

Universidade de Brasília

Departamento de Ciência da Computação

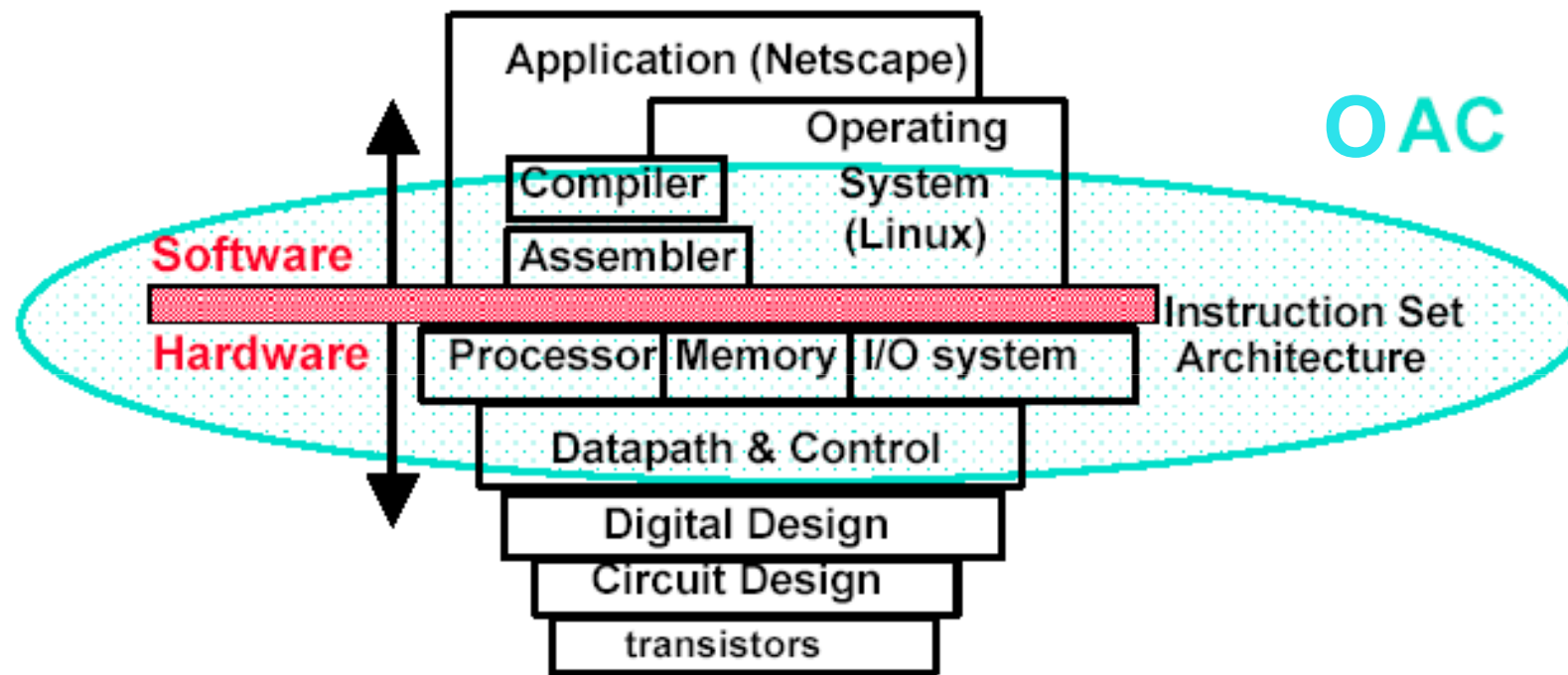
Curso de Licenciatura em Computação

Disciplina: Arquitetura de Processadores Digitais

# Introdução



# O que é Arquitetura e Organização de Computadores?



**Arquitetura** do conjunto de instruções  
+  
**Organização** da máquina

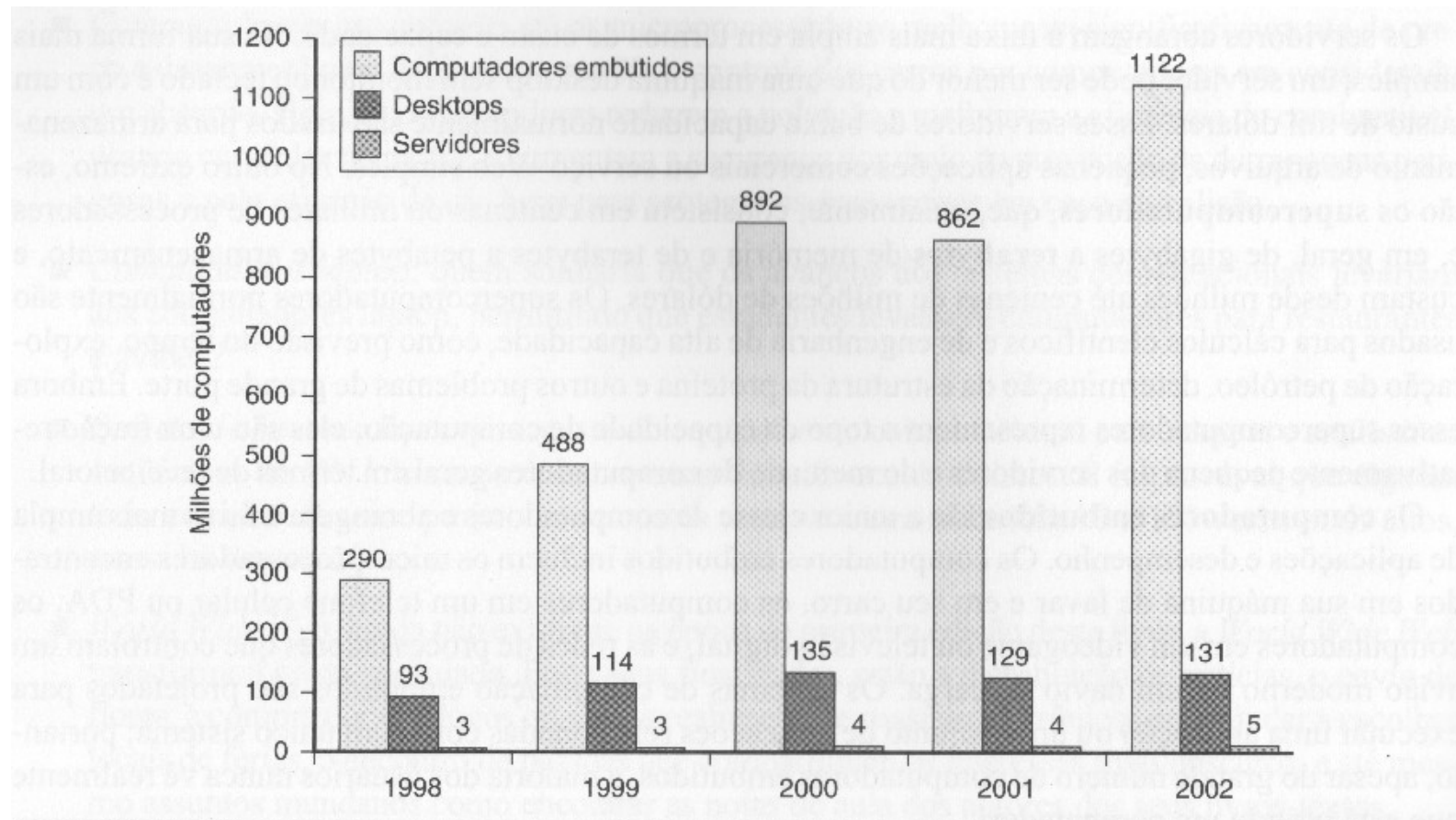


# Introdução

- O objetivo deste curso é mostrar o *link* entre o *Software* e o *Hardware*
- Mas o que entendemos como “computador”?
  - **Diferentes tipos:** desktop, servidores, dispositivos embutidos
  - **Diferentes usos:** automóveis, escritório, design gráfico, finanças, games, games, games etc.
  - **Diferentes fabricantes:** Intel, AMD, Apple, HP, Sun, MIPS, ARM,...
  - **Diferentes tecnologias subjacentes** e diferentes custos!

Analogia: Pense em um curso acerca de “veículos automotivos”

- Muitas semelhanças de um veículo para outro  
por exemplo, todos possuem rodas e volante (antigamente)
  - Grandes diferenças de um veículo para outro  
por exemplo, combustível: gasolina, álcool, GNV, diesel, flex
  - Diferentes fabricantes, diferentes marcas, diferentes usos.
- Melhor maneira de aprender:
    - Concentrar em um exemplo específico e aprender como ele funciona
    - Abordar princípios gerais e perspectivas históricas



**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.

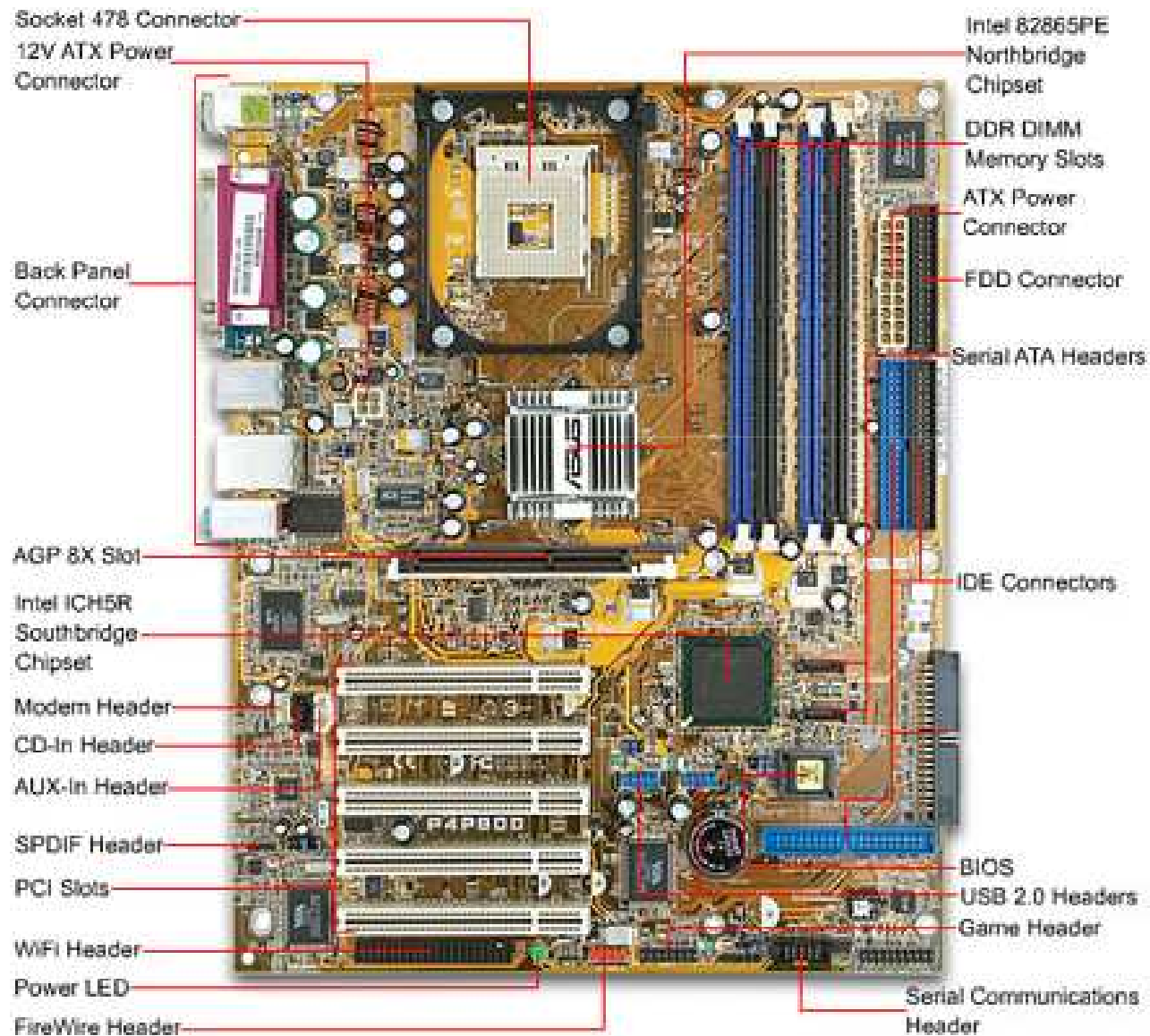


# Por que aprender esse assunto?

- Entender os modernos processadores
- Desenvolver softwares eficientes (super-desempenho)
- Tomar decisões em relação a uma compra ou prestar consultoria (\$\$\$)
- Tanto o hardware quanto o software afetam o desempenho:
  - O algoritmo determina o número de instruções
  - Linguagem/compilador/arquitetura determinam as instruções da máquina (Capítulos 2 e 3 do livro texto)
  - Processador/memória determinam a velocidade com que as instruções são executadas (Capítulos 5, 6 e 7)
- Avaliando e entendendo o desempenho (Capítulo 4)



