

Arquitetura de Computadores

Aula 11 - Multiprocessamento

Prof. Dr. Eng. Fred Sauer
<http://www.fredsauer.com.br>
fsauer@gmail.com

PROCESSAMENTO PARALELO

OBJETIVO: aumentar a capacidade de processamento.

PROCESSADORES

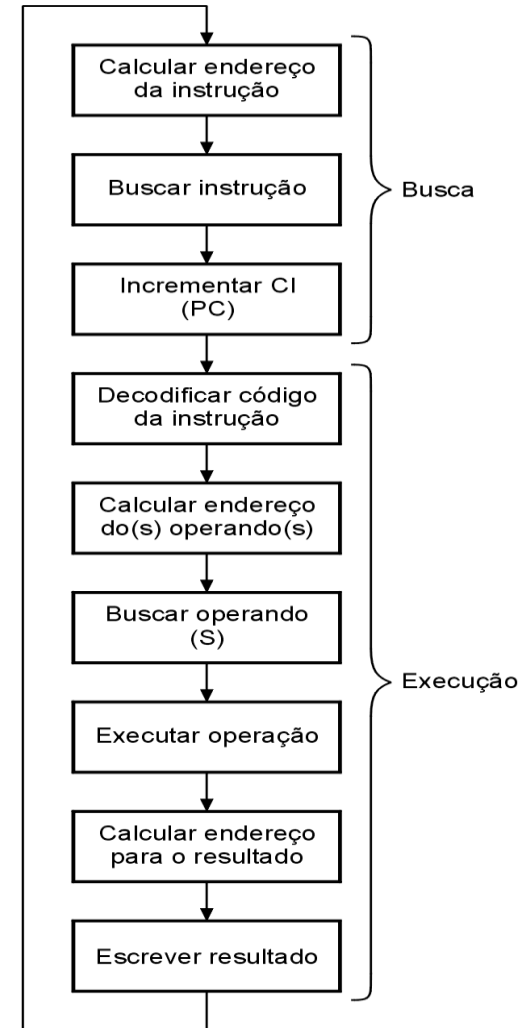
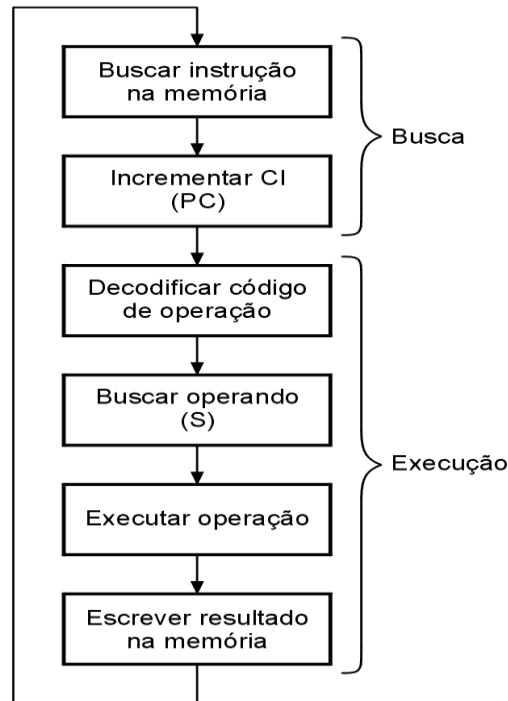
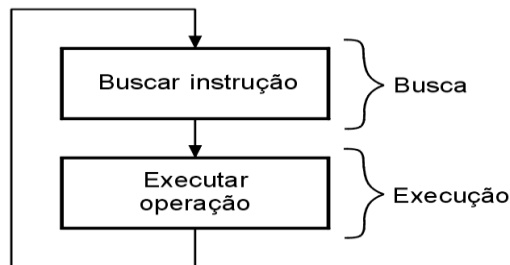
Há diversos métodos de um processador realizar seu trabalho – executar as etapas de um ciclo de instrução.

Processamento simples, sequencial – SISD (uma instrução manipulando um dado)

Processamento paralelo:

- por instrução – SIMD – uma instrução manipulando múltiplos dados
- controle por pipeline – diversas instruções em etapas diferentes de execução “simultânea”
- processamento superescalar – diversas execuções
- multiprocessamento

PROCESSADORES



(a) O ciclo de instrução de forma básica, com duas grandes etapas.

