



Departamento de Ciência da Computação
Arquitetura de Processadores Digitais

Assembly

Tipos de Arquiteturas (RISC e CISC)

Tipos de computadores

Assembly

- Difícil de aprender?
- Difícil de debugar ou manter?
- Consome mais tempo?
- Os compiladores eliminam a necessidade de programar em Assembly?
- Os computadores já são rápidos?
- Melhor usar algoritmos mais rápidos?
- Não é portátil!

Assembly

- Vantagens

- Rapidez
- Espaço
- Capacidade
- Conhecimento

Programa em
linguagem de
alto nível
(em C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

Compilador

Programa em
assembly
(para MIPS)

```
swap:
    muli $2, $5, 4
    add  $2, $4, $2
    lw   $15, 0($2)
    lw   $16, 4($2)
    sw   $16, 0($2)
    sw   $15, 4($2)
    jr   $31
```

Montador

Programa
binário em
linguagem de
máquina
(para MIPS)

```
000000001010000100000000000011000
000000000000110000001100000100001
100011000110001000000000000000000
100011001111001000000000000000100
101011001111001000000000000000000
101011000110001000000000000000100
00000011111000000000000000001000
```

Arquitetura

- Hardware tem evoluído rápido
 - Velocidade dos processadores
 - Tecnologia de processamento (pipelining, superescalar, processamento paralelo)
 - Aumento da largura da palavra (de 32 para 64 bits)
 - Capacidade de armazenamento (caches e registradores)

Arquitetura

- Novas e poderosas linguagens de alto nível surgiram
- Conceitos de programação mais sofisticados:
 - Buscam atender o “raciocínio” humano
 - Mas, se afastam cada vez mais da simplicidade e do primitivismo do hardware, ou seja, das instruções que a máquina entende
- Consequência: Separação acentuada entre as linguagens de alto nível e as linguagens de máquina

Arquitetura

- Um único comando de uma linguagem de alto nível pode gerar muitas instruções de máquina
 - Os compiladores são complexos
 - *Semantic gap* (espaço semântico)
- Consequência: fabricantes (Intel, IBM, DEC, AMD) tomaram algumas medidas estratégicas:
 - Aumentar a quantidade de instruções dos seus processadores para se aproximar da linguagem de alto nível
 - Objetivo: Diminuir o *gap* semântico

CISC

- ***Complex Instruction Set Computers***

- Capazes de executar centenas de instruções complexas diferentes (família x86)
- Microprogramação
 - conjunto de códigos de instruções são gravados no processador
 - as instruções, já em baixo nível, são quebradas em diversas instruções mais próximas do hardware (as instruções contidas no microcódigo do processador)
- Característica: conjunto grande de instruções, a maioria delas em um elevado grau de complexidade.

Características

- Formato de 2 operandos é mais comum
- Usos de modo registrador para registrador, registrador para memória, memória para registrador
- Instruções de largura variável
- Instruções com múltiplos ciclos (de tamanho variável)
- Poucos registradores
- Registradores especializados

RISC

- ***Reduced Instruction Set Computer***
- Otimização do emprego das instruções que consomem mais tempo de execução
- Solução: utilizar arquiteturas simples
- Resultado da pesquisa de vários indivíduos
 - IBM 801, John Cocke
 - Stanford, Hennessy – MIPS
 - Berkley, Patterson – RISC-1 e RISC-2 -- SPARC

Características

- Conjunto pequeno de instruções → baixo nível de complexidade.
- Não há micro-programação, as instruções são executadas diretamente pelo hardware
- Conjunto simples de instruções que levam a mesma quantidade de tempo para serem executadas
- Uso intenso de pipelining
- A maioria dos microprocessadores modernos são RISCs
 - DEC Alpha, SPARC, MIPS, e PowerPC
 - O tipo de microprocessador mais largamente usado em desktops, o x86, é mais CISC do que RISC, embora chips mais novos traduzam instruções x86 baseadas em arquitetura CISC em formas baseadas em arquitetura RISC mais simples, utilizando prioridade de execução.

Consequências

- Menor quantidade de transistores no chip
- Redução da complexidade do decodificador de instruções, reduzindo o tempo de decodificação
- Menor quantidade de bits no campo de código de operação da instrução, reduzindo o tamanho dos programas
 - VAX: 300 instruções com larguras de 4 a 57 bytes
 - SPARC: 65 instruções com 32 bits

Consequências

- Como sobra espaço no chip → mais registradores
- Uso de registradores na passagem de rotinas e funções reduzindo o tempo gasto
 - Lembrando que estas são as funções que gastam mais tempo em alto nível;
 - Se usarmos acesso à memória, o programa fica mais lento.