# Placa mãe para Pentium IV

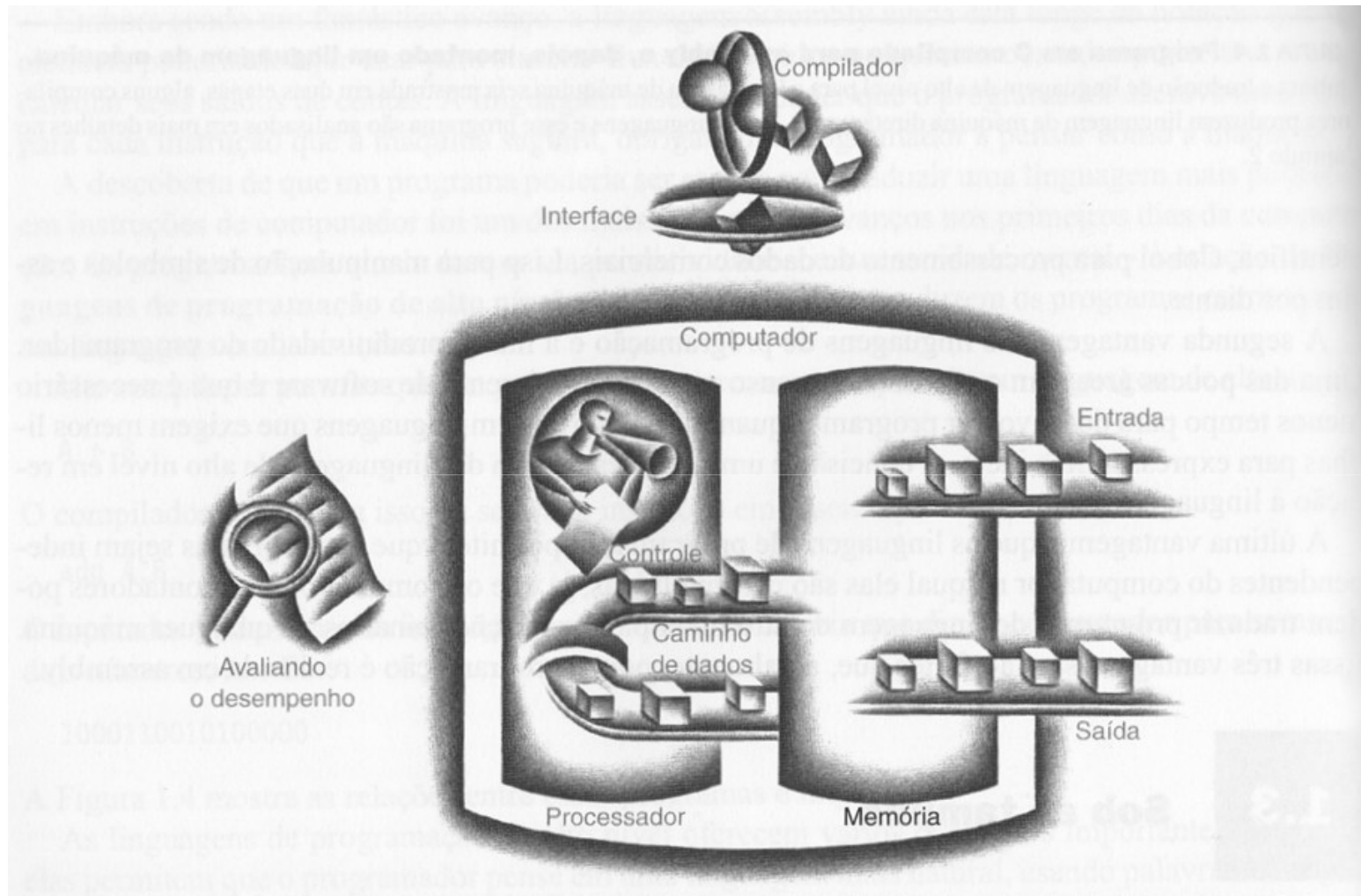




# O que é um computador?

- Componentes:
  - Processador
  - Dispositivos de entrada (mouse, teclado,...)
  - Dispositivos de saída (monitor, impressora,...)
  - Dispositivos de memória (DRAM, SRAM, HD, CD, DVD, pendrive...)
  
- Nosso foco principal: o processador (caminho de dados e controle)
  - Implementado usando milhões de transistores
  - Impossível de entender olhando para os transistores





**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.



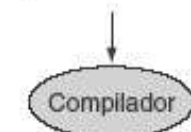


# Abstração

- Maior aprofundamento revela mais informações
- Uma abstração omite detalhes desnecessários e ajuda a entender sistemas complexos
- É a descrição simplificada ou seletiva de um sistema

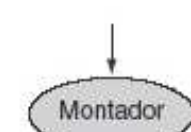
Programa em  
linguagem de  
alto nível  
(em C)