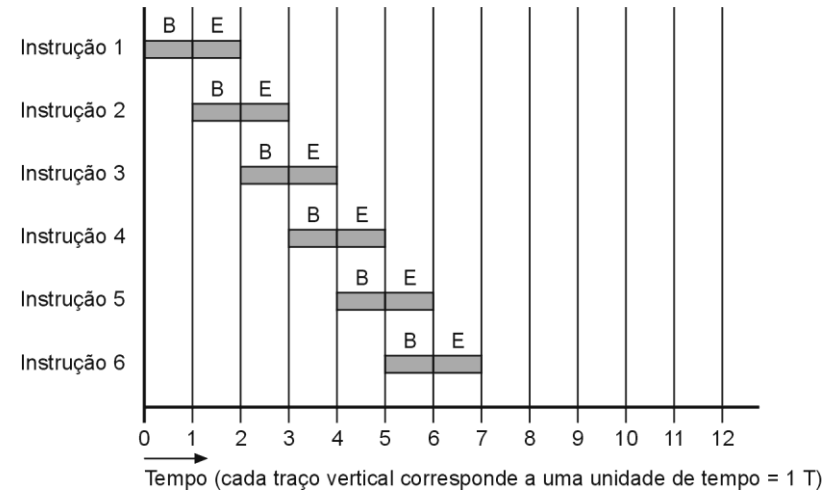
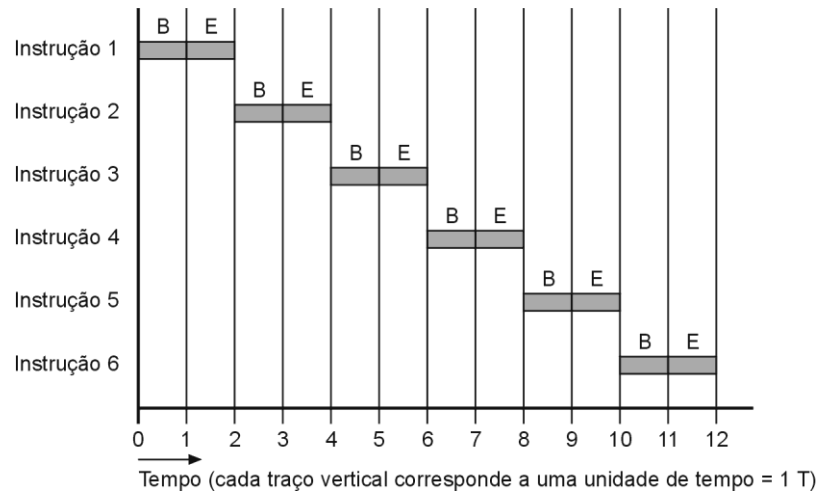
(b) O ciclo de instrução decomposto de forma mais detalhada.

(c) O ciclo de instrução ainda mais detalhado.

