

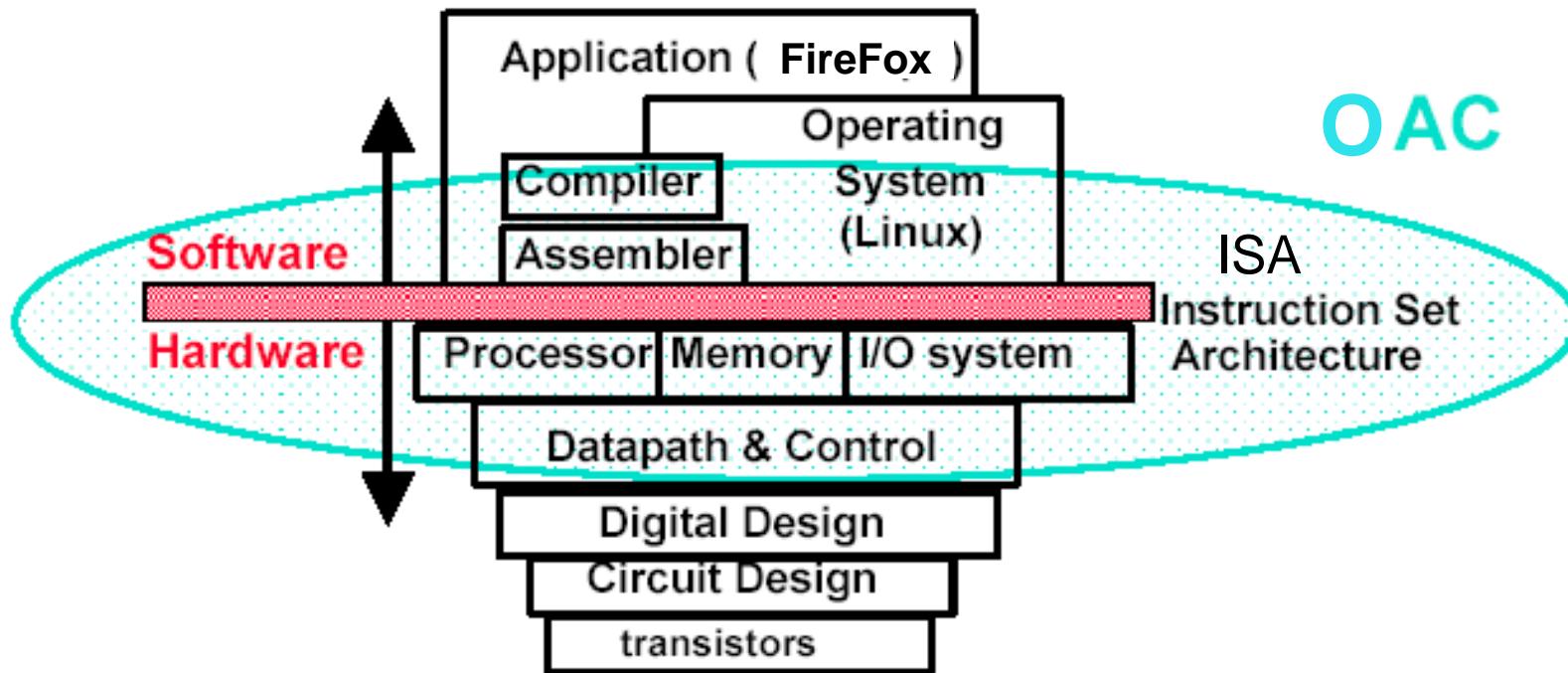


# Aula 1

## Introdução, Abstrações e Histórico



# O que é: Organização e Arquitetura de Computadores?



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



# Revisão de Prefixos

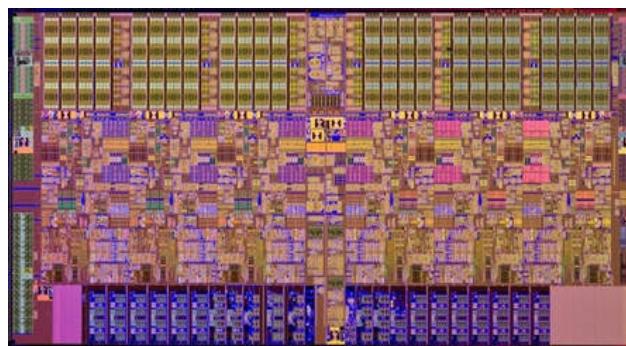
Decimal term	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	kB	$10^3$	kibibyte	KiB	$2^{10}$	2%
megabyte	MB	$10^6$	mebibyte	MiB	$2^{20}$	5%
gigabyte	GB	$10^9$	gibibyte	GiB	$2^{30}$	7%
terabyte	TB	$10^{12}$	tebibyte	TiB	$2^{40}$	10%
petabyte	PB	$10^{15}$	pebibyte	PiB	$2^{50}$	13%
exabyte	EB	$10^{18}$	exbibyte	EiB	$2^{60}$	15%
zettabyte	ZB	$10^{21}$	zebibyte	ZiB	$2^{70}$	18%
yottabyte	YB	$10^{24}$	yobibyte	YiB	$2^{80}$	21%

**FIGURE 1.1 The  $2^x$  vs.  $10^y$  bytes ambiguity was resolved by adding a binary notation for all the common size terms.** In the last column we note how much larger the binary term is than its corresponding decimal term, which is compounded as we head down the chart. These prefixes work for bits as well as bytes, so *gigabit* (Gb) is  $10^9$  bits while *gibibits* (Gib) is  $2^{30}$  bits.