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



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
```



Programa  
binário em  
linguagem de  
máquina  
(para MIPS)

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



# Como os computadores funcionam?

- É preciso entender abstrações como:
  - Software de aplicações
  - Software de sistemas
  - Linguagem assembly
  - Linguagem de máquina
  - Aspectos de arquitetura, como caches, memória virtual, canalização
  - Lógica sequencial, máquinas de estado finito
  - Lógica combinatória, circuitos aritméticos
  - Lógica booleana, 1s e 0s
  - Transistores usados para construir portas lógicas (CMOS/TTL)
  - Semicondutores usados para construir transistores
  - Propriedades dos átomos, elétrons e dinâmica quântica

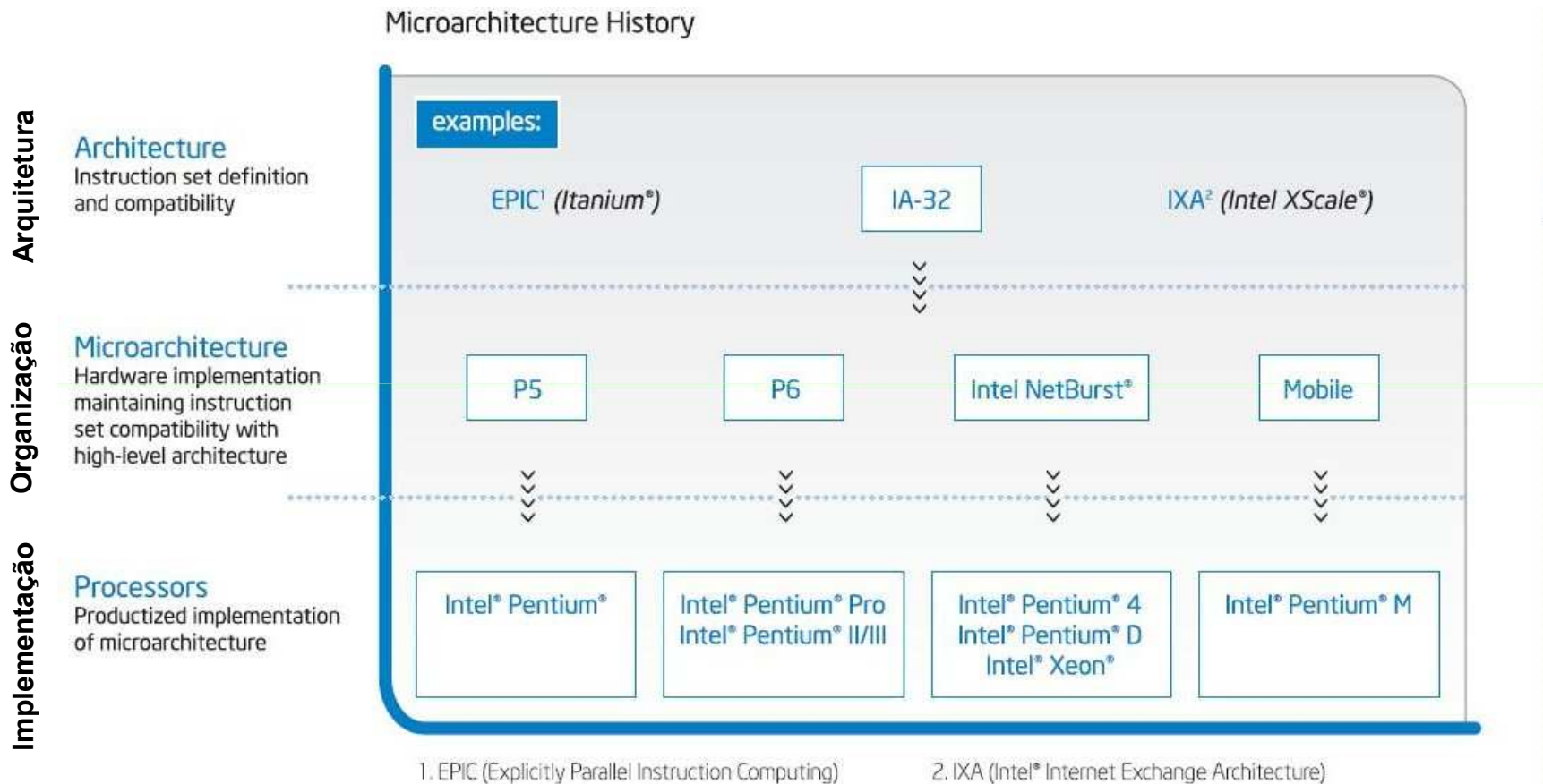


# Arquitetura do conjunto de instruções

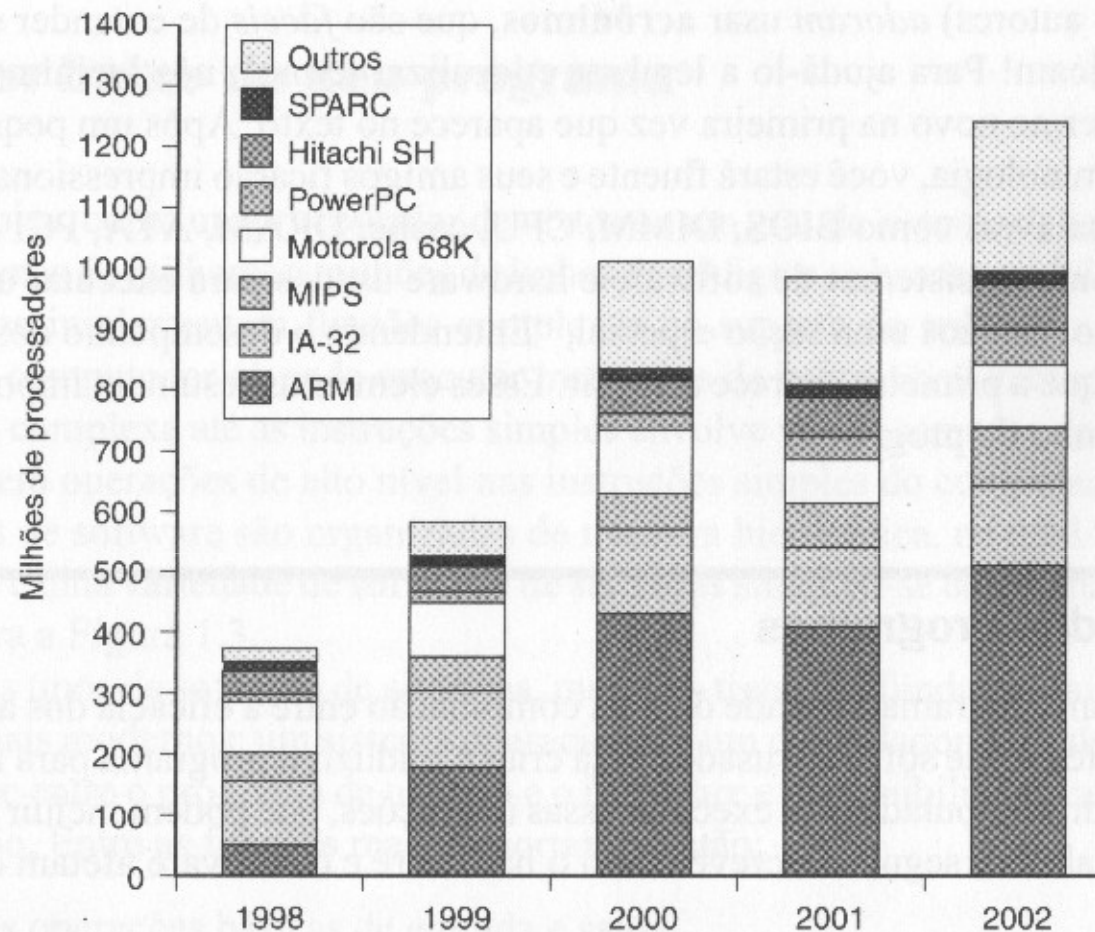
- Uma abstração muito importante
  - interface entre o hardware e o software de baixo nível
  - padroniza instruções, padrões de bits de linguagem de máquina, etc.
  - vantagem: *diferentes implementações da mesma arquitetura*
  - desvantagem: *algumas vezes impede o uso de inovações*
  
- Arquiteturas de conjunto de instruções modernas: IA-32 (x86), PowerPC, SGI, MIPS, SUN SPARC, ARM, HP PA-RISC e outras



# Arquitetura x Organização x Implementação



**Figure 1.** This diagram shows the difference between processor architecture and microarchitecture. **Processor Architecture** refers to the instruction set, registers, and memory data-resident data structures that are public to a programmer. Processor architecture maintains instruction set compatibility so processors will run code written for processor generations, past, present, and future. **Microarchitecture** refers to the implementation of processor architecture in silicon. Within a family of processors, the microarchitecture is often enhanced over time to deliver



**FIGURA 1.2 Vendas de microprocessadores entre 1998 e 2002 por arquitetura combinando todos os usos.** A categoria “outros” se refere aos processadores específicos à aplicação ou às arquiteturas personalizadas. No caso do ARM, aproximadamente 80% das vendas são para telefones celulares, sendo que um núcleo do ARM é usado em conjunto com lógica específica da aplicação em um chip.





## No princípio (pré-computadores)

No início do século 17 iniciou-se a automação de tarefas com máquinas, com resultados utilizados até hoje

### **Por exemplo: Máquina de Pascal (1642-1644)**



Fazia soma e subtração em decimal mecanicamente.

Mais tarde, no mesmo século foram adicionadas multiplicação e divisão à máquina.

Cartões perfurados codificados com instruções para a máquina vieram da indústria de tecelagem.



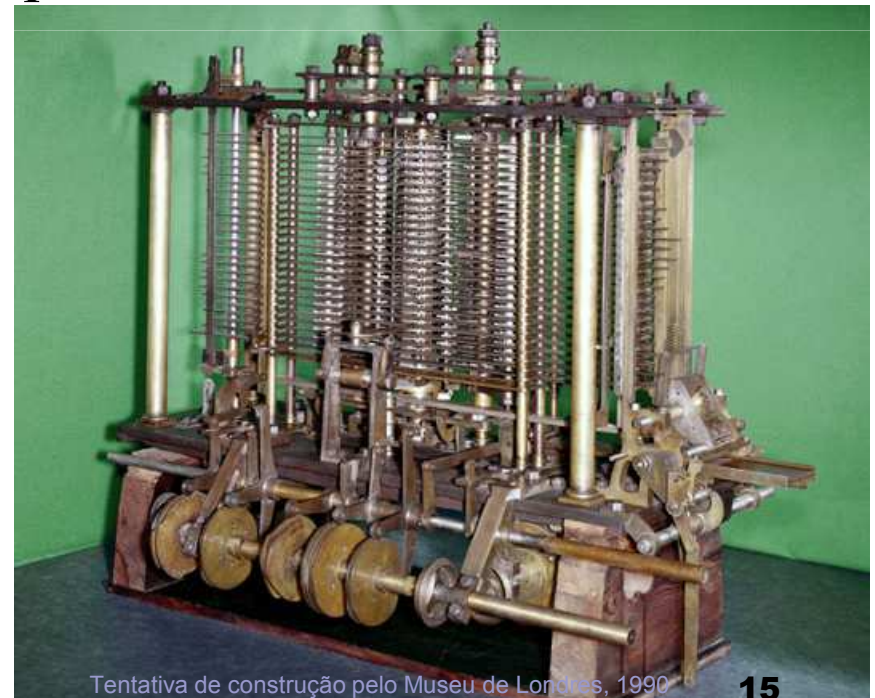
# *A Calculadora de Babbage (1837~†1871)*

Um dos grandes sucessos e fracassos no caminho do desenvolvimento de computadores.