Pipelining → Paralelismo

Em relação ao CISC ...

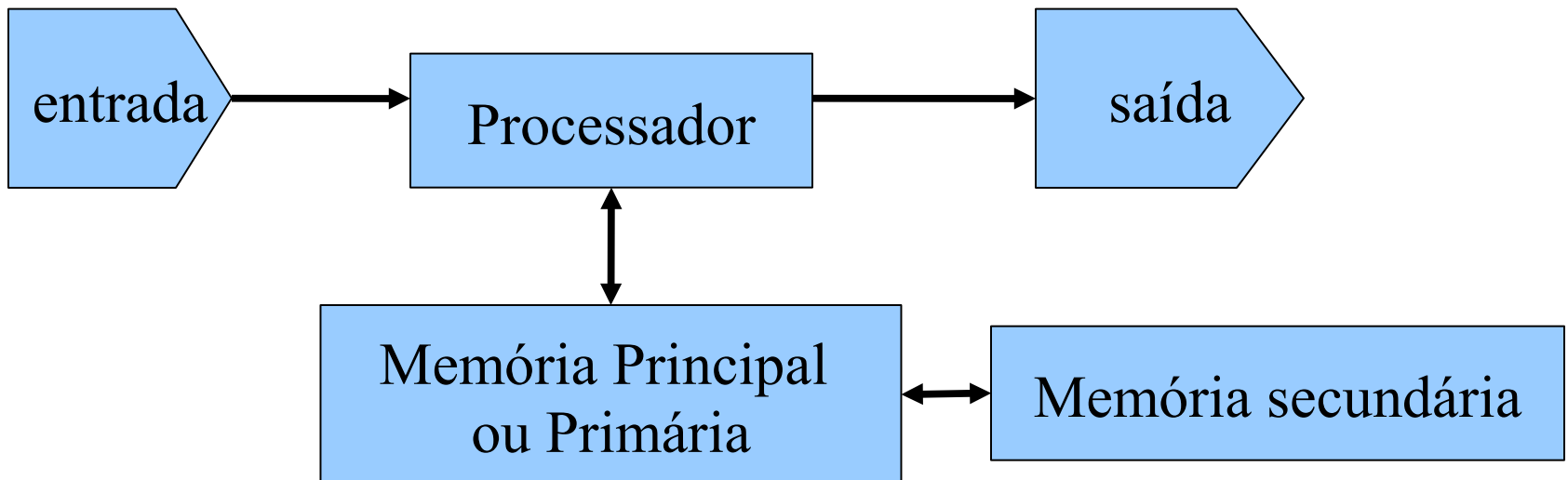
Característica	Considerações
Menor quantidade de instruções que as máquinas CISC	<ul style="list-style-type: none">• Simplifica o processamento de cada instrução e torna este item mais eficaz• Maioria das instruções realizada em um ciclo
Execução otimizada da chamada de funções	<ul style="list-style-type: none">• Registradores para armazenar parâmetros e variáveis em chamada de rotinas e funções• CISC usam memória
Menor quantidade de modos de endereçamento	<ul style="list-style-type: none">• Instruções load/store, desvio e operações aritméticas
Pipelining	<ul style="list-style-type: none">• Para que o objetivo de completar uma instrução a cada ciclo seja atingido, utiliza-se pipelining em larga escala

Processadores CISC e RISC

Características	RISC		CISC	
	MIPS R4000	RS/6000	VAX11/ 780	Intel 486
Quant. de instruções	94	183	303	235
Modos de endereçamento	1	4	22	11
Largura de cada instrução	4	4	2-57	1-12
Quantidade de registradores	32	32	16	8

O Computador

- O 1º computador tinha Processador, Entrada e Saída
- John Von Neumann adicionou a memória
 - Serve para guardar dados e instruções
 - Torna o processo mais rápido e eficiente.



