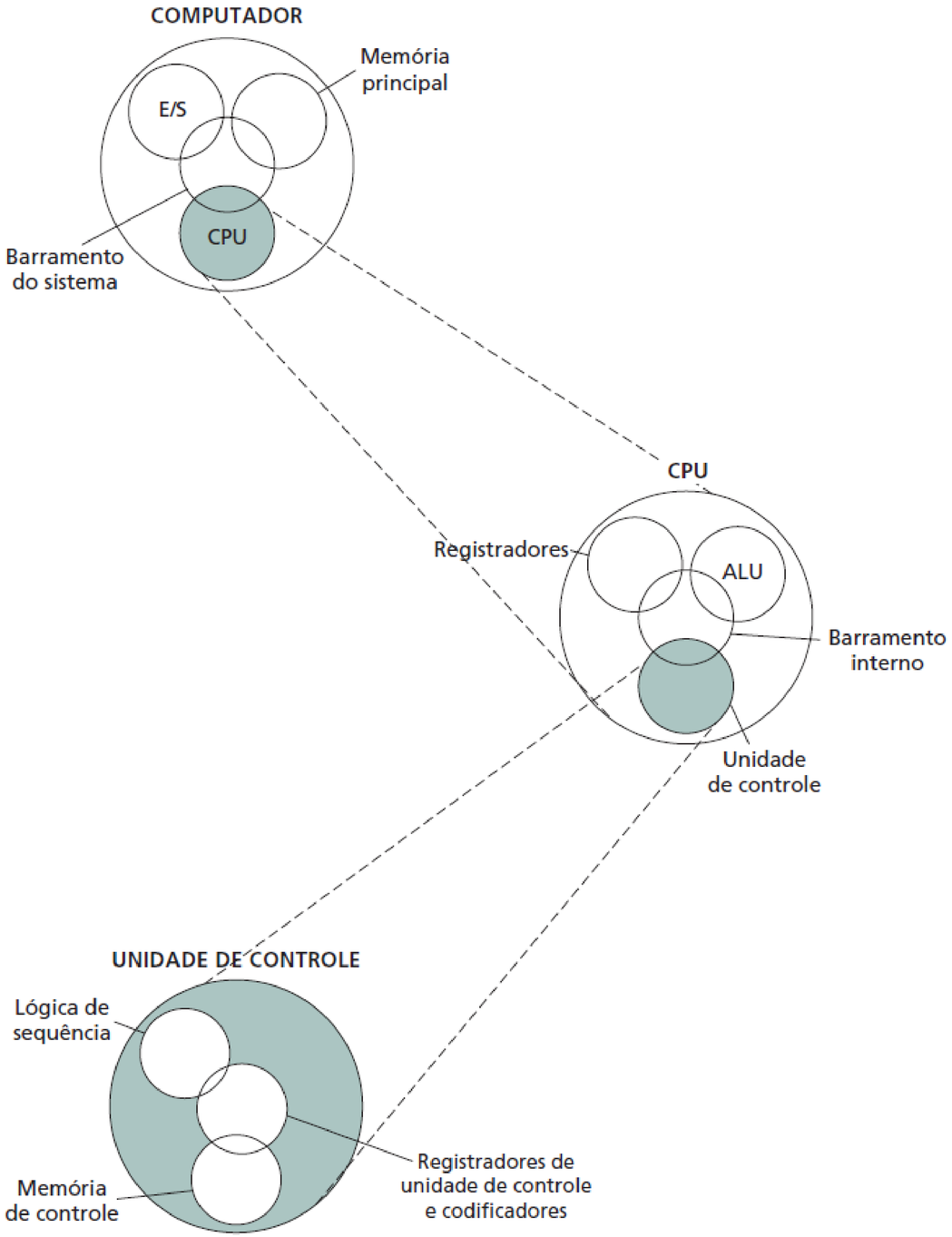
- Memória
- Barramento
- Processadores
- Dispositivos de Entrada/Saída