PROCESSADORES

PROCESSAMENTO PIPELINE

Diagrama de tempo de um ciclo de instrução com 2 estágios

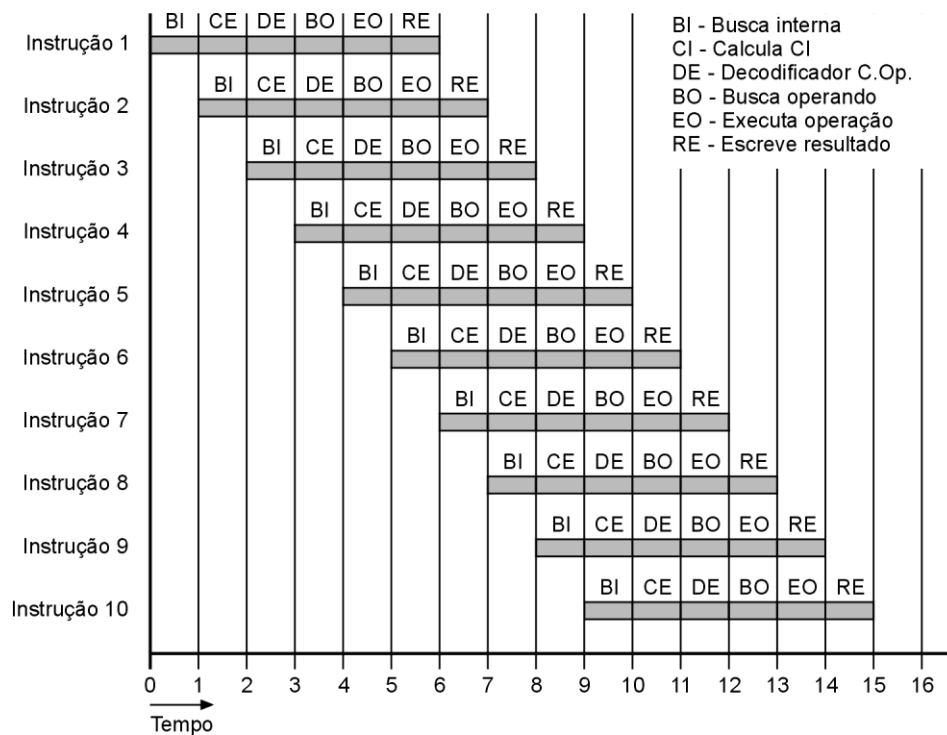


B – estágio de busca
E – estágio de execução

PROCESSADORES

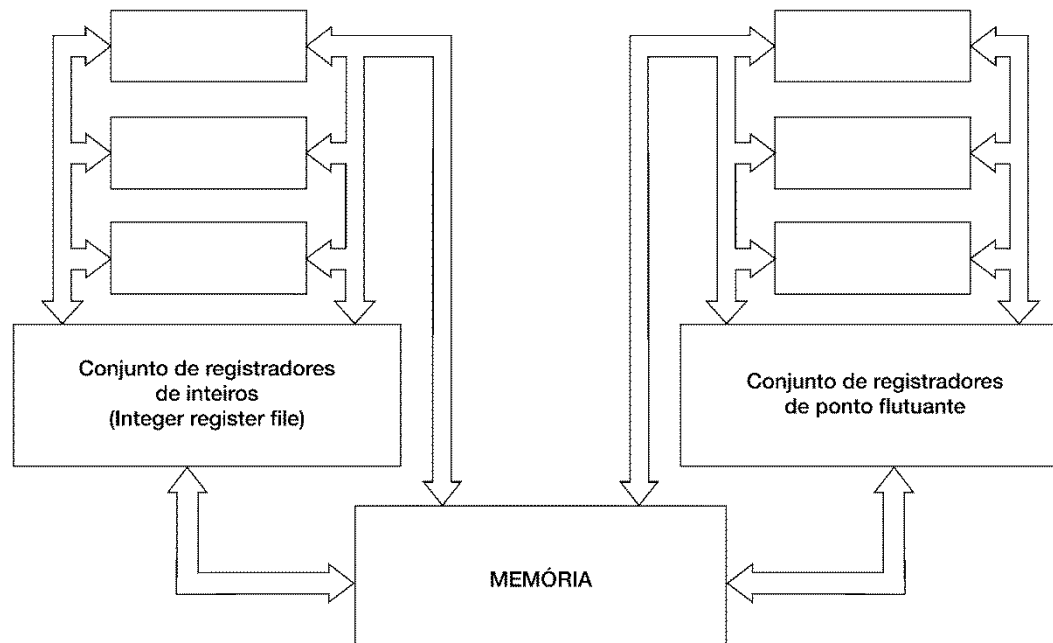
PROCESSAMENTO PIPELINE

Diagrama de tempo de um ciclo de instrução com 6 estágios

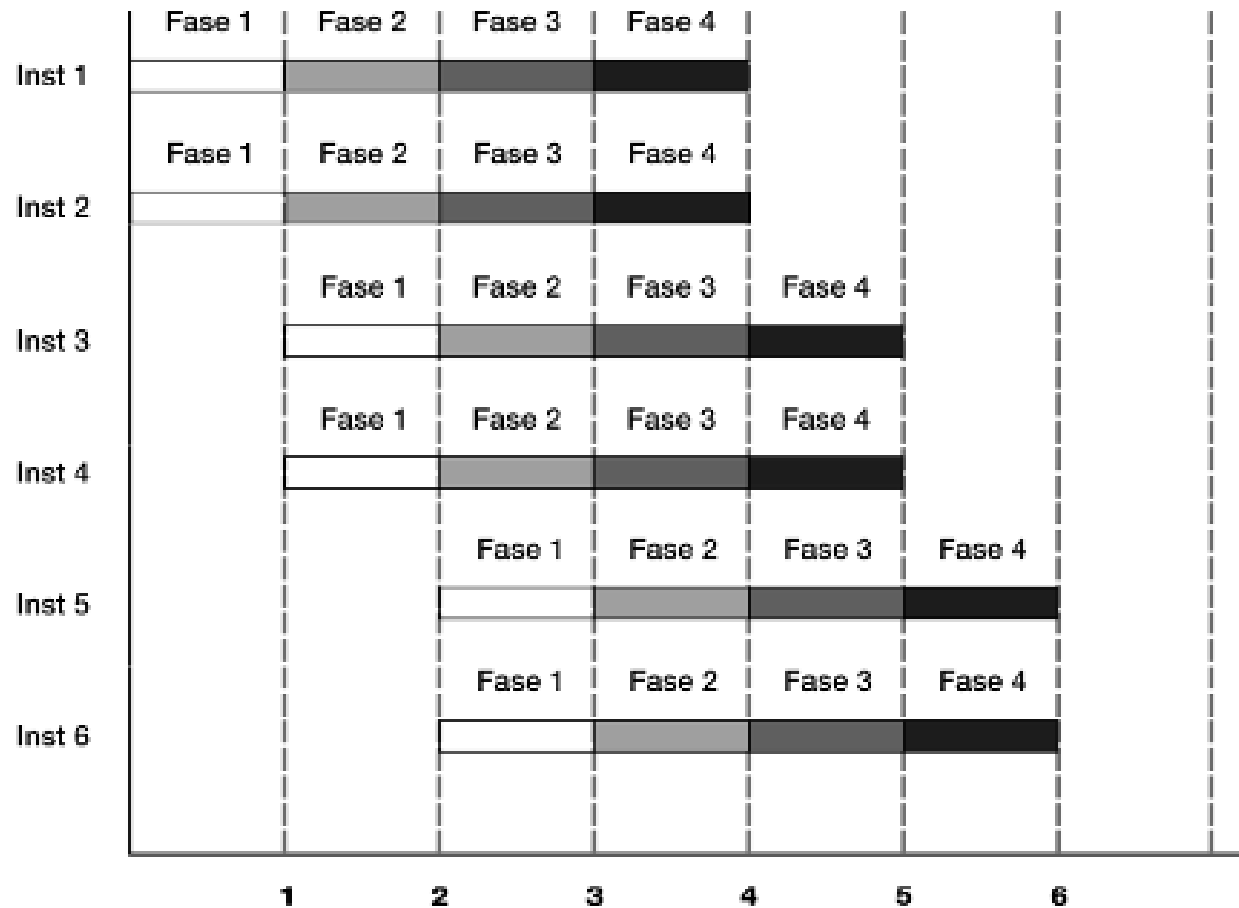


PROCESSADORES

EXEMPLO DE ARQUITETURA SUPERESCALAR

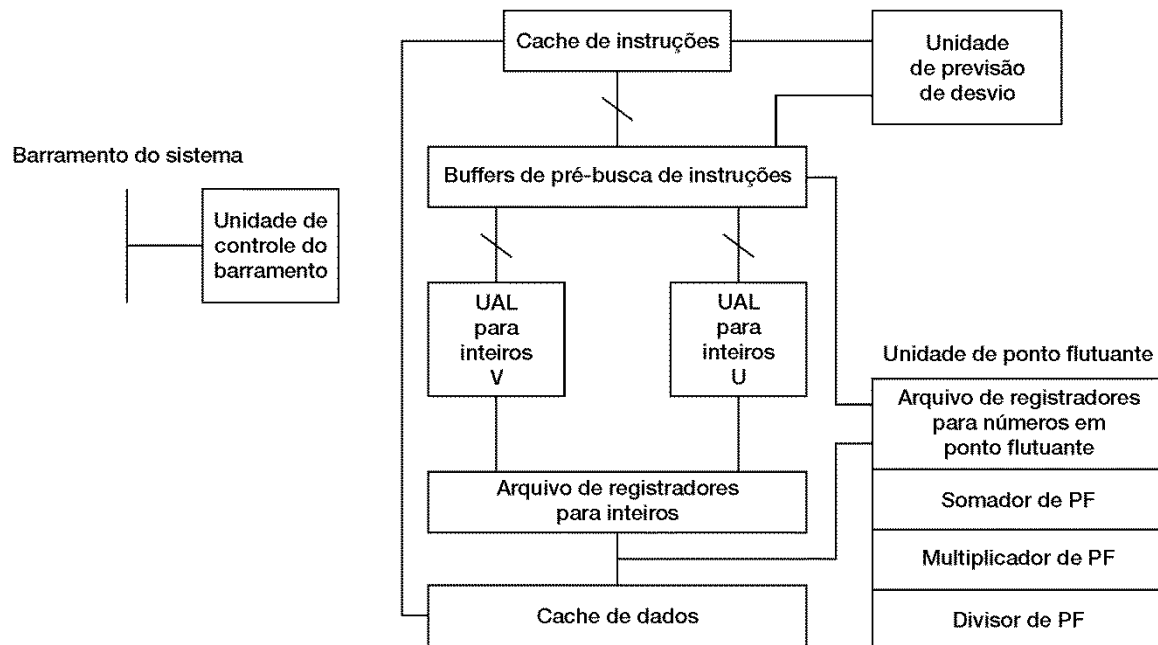


Paralismo Explorando o Pipeline