Uma calculadora mecânica automática que nunca funcionou. A *Analytical Engine* foi a 3ª máquina de calcular projetada por Babbage e a que mais contribuiu para o desenvolvimento da computação.

Charles Babbage não conseguiu solucionar problemas mecânicos tentando construir sua máquina.

A máquina de Babbage ficou num sonho. Era muito complexa para os profissionais da época.



Tentativa de construção pelo Museu de Londres, 1990



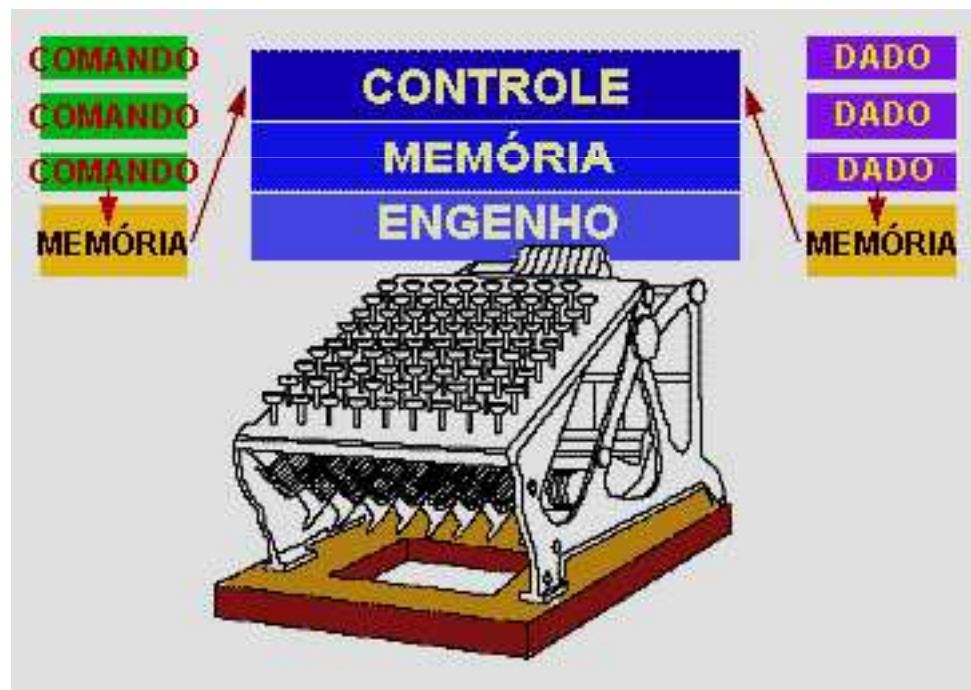


# Calculadora de Babbage

O sucesso de sua máquina, utilizado até os dias de hoje, foi a idéia que Babbage teve de como ela deveria processar as informações.

Babbage dividiu sua máquina em três partes:

- armazenamento
- engenho
- controle





# Calculadora de Babbage

## Engenho e Controle

O engenho é a parte da máquina que executa as instruções sobre os dados.

O controle supervisiona o movimento dos dados para dentro e para fora dos dois armazenamentos, para dentro e para fora do engenho.



# Calculadora de Babbage

## Armazenamento

O armazenamento é o que hoje chamamos de memória.

Babbage propôs dois tipos de armazenamento para sua máquina: um para os números (dados) a serem trabalhados e outro (armazenamento de controle) para as instruções que diziam o que fazer com os números.

O projeto de Babbage utilizaria cartões perfurados com formatos diferentes para dados e controle.

Cada padrão de furos no cartão de dados representaria um número enquanto cada furo no cartão de controle representaria uma possível ação que o engenho poderia tomar.





# Calculadora de Babbage

O projeto de Babbage teria um conjunto de instruções bem simples, limitado a operações como:

“pegar um número do cartão de dados em curso”

“somar 1 ao número em curso”

“subtrair 1 do número em curso”

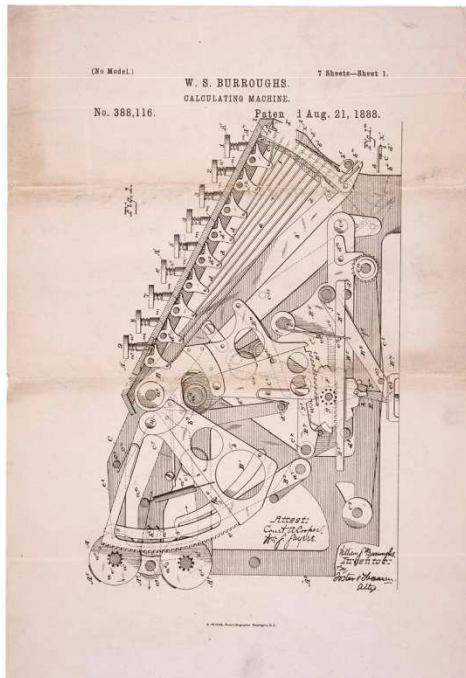
“ir para o próximo cartão de dados” ...



## Mas ...

Por outro lado, o primeiro calculador mecânico - a primeira máquina comercial de somar - foi produzida em 1885 por William Seward Burroughs.

Apesar da máquina de Burroughs utilizar um projeto diferente, a idéia de Babbage vingou ...



1890



# Indo adiante

A idéia de Babbage sobre a estruturação de informação dentro de um dispositivo foi utilizada, finalmente com algum sucesso, no início do século 20.

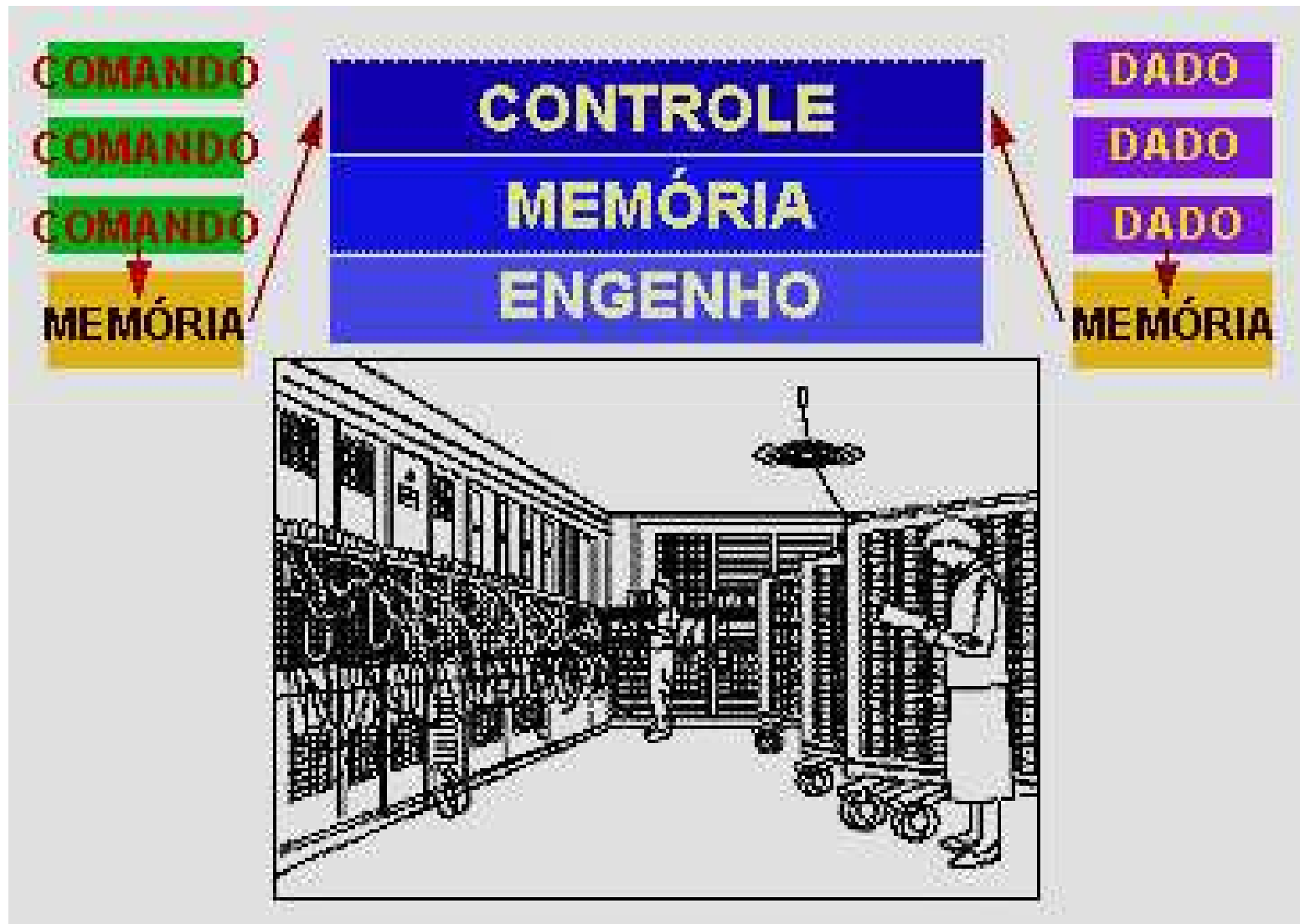
Algumas calculadoras foram construídas a partir de componentes de chaveamento de equipamentos telefônicos: os relés.

Estas máquinas eram capazes de executar cálculos em milissegundos.

Contudo, foi em 1940 com o impulso da guerra e o desenvolvimento de novas tecnologias que foi desenvolvido o que hoje chamamos de computador.



# Indo adiante



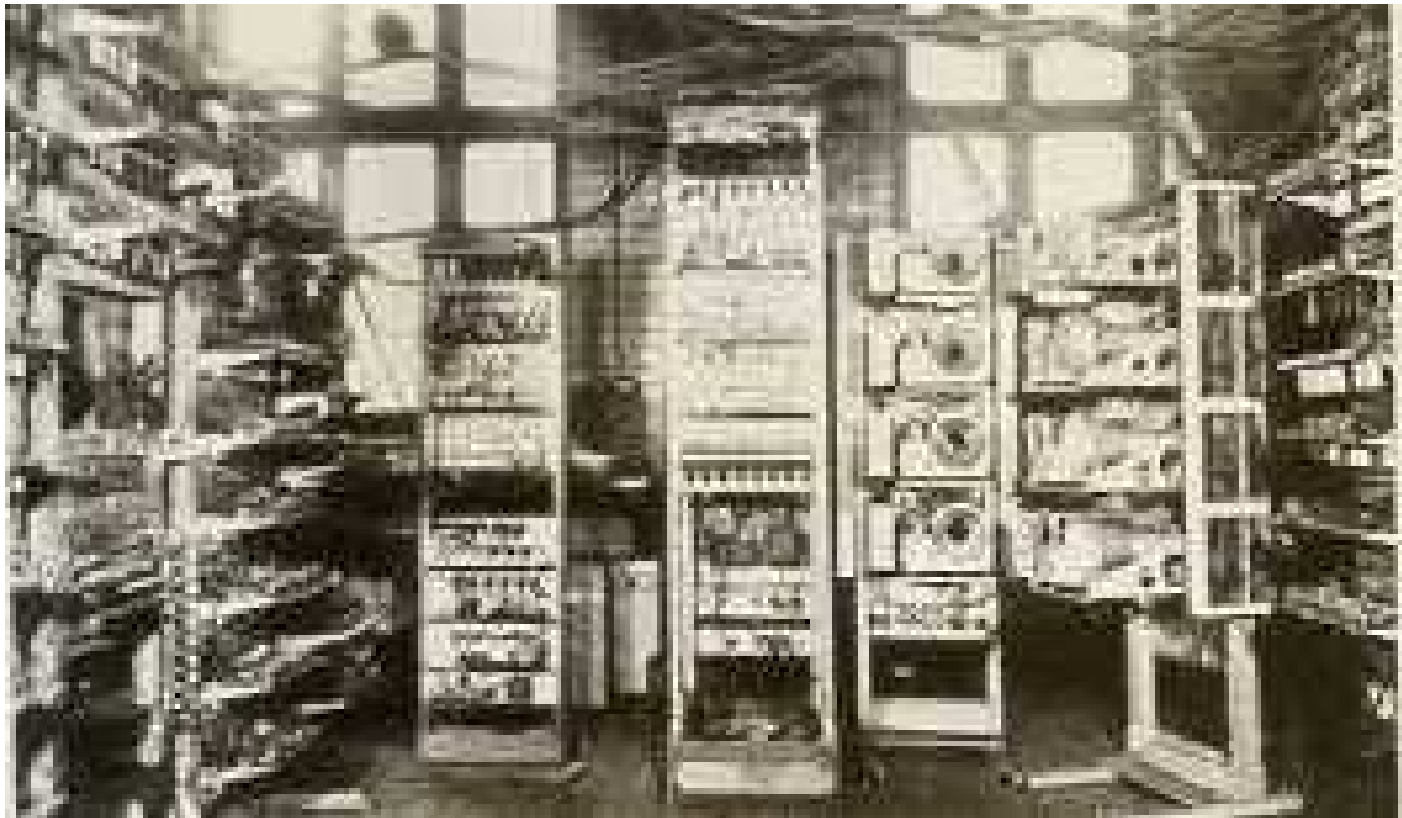




# MARK I

A série de computadores Mark foi desenvolvida na Universidade de Harvard durante os anos 40.

O primeiro, Mark I, entrou em operação em 1944 e foi utilizado até 1959.





# O MARK I

Armazenava e contava números mecanicamente, utilizando 3.000 discos de armazenamento decimais, 1.400 chaves circulares (*rotary dial switches*) e 500 milhas de fios. Transmítia e lia os dados eletricamente.