Elementos do sistema de computação

- Memória
- Barramento
- Processadores
- Dispositivos de Entrada/Saída

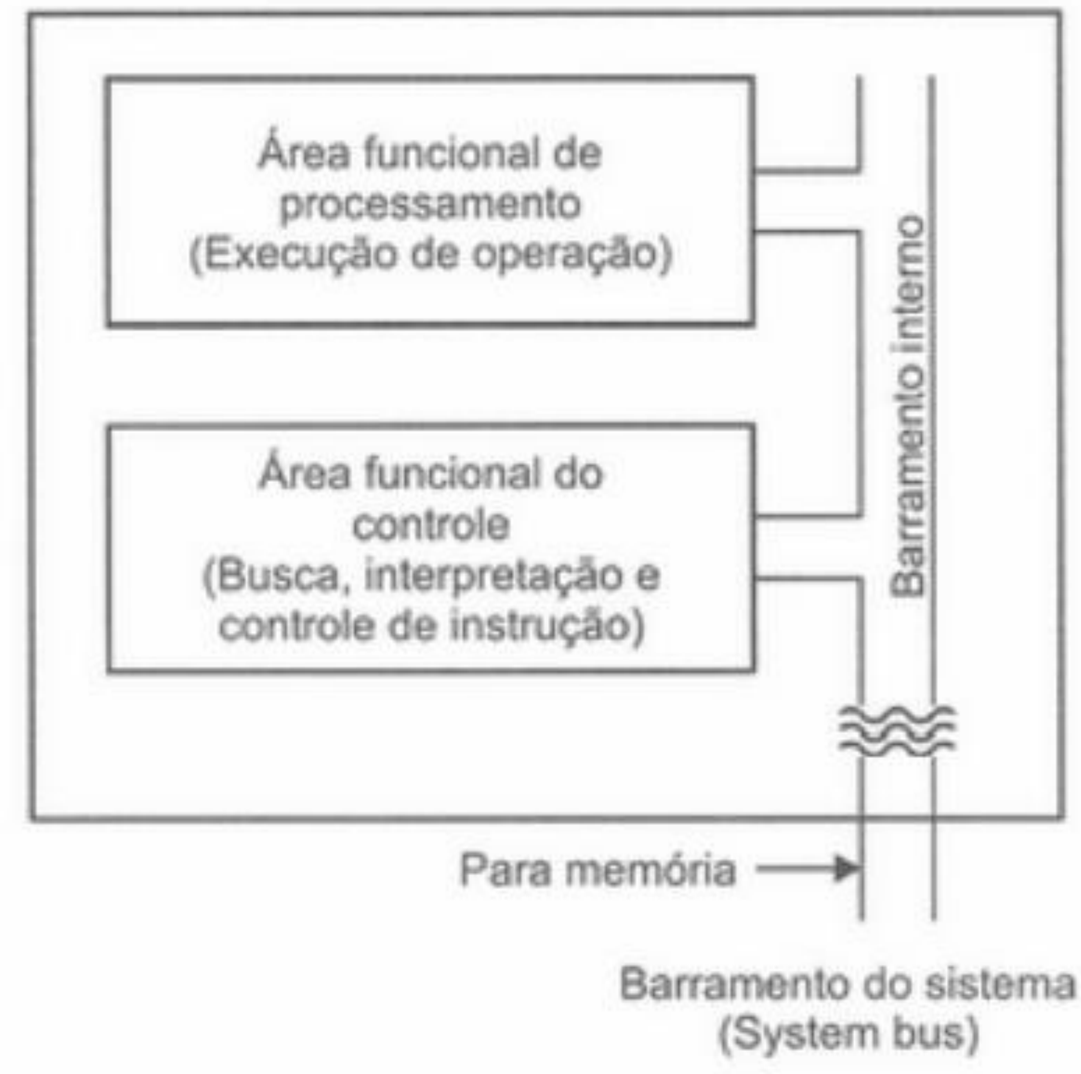
Visão geral

- **Unidade de controle**: controla a operação da CPU e, portanto, do computador.
- **Unidade lógica e aritmética (ULA)**: executa as funções de processamento de dados do computador.
- **Registradores**: proporciona armazenagem interna na CPU.
- **Interconexão da CPU**: alguns mecanismos que proporcionam comunicação entre a unidade de controle, ULA e registradores.