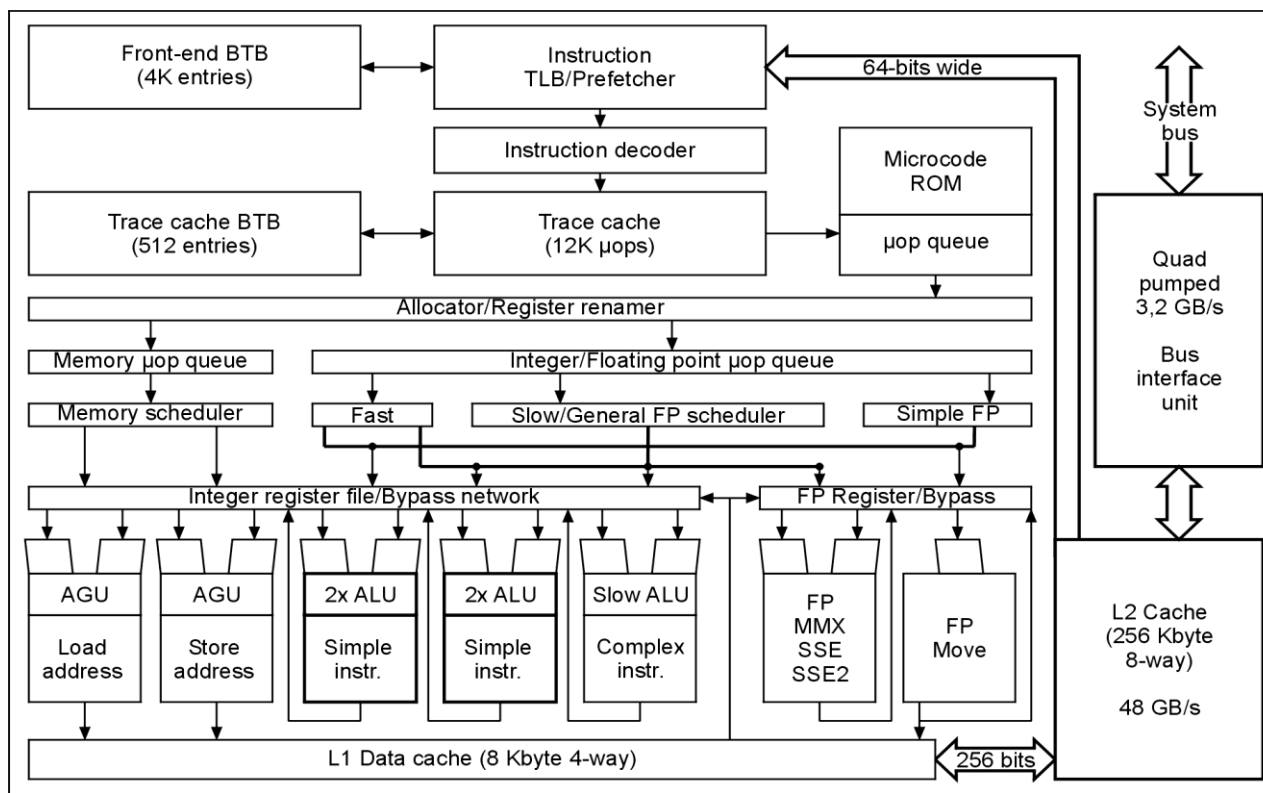
PROCESSADORES

DIAGRAMA EM BLOCOS DE UM PROCESSADOR PENTIUM



PROCESSADORES

PROCESSADOR PENTIUM 4 (P4)



EXECUÇÃO DE PROGRAMAS

EXECUÇÃO DE PROGRAMAS

- Um programa executável (.EXE) é constituído de um conjunto de instruções de máquina (código binário).
- O processador executa instrução por instrução (ciclo de instrução) e NÃO o programa inteiro. Isto é, ele não entende o programa, mas sim a instrução que vai ser executada no momento.