# O que é um computador?

- Componentes:
  - Processador(es)
  - Dispositivos de entrada (mouse, teclado,...)
  - Dispositivos de saída (monitor, impressora,...)
  - Dispositivos de memória (DRAM, SRAM, HD, CD, DVD,...)
  - Dispositivos de comunicação (Ethernet, USB, IEEE1394, ...)
- Nosso foco principal: o processador (caminho de dados e controle)
  - Implementado usando milhões de transistores
  - Impossível de entender olhando para os transistores

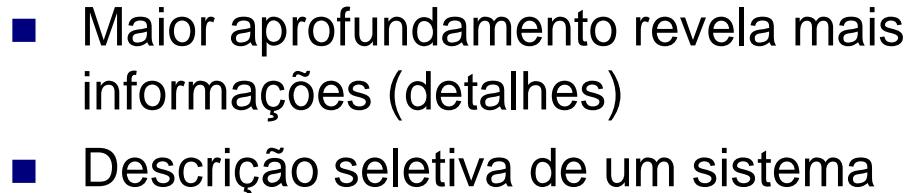




# Sistema Computacional

- Software de Aplicação, aplicação do usuário.
- Software de Base
  - Sistema Operacional, interface da máquina com o usuário,
    - Manipulam os sistemas básicos de I/O.
    - Alocação de memória e armazenamento.
    - Gerencia (proteção, escalonamento, controle, etc.) a realização de múltiplas tarefas.
  - Compiladores/Interpretadores, usuário desenvolve programa
    - Transforma as instruções de alto nível para linguagem de máquina
- Hardware, executa as instruções em linguagem de máquina

# Abstração



## Ex.: Tradução

The diagram illustrates the addition of two binary numbers, A and B. The top row shows the binary digits of A and B: A = 1000110010100000 and B = 1000110010100000. The middle row shows the sum of A and B: add A,B = 1000110010100000. A blue curved arrow points from the first column of A and B to the first column of the sum, indicating the addition of the least significant bits.

# Programar em Linguagem de Alto nível:

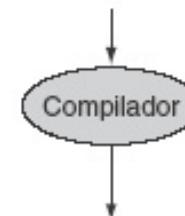
- 1) Linguagem próxima à humana
  - 2) Aumenta a produtividade
  - 3) Independência da máquina
  - 4) Hoje, compiladores muito eficientes

Programa em  
linguagem de  
alto nível  
(em C)

```

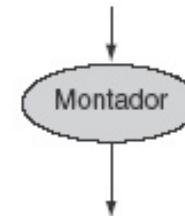
swap(int v[], int k)
{int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}