Era programado por cartões perfurados, pesava 5 toneladas e realizava uma operação de multiplicação em 6 segundos.

Os dados eram armazenados em local diferente das instruções (programa).

As instruções também eram armazenadas num formato diferente dos dados.



# A arquitetura Harvard

A técnica de armazenamento de dados e instruções separadamente tornou-se conhecida como Arquitetura Harvard.

Apesar desta arquitetura ter sido formalmente descartada nos sistemas modernos de computação, foi ressuscitada em muitos processadores de sinais digitais atuais (DSPs, Processadores Gráficos, PS2, PS3) e de forma implícita em quase todos os processadores (*Pipeline*).



# ENIAC – 1º Computador Eletrônico

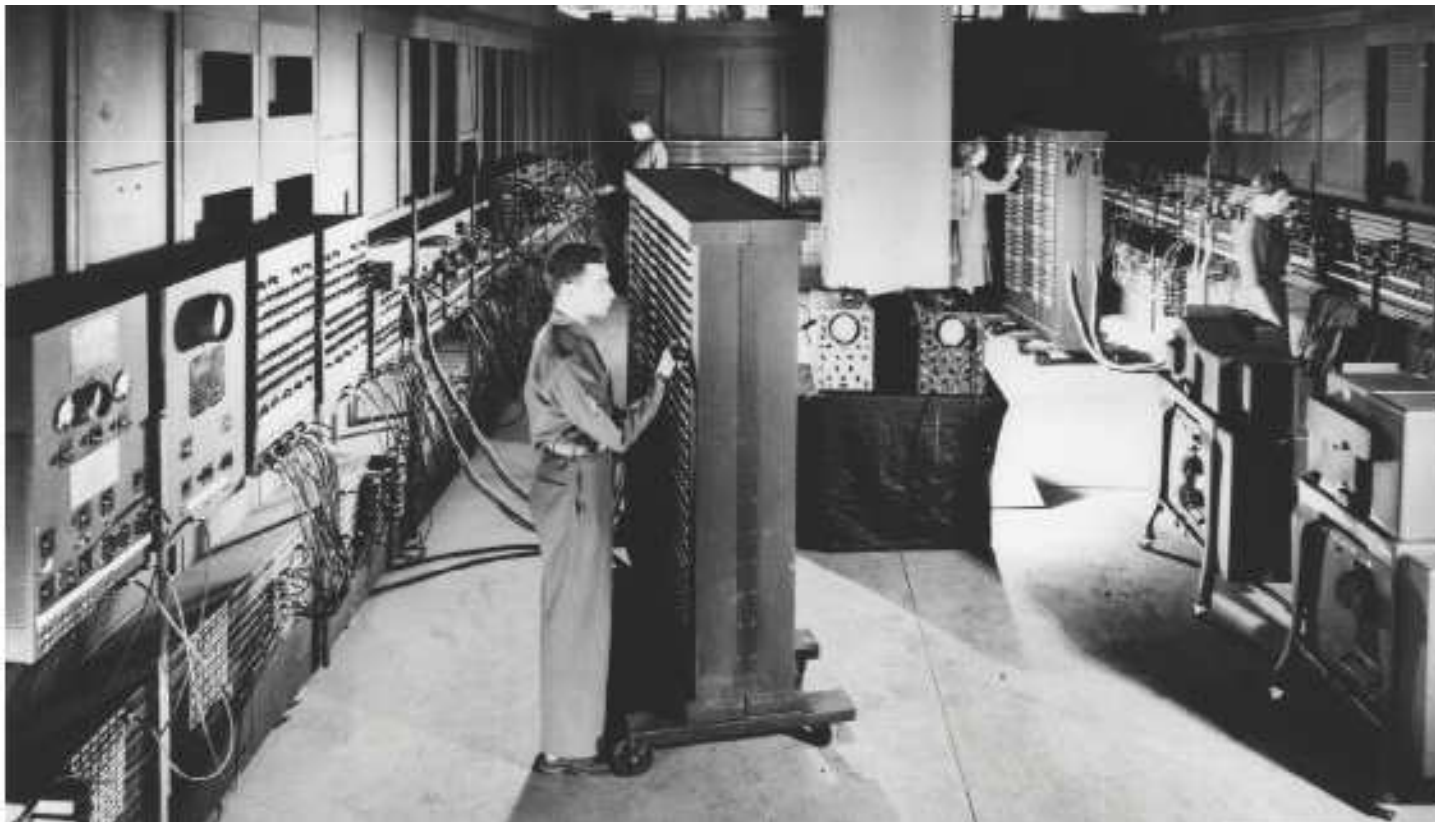
No início dos anos 40 este computador foi desenvolvido na Universidade da Pennsylvania, utilizando 18.000 válvulas e 1.500 relés para movimentar a informação na máquina, chamado de *Electronic Numerical Integrator And Calculator*.

Podia fazer 5.000 adições por segundo ou 357 multiplicações por segundo.

Era programado por cartões perfurados e podia ler dois números por segundo.



- O ENIAC, construído na Segunda Guerra Mundial, foi o primeiro computador de finalidade geral.
  - Usado para calcular tabelas de disparo de artilharia.
  - 24 metros de comprimento por 2,5 metros de altura e dezenas de centímetros de profundidade.
  - Cada um dos 20 registradores de 10 dígitos tinha 60 centímetros de comprimento.
  - Usava 18.000 válvulas.





## Ingressando nos anos 60

Os computadores dos anos 40 funcionavam. Em frações de segundos podiam executar cálculos que levariam semanas para serem efetuados por equipes utilizando máquinas de somar.

Também eram máquinas imensas, necessitando de salas inteiras para si próprias e seus sistemas de ar-condicionado.

Sua programação era lenta e complicada.

E a portabilidade dos programas ....



# Ingressando nos anos 60

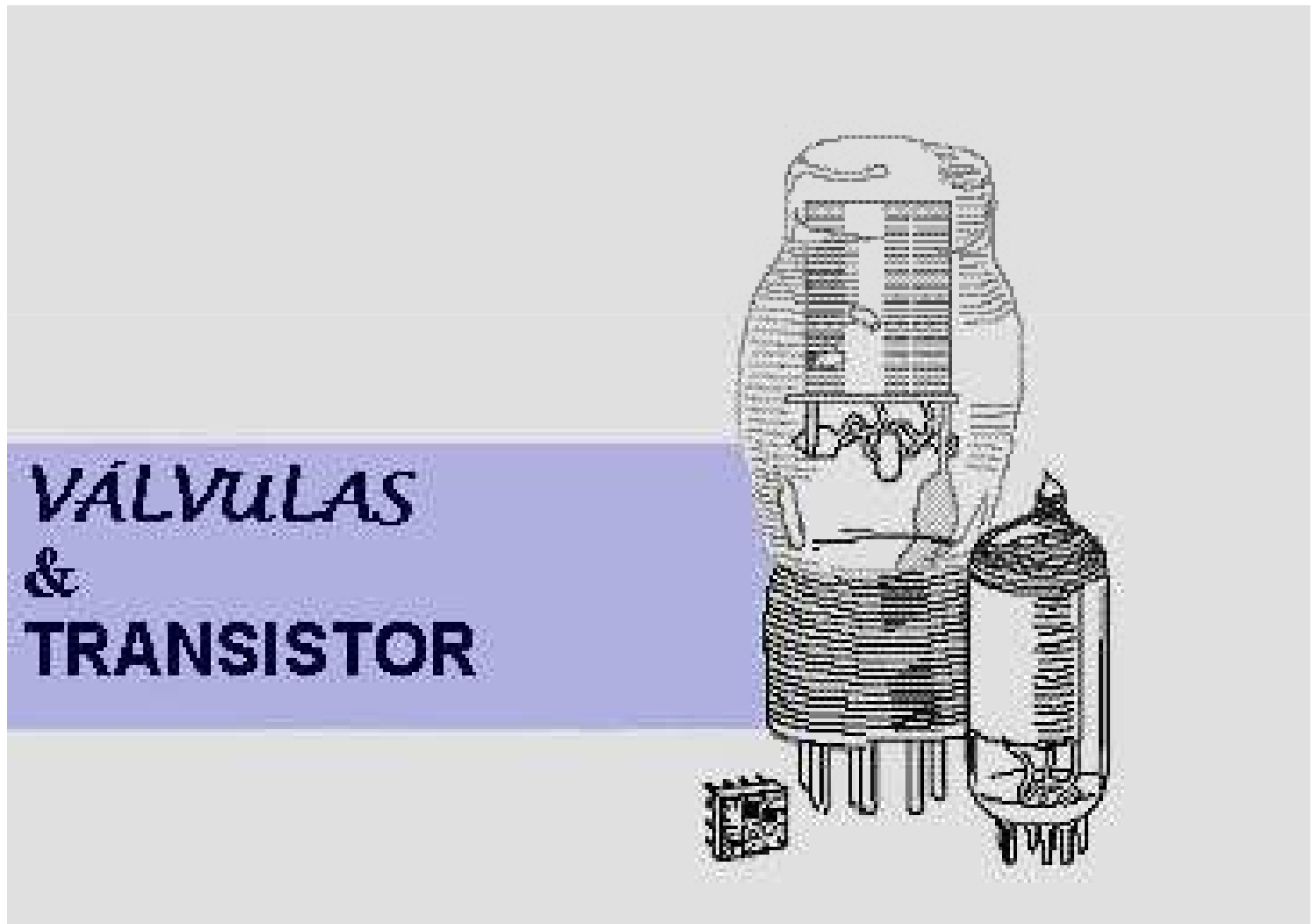
Dos anos 40 aos anos 60, os projetistas de computadores trabalharam na mudança dos computadores.

As máquinas deveriam ser menores, com programas intercambiáveis e fáceis de escrever, maior quantidade de memória, maior “velocidade”, maior capacidade de processamento, ...





# Mudanças no hardware





# Mudanças no hardware

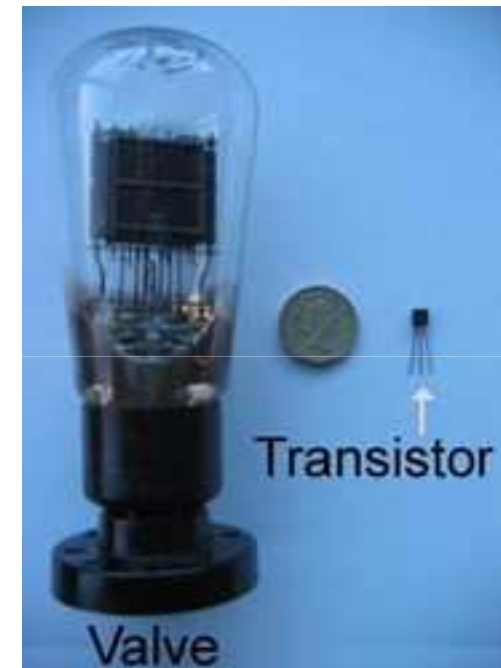
Alguns problemas dos primeiros computadores tinham a ver com o tamanho de seus componentes, que os tornavam grandes, lentos, quentes e pesados.



# Mudanças no hardware

Para o final dos anos 50 foi introduzido o uso do transistor. 1/200 do tamanho da válvula.

Geravam muito menos calor que as válvulas e eram muito, muito mais rápidos (as distâncias entre eles eram muito menores): podiam efetuar até 100.000 instruções por segundo.





Os programas lógicos eram introduzidos nos primeiros computadores durante a operação: rolos de papel eram perfurados para codificar 0s e 1s para que a máquina pudesse ler, ou bancos de chaves eram posicionados em longos padrões de 0s e 1s.

