

Arquitetura de Computadores

EST-UEA

Prof. Carlos Maurício Seródio Figueiredo

Parte I: Fundamentos de Projetos de Computadores

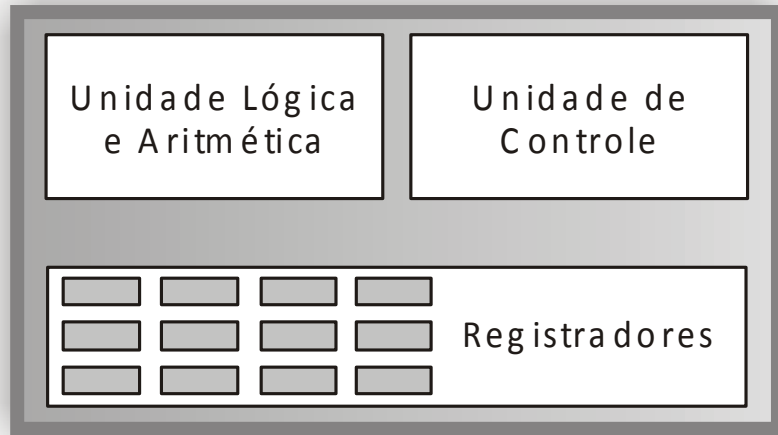
Introdução

Introdução

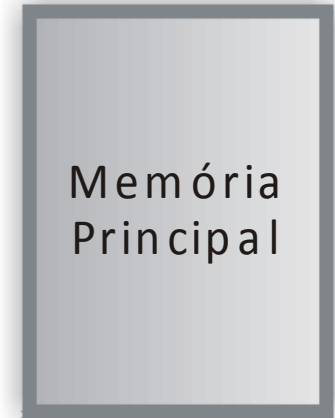
- Arquitetura de Computadores
 - “os atributos de um sistema de computação na visão do programador, i.e., a estrutura conceitual e o comportamento funcional, ... em oposição à implementação física.” - Amdahl, Blaaw, and Brooks, 1964.
- Importante:
 - Enteder a arquitetura de computadores do ponto de vista do programador;
 - O que o programador escreve;
 - Como é convertido para algo que o computador entende;
 - Como o computador interpreta o programa;
 - O que torna programas lentos ou rápidos.

Estrutura Básica de um Computador

Processador / UCP



Memória Principal



Barramento

Dispositivos de E/S



Estrutura Básica

- Processador:
 - Gerencia todo o sistema computacional controlando as operações realizadas por cada unidade funcional.
 - Sua função é executar instruções presentes na memória principal.
 - São compostos por unidade de controle, unidade lógica e aritmética, e registradores.
 - RISC x CISC
 - RISC: Poucas instruções simples executadas diretamente pelo hardware.
Ex: Sparc, Alpha, MIPS
 - CISC: muitas instruções complexas interpretadas por microprogramas.
Ex: x86

Estrutura Básica

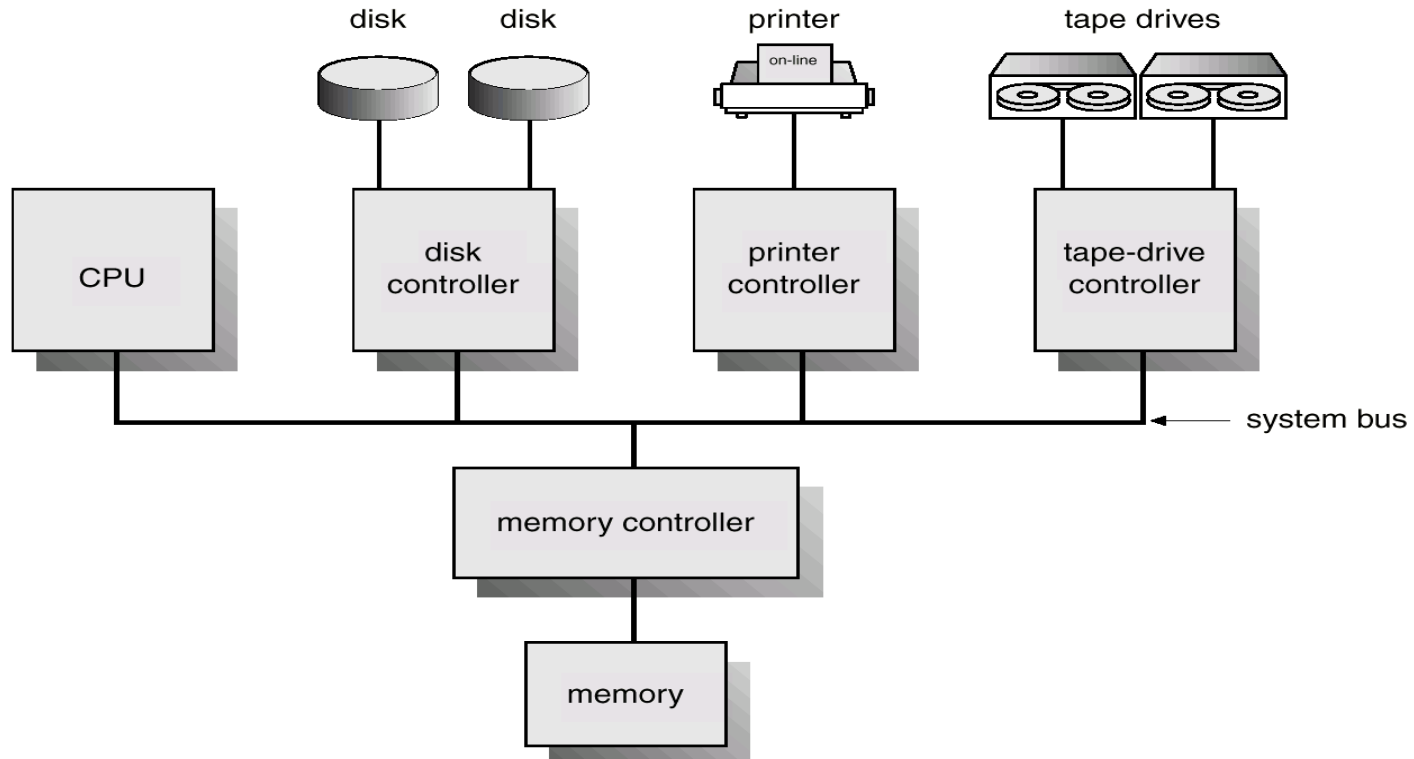
- Processador:
 - Tipos básicos de comandos:
 - carregar dados em registradores ou armazená-los, de/para memória ou I/O;
 - efetuar operações com dados, indo buscar os operandos, se necessário, e guardando o resultado no fim;
 - re-definir o (local do) próximo comando a executar.

Estrutura Básica

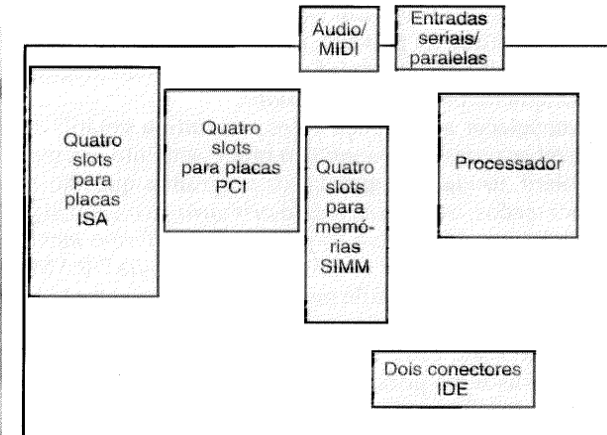
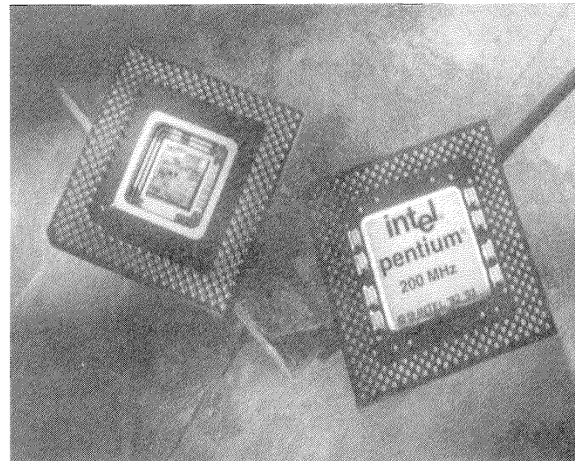
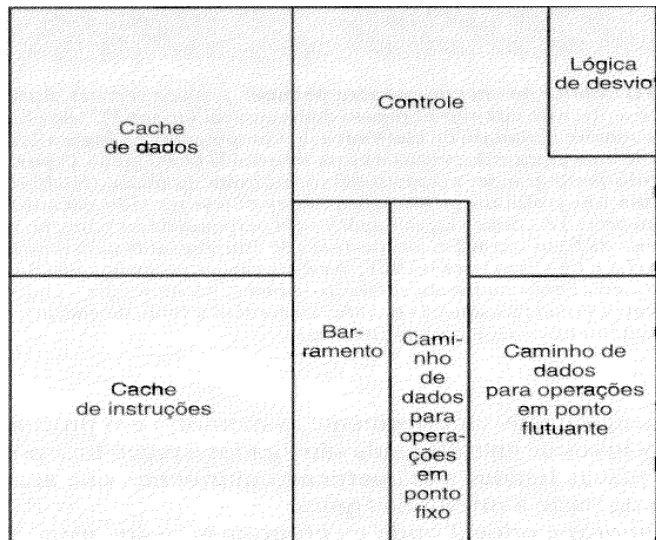
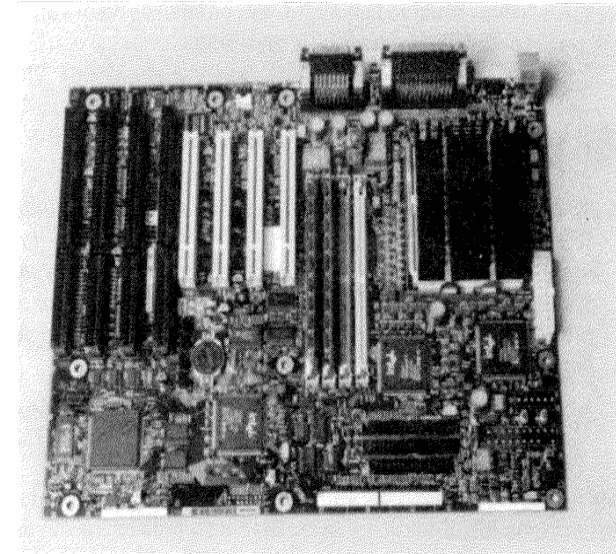
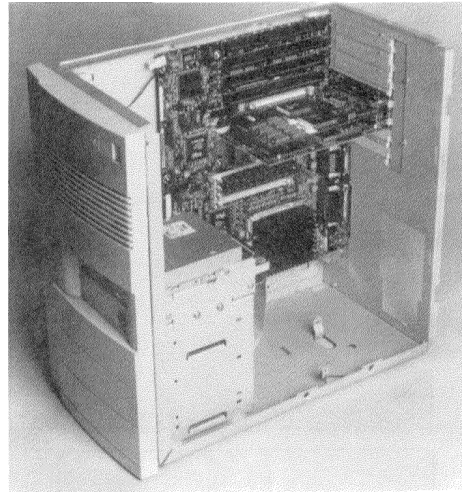
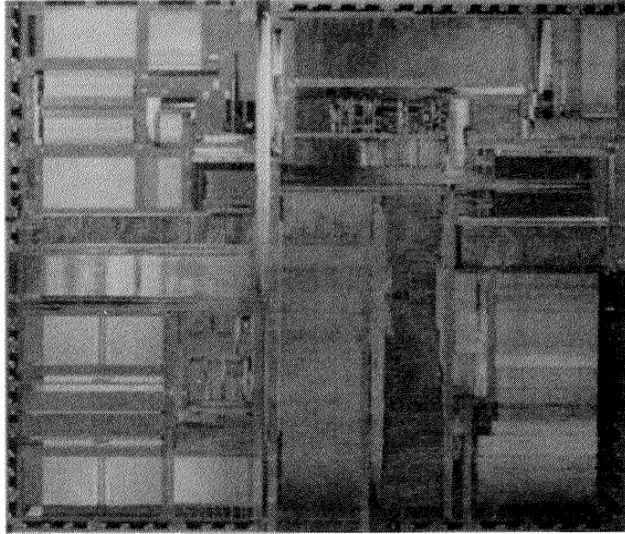
- Memória Principal:
 - Armazena instruções e dados em um esquema de endereçamento.
 - Único tipo de memória acessado diretamente pela UCP.
- Dispositivos de E/S:
 - Permite comunicação com o mundo externo.
 - Comuns: Dispositivos de armazenamanto secundário, dispositivos de interface homem-máquina e interfaces de rede.
 - Normalmente são manipulados através das controladoras de dispositivos.