EM RESUMO

O processador busca uma instrução da memória, interpreta o que ela faz (qual a operação) e executa; em seguida, repete as mesmas atividades para instrução seguinte... e assim por diante.

EXECUÇÃO DE PROGRAMAS

**O DESEMPENHO DE UM PROCESSADOR É
DEPENDENTE DE DIVERSOS FATORES.**

**Um deles refere-se ao tempo de execução de
cada instrução – quantidade de ciclos de
relógio (“clock”)**

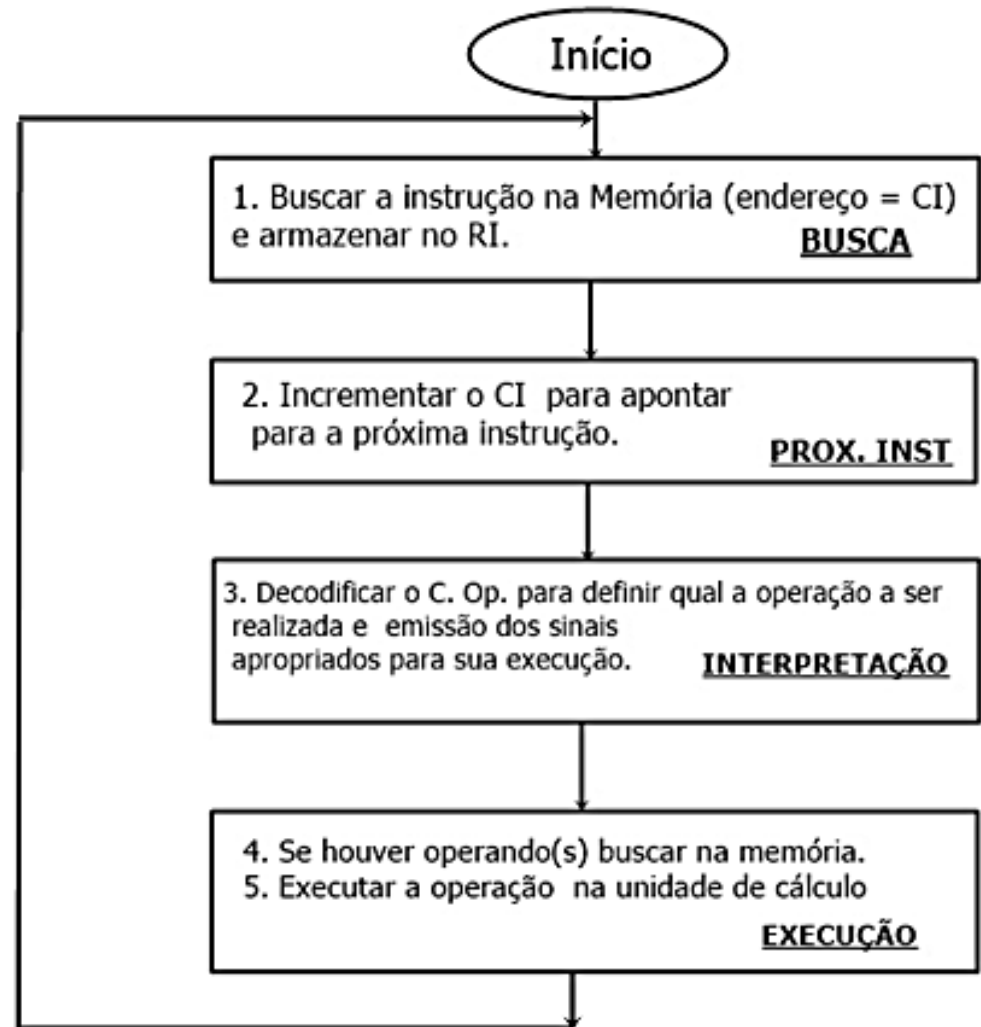
EXECUÇÃO DE PROGRAMAS

**O PONTO CRÍTICO DO PROJETO DE UM PROCESSADOR É,
ENTÃO :**

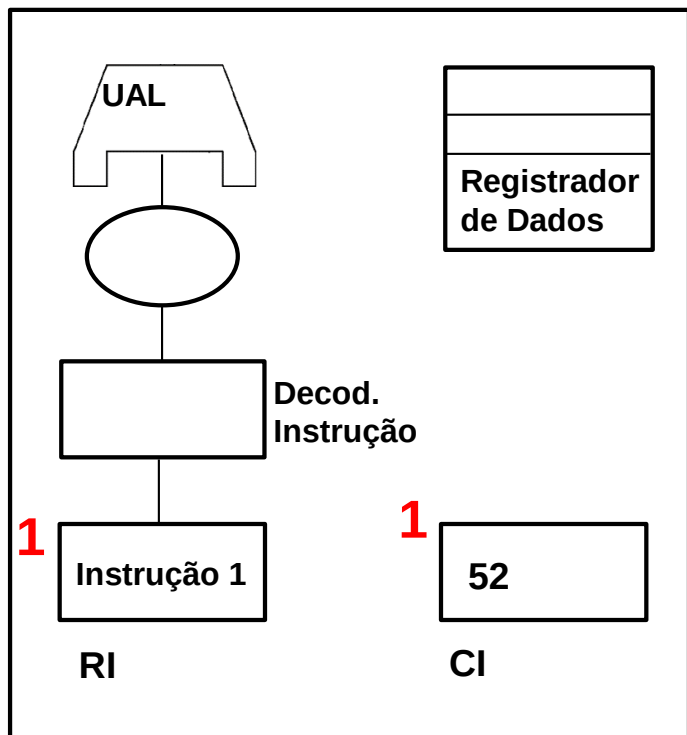
O CICLO DE INSTRUÇÃO

PROCESSADORES

CICLO DE INSTRUÇÃO



Processador



Mem Principal

52	Instrução 1
53	Instrução 2
54	Instrução 3

Ciclo de Instrução

1. Buscar Instrução
 $RI \leftarrow (CI)$

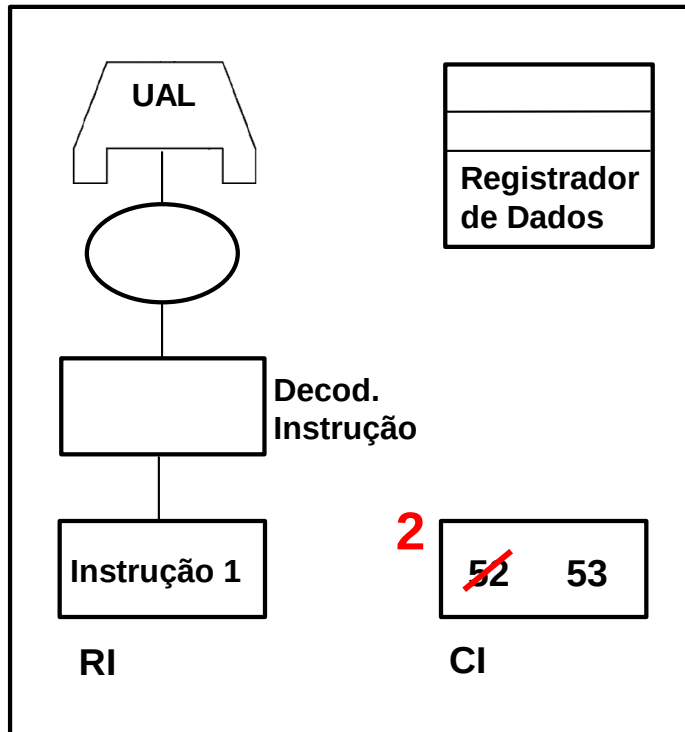