Visão geral

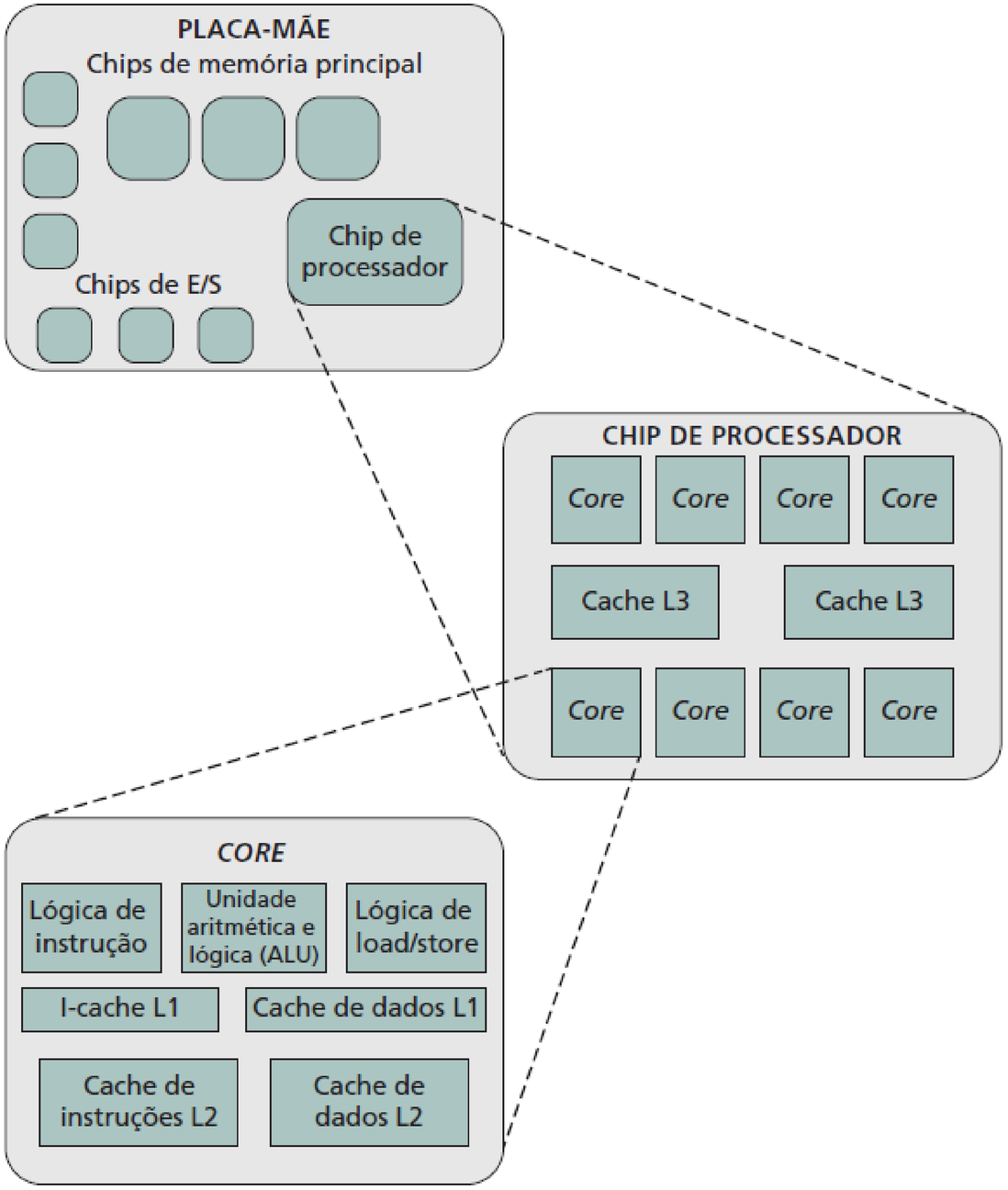
Áreas funcionais



Visão geral

Múltiplos núcleos (*cores*)

- A placa-mãe contém um *slot* ou um soquete para o chip do processador, o que geralmente contém múltiplos núcleos (cores) individuais, que é conhecido como processador *multicore*.
- Cada núcleo contém ULA, Lógica de instrução, cache L1
- Há também *slots* para os chips da memória, chips de controlador E/S e outros componentes-chave do computador.