```



## Programa em assembly (para MIPS)

```
swap:    mul $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)



# Arquitetura do Conjunto de Instruções (ISA)

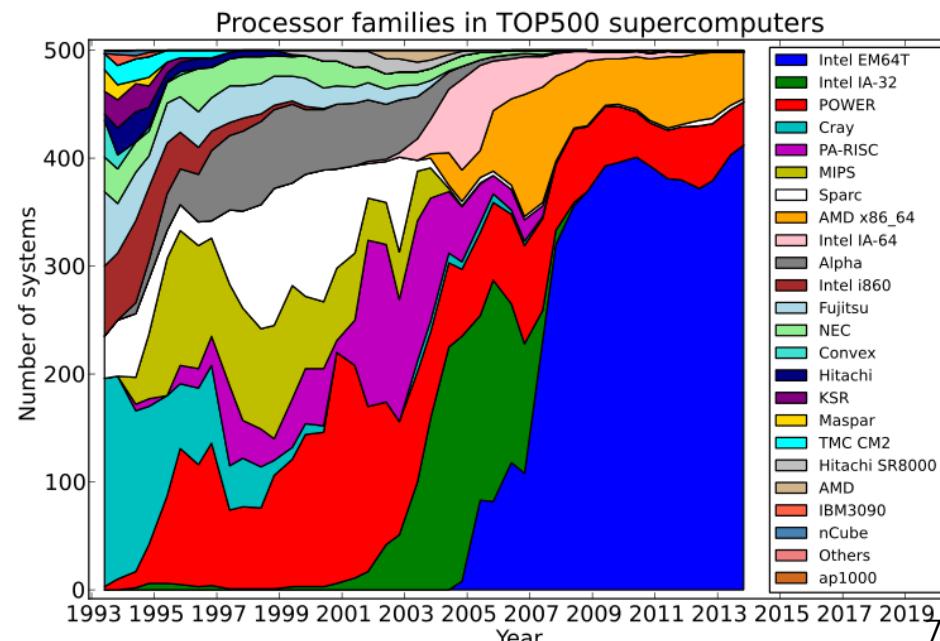
## ■ Uma abstração muito importante

- interface entre o hardware e o software de baixo nível
- padroniza instruções, padrões de bits de linguagem de máquina, etc.
- vantagem: *permite diferentes implementações de um processador*
- desvantagem: *algumas vezes impede o uso de inovações*



## ■ Arquiteturas de conjunto de instruções modernas:

- IA-32 (x86),
- EM64T, AMD64, x86-64 ou x64  
(Obs. IA-64 foi usada no Itanium)
- PowerPC,
- SGI,
- MIPS32 e MIPS64,
- SUN SPARC,
- ARMv7-A e ARMv8,
- HP PA-RISC
- e outras





# Arquitetura x Organização x Implementação

Arquitetura

## Architecture

Instruction set definition  
and compatibility

Organização

## Microarchitecture

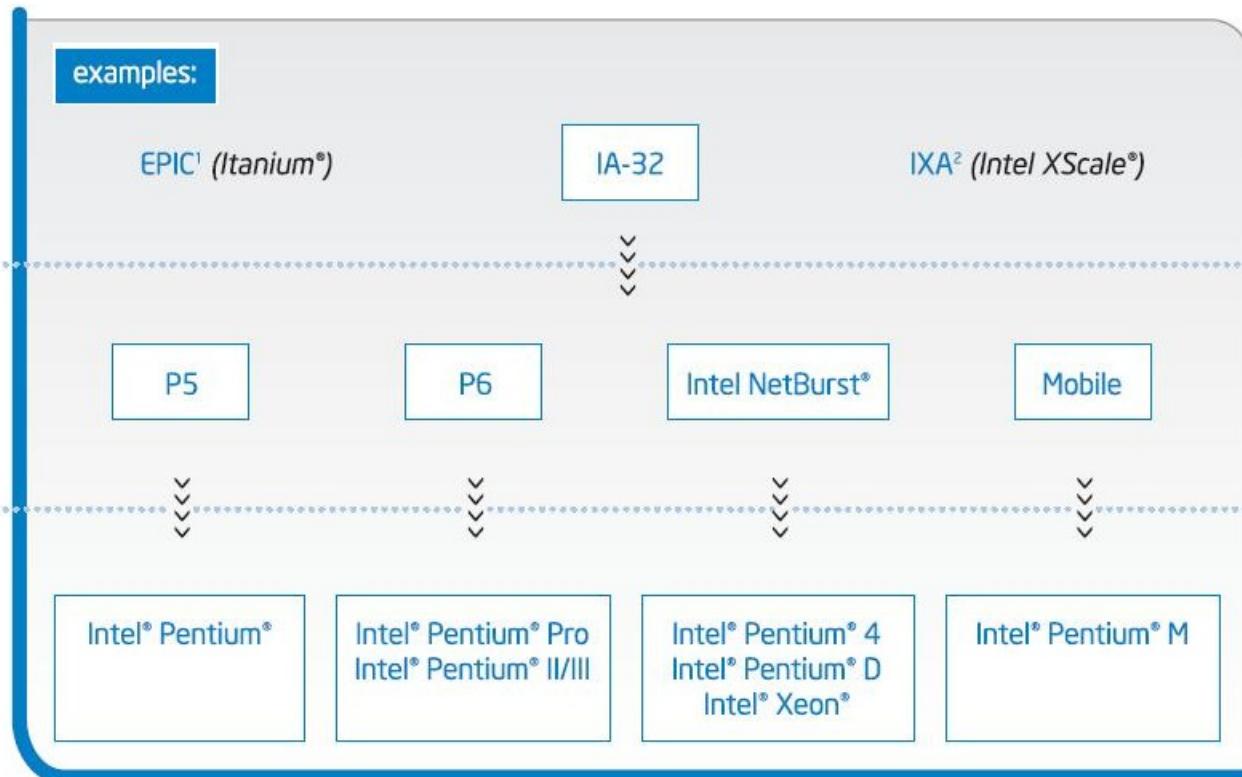
Hardware implementation  
maintaining instruction  
set compatibility with  
high-level architecture

Implementação

## Processors

Productized implementation  
of microarchitecture

### Microarchitecture History



1. EPIC (Explicitly Parallel Instruction Computing)

2. IXA (Intel® Internet Exchange Architecture)