2. Incrementa o CI para o end. da próx. instrução
 $CI \leftarrow CI + 1$

Decodificar COD. Operação

4. Executar a Operação

5. Voltar para etapa 1

Processador



Mem Principal

52	Instrução 1
53	Instrução 2
54	Instrução 3

Ciclo de Instrução

1. Buscar Instrução
 $RI \leftarrow (CI)$

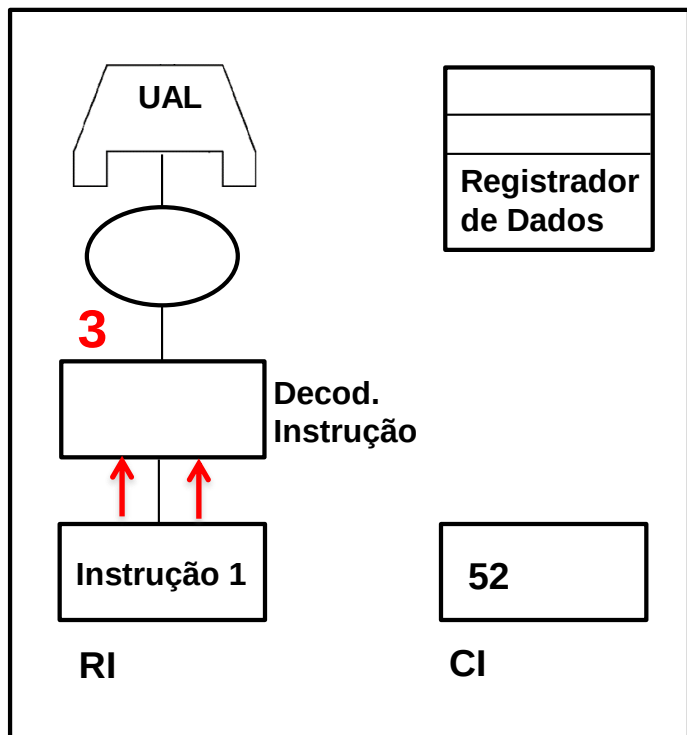
2. Incrementa o CI para o
end. da próx. instrução
 $CI \leftarrow CI + 1$

Decodificar COD.
Operação

4. Executar a Operação

5. Voltar para etapa 1

Processador



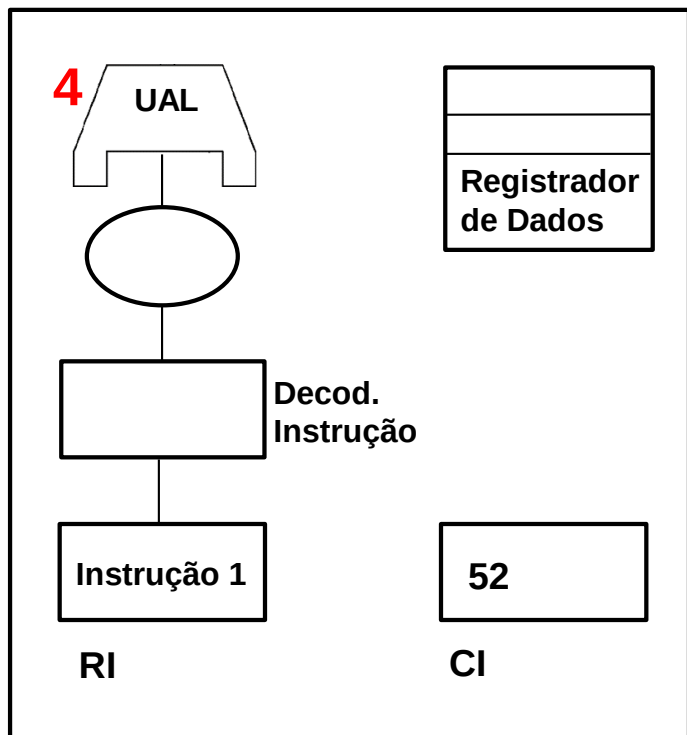
Mem Principal

52	Instrução 1
53	Instrução 2
54	Instrução 3

Ciclo de Instrução

1. Buscar Instrução
 $RI \leftarrow (CI)$
2. Incrementa o CI para o end. da próx. instrução
 $CI \leftarrow CI + 1$
3. **Decodificar COD. Operação**
4. Executar a Operação
5. Voltar para etapa 1