Estrutura interna

- Os componentes são interligados por um **barramento interno do processador**
- Registradores visíveis ao usuário: **Uso geral**; **Dados**; **Endereços**.
- Registradores de controle e de estado: PC, IR, MAR, MBR e de estado (PSW).
- O registrador PSW pode conter bits específicos para: sinal, zero, *carry*, ...

Estrutura interna

- Os componentes são interligados por um **barramento interno do processador**
- Registradores visíveis ao usuário: **Uso geral**; **Dados**; **Endereços**.
- Registradores de controle e de estado: PC, IR, MAR, MBR e de estado (PSW).
- O registrador PSW pode conter bits específicos para: sinal, zero, *carry*, ...

Set de instruções

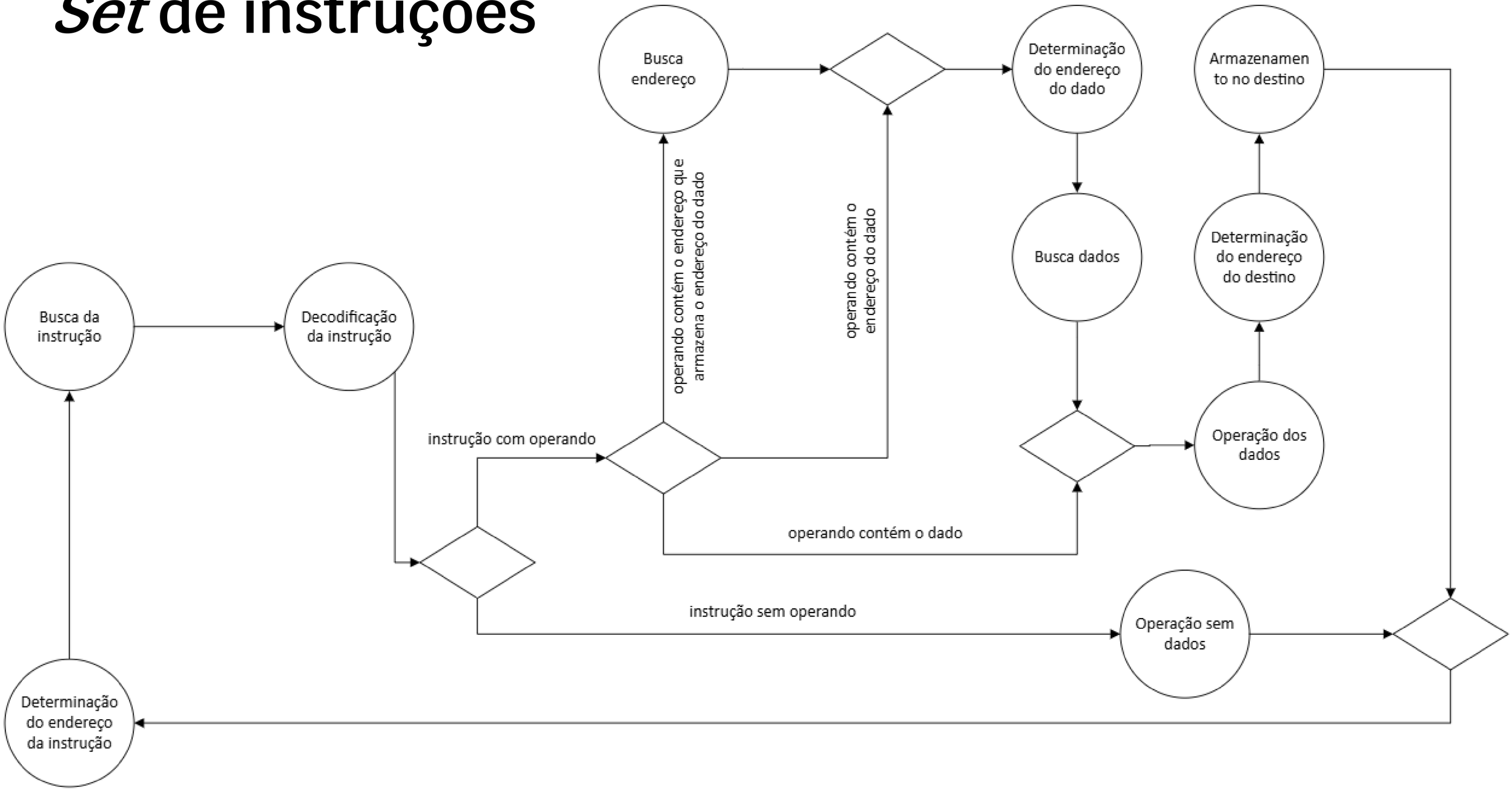
- A operação do processador é determinada pelas instruções que ele executa, conhecidas como **instruções de máquina** ou instruções de computador.
- A coleção de diferentes instruções que o processador pode executar é conhecida como *set* (conjunto) de instruções.
- As instruções que podem ser **interpretadas** pelo processador, **decodificadas** pela UC.
- Cada instrução corresponde um comportamento específico
- Armazenadas em posições lineares de endereço
- Normalmente possuem **visibilidade sobre alguns registradores**