Os dados necessários se encontravam dentro da máquina.

Os contadores em anel do ENIAC eram representações físicas dos números que os programas deveriam manipular.



# A arquitetura von Neumann

Em meados dos anos 40, John von Neumann mostrou que as instruções poderiam ser representadas na mesma linguagem utilizada para os dados.

Instruções e dados poderiam, então, ser armazenados juntos dentro do computador.

O primeiro computador com esta Arquitetura von Neumann foi o *Electronic Discrete Variable Automatic Computer*, o EDVAC que tornou-se operacional em 1952.



# A arquitetura von Neumann

A arquitetura von Neumann tornou-se padrão para os sistemas computacionais modernos.

A maioria dos computadores desde então possui alguma versão desta arquitetura.

Num típico sistema von Neumann, instruções e dados estão inseridos juntos na mesma memória. Muitas vezes com os dados seguindo imediatamente as instruções.

Instruções são apenas números, não podendo ser distinguidas dos dados.



# A arquitetura von Neumann

**Combinar instruções e dados na mesma memória traz algumas vantagens:**

- **Uso eficiente da memória.**  
Um único bloco (grande) de memória em vez de dois menores.
- **Instruções são facilmente manipuláveis (como os dados).**  
Como instruções e dados estão armazenados juntos, movimentar blocos de instruções (programas) é mais simples, ou ...
- **Facilidade em carregar programas na memória.**  
Basta ler as instruções do disco ou outra memória secundária e executá-las.





# A arquitetura von Neumann

**Combinar instruções e dados na mesma memória traz algumas desvantagens:**

- **Dados podem sobrescrever instruções.**

Sem alguma precaução especial do hardware (proteção de memória), uma escrita incorreta na memória pode sobrescrever algumas instruções. Como os sistemas von Neumann não fazem distinção entre dados e instruções, a máquina pode tentar executar dados como instruções, com resultados indesejados.



# A arquitetura von Neumann

**Combinar instruções e dados na mesma memória traz algumas desvantagens:**

- **Largura de banda limitada.**

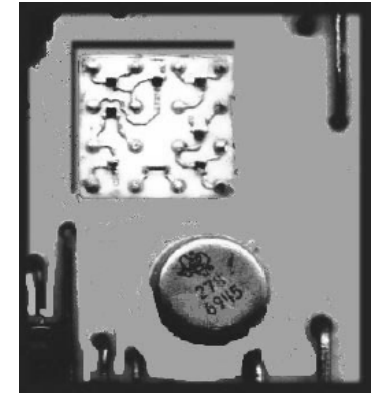
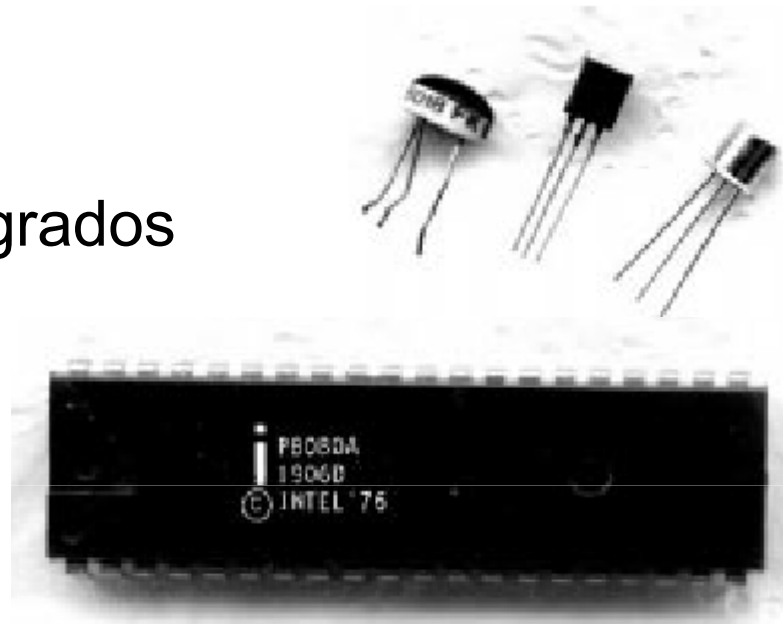
Armazenar instruções e dados juntos significa que ambos percorrem o mesmo caminho até o processador.

Este é o gargalo da arquitetura von Neumann. O processador deve executar um grande número de instruções por segundo e ler uma grande quantidade de dados ao mesmo tempo.



# Revolução da Eletrônica

- Válvulas
- Transistores
- Circuitos Integrados
- LSI
- VLSI



**Conceitos de organização, paralelismo e hierarquia de memória são os mesmos de mainframes das décadas de 60 e 70**

**• diferença está na tecnologia**

- **1970: poucos milhares de transistores num chip**
- **2005: dezenas a poucas centenas de milhões de transistores num chip**
- **2010: mais de 1 bilhão de transistores num chip**