Processador



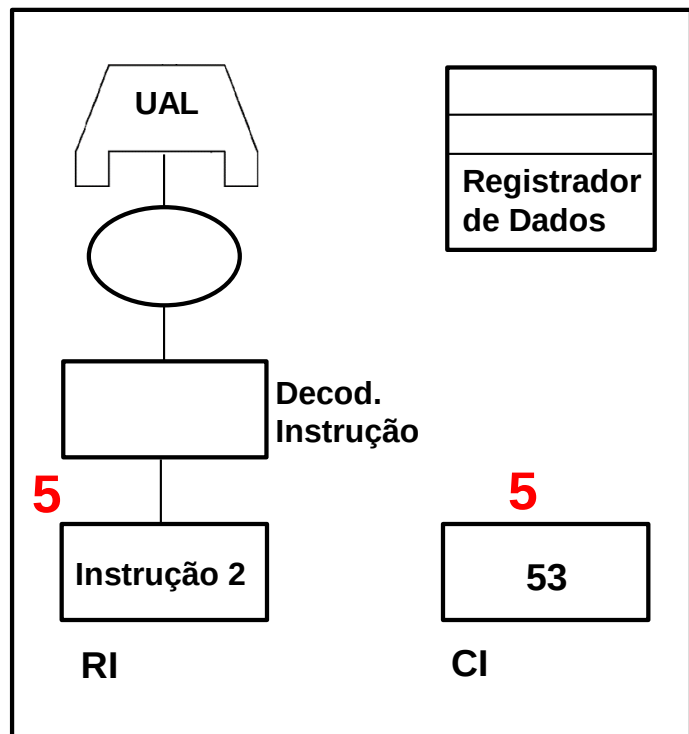
Mem Principal

52	Instrução 1
53	Instrução 2
54	Instrução 3

Ciclo de Instrução

1. Buscar Instrução
 $RI \leftarrow (CI)$
2. Incrementa o CI para o end. da próx. instrução
 $CI \leftarrow CI + 1$
3. Decodificar COD. Operação
- 4. Executar a Operação**
5. Voltar para etapa 1