Set de instruções

Os elementos de uma instrução de máquina são:

- A instrução pode ser dividida em dois campos, ***opcode***, ou código da instrução; e ***operando***.
- A referência aos dados usados pela operação.
- A referência ao resultado, ou seja, o destino da operação.
- A referência à próxima instrução.

Set de instruções



Set de instruções

Alternativas de projeto

RISC (*Reduced Instruction Set Computing*)

x

CISC (*Complex Instruction Set Computing*)

Aspecto	RISC (Reduced Instruction Set Computing)	CISC (Complex Instruction Set Computing)
Complexidade das Instruções	Conjunto reduzido de instruções simples e de execução rápida.	Conjunto maior e mais complexo de instruções, permite operações complexas em uma instrução.
Ciclo de Instrução	Instruções executadas em um único ciclo de clock, aumentando a previsibilidade e eficiência no pipeline.	Instruções podem levar múltiplos ciclos, dificultando o pipeline, mas simplificando operações complexas.
Uso de Registradores	Maior uso de registradores, com mais operações ocorrendo diretamente neles, reduzindo acessos à memória.	Menor quantidade de registradores, com acesso direto à memória para muitas instruções.
Eficiência no Pipeline	Simplicidade das instruções e ciclo fixo favorecem a eficiência do pipeline, promovendo paralelismo.	Complexidade das instruções pode dificultar o pipeline, mas menos instruções são necessárias para operações complexas.
Tamanho do Código	Requer mais instruções para tarefas complexas, resultando em um código geralmente maior.	Menor quantidade de instruções necessárias para operações complexas, gerando código mais compacto.
Aplicações e Desempenho	Ideal para dispositivos móveis e sistemas embarcados devido à eficiência energética e velocidade.	Preferido em desktops e servidores, onde a compatibilidade e execução de tarefas complexas são essenciais.
Exemplos de Processadores	ARM, MIPS, SPARC	x86 (Intel e AMD)