Estrutura Básica

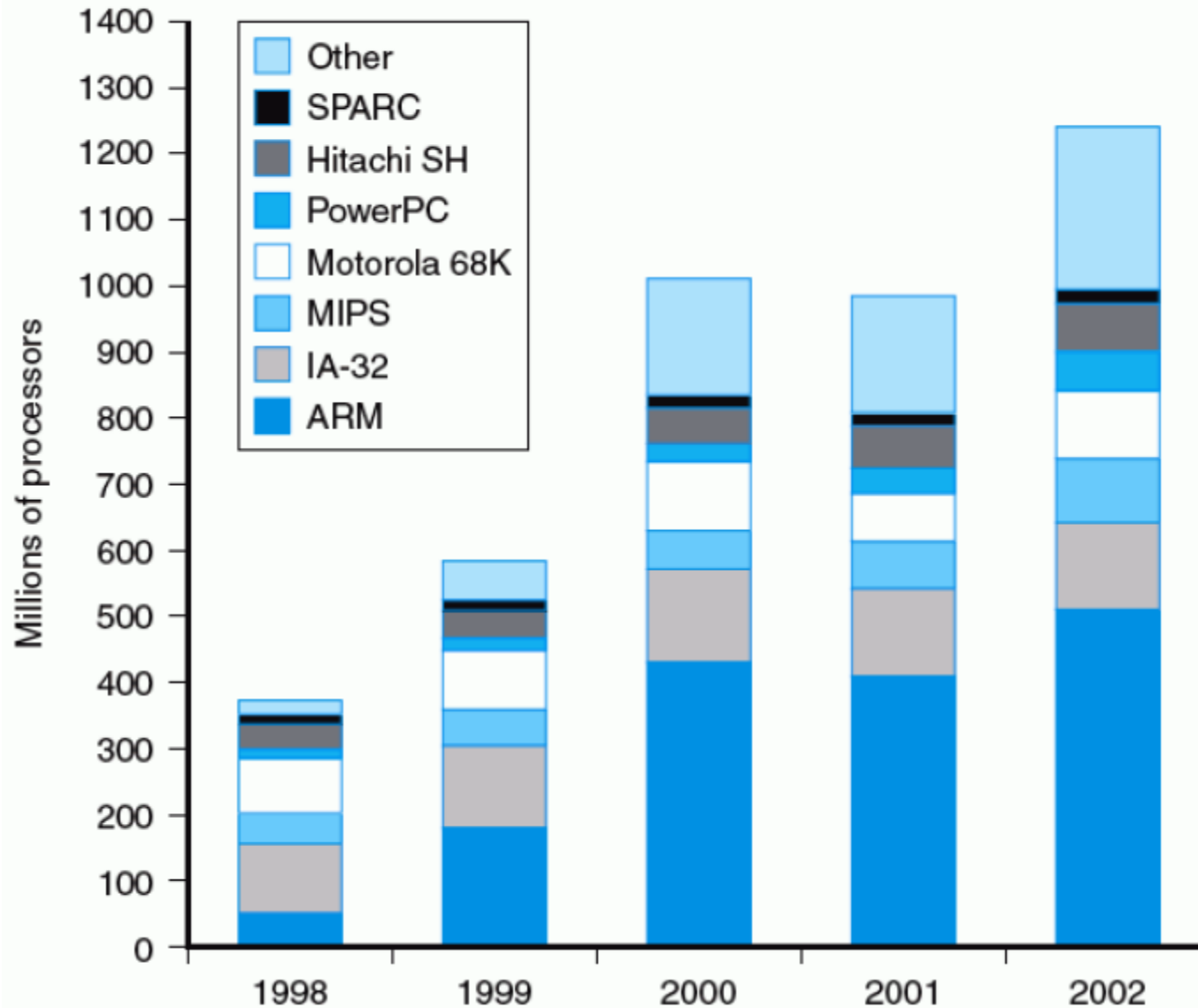
- Barramento:
 - Meio de comunicação compartilhado que interliga as unidades funcionais.



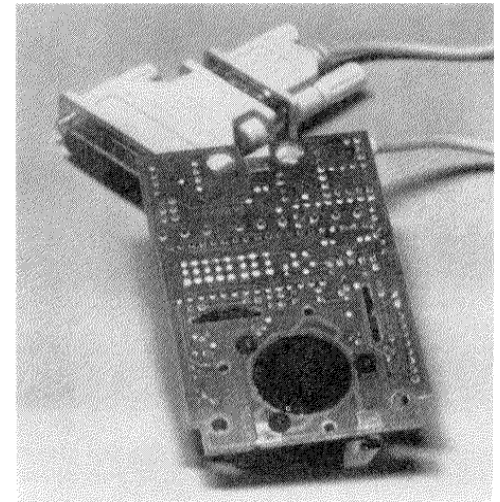
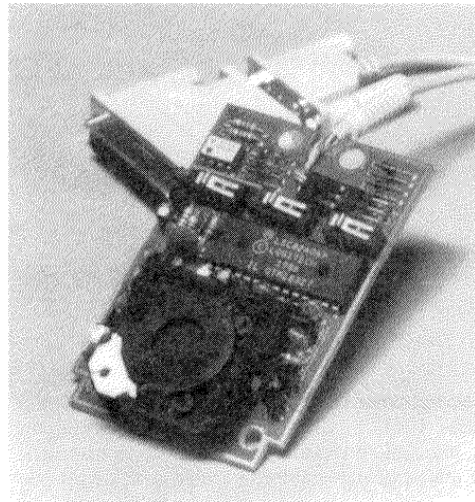
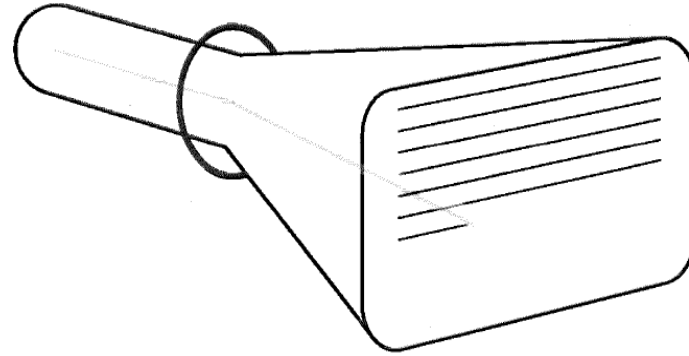
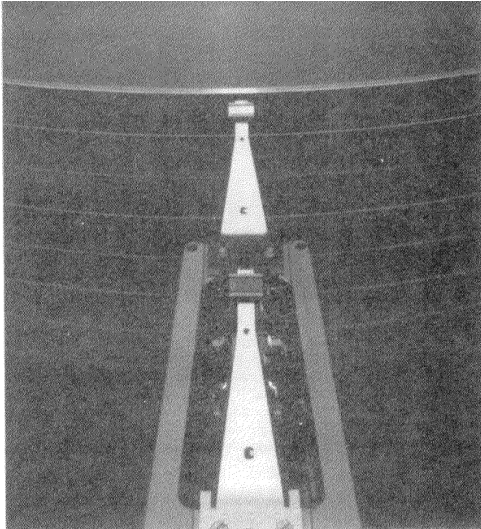
Olhando por dentro