**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



# No princípio (pré-computadores)

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

## 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.

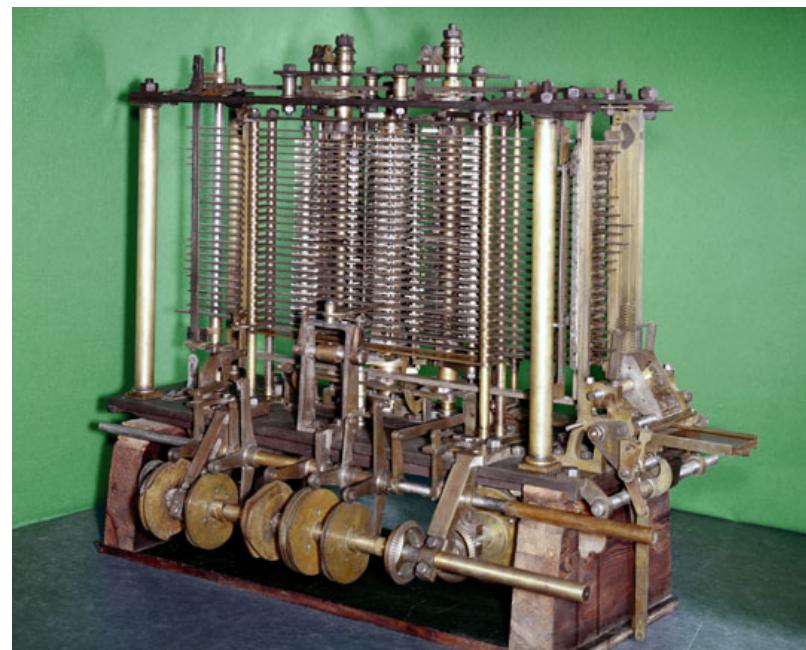
# A Calculadora de Babbage (1791~†1871)

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



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

Charles Babbage não conseguiu solucionar problemas mecânicos (precisão) devido à tecnologia da época.



Reconstrução pelo Museu de Londres, 1990



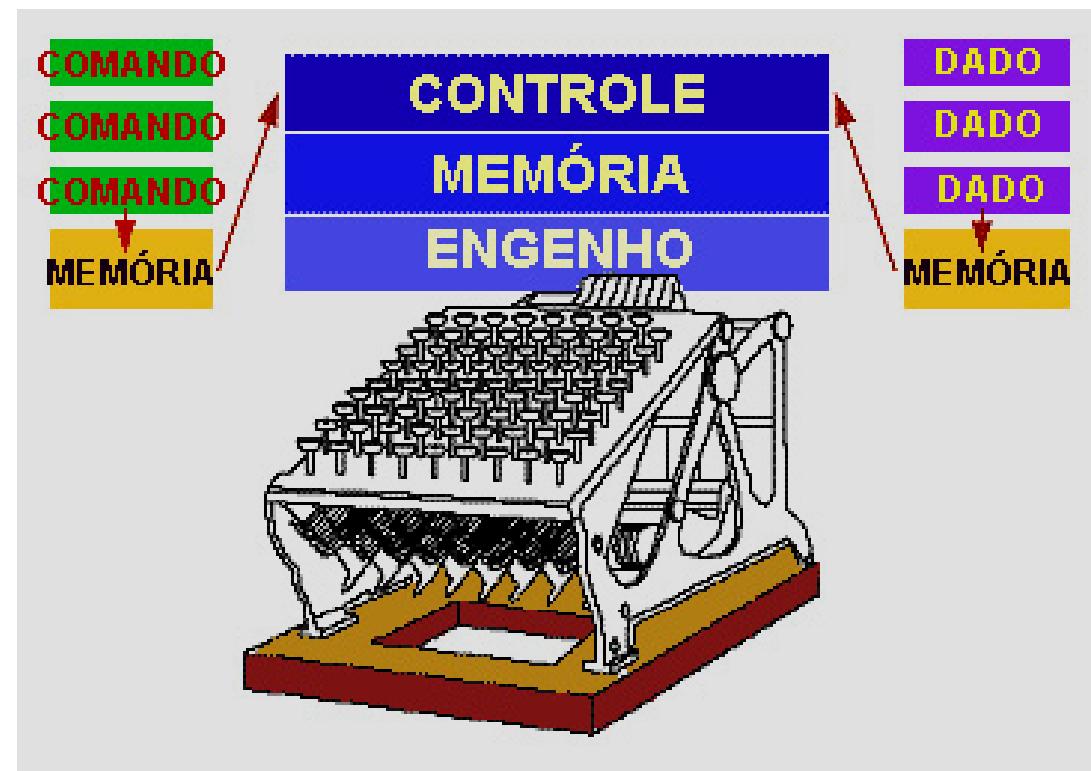
# A Calculadora de Babbage

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

Babbage dividiu sua máquina em três partes:

- Armazenamento
- Engenho
- Controle

Cartões perfurados:





# 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”

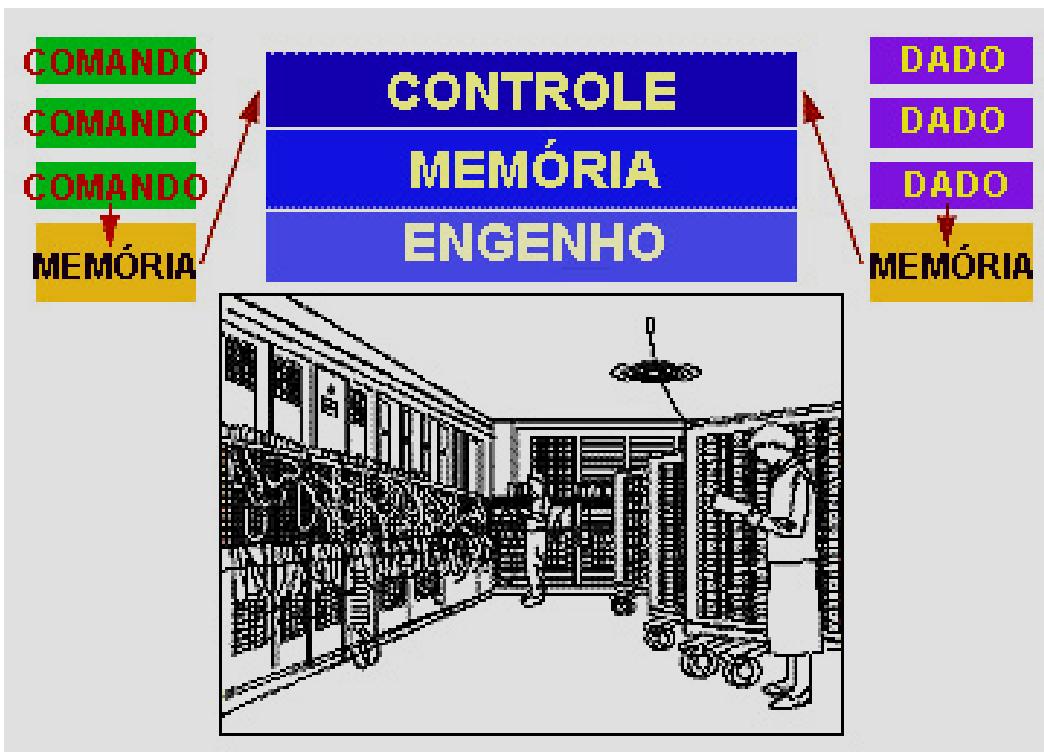
“subtrair 1 do número em curso”

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

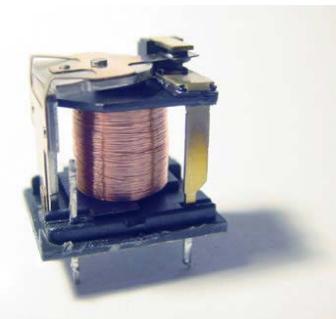
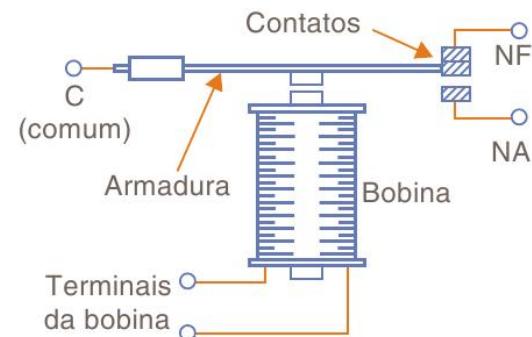


# Indo adiante

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



## Chave eletromecânica (Relé)

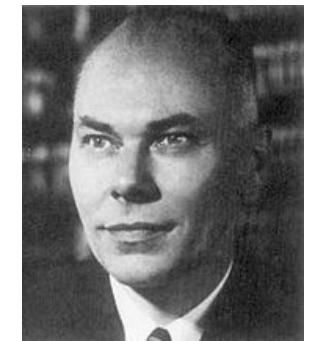




# MARK I

A série de computadores Mark (I,II,II,IV) foi desenvolvida na Universidade de Harvard durante os anos 40, sob liderança de Howard Aiken