Pipeline de instruções

Estágios do ciclo de instruções

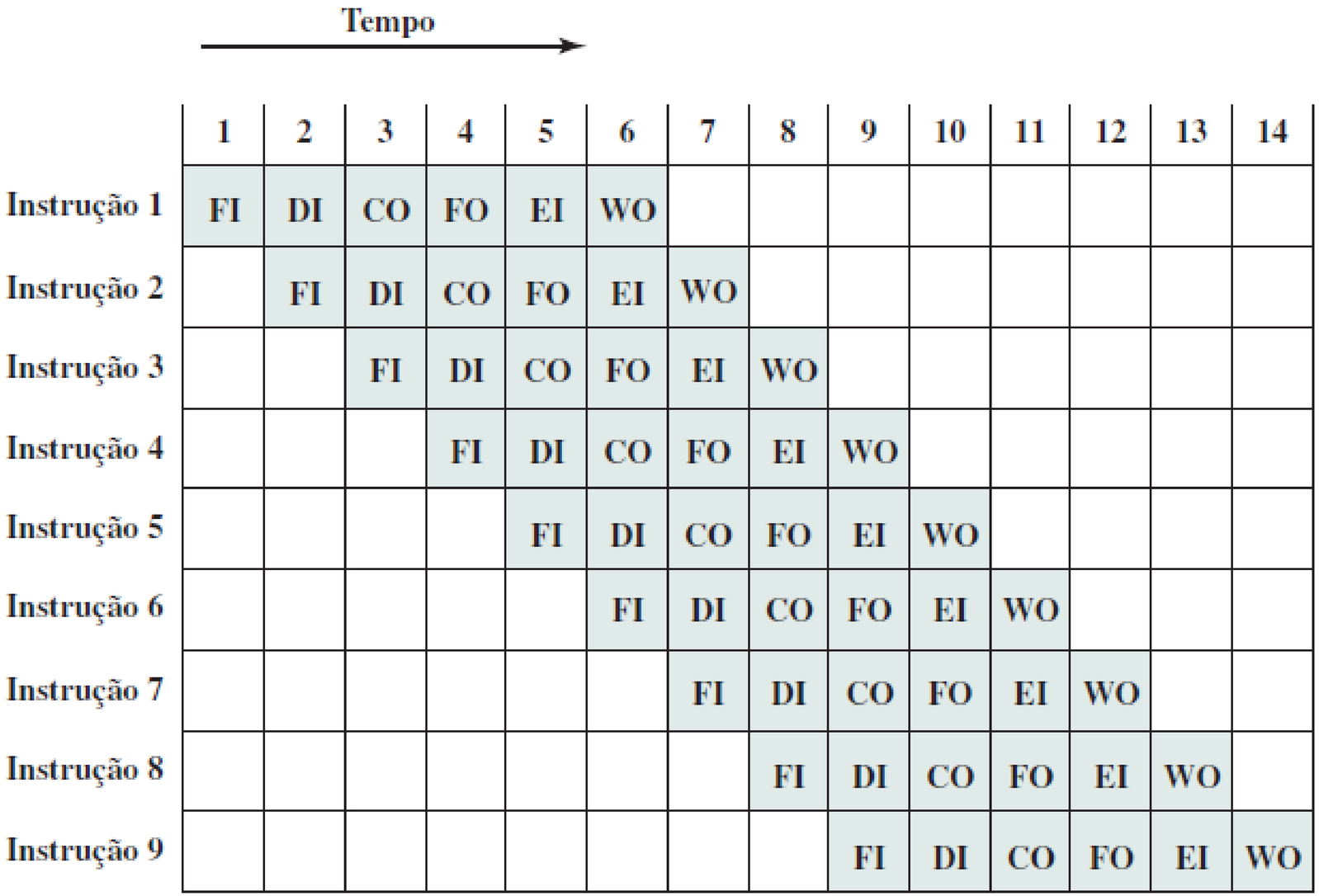
- **Buscar a instrução (FI — do inglês, *Fetch Instruction*):** ler a próxima instrução esperada em um *buffer*.
- **Decodificar a instrução (DI):** decodificar o opcode e o operando.
- **Calcular endereço do operando (CO):** calcular o endereço efetivo de cada operando origem.
- **Buscar os operandos (FO — do inglês, *Fetch Operands*):** ler cada operando da memória/registradores.
- **Executar a instrução (EI):** efetuar a operação indicada e armazenar o resultado, se houver.
- **Escrever o operando (WO — do inglês, *Write Operands*):** armazenar o resultado na memória/registradores.