Processadores

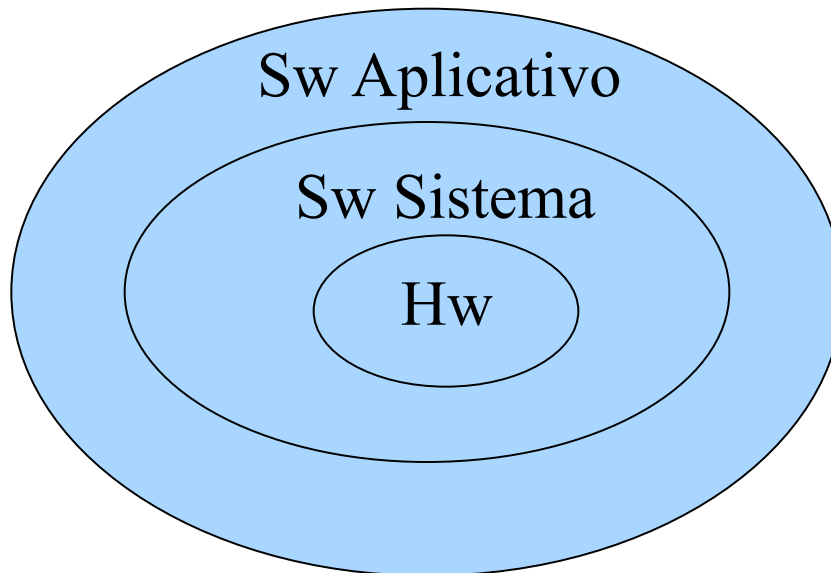


Olhando por dentro

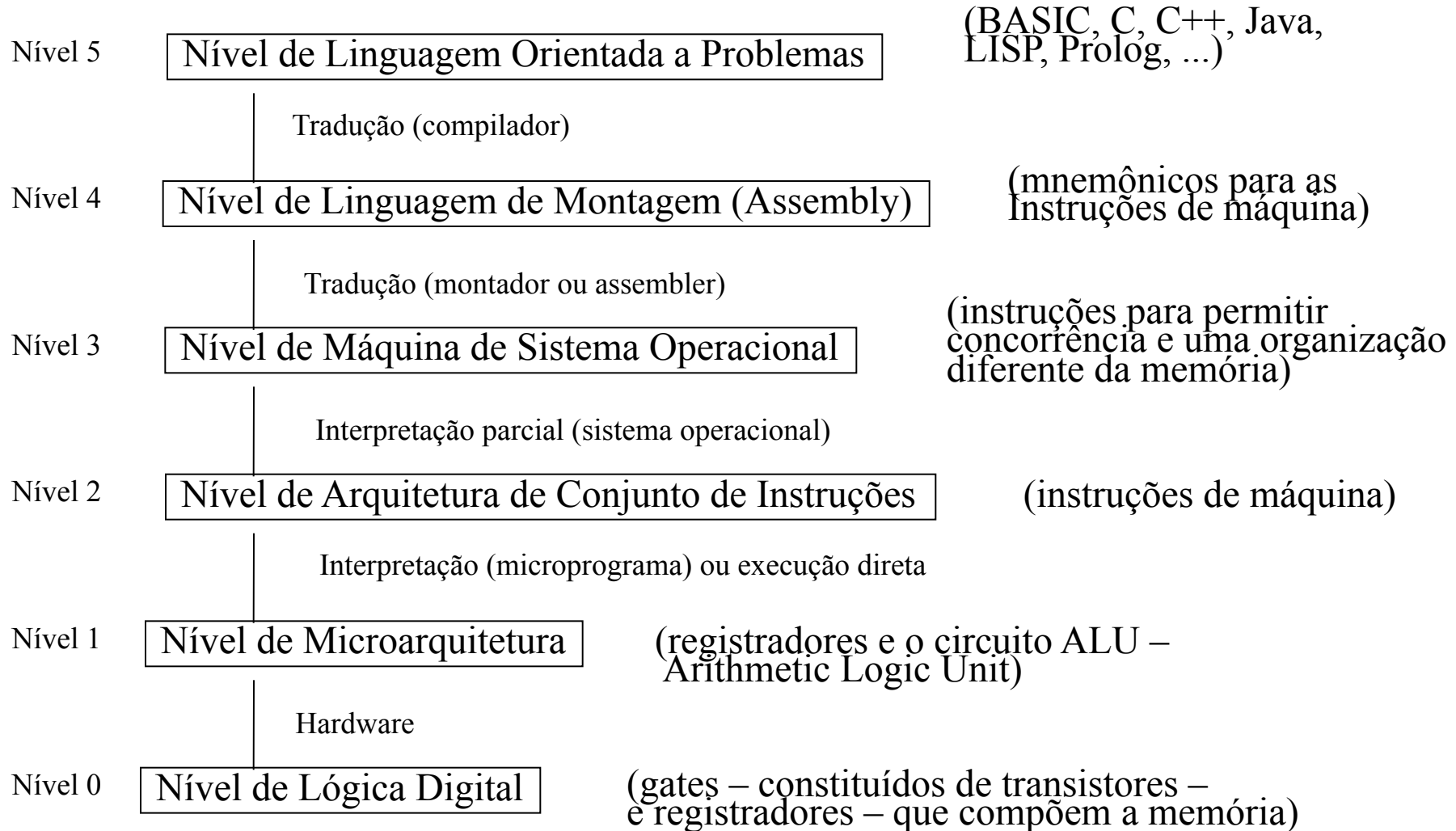


Execução de Programas

- É a finalidade dos computadores
- Computadores são o Hardware de uso geral
- Programas e Aplicativos são o Software
- Visão em Camadas:



Máquina Multinível Contemporânea



Execução de Programas

Processo para geração de código binário:

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 linguagem
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
```

↓

Assembler

↓

Programa em
linguagem de
máquina binária
(para MIPS)

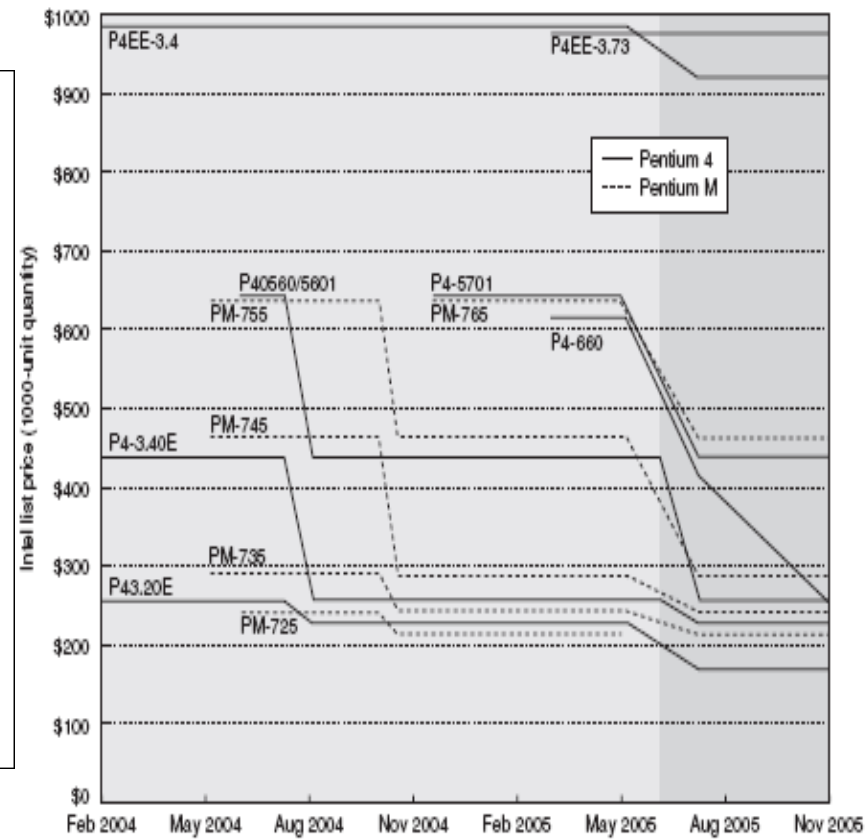
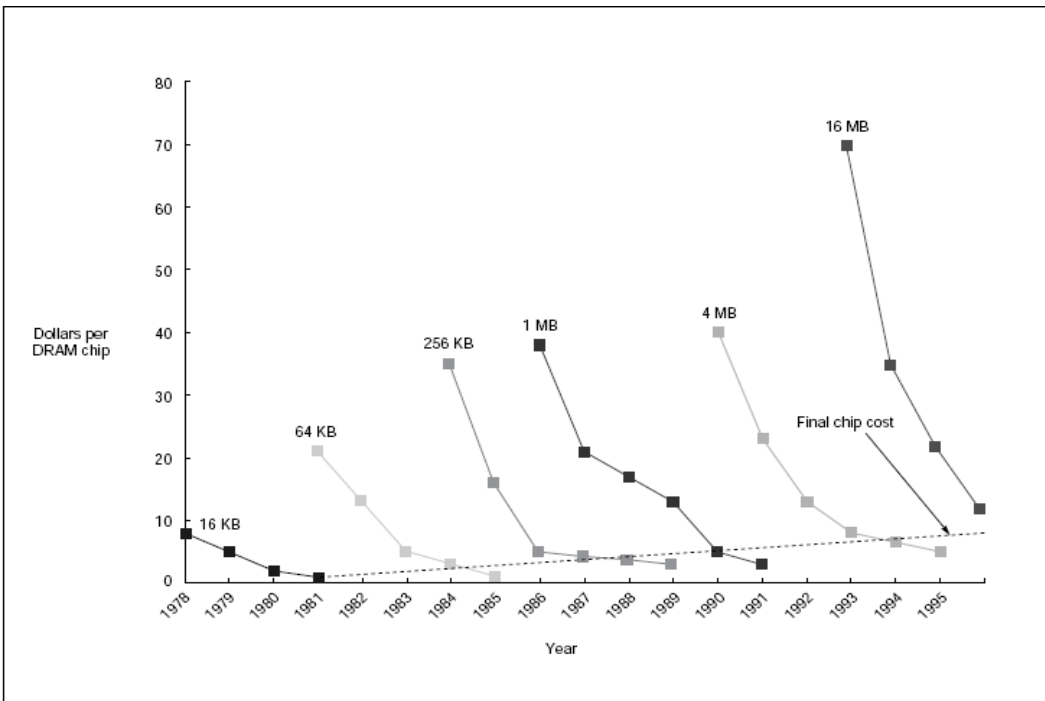
```
000000001010000100000000000000011000
0000000000001100110000001100000100001
100011000110001000000000000000000000
100011001111001000000000000000000100
101011001111001000000000000000000000
101011000110001000000000000000000100
000000111110000000000000000000000100
```

Custos

Custos

- Vários aspectos do projeto de um computador afetam em seu custo
- Entender esses fatores permitem a tomada de decisões melhores
- Fator TEMPO
 - Custos de manufatura tendem a diminuir no tempo
 - Aprendizado, melhorias de processo
 - Ex. Custo do MByte DRAM cai 40% por ano
 - Preços de processadores também caem, mas não tão uniformemente

Custos



Custos

- Fator Volume

- Maior volume de vendas causa diminuição de preços
- Amortização dos custos de projeto e testes pelo mercado de massa

- Fator Comódites

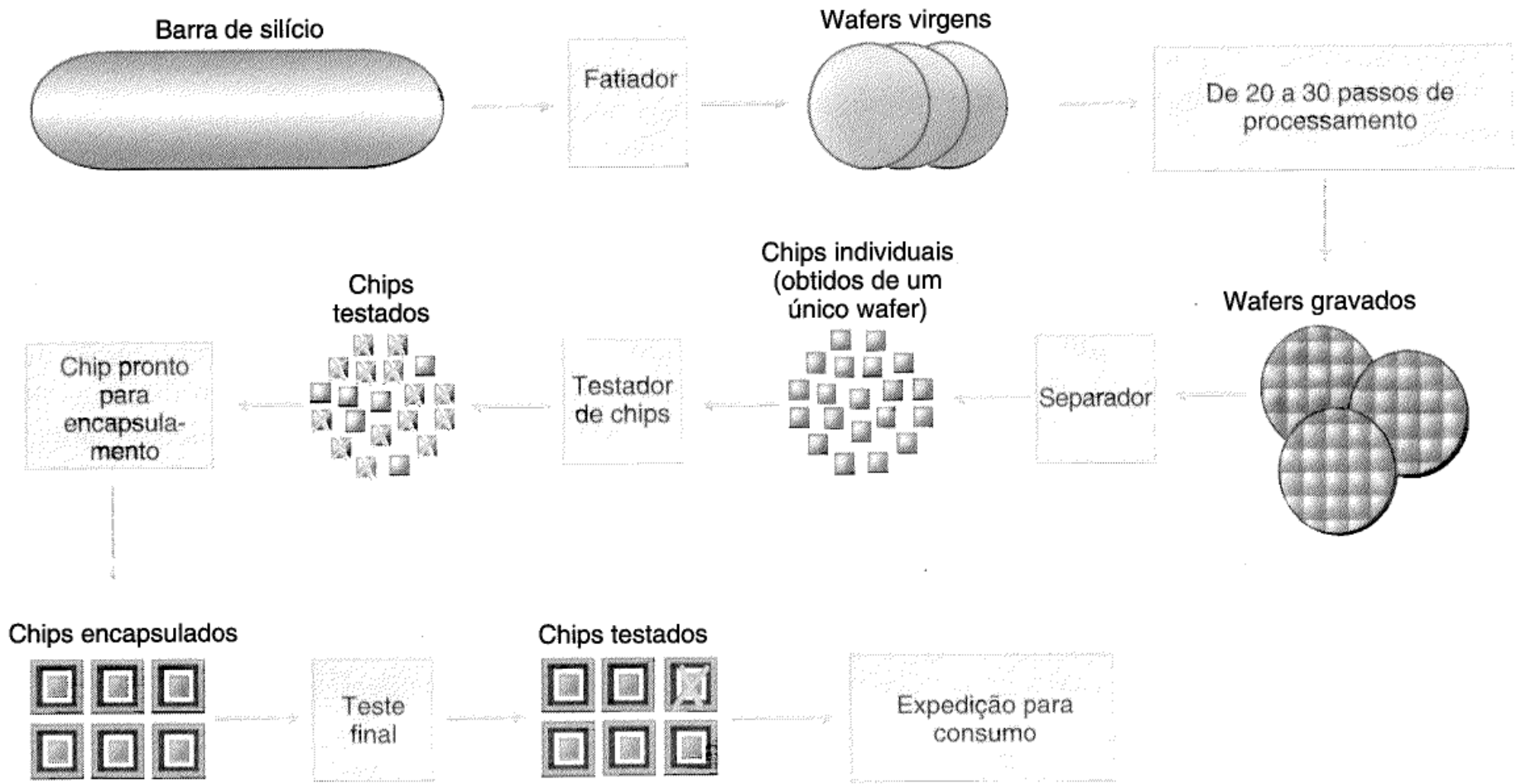
- Mais pontos de venda e mais fabricantes tendem a causar queda de preços
- Concorrência força eficiência no processo

Custo de Circuitos Integrados

- Boa parte dos custos de um computador se referem aos circuitos integrados
- Exemplo de distribuição de custos

System	Subsystem	Fraction of total
Cabinet	Sheet metal, plastic	1%
	Power supply, fans	2%
	Cables, nuts, bolts	1%
	Shipping box, manuals	0%
	Subtotal	4%
Processor board	Processor	6%
	DRAM (64 MB)	36%
	Video system	14%
	I/O system	3%
	Printed circuit board	1%
	Subtotal	60%
I/O devices	Keyboard and mouse	1%
	Monitor	22%
	Hard disk (1 GB)	7%
	DAT drive	6%
	Subtotal	36%

Processo de Fabricação de CIs



Exemplo de um Wafer

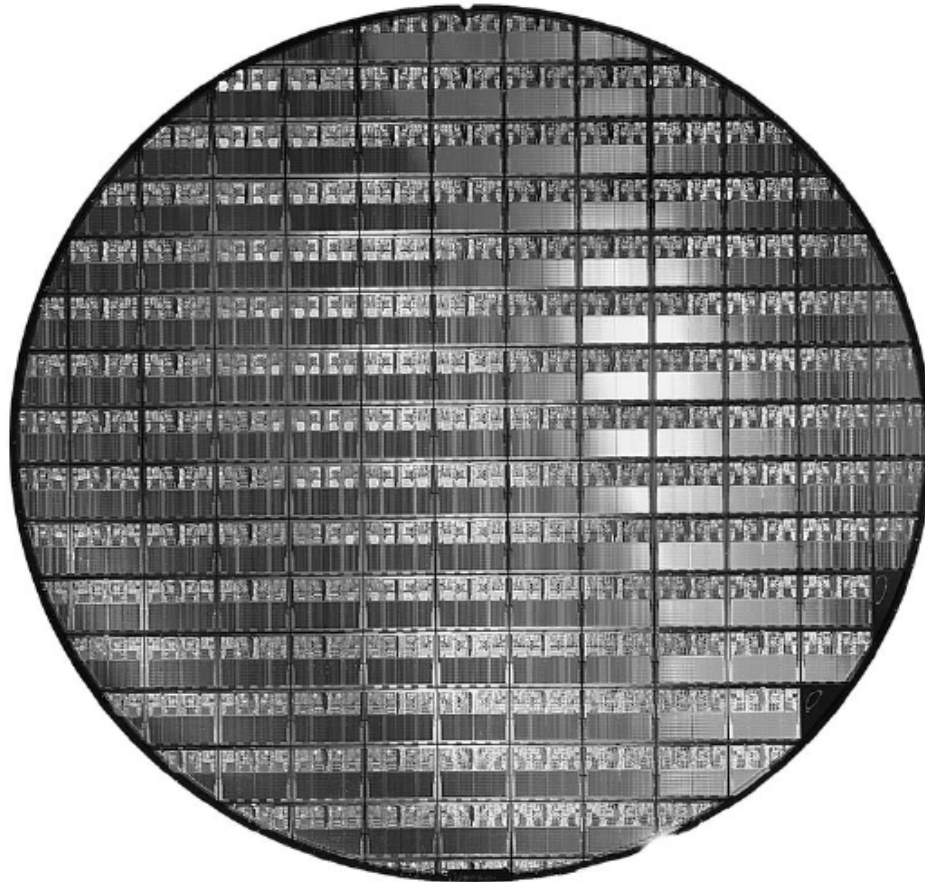


Figure 1.12 This 300mm wafer contains 117 AMD Opteron chips implemented in a 90 nm process. (Courtesy AMD.)

Fórmulas de custo de CIs

- A prática mostrou que há muitas perdas de CIs por defeitos em wafers.
 - Portanto, rendimento é levado em consideração.
 - Fórmula do rendimento obtida de forma empírica após anos de observação

$$\text{Custo por chip} = \frac{\text{Custo por wafer}}{\text{Chips por wafer} \times \text{rendimento}}$$

$$\text{Chips por Wafer} \approx \frac{\text{Área do wafer}}{\text{Área do chip}}$$

$$\text{Rendimento} = \frac{1}{\left(1 + \left(\text{Defeitos por área} \times \text{Área do chip}/2\right)\right)^2}$$

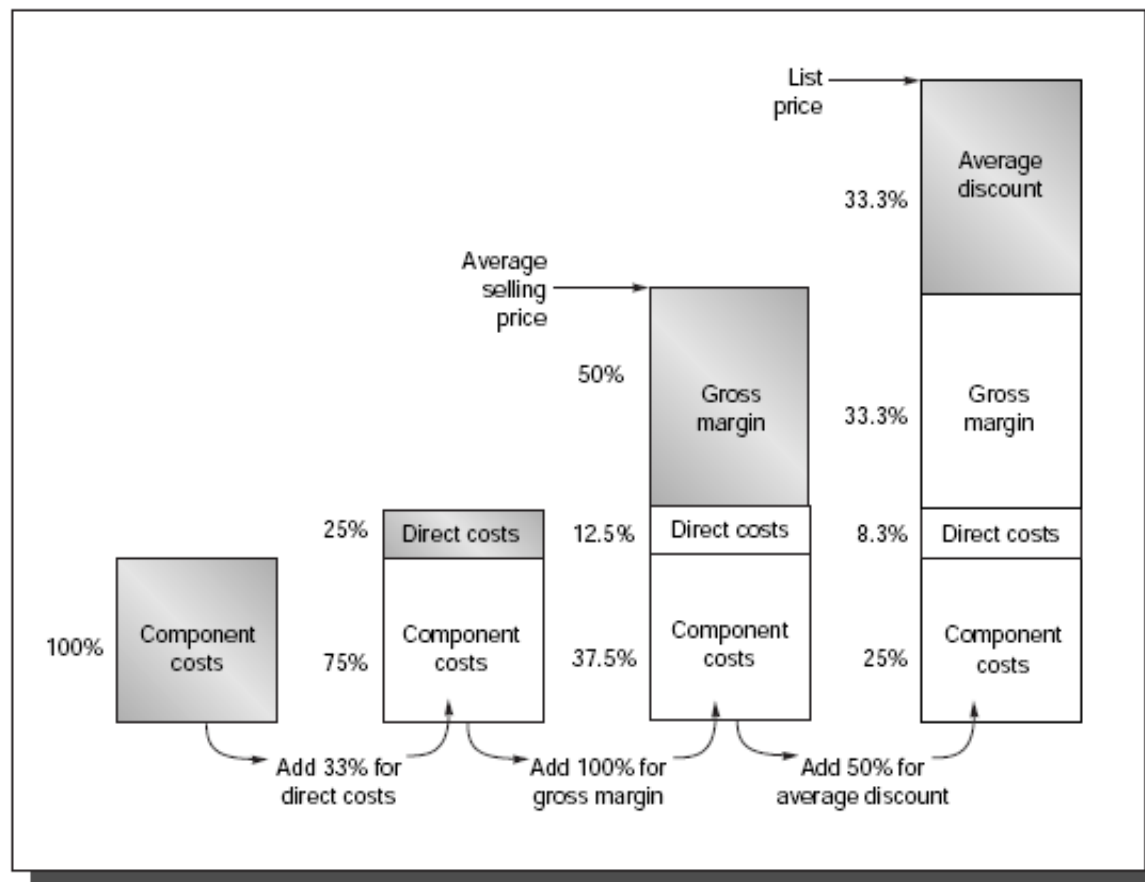
- Quantidade de chips é um aproximação, pois wafer são redondos e chips são quadrados.

Custos de CIs

- Ex. 1: Qual o rendimento de um Wafer para um chip de lado 1cm e para um chip de lado 2cm, assumindo um defeito por área de 0.5 por cm^2 ?
- Ex. 2: Qual o custo de cada um desses chips assumindo um wafer de 20 cm de diâmetro a \$ 100?

Formação de preço

- Obviamente, o preço de um produto leva em consideração margens de lucro e outros fatores comerciais.



Desempenho: Busca Eterna

Evolução de Processadores

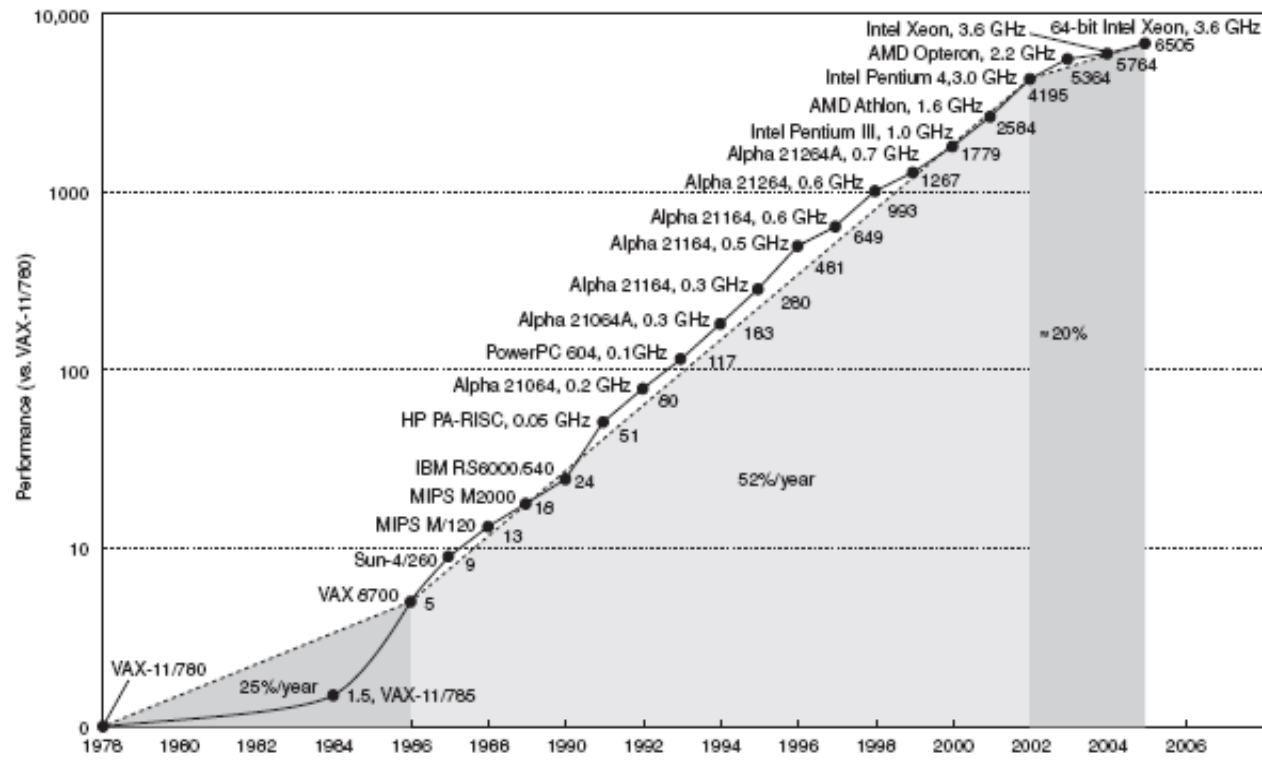
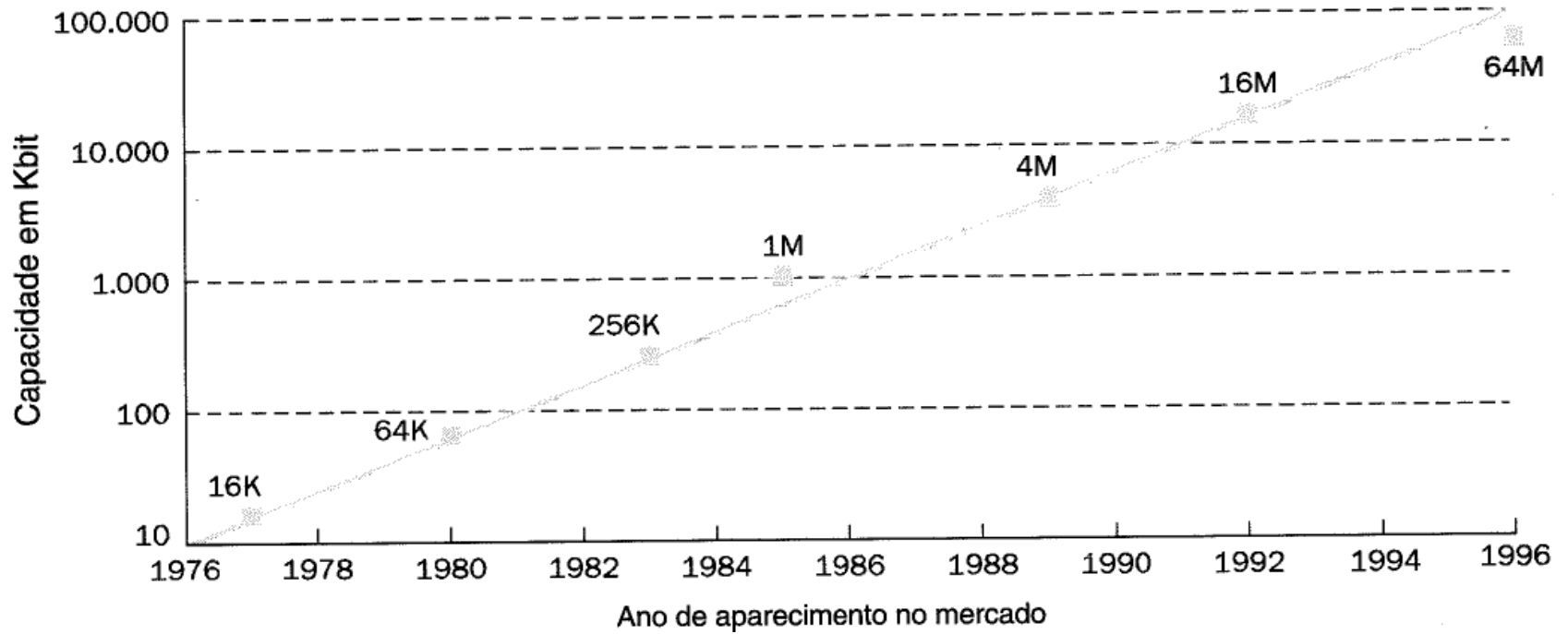


Figure 1.1 Growth in processor performance since the mid-1980s. This chart plots performance relative to the VAX 11/780 as measured by the SPECint benchmarks (see Section 1.8). Prior to the mid-1980s, processor performance growth was largely technology driven and averaged about 25% per year. The increase in growth to about 52% since then is attributable to more advanced architectural and organizational ideas. By 2002, this growth led to a difference in performance of about a factor of seven. Performance for floating-point-oriented calculations has increased even faster. Since 2002, the limits of power, available instruction-level parallelism, and long memory latency have slowed uniprocessor performance recently, to about 20% per year. Since SPEC has changed over the years, performance of newer machines is estimated by a scaling factor that relates the performance for two different versions of SPEC (e.g., SPEC92, SPEC95, and SPEC2000).

Evolução de Memórias



Marcos da Evolução

Microprocessor	16-bit address/bus, microcoded	32-bit address/bus, microcoded	5-stage pipeline, on-chip I & D caches, FPU	2-way superscalar, 64-bit bus	Out-of-order 3-way superscalar	Out-of-order superpipelined, on-chip 1.2 cache
Product	Intel 80286	Intel 80386	Intel 80486	Intel Pentium	Intel Pentium Pro	Intel Pentium 4
Year	1982	1985	1989	1993	1997	2001
Die size (mm ²)	47	43	81	90	308	217
Transistors	134,000	275,000	1,200,000	3,100,000	5,500,000	42,000,000
Pins	68	132	168	273	387	423
Latency (clocks)	6	5	5	5	10	22
Bus width (bits)	16	32	32	64	64	64
Clock rate (MHz)	12.5	16	25	66	200	1500
Bandwidth (MIPS)	2	6	25	132	600	4500
Latency (ns)	320	313	200	76	50	15
Memory module	DRAM	Page mode DRAM	Fast page mode DRAM	Fast page mode DRAM	Synchronous DRAM	Double data rate SDRAM
Module width (bits)	16	16	32	64	64	64
Year	1980	1983	1986	1993	1997	2000
Mbits/DRAM chip	0.06	0.25	1	16	64	256
Die size (mm ²)	35	45	70	130	170	204
Pins/DRAM chip	16	16	18	20	54	66