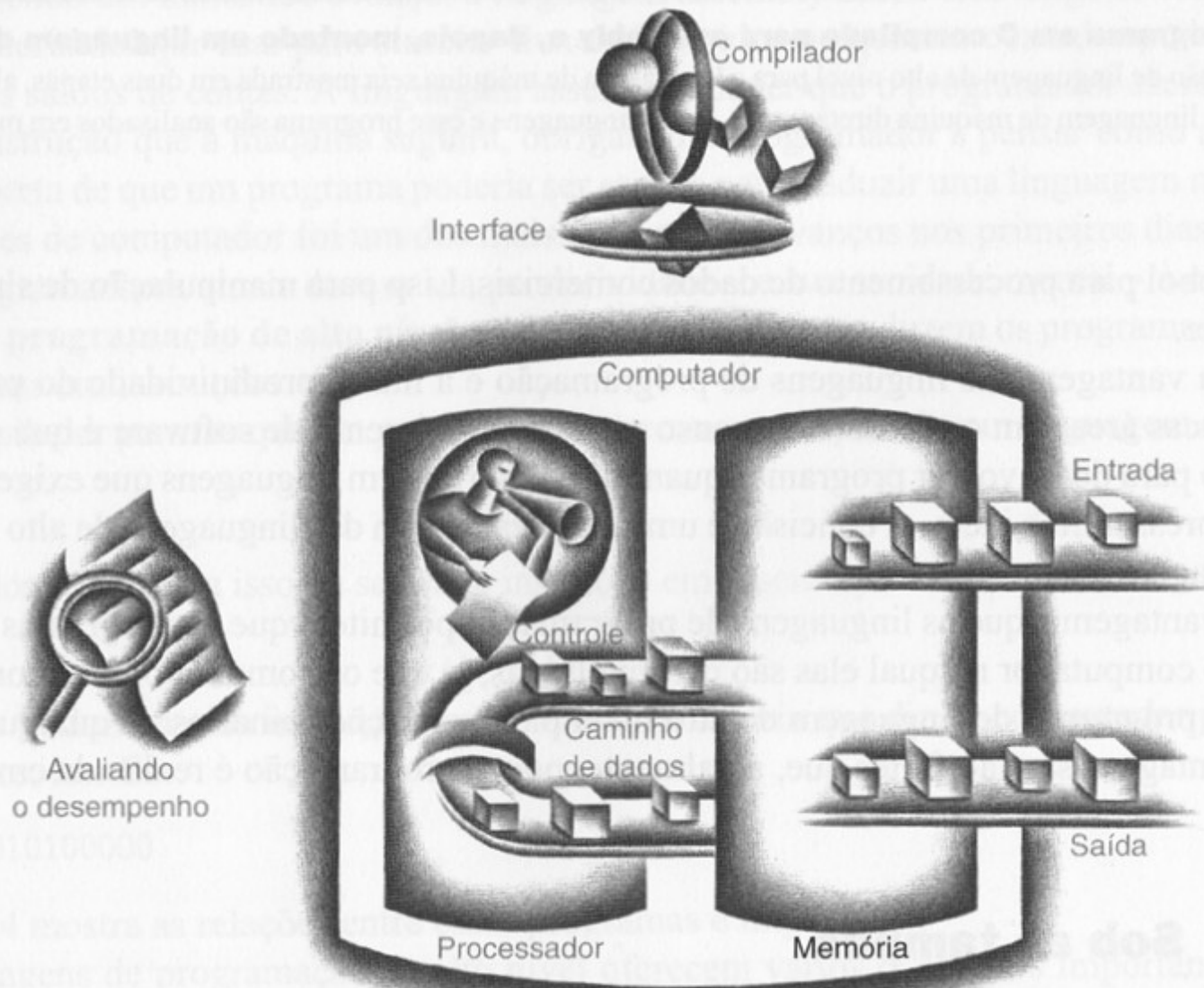
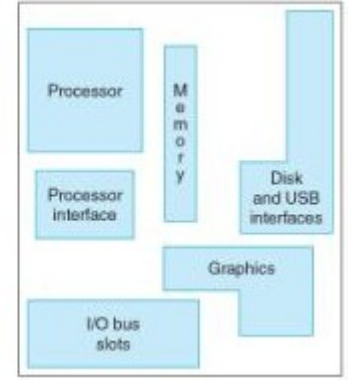
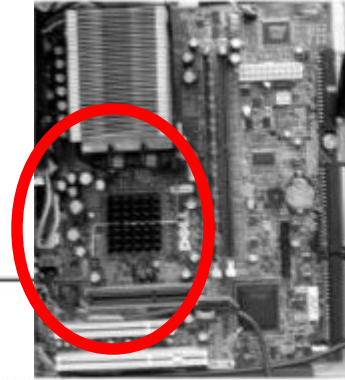