Processador



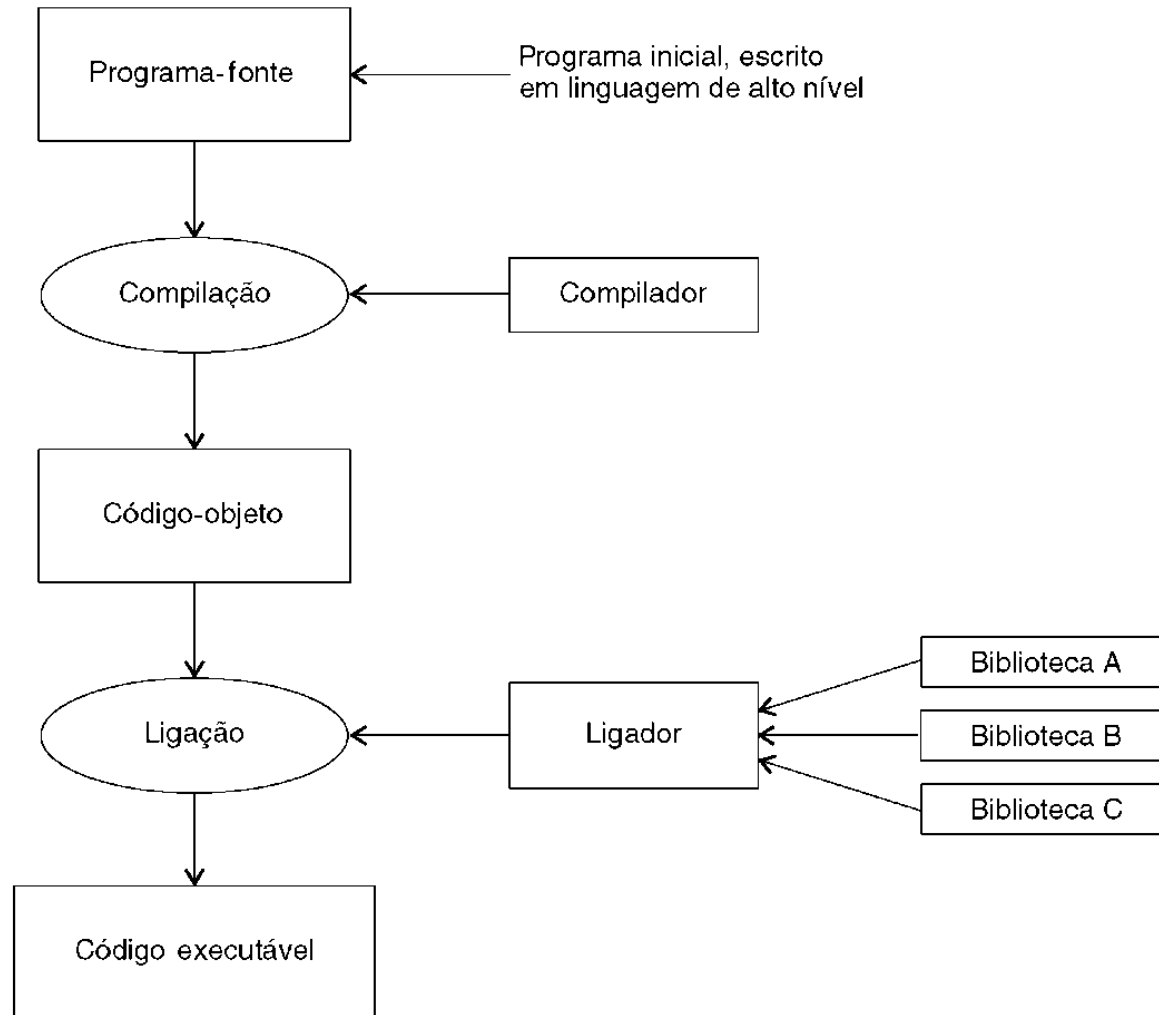
Mem Principal

52	Instrução 1
53	Instrução 2
54	Instrução 3

Ciclo de Instrução

1. Buscar Instrução
 $RI \leftarrow (CI)$
2. Incrementa o CI para o end. da próx. instrução
 $CI \leftarrow CI + 1$
3. Decodificar COD. Operação
4. Executar a Operação
5. Voltar para etapa 1

ETAPAS DE EXECUÇÃO DE UM PROGRAMA



ARQUITETURAS CISC X RISC

CISC – Complex Instruction Set Computer (computador com conjunto complexo de instruções)

RISC – Reduced Instruction Set Computer (computador com conjunto reduzido de instruções)

CARACTERÍSTICAS DE ARQUITETURAS CISC

- Poucos Registradores de Dados
- Conjunto de Instruções de Máquina
 - * O controle é implementado por microprogramação.
 - * Muitas Instruções (alguns processadores chegaram a ter mais de 300 instruções)
 - * Maior tempo de decodificação
 - * Muitos modos de endereçamento, a maioria não sendo usados pelos compiladores.
 - * Diversas instruções com acesso à memória (lento)

CARACTERÍSTICAS DE ARQUITETURAS CISC

POR QUE MUITAS INSTRUÇÕES?

Para facilitar a tarefa dos compiladores (bem como seu projeto e implementação)

Reduz o GAP na tradução dos comando do código fonte nas instruções do código objeto.

CARACTERÍSTICAS DE ARQUITETURAS RISC

Característica	Considerações
Menor quantidade de instruções que as máquinas CISC	• Simplifica o processamento de cada instrução e torna este item mais eficaz. • Embora o processador RS/600 possua 184 instruções, ainda assim é bem menos que as 303 instruções dos sistemas VAX-11. Além disso, a maioria das instruções é realizada em 1 ciclo de relógio, o que é considerado o objetivo maior dessa arquitetura.
Execução otimizada de chamada de funções	• As máquinas RISC utilizam os registradores da UCP (em maior quantidade que os processadores CISC) para armazenar parâmetros e variáveis em chamadas de rotinas e funções. Os processadores CISC usam mais a memória para a tarefa.
Menor quantidade de modos de endereçamento	• As instruções de processadores RISC são basicamente do tipo Load/Store, desvio e de operações aritméticas e lógicas, reduzindo com isso seu tamanho. • A grande quantidade de modos de endereçamento das instruções de processadores CISC aumenta o tempo de execução das mesmas.
Utilização em larga escala de <i>pipelining</i>	• Um dos fatores principais que permite aos processadores RISC atingir seu objetivo de completar a execução de uma instrução pelo menos a cada ciclo de relógio é o emprego de <i>pipelining</i> em larga escala.

CARACTERÍSTICAS DE ARQUITETURAS RISC

- * Devem completar cada instrução em 1 ciclo de relógio
- * Possuem grande quantidade de registradores, com o propósito de reduzir acesso à memória. Os dados permanecem nos registradores sendo reutilizados de lá e não da memória.
- * Acesso à memória apenas para buscar (inicialmente) o dado e retornar eventual resultado de operação)
Somente instruções LOAD e STORE com a RAM.

Todas as operações usam instruções que manipulam dados nos registradores (são menores em largura e mais rápidas).

EXEMPLOS DE PROCESSADORES

CISC – IBM/64; Intel Pentium 1, II, III e alguns modelos P4, bem como os correspondentes AMD (mesma arquitetura X86)

**RISC - Berkeley RISC 1 – SPARC - MIPS – RS6000 – Power PC
Intel i3, i5, i7 - ARM (celulares, palms, embutidos, etc)**

A maioria dos processadores atuais são híbridos, com instruções RISC e CISC.

CARACTERÍSTICAS DA ARQUITETURA ARM

(Advanced Risc Machine)

- **Desenvolvida (1983) pela empresa Acorn (atualmente licencia)**
- **Baixo custo – baixo consumo de energia (poucos transistores)**
- **Exemplos: celulares – tablets – console jogos**

- **Características, típicas de arquitetura RISC:**
 - **Palavra de 32 bits**
 - **Muitos registradores (32 de emprego geral)**
 - **Apenas instruções LOAD/STORE para acesso memória**
 - **Instruções de tamanho fixo**
 - **Instruções aritméticas com 3 operandos**
 - **Coprocessador (extende uso do processador)**