Bandwidth (MBit/sec)	13	40	160	267	640	1600
Latency (ns)	225	170	125	75	62	52
Local area network	Ethernet	Fast Ethernet	Gigabit Ethernet	10 Gigabit Ethernet		
IEEE standard	802.3	803.3u	802.3ab	802.3ac		
Year	1978	1995	1999	2003		
Bandwidth (MBit/sec)	10	100	1000	10000		
Latency (μsec)	3000	500	340	190		
Hard disk	3600 RPM	5400 RPM	7200 RPM	10,000 RPM	15,000 RPM	
Product	CDC WrenI 94145-36	Seagate ST41600	Seagate ST15150	Seagate ST39102	Seagate ST373453	
Year	1983	1990	1994	1998	2003	
Capacity (GB)	0.03	1.4	4.3	9.1	73.4	
Disk form factor	5.25 inch	5.25 inch	3.5 inch	3.5 inch	3.5 inch	
Media diameter	5.25 inch	5.25 inch	3.5 inch	3.0 inch	2.5 inch	
Interface	ST-412	SCSI	SCSI	SCSI	SCSI	
Bandwidth (MBit/sec)	0.6	4	9	24	86	
Latency (ms)	48.3	17.1	12.7	8.8	5.7	

Introdução

- Como medir e informar desempenho?
- Tarefa não trivial
 - Hardware moderno diverso e complexo
 - Diversidade gigantesca de programas e aplicações
 - Difícil determinar a velocidade de como determinado programa rodará em determinado computador
 - Mas o desempenho é o principal fator de escolha de determinada arquitetura de computador
- Interesse mais amplo!
 - Entender como determinado software se comporta
 - Como implementar determinado conjunto de instruções para melhor desempenho
 - Características do hardware que afetam significativamente o desempenho

Definição de Desempenho

- Depende da necessidade do “cliente”
 - Ex. Aviões de diferentes tamanhos e velocidades
- Métricas importantes a computadores:
 - Tempo de resposta: Tempo que um programa leva para ter sua execução concluída. Quanto menor melhor.
 - Throughput: Quantidade de trabalho executado em um determinado intervalo de tempo. Quanto maior melhor.
- Ex. Processadores podem executar um programa por vez ou executar vários em esquema de tempo compartilhado.
 - 1. Uma máquina que tem seu processador substituído por um mais rápido, como afeta essas métricas?
 - 2. E uma máquina que tem mais processadores iguais podendo rodar mais processos?
- Normalmente, uma métrica afeta a outra e vice-versa.

Tempo de Execução

- Métrica mais básica
 - Quanto menor, melhor
- Se compararmos Máq. X e Y

$$\text{Performance}_x = \frac{1}{\text{Tempo de execução}_x}$$

$$\frac{\text{Performance}_x}{1} > \frac{\text{Performance}_y}{1}$$
$$\frac{1}{\text{Tempo de execução}_x} > \frac{1}{\text{Tempo de execução}_y}$$

$$\text{Tempo de execução}_y > \text{Tempo de execução}_x$$

- Quantas vezes X é mais rápida que Y (n)?

$$\frac{\text{Performance}_x}{\text{Performance}_y} = \frac{\text{Tempo de execução}_y}{\text{Tempo de execução}_x} = n$$

Tempo de Execução

- Ex. A roda um programa em 10 s, e B em 15 s.
Quantas vezes A é **mais rápida** que B?
- O tempo de execução de um programa pode depender de vários outros fatores, como o tempo gasto pelo SO, o tempo gasto do processador com I/O ou tempo gasto com outros programas concorrentes.
 - Uma medida do tempo de término de um programa pode mudar em uma mesma máquina. É impreciso.
 - Necessidade de outras métricas para focar o hardware.

Clocks/Ciclos

- Processadores executam suas atividades mais fundamentais em ciclos com períodos constantes.
- O período de um ciclo de clock é o tempo necessário para completá-lo.
- Ex. Um processador de clock de 1 GHz, cujo período é de 1 ns.
- Período é o inverso da frequência de clock.
- Relação com o tempo de execução:

$$\begin{array}{l} \text{Tempo de execução} \\ \text{no processador} \\ \text{para o programa} \end{array} = \begin{array}{l} \text{Número de ciclos do} \\ \text{clock do processador} \\ \text{para o programa} \end{array} \times \text{Ciclo do clock}$$

ou, alternativamente,

$$\begin{array}{l} \text{Tempo de execução} \\ \text{no processador} \\ \text{para o programa} \end{array} = \frac{\begin{array}{l} \text{Número de ciclos do clock do} \\ \text{processador para o programa} \end{array}}{\text{Frequência do clock}}$$

- > Desempenho: Menor ciclo ou menos ciclos por programa

Exemplo

Nosso programa favorito roda em 10 segundos em um computador A, que tem um clock de 400 MHz. Estamos tentando ajudar um projetista de computador em início de carreira a construir uma máquina B, que deverá rodar nosso programa em 6 segundos. O projetista já sabe que pode contar com a tecnologia para aumentar de modo significativo a frequência do clock da sua máquina, mas este aumento vai provocar reflexos em outros parâmetros da performance, fazendo com que a máquina B precise de 1,2 vez mais ciclos do que a máquina A para executar o tal programa. Qual a frequência do clock que nosso projetista deve implementar em sua máquina?

Solução

- Número de ciclos em A:

$$\text{Tempo de processador}_A = \frac{\text{Número de ciclos de clock do processador}_A}{\text{Frequência do clock}_A}$$

$$10 \text{ segundos} = \frac{\text{Número de ciclos de clock do processador}_A}{400 \times 10^6 \frac{\text{ciclos}}{\text{segundo}}}$$

$$\begin{aligned} \text{Número de ciclos de clock do processador}_A &= 10 \text{ segundos} \times 400 \times 10^6 \frac{\text{ciclos}}{\text{segundo}} \\ &= 4000 \times 10^6 \text{ ciclos} \end{aligned}$$