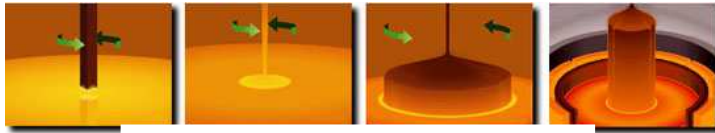
- **Cronologia: Microprocessadores integrados em um chip da Intel**

- 1971: Intel 4004 - 4 bits, 2.300 transistores, 740kHz
- 1972: Intel 8008 - 8 bits, 3.500, 500kHz
- 1976, Intel 8085 – 8 bits, 6.500, 5MHz
- 1978: Intel 8086 - 16 bits, 29.000, 10MHz
- 1982: Intel 80186 – 16bits
- 1982: Intel 80286 – 16 bits, 134.000, 25MHz
- 1985: Intel 80386 - 32 bits, 275.000, 33MHz (cache externa)
- 1989: Intel 40486 – 32 bits, 1.200.000, 50MHz (L1 cache)
- 1993: Pentium – 32 bits, 3.100.000, até 233MHz
- 1995: Pentium Pro e MMX – 32 bits, 4.500.000, 200MHz (L2 cache)
- 1997: Pentium II– 32 bits, 7.5000.000, 450MHz
- 1999: Pentium III – 32 bits, 28.000.000, 1.13GHz
- 2000: Pentium IV – 32 bits, 42.000.000, até 3.4GHz
- 2001: Intel Itanium – 64 bits (AMD antes)
- 2003: Pentium-M (Centrino, Celeron M) – 32 bits, 77.000.000, 2.1GHz
- 2005: Pentium-D (Extreme HT) – 64bits, 230.000.000, 3.4GHz
- 2006: Core (Duo,Solo) – 32bits, 151.000.000, 1.66GHz
- 2006: Core2 (Duo) - 64 bits, 291.000.000, 3GHz
- 2007: Core2 (Quad) – 64 bits 830.000.000 3GHz (só L2)
- 2008: Core i7 (Quad) – 64 bits 731.000.000 3GHz (L3 cache)

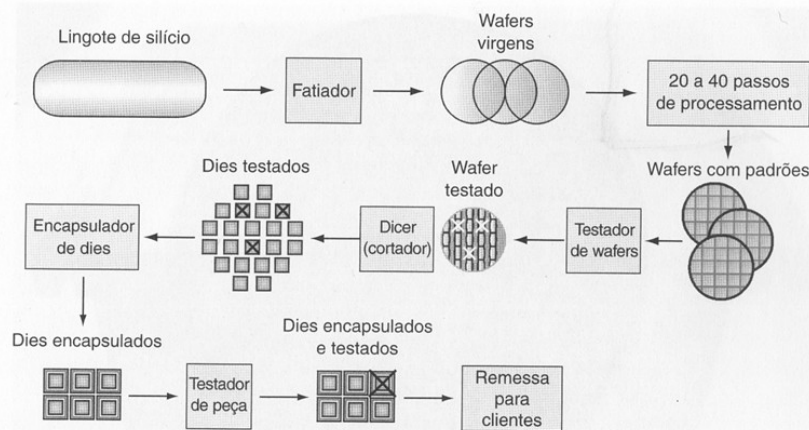
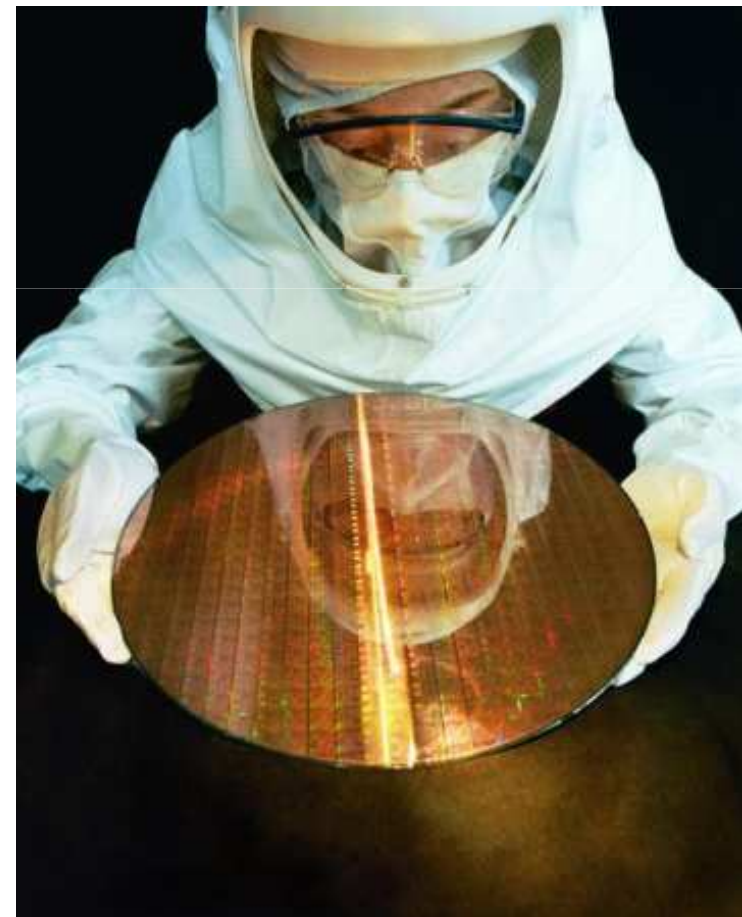
[http://www.intel.com/portugues/products/processor/core2quad/?cid=lar:ggl|c2d\\_br\\_quad|k8B85|s](http://www.intel.com/portugues/products/processor/core2quad/?cid=lar:ggl|c2d_br_quad|k8B85|s)



# Fabricação de um chip

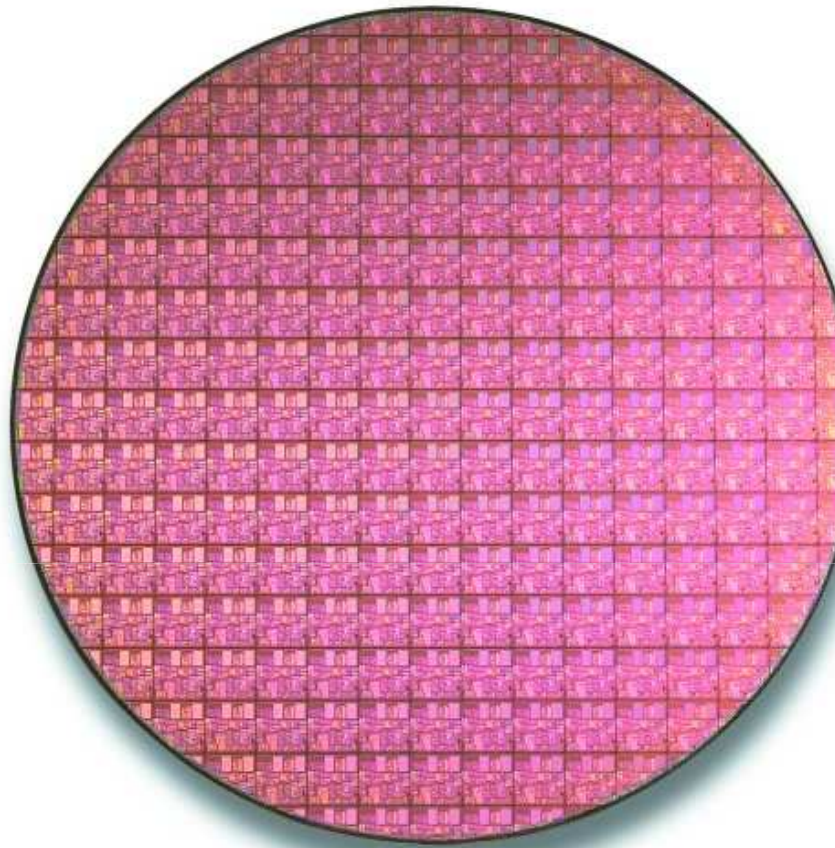


Lingote de Silício



**FIGURA 1.14 Processo de fabricação de um chip.** Após ser fatiado de um lingote de silício, os wafers virgens passam por 20 a 40 passos para criar wafers com padrões (veja a Figura 1.15). Esses wafers com padrões são testados com um testador de wafers e é criado um mapa das partes boas. Depois, os wafers são divididos em dies (moldes) (veja a Figura 1.9). Nessa figura, um wafer produziu 20 dies, dos quais 17 passaram no teste. (X significa que o die está ruim.) O aproveitamento de dies bons nesse caso foi de 17/20, ou 85%. Esses dies bons são soldados a encapsulamentos e testados outra vez antes de serem remetidos para os clientes. Um die encapsulado ruim foi encontrado nesse teste final.

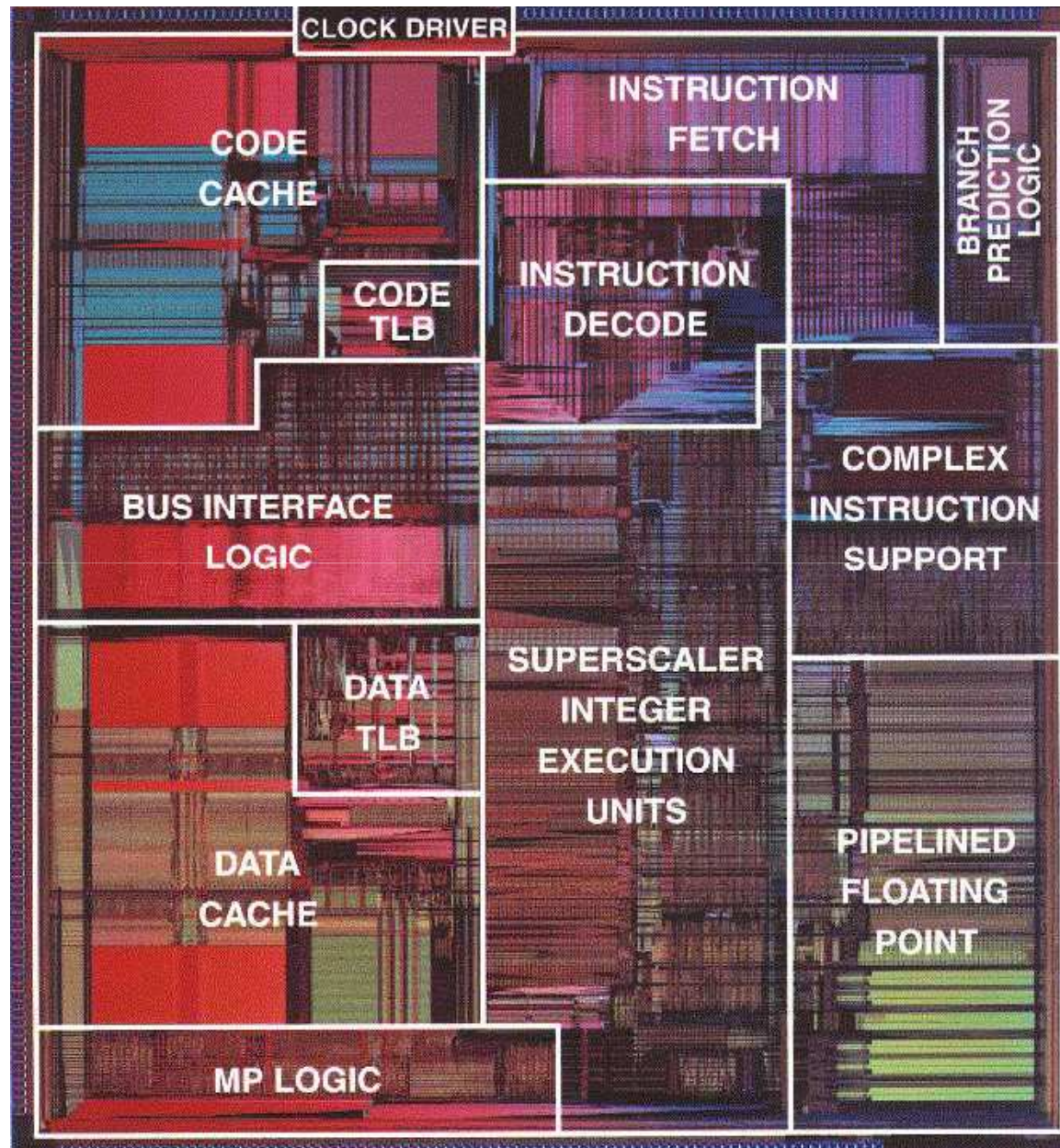


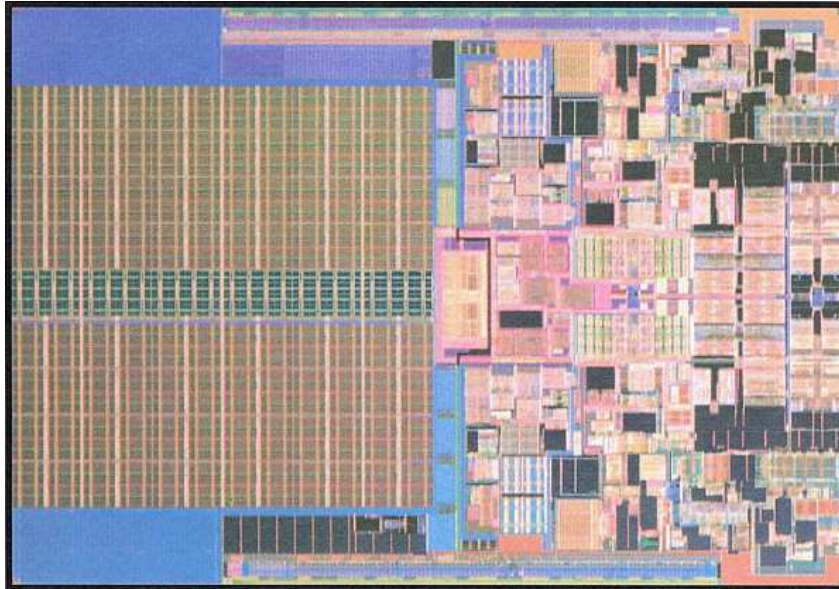


Core i7

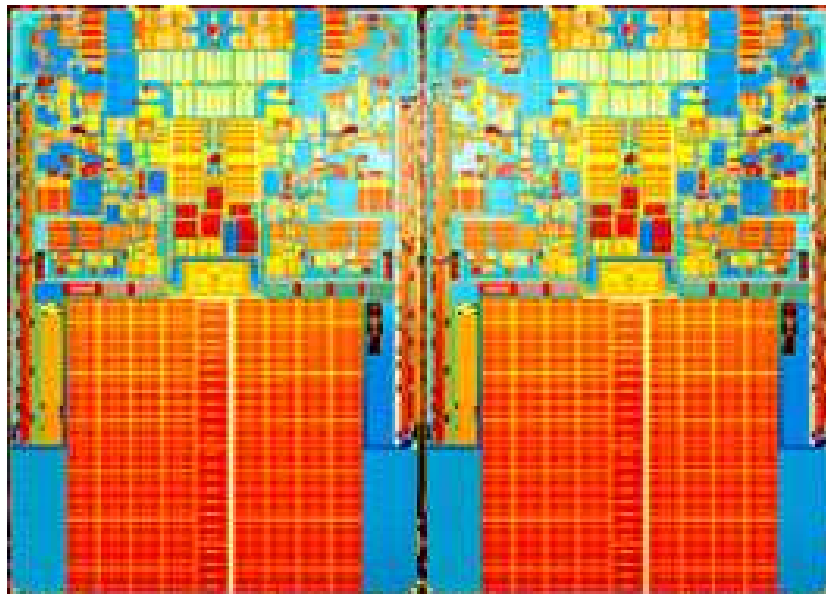
**Figura 1.15 Um wafer de 200mm de diâmetro com processadores Intel Pentium 4.** O número de dies de Pentium por wafer em 100% de aproveitamento é 165. A Figura 1.9 é uma microfotografia de um desses dies de Pentium 4. A área do die é de  $250\text{mm}^2$  e contém cerca de 55 milhões de transistores. Esse die usa uma tecnologia de 0,18 micron, o que significa que os menores transistores possuem um tamanho de aproximadamente 0,18 micron, embora normalmente sejam um pouco menores do que o tamanho real catalogado, que se refere ao tamanho dos transistores como “desenhados” *versus* o tamanho final fabricado. As várias dezenas de chips parcialmente arredondados nas bordas do wafer são inúteis; são incluídas porque é mais fácil criar as máscaras usadas para imprimir os padrões desejados ao silício.



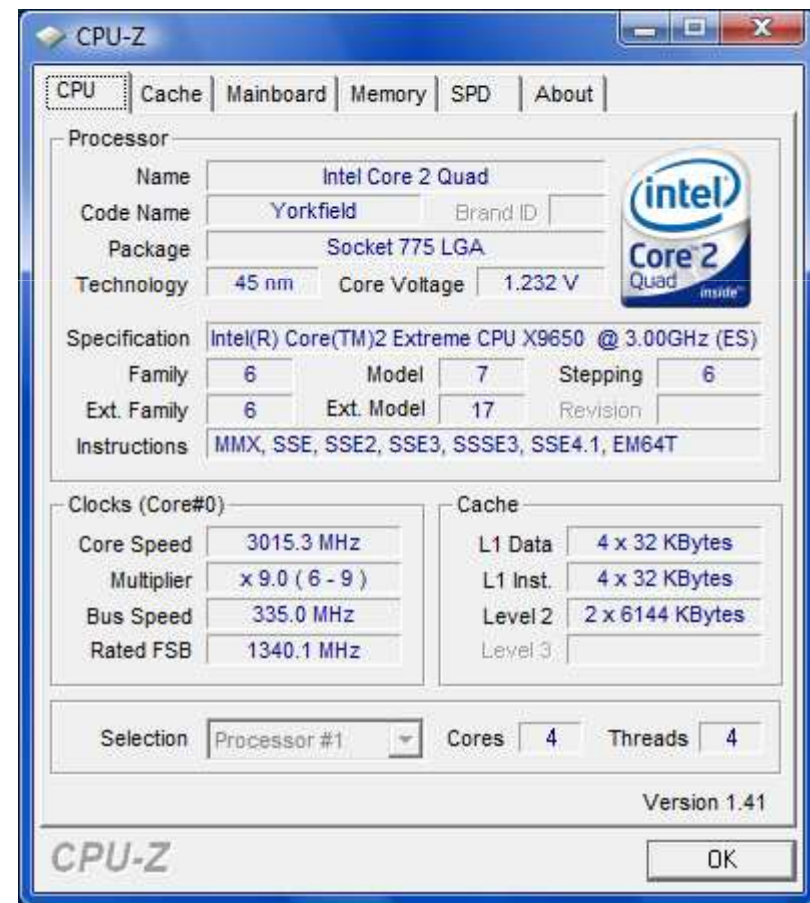