O primeiro, Mark I, foi desenvolvido com o apoio da recém criada IBM e da Marinha dos EUA, entrou em operação em 1944 e foi utilizado até 1959. Realizava uma operação de multiplicação em 6 segundos





# O MARK I – Arquitetura Harvard

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

- Programa: Papel perfurado
- Dados: Dispositivos Eletromecânicos



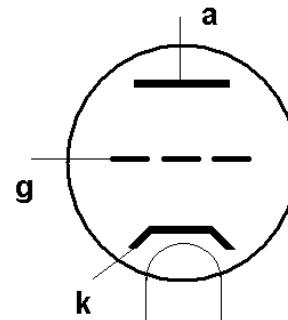
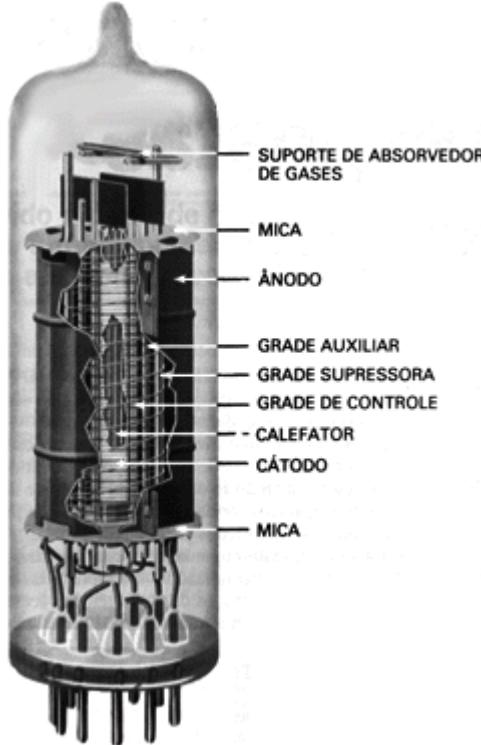
As instruções também eram armazenadas num formato diferente dos dados.

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



# Indo adiante

1906: Invenção da Válvula Termiônica.



Alta tensão entre A e K com corrente controlada pela tensão da grade ( $V_{gk}$ ).

Vantagem: tempo de comutação (on/off) muito menor que relés eletromecânicos.

Desvantagem: Alta tensão e dissipação térmica.

Vantagens hoje em dia: Robusta a transientes elétricos, fortes impulsos eletromagnéticos, tempestade solar, guerra nuclear, etc.

Uso: Transmissores de Rádio e TV (alta potência), forno de microondas, amplificadores de áudio (alta potência), etc



# 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 através da máquina, chamado de *Electronic Numerical Integrator And Calculator*.

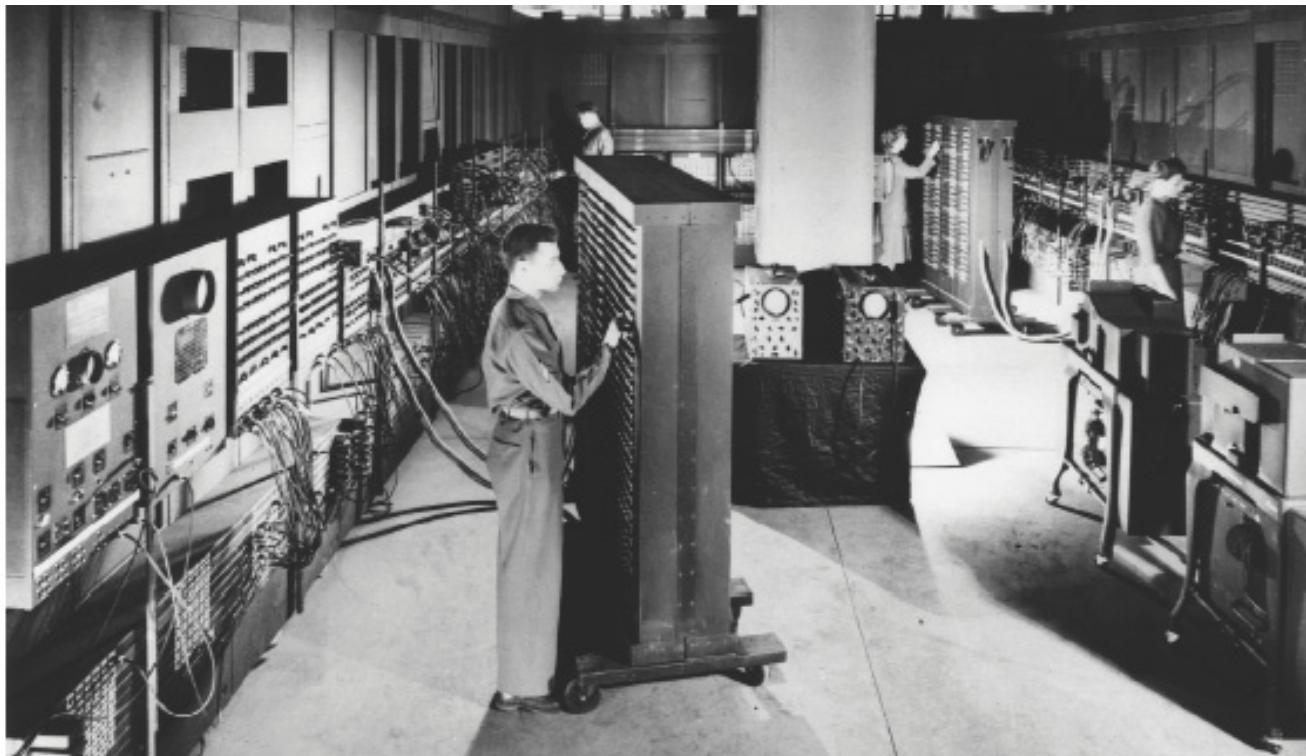