Pipeline de instruções

Estágios do ciclo de instruções

- Os estágios possuem duração aproximadamente igual
- Os estágios são independentes
- A execução da tarefa em cada estágio é simultânea, ou seja, paralela
- Não há conflito quando estágios diferentes precisam acessar memória
- Deve-se prever uma saída para instruções de desvio

Pipeline de instruções



Pipeline de instruções

Ganho de desempenho

- Pode-se comparar o ganho de desempenho entre uma arquitetura sem e uma com pipeline pela relação

$$G = \frac{T_{s/pipe}}{T_{c/pipe}}$$

- O tempo *s/pipe* é calculado admitindo *n* instruções e um tempo *t* em cada estágio, logo para 6 estágios: $T_{s/pipe} = 6.n.t$
- O tempo c/pipe será $T_{c/pipe} = 6.t + (n-1)t = (5+n)t$
- Portanto o ganho será

$$G = \frac{6n}{(5+n)}$$

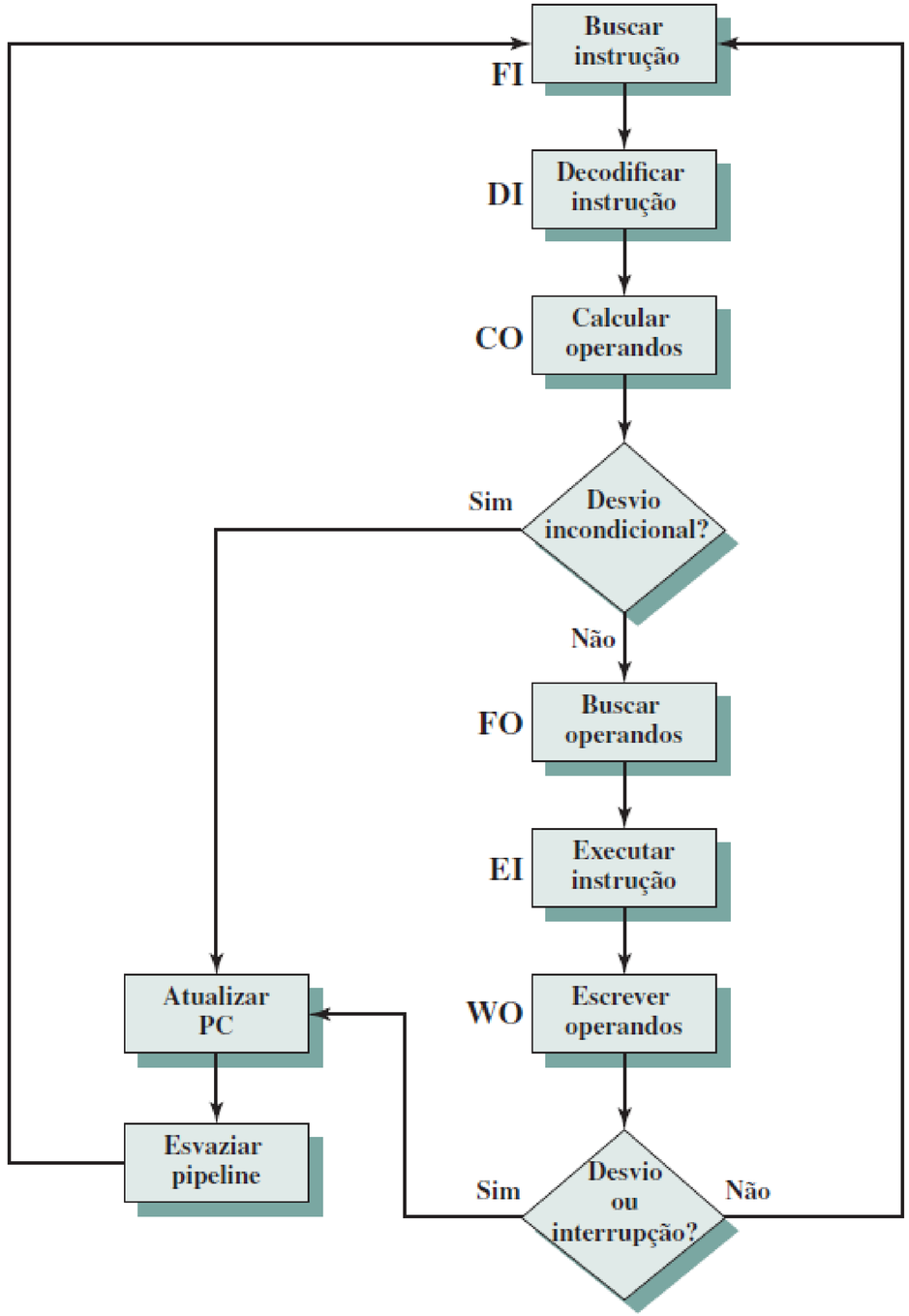
Pipeline de instruções

com desvio incondicional

Tempo

	FI	DI	CO	FO	EI	WO
1	I1					
2	I2	I1				
3	I3	I2	I1			
4	I4	I3	I2	I1		
5	I5	I4	I3	I2	I1	
6	I6	I5	I4	I3	I2	I1
7	I7	I6	I5	I4	I3	I2
8	I8	I7	I6	I5	I4	I3
9	I9	I8	I7	I6	I5	I4
10		I9	I8	I7	I6	I5
11			I9	I8	I7	I6
12				I9	I8	I7
13					I9	I8
14						I9

	FI	DI	CO	FO	EI	WO
1	I1					
2	I2	I1				
3	I3	I2	I1			
4	I4	I3	I2	I1		
5	I5	I4	I3	I2	I1	
6	I6	I5	I4	I3	I2	I1
7	I7	I6	I5	I4	I3	I2
8	I15					I3
9	I16	I15				
10		I16	I15			
11			I16	I15		
12				I16	I15	
13					I16	I15
14						I16



Classificação das instruções

Modos de endereçamento

como a instrução **indica a localização do dado** na arquitetura

- imediato
- direto
- indireto
- relativo ou indexado
- ...

Tipos de ação

- Aritméticas e lógicas
- Desvio condicional e incondicional
- Transferência de dados
- Controle

Classificação das instruções

Número de campos do operando

- Sem operando (zero-op): Não possuem operandos explícitos, geralmente operando com acumuladores ou registradores padrão.