- Clock em B a partir do tempo de proc. Em B

$$\text{Tempo de processador}_B = \frac{1,2 \times \text{Número de ciclos de clock do processador}_A}{\text{Frequência do clock}_B}$$

$$6 \text{ segundos} = \frac{1,2 \times 4000 \times 10^6 \text{ ciclos}}{\text{Frequência do clock}_B}$$

$$\begin{aligned} \text{Frequência do clock}_B &= \frac{1,2 \times 4000 \times 10^6 \text{ ciclos}}{6 \text{ segundos}} \\ &= \frac{800 \times 10^6 \text{ ciclos}}{\text{segundo}} = 800 \text{ MHz} \end{aligned}$$

Relação com Número de Instruções

- É importante relacionar desempenho do hardware com o software que eles executa.
- Programas são um conjunto de instruções.
- O tempo de execução de um programa é o tempo de execução de todas as suas instruções, uma a uma.
- Uma forma de relacionar a execução de instruções é considerar o tempo médio de execução destas
 - Obs. Instruções diferentes podem levar diferentes nros de ciclos
- Em número de ciclos de clock:
 - Número médio de ciclos por instrução é chamado CPI

$$\begin{array}{l} \text{Número de} \\ \text{ciclos de clock} \\ \text{do processador} \end{array} = \begin{array}{l} \text{Número de} \\ \text{instruções de} \\ \text{um programa} \end{array} \times \begin{array}{l} \text{Número} \\ \text{médio de ciclos} \\ \text{por instrução} \end{array}$$

Exemplo

Suponha que temos duas implementações diferentes da mesma arquitetura do conjunto de instruções. A máquina A tem um ciclo de clock de 1 ns e uma CPI de 2,0 considerando um programa qualquer. A máquina B tem um ciclo de clock de 2 ns e uma CPI de 1,2, para o mesmo programa. Qual das duas máquinas executa esse programa mais rapidamente? Calcule também quanto uma é mais rápida que a outra.

Solução

- As duas máquinas executam o mesmo número de instruções (I), pois trata-se do mesmo programa

$$\text{Ciclos}_A = I \times 2,0$$

$$\text{Ciclos}_B = I \times 1,2$$

- Tempo de proc. A e B

$$\begin{aligned}\text{Tempo}_A &= \text{Ciclos}_A \times \text{Clock}_A = I \times 2,0 \times 1 \text{ ns} \\ &= 2 \times I \text{ ns}\end{aligned}$$

$$\begin{aligned}\text{Tempo}_B &= \text{Ciclos}_B \times \text{Clock}_B = I \times 1,2 \times 2 \text{ ns} \\ &= 2,4 \times I \text{ ns}\end{aligned}$$

- A mais rápido que B, pelo fator de:

$$\begin{aligned}\frac{\text{Performance do processador}_A}{\text{Performance do processador}_B} &= \frac{\text{Tempo de execução}_B}{\text{Tempo de execução}_A} \\ &= \frac{2,4 \times I \text{ ns}}{2 \times I \text{ ns}} = 1,2\end{aligned}$$

Fórmula básica

- Relação dos principais fatores que afetam o desempenho de um computador:
 - Clock, CPI e Nro de Instruções

$$\text{Tempo de processador} = \text{Número de instruções} \times \text{CPI} \times \text{ciclo de clock}$$

ou

$$\text{Tempo de processador} = \frac{\text{Número de instruções} \times \text{CPI}}{\text{Frequência de clock}}$$

CPI

- Como obter o CPI?
 - Medição de programas
- Determinadas arquiteturas têm quantidades de ciclos diferentes para diferentes instruções
 - CPI depende dos programas
 - CPI pode ser levado em consideração individualmente ou em classes, conhecendo-se os diferentes tipos de instruções e suas frequências de execução (C_i = Nro de instruções de um tipo ou classe)

$$\text{Número de ciclos do processador} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

Exemplo

Um projetista de compilador está tentando decidir entre duas seqüências de código para uma determinada máquina. Para tanto, obteve com a equipe de hardware os seguintes dados:

Classe de instrução	CPI para esta classe de instrução
A	1
B	2
C	3

Considerando o código a ser gerado para uma particular declaração de uma linguagem de alto nível, o responsável pelo projeto do compilador está considerando duas possíveis seqüências de código, com as seguintes contagens de instruções:

Seqüência de código	Número de instrução para a classe		
	A	B	C
1	2	1	2
2	4	1	1

Qual das duas seqüências executa mais instruções? Qual a mais rápida? Qual a CPI para cada seqüência?

Solução

- Seq. 1 executa 5 instruções. Seq. 2, 6 instruções.
- Total de ciclos para as seqs.:

$$\begin{aligned}\text{Ciclos de clock}_1 &= (2 \times 1) + (1 \times 2) + (2 \times 3) = \\ &= 2 + 2 + 6 = 10 \text{ ciclos}\end{aligned}$$

$$\begin{aligned}\text{Ciclos de clock}_2 &= (4 \times 1) + (1 \times 2) + (1 \times 3) = \\ &= 4 + 2 + 3 = 9 \text{ ciclos}\end{aligned}$$

- Seq. 2 mais rápida com mais instruções!!
- CPIs:

$$\text{CPI} = \frac{\text{Ciclos de clock do processador}}{\text{Número de instruções}}$$

$$\text{CPI}_1 = \frac{\text{Ciclos de clock do processador}_1}{\text{Número de instruções}_1} = \frac{10}{5} = 2$$

$$\text{CPI}_2 = \frac{\text{Ciclos de clock do processador}_2}{\text{Número de instruções}_2} = \frac{9}{6} = 1,5$$

Programas especiais para análise de desempenho

- Workload: Conjunto de programas que um usuário usa
- Para comparar computadores, bastaria rodar o mesmo workload nas máquinas
- Mas usuários diferente têm workloads diferentes
- Alternativa: Bechmarks
 - Conjunto de programas próprios para avaliar desempenho