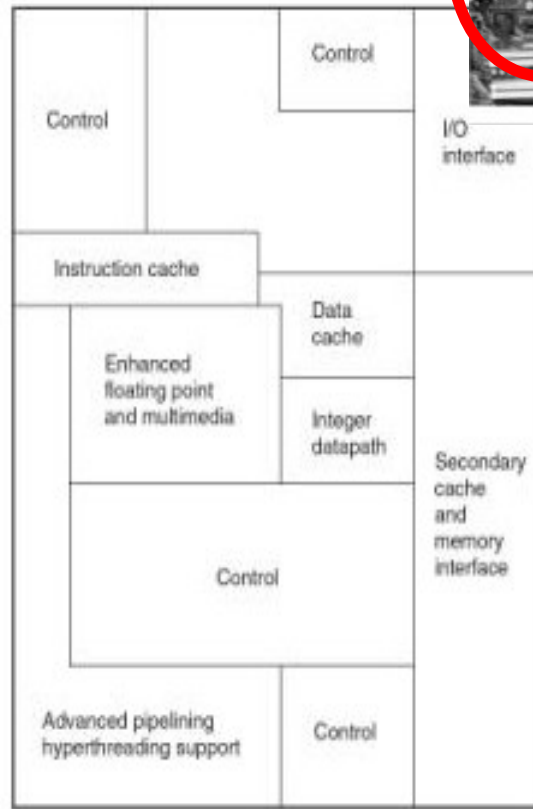
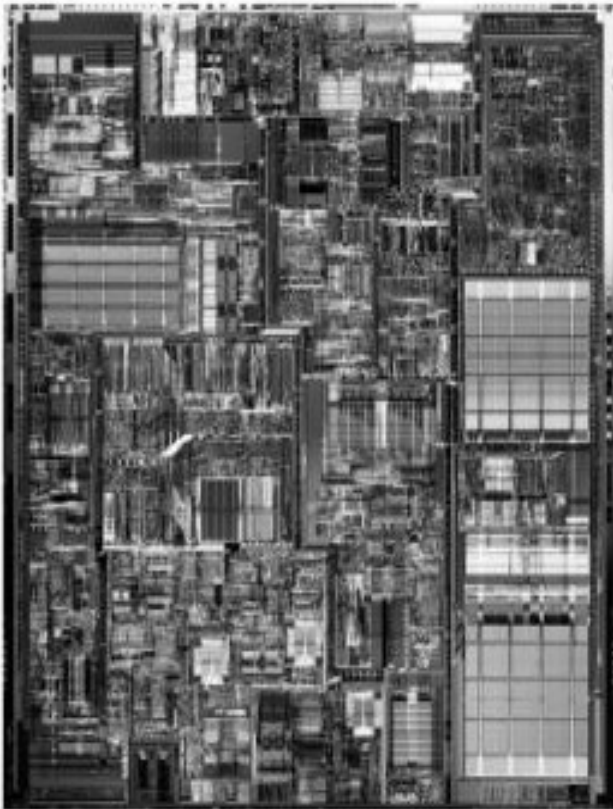