- Instruções monádicas (um operando): Contêm um único operando explícito.
- Instruções diádicas (dois operandos): Possuem dois operandos explícitos. Comum em operações aritméticas ou lógicas.
- Instruções triádicas (três operandos): Possuem três operandos explícitos. Por exemplo, instruções que especificam dois operandos de entrada e um de saída.

Endereçamento imediato, desvio, monádica

- operando contém o próprio dado
- dado é buscado juntamente na busca da instrução
- a execução da instrução tipicamente mais rápida

Exemplo

Considerando o *set* de instruções do Atmeg2560,

JMP k

operação: PC ← k, carrega no Contador de Programa o valor do operando

Endereçamento direto, transferência de dados, diádica

- operando contém o endereço do dado em memória
- execução da instrução requer, em consequência, uma busca da instrução, seguida da busca ao dado armazenado no endereço

Exemplo

Considerando o *set* de instruções do Atmeg2560,

LDS Rd, k

operação: Rd $\leftarrow$ [k], carrega no registrador de uso geral Rd o valor armazenado no endereço k da memória

Endereçamento indireto, transferência de dados, diádica

- operando carrega um dado ou lê um dado em memória
- endereço que aponta ao dado está contido em um registrador.
- endereço nessa forma é chamado de **ponteiro**.

Exemplo

Considerando o *set* de instruções do Atmeg2560,

LD Rd, X

operação: Rd ← [X], carrega no registrador de uso geral Rd o valor armazenado no endereço da memória cujo valor está armazenado no registrador de endereço X – r27,r26

Endereçamento relativo, desvio, monádica

- operando indica o desvio relativo da execução
- altera o PC em um valor k

Exemplo

Considerando o *set* de instruções do Atmeg2560,

RJMP k

operação: PC ← PC + 1 + k, desvia a próxima instrução para o endereço 2k+1

IBMEC.BR

/IBMEC

IBMEC

@IBMEC_OFICIAL

@IBMEC

Ibmec