 - Tentam representar sequências de instruções importantes e comuns em programas reais
 - Como são muito conhecidos, podem ser burlados
 - Otimizações podem ser feitas visando apenas trechos muito específicos do benchmark
 - Exemplo da Intel Pentium, em 92, que uso um compilador especial no teste matrix300 do SPEC
 - Sempre será melhor avaliar com base em programas reais

SPEC92 System Performance Evaluation Cooperative

- Benchmark baseado em alguns programas reais

Benchmark	Source	Lines of code	Description
espresso	C	13,500	Minimizes Boolean functions.
li	C	7,413	A lisp interpreter written in C that solves the 8-queens problem.
eqntott	C	3,376	Translates a Boolean equation into a truth table.
compress	C	1,503	Performs data compression on a 1-MB file using Lempel-Ziv coding.
sc	C	8,116	Performs computations within a UNIX spreadsheet.
gcc	C	83,589	Consists of the GNU C compiler converting preprocessed files into optimized Sun-3 machine code.
spice2g6	FORTRAN	18,476	Circuit simulation package that simulates a small circuit.
doduc	FORTRAN	5,334	A Monte Carlo simulation of a nuclear reactor component.
mdljdp2	FORTRAN	4,458	A chemical application that solves equations of motion for a model of 500 atoms. This is similar to modeling a structure of liquid argon.
wave5	FORTRAN	7,628	A two-dimensional electromagnetic particle-in-cell simulation used to study various plasma phenomena. Solves equations of motion on a mesh involving 500,000 particles on 50,000 grid points for 5 time steps.
tomcatv	FORTRAN	195	A mesh generation program, which is highly vectorizable.
ora	FORTRAN	535	Traces rays through optical systems of spherical and plane surfaces.
mdljsp2	FORTRAN	3,885	Same as mdljdp2, but single precision.
alvinn	C	272	Simulates training of a neural network. Uses single precision.
ear	C	4,483	An inner ear model that filters and detects various sounds and generates speech signals. Uses single precision.
swm256	FORTRAN	487	A shallow water model that solves shallow water equations using finite difference equations with a 256×256 grid. Uses single precision.
su2cor	FORTRAN	2,514	Computes masses of elementary particles from Quark-Gluon theory.
hydro2d	FORTRAN	4,461	An astrophysics application program that solves hydrodynamical Navier Stokes equations to compute galactical jets.
nasa7	FORTRAN	1,204	Seven kernels do matrix manipulation, FFTs, Gaussian elimination, vortices creation.
fp PPP	FORTRAN	2,718	A quantum chemistry application program used to calculate two electron integral derivatives.

Exemplo de Descrição de Procedimento de Análise

Hardware		Software	
Model number	Powerstation 590	O/S and version	AIX version 3.2.5
CPU	66.67 MHz POWER2	Compilers and version	C SET++ for AIX C/C++ version 2.1 XL FORTRAN/6000 version 3.1
FPU	Integrated	Other software	See below
Number of CPUs	1	File system type	AIX/JFS
Primary cache	32KBI+256KBD off chip	System state	Single user
Secondary cache	None		
Other cache	None		
Memory	128 MB		
Disk subsystem	2x2.0 GB		
Other hardware	None		
SPECbase_fp92 tuning parameters/notes/summary of changes: FORTRAN flags: -O3 -qarch=pwrx -qhsflt -qnofold -bnso -bI:/lib/syscalss.exp C flags: -O3 -qarch=pwrx -Q -qtune=pwrx -qhssngl -bnso -bI:/lib/syscalls.exp			

Exemplo de Relatório de Desempenho

- Programas usados podem ser apresentados individualmente em A, B e C
 - A é 10 x mais rápido que b, A é 20 x C
- Resultados podem ser resultados de médias aritméticas, inclusive com diferentes pesos
 - W1: Média sem peso
 - W2: Média com peso inversamente propor. aos tempos de B
 - W3: Média com peso inv. Prop. A P1 e P2 em A

	A	B	C	W(1)	W(2)	W(3)
Program P1 (secs)	1.00	10.00	20.00	0.50	0.909	0.999
Program P2 (secs)	1000.00	100.00	20.00	0.50	0.091	0.001
Arithmetic mean:W(1)	500.50	55.00	20.00			
Arithmetic mean:W(2)	91.91	18.19	20.00			
Arithmetic mean:W(3)	2.00	10.09	20.00			

$$w_i = \frac{1}{\text{Time}_i \times \sum_{j=1}^n \left(\frac{1}{\text{Time}_j} \right)}$$

Princípios para Melhora de Desempenho

- Faça os casos mais comuns melhores!
 - Quanto e onde gasta recursos para melhorias?
- Comparação de melhoria: Lei de Amdahl
 - Define quando é ganho com determinada melhoria

$$\text{Speedup} = \frac{\text{Performance for entire task using the enhancement when possible}}{\text{Performance for entire task without using the enhancement}}$$

Alternatively,

$$\text{Speedup} = \frac{\text{Execution time for entire task without using the enhancement}}{\text{Execution time for entire task using the enhancement when possible}}$$

Lei de Amdahl

- Outra forma, em relação à fração melhorada:

$$\text{Execution time}_{\text{new}} = \text{Execution time}_{\text{old}} \times \left((1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right)$$

The overall speedup is the ratio of the execution times:

$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

Exemplo

- Suponha que estamos considerando uma melhoria que roda 10 vezes mais rápido do que a solução original, mas que só é empregada 40% do tempo de execução total. Qual o speedup geral devido à melhoria.

Solução

$$\text{Fraction}_{\text{enhanced}} = 0.4$$

$$\text{Speedup}_{\text{enhanced}} = 10$$