FIGURA 1.5 A organização de um computador, mostrando os cinco componentes clássicos. O processador obtém instruções e dados da memória. A entrada escreve dados na memória, e a saída lê os dados da memória. O controle envia os sinais que determinam as operações do caminho de dados, da memória, da entrada e da saída.

Dentro do PC



Processador



CPU ou UCP
Unidade Cental
de Processamento

Caminho de dados e Processamento

Memória

- DRAM (Dynamic Random Access Memory)
 - Guardam as instruções e dados do programa
 - RAM -> acesso aleatório e, portanto, mais rápido
- Memória Cache
 - Buffer para DRAM (dentro do processador)
 - SRAM (Static Random Access Memory)
 - Mais rápida, menos densa, mais cara
- Memória secundária (HD, CD, etc.)
- Memória ROM

Tipos de Computadores

- Computadores pessoais (desktop)
 - Forma mais conhecida e de baixo custo;
 - Bom desempenho para um único usuário;
 - 30 anos.
- Servidores
 - Antigos Mainframes;
 - Grandes cargas de trabalhos e Caros.

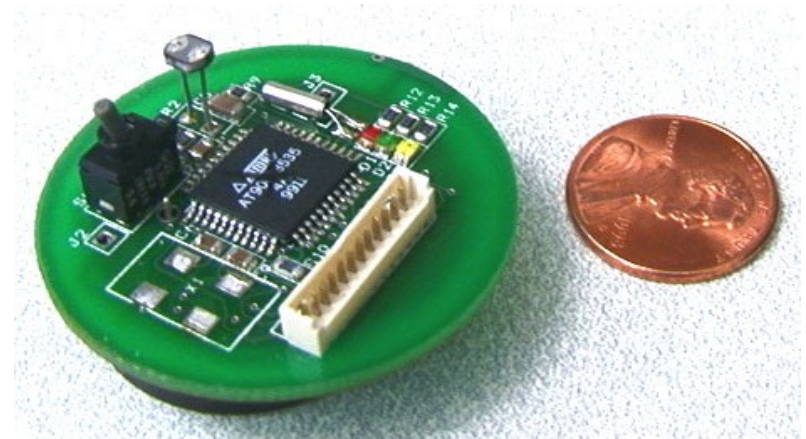
Tipos de Computadores

- Computadores Embutidos

- Computador dentro de outro dispositivo;
- Execução de uma aplicação determinada;
- Desempenho mínimo e limitações rígidas;
- Minimiza custo e potência;
- O crescimento do número de computadores embutidos é bem maior que os dos desktops.