Podia fazer 5000 adições por segundo ou 357 multiplicações por segundo.

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



O ENIAC 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 decimais tinha 60 centímetros de comprimento
- Usava 18.000 válvulas > Calor!





# A arquitetura von Neumann

Em meados dos anos 40, o matemático húngaro, John von Neumann mostrou que as instruções poderiam ser representadas na mesma forma utilizada para os representar os dados.



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

## Conceito de Programa Armazenado

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

Combinar instruções e dados na mesma memória traz algumas vantagens:

- **Uso eficiente da memória.**

Um único bloco (grande) de memória ao invés 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 instruções. Como os sistemas von Neumann não fazem distinção entre dados e instruções, a máquina pode tentar executar dados como instruções, com resultados imprevisíveis.

- **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.



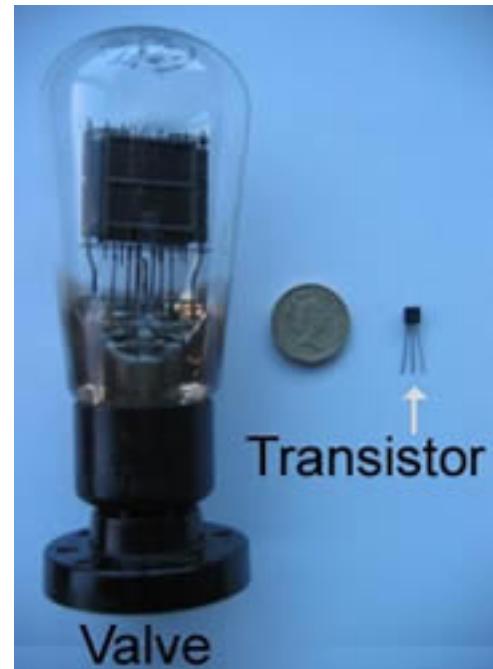
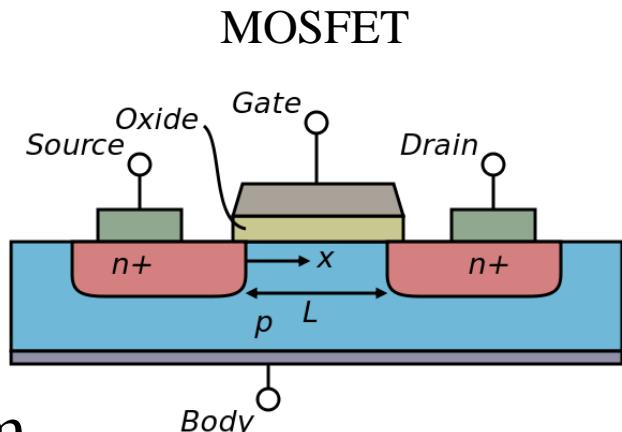
# Indo adiante

1950: Invenção do Transistor

Chave eletrônica.

Tecnologia :Tamanho do canal (L) nm