$$\text{Speedup}_{\text{overall}} = \frac{1}{0.6 + \frac{0.4}{10}} = \frac{1}{0.64} \approx 1.56$$

Outros Métricas de Desempenho

- MIPS (Milhões de Instruções por segundo)
 - Muito comum para marketing e fácil de entender
 - Depende das instruções, dificultando comparação de diferentes máquinas
 - Varia de programa a programa
 - Pode variar de forma inversa ao desempenho se escolher um conjunto de interesse de instruções

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

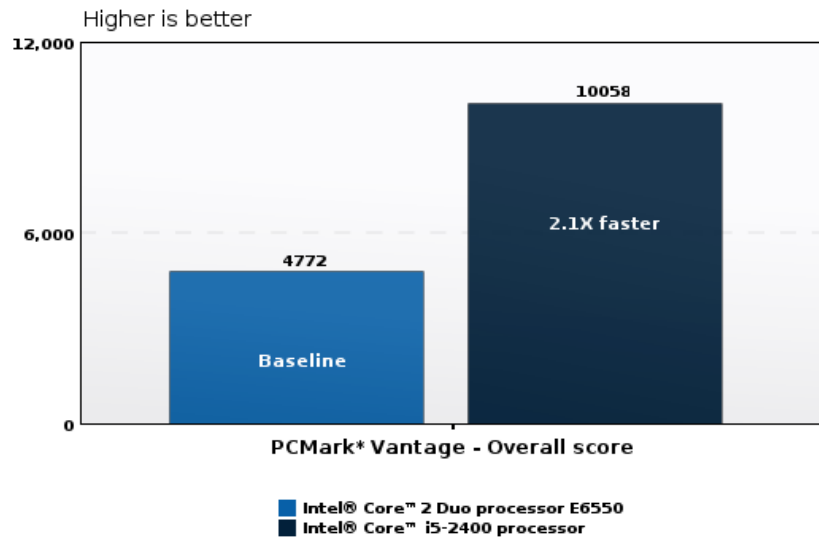
Outros Métricas de Desempenho

- MFLOPs (Milhões de Operações de Ponto Flutuante por segundo)
 - Também muito popular para divulgação
 - Sofre dos problemas anteriores. Ex. Compiladores quase nunca usam operações de ponto flutuante

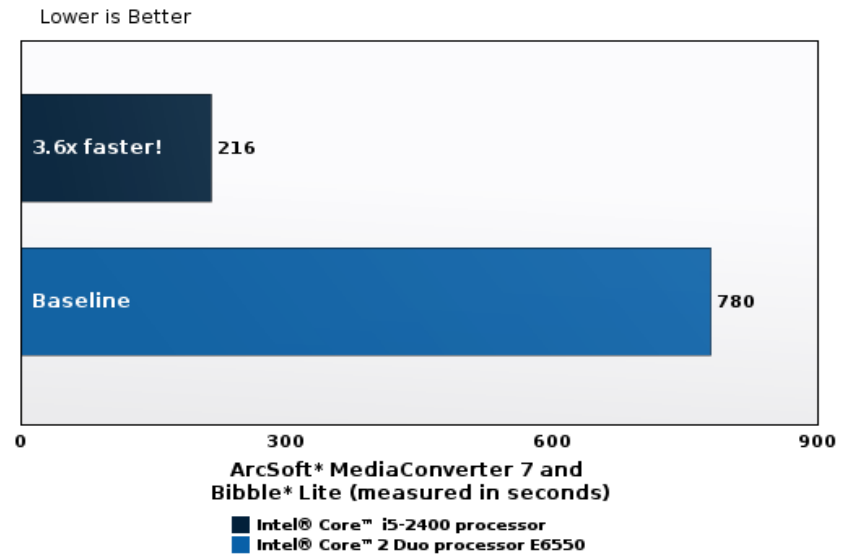
$$\text{MFLOPS} = \frac{\text{Number of floating-point operations in a program}}{\text{Execution time in seconds} \times 10^6}$$

Exemplo

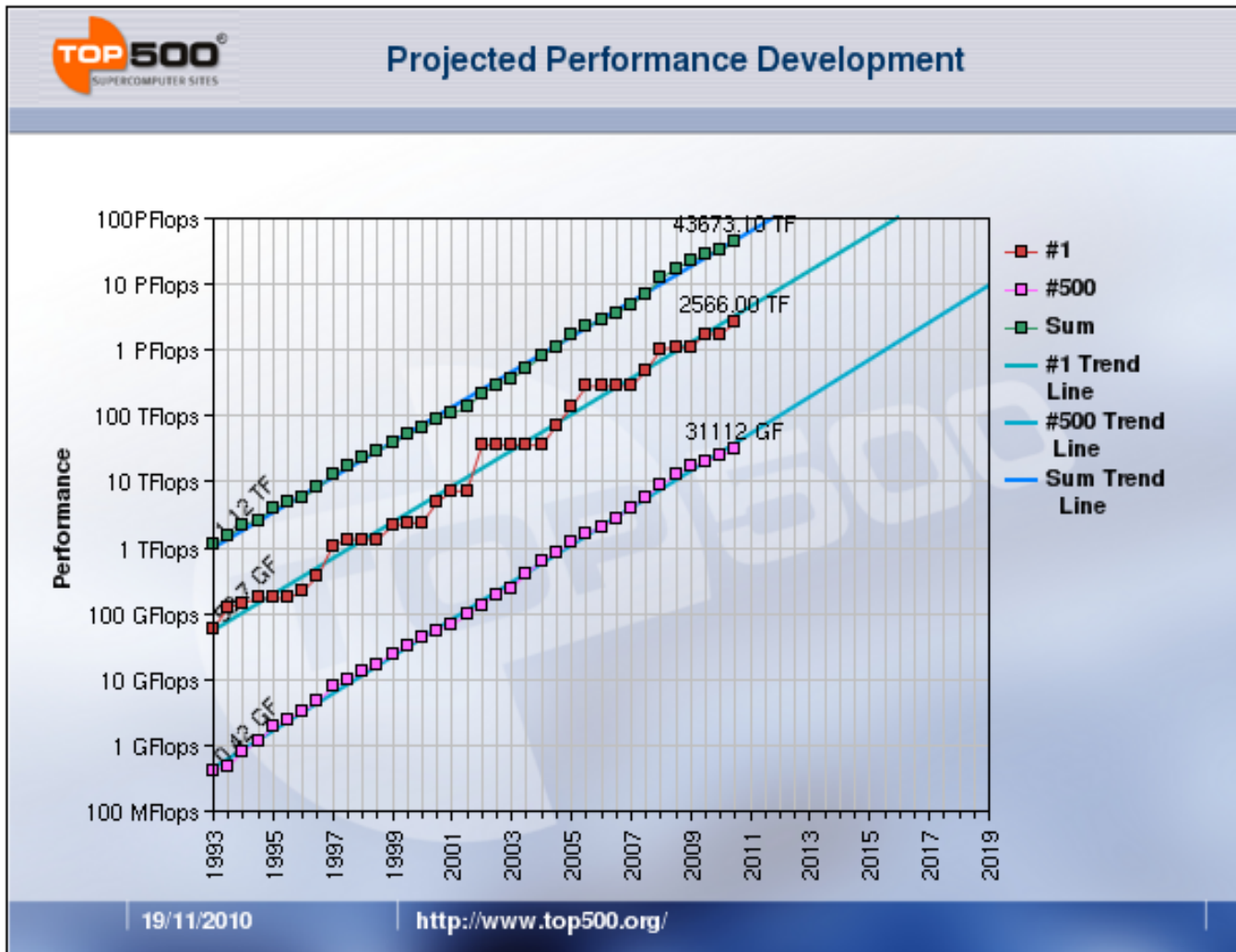
Basic applications



HD video to iPod* while archiving vacation photos



Exemplo



Projected Performance Development

Exemplo Top 10 Supercomputers

November 2010 list

[\[edit\]](#)

The following table gives the Top 10 positions of the 36th TOP500 List released on November 14, 2010 during SC10 in [New Orleans, LA](#).^[5]

Rank ⌵	Rmax Rpeak (Tflops) ⌵	Name ⌵	Computer Processor cores ⌵	Vendor ⌵	Site Country, Year ⌵	Operating System ⌵
1	2566.00 4701.00	<i>Tianhe-1A</i>	NUDT YH Cluster 14,336x6 Xeon + 7168x14 Fermi, Arch (Proprietary) ^[6]	NUDT	National Supercomputing Center in Tianjin  China, 2010	Linux
2	1759.00 2331.00	<i>Jaguar</i>	Cray XT5 224,162 Opteron	Cray	Oak Ridge National Laboratory  United States, 2009	Linux (CLE)
3	1271.00 2984.30	<i>Nebulae</i>	Dawning TC3600 Blade 55,680 Xeon + 64,960 Tesla, InfiniBand	Dawning	National Supercomputing Center in Shenzhen (NSCS)  China, 2010	Linux
4	1192.00 2287.63	<i>TSUBAME 2.0</i>	HP Cluster Platform 3000SL 73,278 Xeon, Fermi	NEC/HP	GSIC Center, Tokyo Institute of Technology  Japan, 2010	Linux (SLES 11)
5	1054.00 1288.63	<i>Hopper</i>	Cray XE6 153,408 Opteron	Cray	DOE/SC/LBNL/NERSC  United States, 2010	Linux (CLE)
6	1050.00 1254.55	<i>Tera 100</i>	Bull Bullx 138,368 Xeon, InfiniBand	Bull SA	Commissariat à l'énergie atomique (CEA)  France, 2010	Linux
7	1042.00 1375.78	<i>Roadrunner</i>	BladeCenter QS22/LS21 122,400 Cell/Opteron	IBM	Los Alamos National Laboratory  United States, 2009	Linux
8	831.70 1028.85	<i>Kraken</i>	Cray XT5 98,928 Opteron	Cray	National Institute for Computational Sciences  United States, 2009	Linux (CLE)
9	825.50 1002.70	<i>JUGENE</i>	Blue Gene/P Solution 294,912 Power	IBM	Forschungszentrum Jülich  Germany, 2009	Linux (SLES 11)
10	816.60 1028.66	<i>Cielo</i>	Cray XE6 107,152 Opteron	Cray	DOE/NNNSA/LANL/SNL  United States, 2010	Linux (CLE)

Supercomputador brasileiro

Supercomputador brasileiro está entre os 30 mais rápidos do mundo



Lembra daquele [supercomputador de 50 milhões de reais](#) que o Inpe comprou para aguçar a previsão do tempo? Pois bem, de lá para cá as coisas só melhoraram para o novo grande cérebro brasileiro: ele foi batizado de Tupã, foi chamado pela [revista Nature](#) de "o [futuro](#) da previsão de tempo global" e, para completar, foi considerado hoje o 29º computador mais rápido do mundo, colocando pela primeira vez uma máquina do hemisfério sul no top30 dos gigantes da supercomputação.

Como [nós já falamos por aqui](#), a nova aquisição do Inpe, um XT6, da Cray, está instalada em Cachoeira Paulista, a 202km de São Paulo, e veio direto de Winsconsin. Depois de se adaptar ao novo clima brasileiro e [aprender](#) algumas palavras em

português, a máquina ganhou o nome de Tupã – referência ao deus indígena que surgia em forma de trovão – e em pouco menos de dois meses já está entre os trinta computadores mais rápidos do mundo, segundo o TOP500, [especializado](#) na avaliação. Era de se esperar, já que o supercomputador é capaz de atingir nada menos do que 258 teraflops. Nada mal, hein?

E se estar entre os 30 melhores computadores do mundo ainda é pouco, o Tupã foi [citado na revista Nature](#), principal publicação de ciência nos dias de hoje, como uma das soluções computacionais para controle e monitoramento climático. A ideia dos pesquisadores brasileiros é entender cada vez mais e [criar](#) novos modelos sobre as mudanças climáticas no Brasil, principalmente em regiões de maior impacto, como na Amazônia. Ponto para o Inpe e para o Brasil, que agora começa a entrar de vez [nesse infográfico](#). [TOP500 via G1]

Como avançar no desempenho

- Esforços, até então, concentram-se em aumentar frequência de clock e construir arquiteturas que favoreçam CPI.
- Mas estamos chegando no limite do silício
- Clock alto aumenta aquecimento e consumo de energia
- Mais recente, o esforço passou a considerar mais CPUs ou núcleos, sendo o assunto relevante de P&D.
- Futuro?
 - Spintrônica: Novos materiais com novas propriedades e com menor dissipação
 - Bio-computação: Ex. Armazenamento em bactérias
 - Nano-Computação: Nano-materiais e nano-máquinas
 - ???