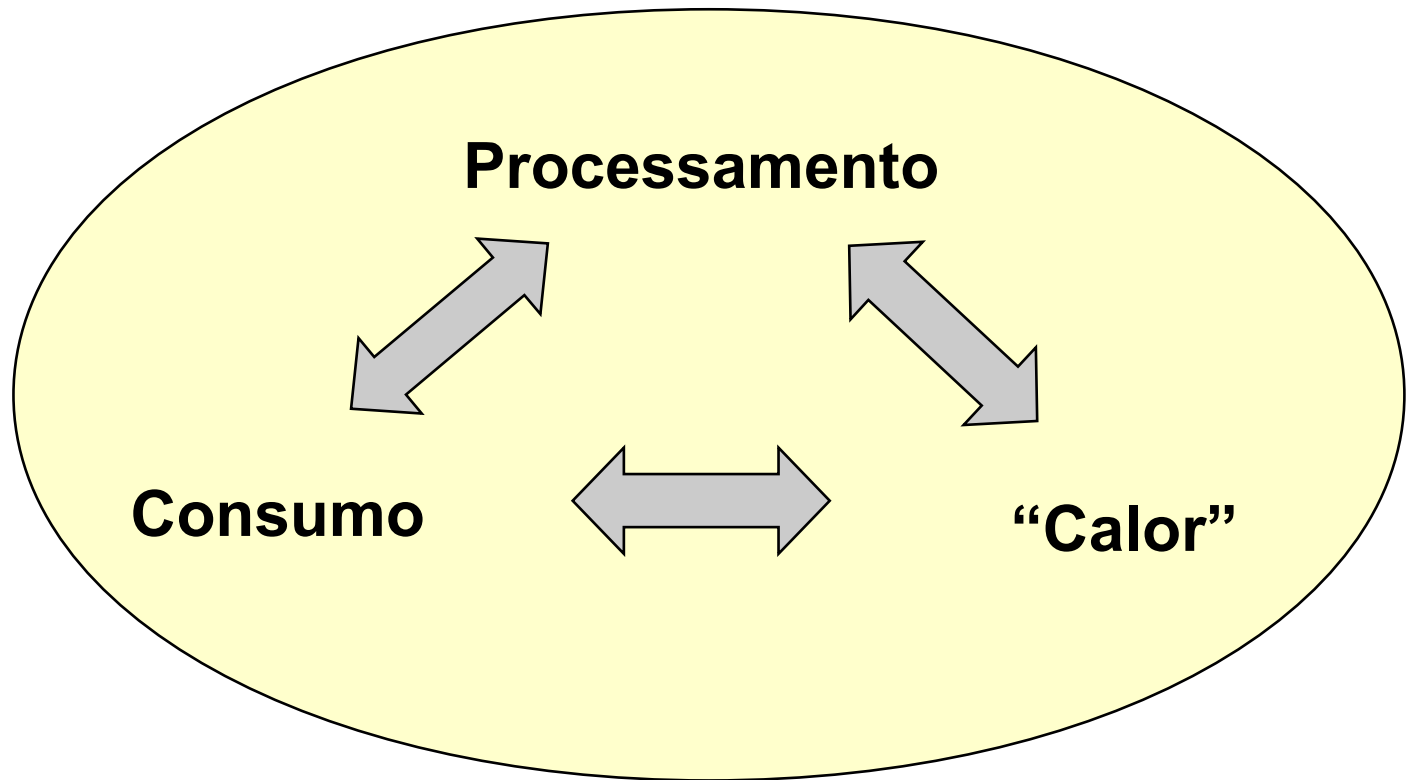
- ITRON é o SO mais popular do mundo

Computadores Embutidos



Sistemas Embarcados

- **Desafio**



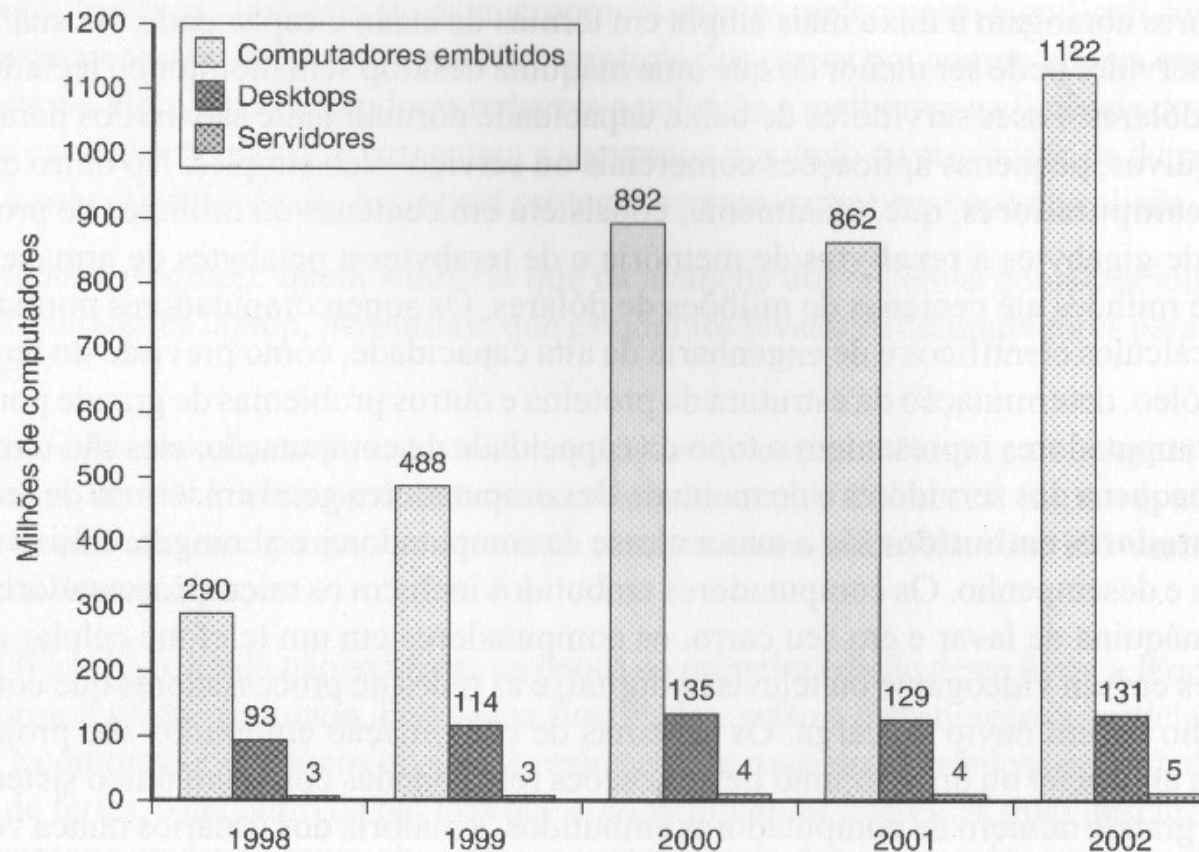


FIGURA 1.1 Número de processadores diferentes vendidos entre 1998 e 2002. Esses números são obtidos de maneira um tanto diferente e, portanto, é necessário algum cuidado na interpretação dos resultados. Por exemplo, os totais para os desktops e servidores consideram sistemas computacionais completos, pois uma parte deles inclui processadores múltiplos; o número de processadores vendidos é um pouco maior, mas provavelmente em apenas 10 a 20% no total (já que os servidores, que podem ter em média mais de um processador por sistema, são apenas cerca de 3% das vendas de desktops, que são predominantemente sistemas monoprocessados). Os totais para computadores embutidos consideram realmente processadores, muitos dos quais não são sequer visíveis e, em alguns casos, pode haver vários processadores por dispositivo.