Menores, geram menos calor e são muito mais rápidos que as válvulas

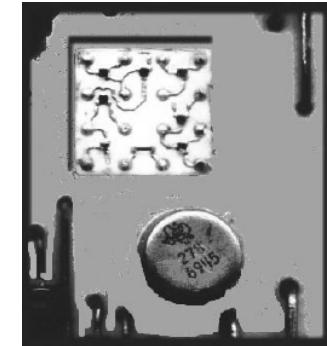




# Indo adiante

## Circuitos Integrados, CI, Chip

- SSI, MSI, LSI
- VLSI
- ULSI
- SLSI
- ...



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: centenas de milhões de transistores num chip
  - 2010: mais de 2 bilhões de transistores num chip
  - 2015: cerca de 10 bilhões de transistores num chip



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

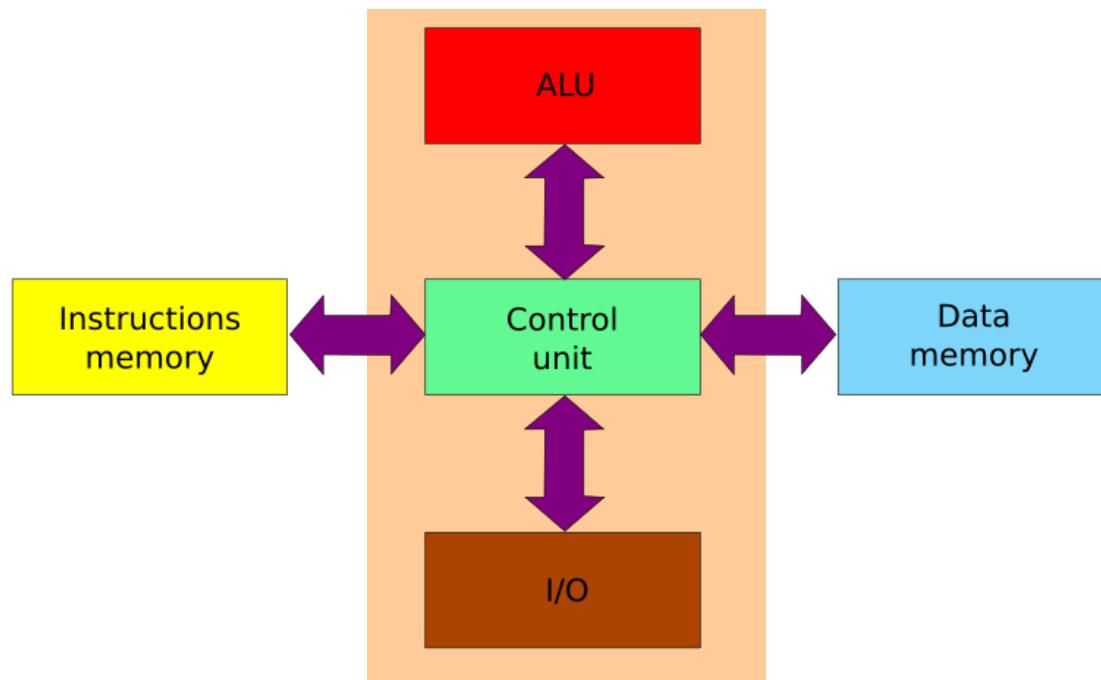
- Ano: Modelo - Tamanho da Palavra, Num. Trans., Frequência de clock
- 1971: Intel 4004 - 4 bits, 2.300 transistores, 740kHz
- 1972: Intel 8008 - 8 bits, 3.500, 500kHz
- 1976, Intel 8085 – 8 bits, 6.500, 5MHz
- 1978: Intel 8086 - 16 bits, 29.000, 10MHz
- 1982: Intel 80186 – 16bits
- 1982: Intel 80286 – 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, 7.500.000, 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 lançou arquitetura EM64T 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 (Hex/Quad) – 64 bits 731.000.000 3GHz (L3 cache)
- 2011: Core i7 2<sup>a</sup> Geração (Quad) – 64 bits 1.160.000.000 3.4GHz
- 2012: Core i7 3<sup>a</sup> Geração (Quad+GPU) – 64 bits 1.400.000.000 3.9GHz (3D)