Penryn: Core2 Duo ,45nm, SSE4, 6MB L2 06/01/2008



Yorkfield: Core2 Quad 2x Penryn



Obs: Core i7 : Nehalem lançado dia 11/08/2008



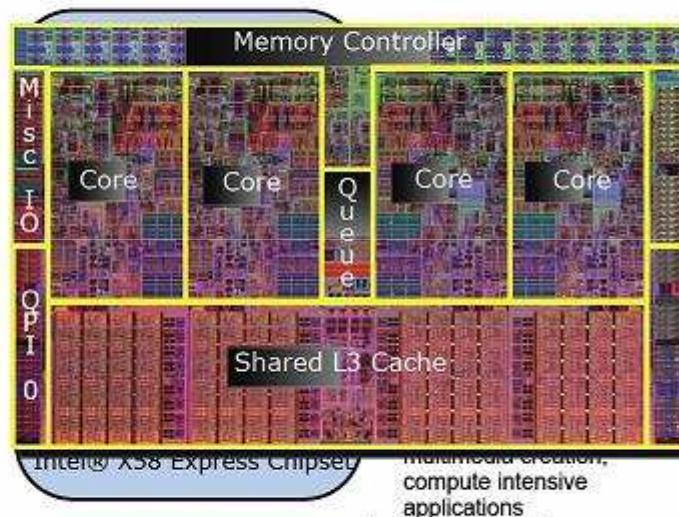


## Família Nehalem – codename Bloomfield

### Intel® Core™ i7 Processor

#### Performance/Features:

- 8 processing threads via Intel® Hyper-Threading Technology (HT)
- 4 cores
- Turbo Mode operation
- Intel® QuickPath Interconnect (Intel® QPI) to Intel® X58 Express Chipset
- Integrated Memory Controller (IMC) – 3ch DDR3
- 7 more SSE4 instructions
- Overspeed Protection Removed



Intel's Next Gen Computing Genius!

Intel Developer  
DEMO FORUM

The screenshot shows the CPU-Z software interface. The 'Processor' tab is selected, displaying the following information:

Processor			
Name	Intel Core i7 920		
Code Name	Bloomfield	Brand ID	
Package	Socket 1366 LGA		
Technology	45 nm	Core Voltage	1.208 V
Specification: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz			
Family	6	Model	A
Ext. Family	6	Ext. Model	1A
Instructions	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T		
Stepping: 4			
Revision: C0/C1			

The 'Clocks (Core #0)' section shows:

Clocks (Core #0)	
Core Speed	2659.5 MHz
Multiplier	x 20.0
Bus Speed	133.6 MHz
QPI Link	3207.2 MHz

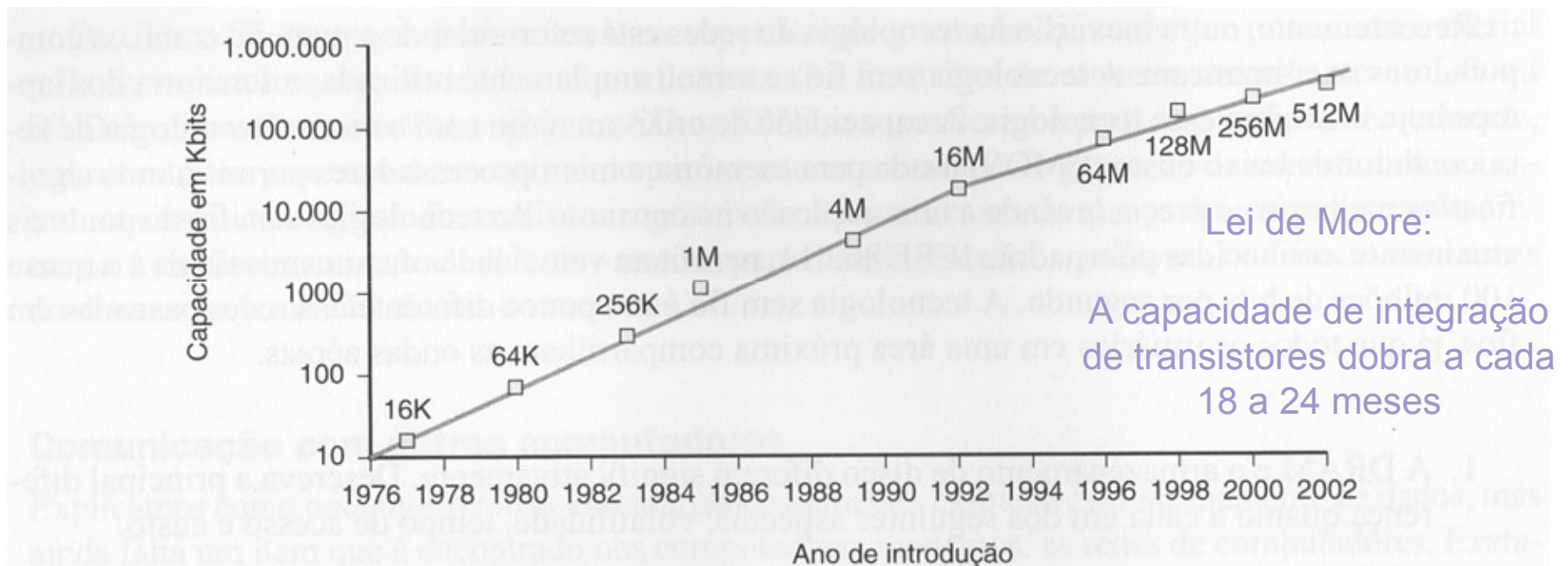
The 'Cache' section shows:

Cache	
L1 Data	4 x 32 KBytes
L1 Inst.	4 x 32 KBytes
Level 2	4 x 256 KBytes
Level 3	8 MBytes

The 'Selection' dropdown is set to 'Processor #1'. The 'Cores' field shows 4, and the 'Threads' field shows 8. The version is 1.50.



# Capacidade da Memória em chip DRAM

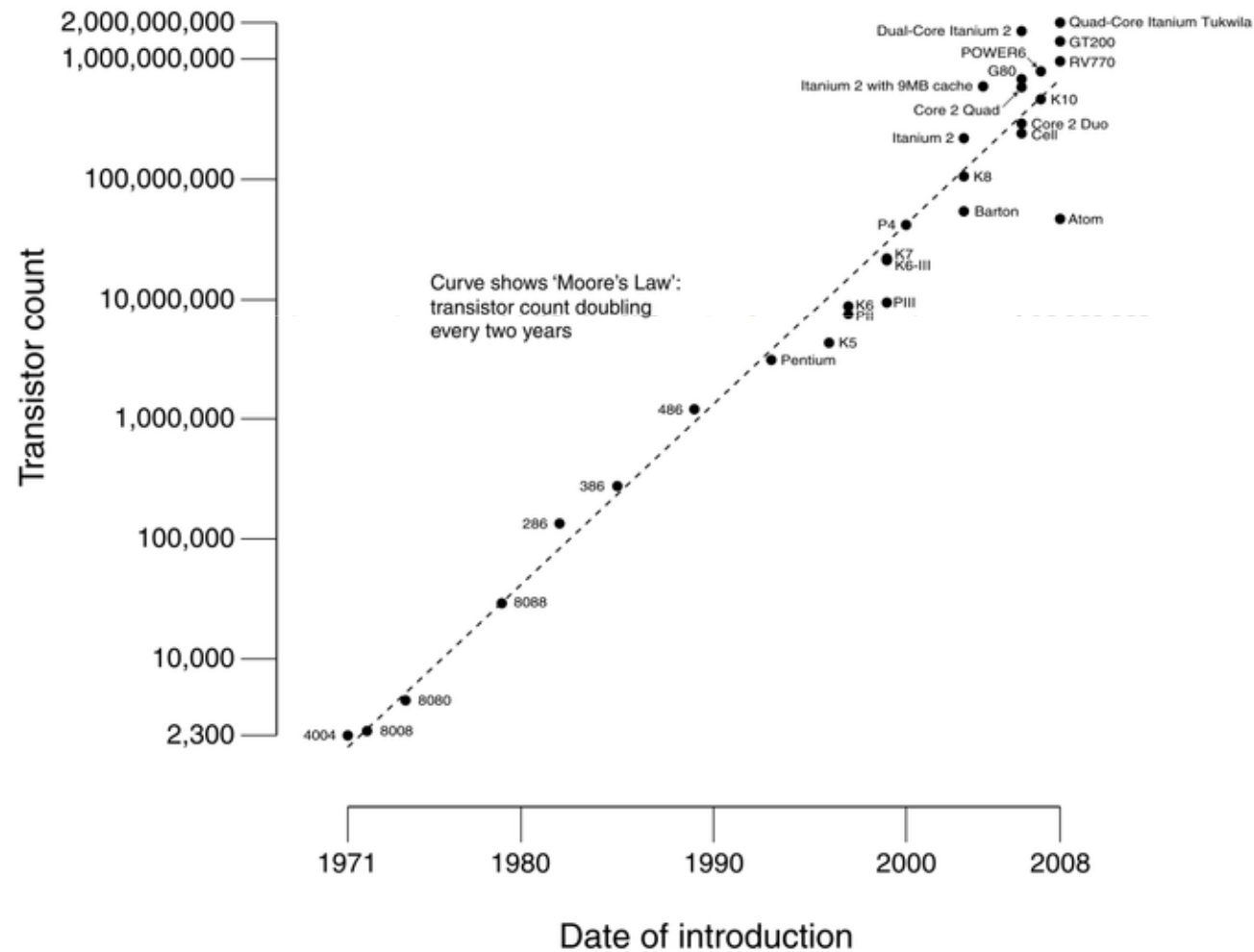


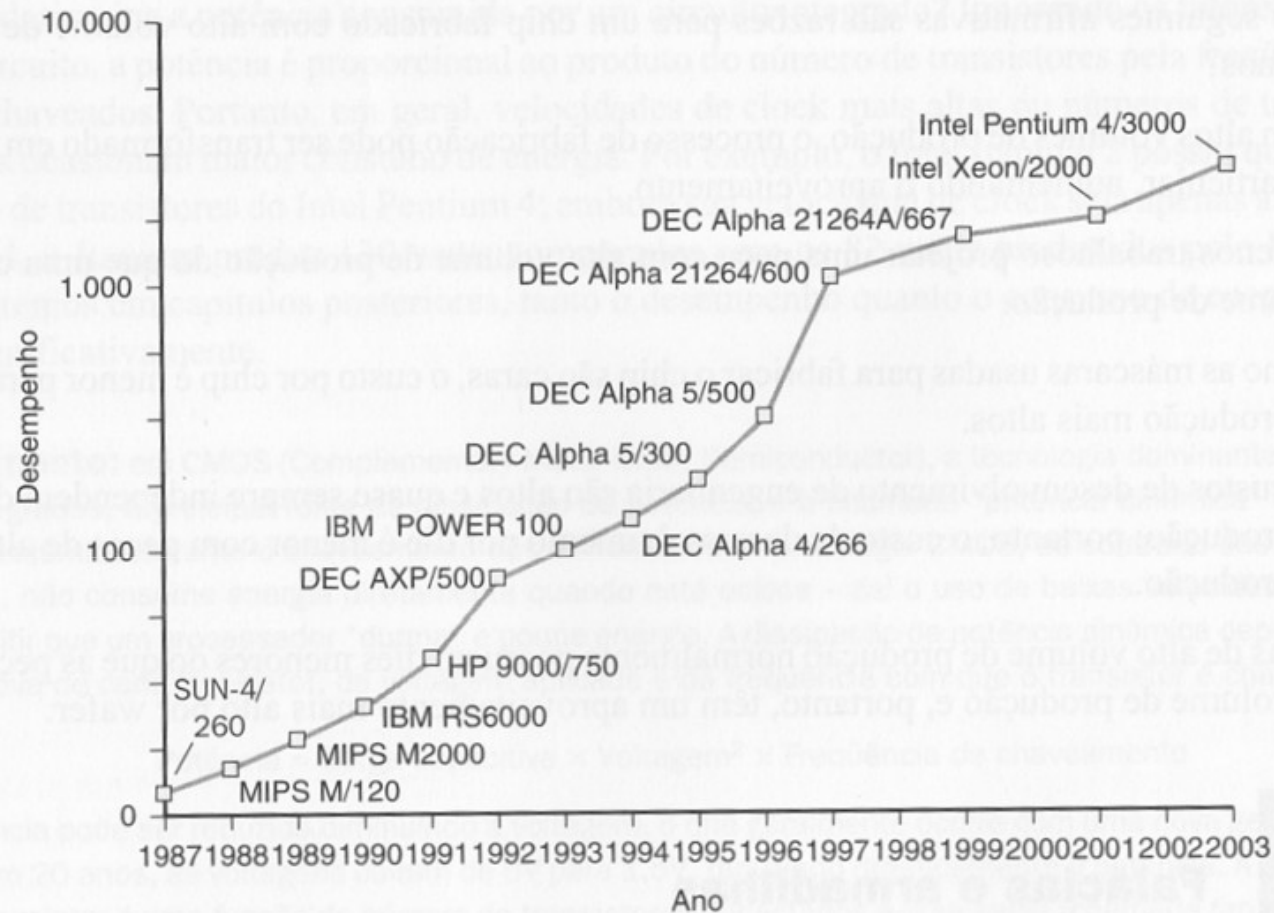
**FIGURA 1.13 Crescimento da capacidade por chip de DRAM ao longo do tempo.** O eixo y é medido em Kbits, onde K = 1024 ( $2^{10}$ ). A indústria de DRAM quadruplicou a capacidade a cada quase 3 anos, um aumento de 60% por ano, durante 20 anos. Essa estimativa de “quatro vezes a cada três anos” foi conhecida como a *regra do crescimento da DRAM*. Nos últimos anos, essa taxa diminuiu um pouco e está próximo do dobro a cada dois anos.



# Lei de Moore

## CPU Transistor Counts 1971-2008 & Moore's Law





**FIGURA 1.17 Aumento no desempenho das estações de trabalho entre 1987 e 2003.** Aqui, o desempenho é fornecido aproximadamente como o número de vezes que o computador é mais rápido do que o VAX-11/780, que é um padrão comumente usado. O índice de melhoria de desempenho está entre 1,5 e 1,6 vezes por ano. Esses números de desempenho baseiam-se na execução do SPECint (consulte o Capítulo 2) e escalado ao longo do tempo para resolver as alterações do *benchmark*. Para os processadores assinalados com x/y ao lado do nome, x é o modelo e y é a velocidade em megahertz.