- 2013: Core i7 4<sup>a</sup> Geração (Quad+GPU) – 64 bits 1.400.000.000 3.9GHz
- 2014: Core i7 5<sup>a</sup> Geração (Quad+GPU) – 64 bits 1.900.000.000 4GHz
- 2015: Core i7 6<sup>a</sup> Geração (Quad+GPU) – 64 bits ? 4GHz



# A Arquitetura Harvard Original

O retorno da arquitetura Harvard foi impulsionada inicialmente pelos Processadores Digitais de Sinais, e utilizada ainda hoje, na sua forma pura, em diversos processadores e microcontroladores de baixo custo.

Ex.: Microcontroladores PIC, 8051, AVR8, etc.

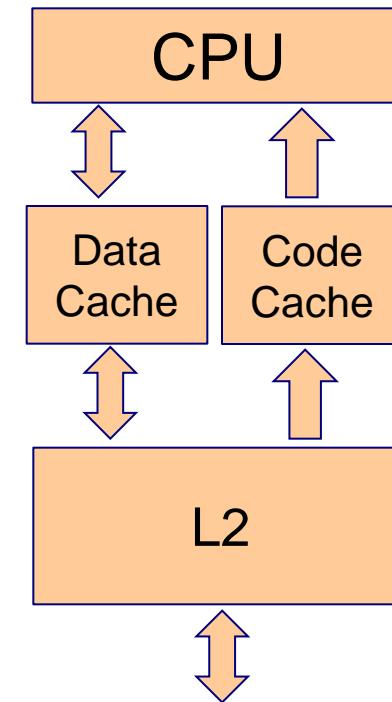
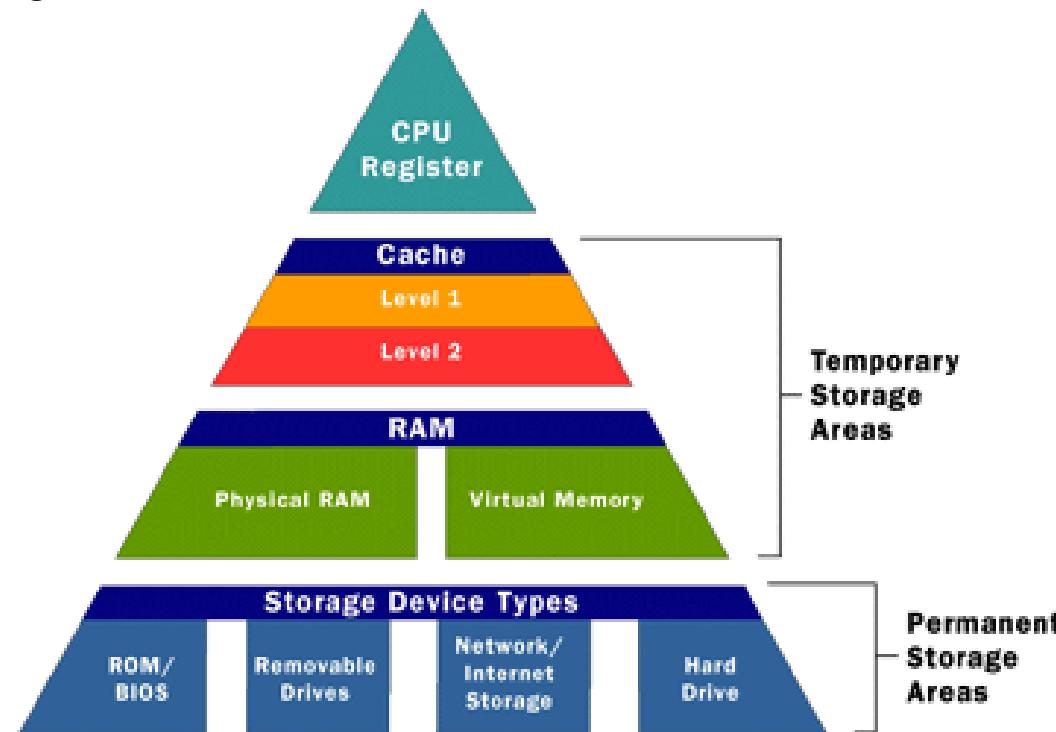


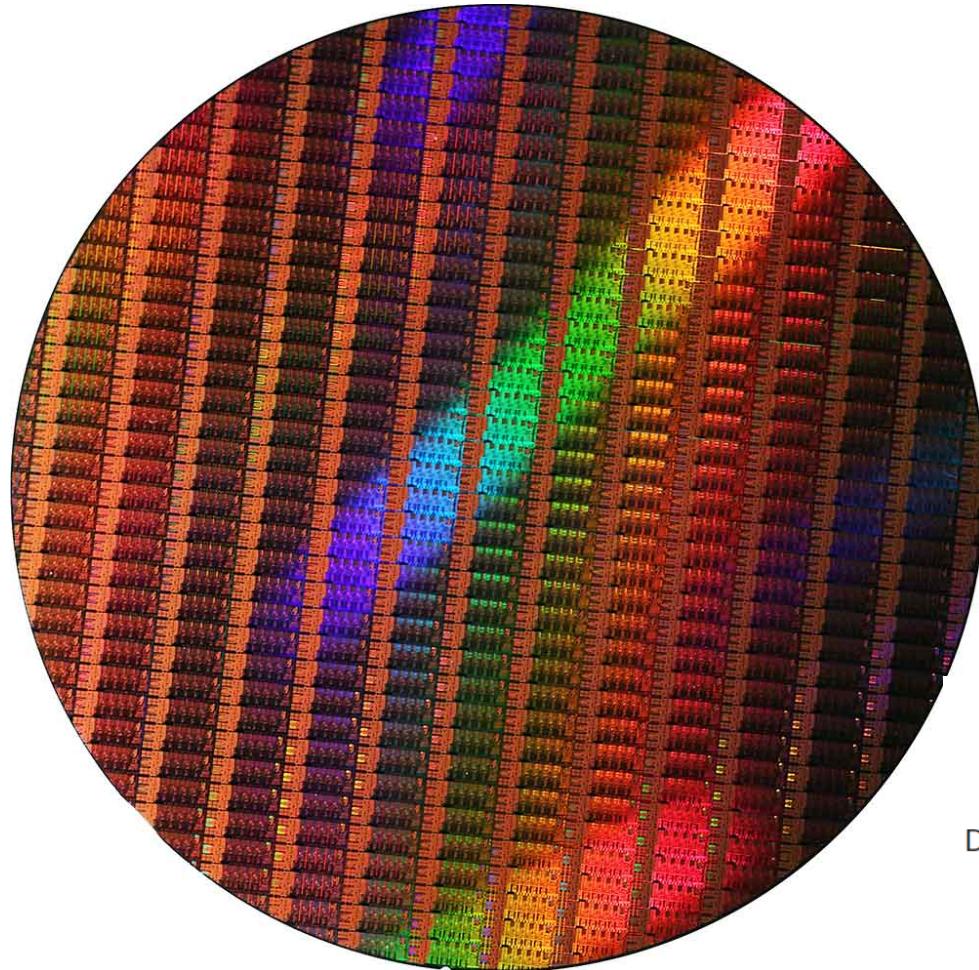


# A Arquitetura Harvard Modificada

É atualmente utilizada em praticamente todos os sistemas computacionais.

Une os benefícios da maior largura de banda (acesso a instruções e dados simultaneamente) com o conceito de programa armazenado.





$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{yield}}$$

$$\text{Dies per wafer} \approx \frac{\text{Wafer area}}{\text{Die area}}$$

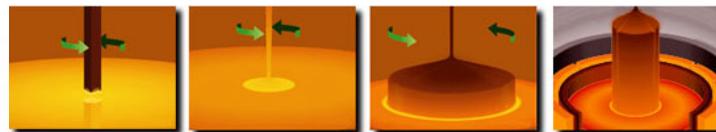
$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area}/2))^2}$$

**FIGURE 1.13 A 12-inch (300 mm) wafer of Intel Core i7 (Courtesy Intel).** The number of dies on this 300 mm (12 inch) wafer at 100% yield is 280, each 20.7 by 10.5 mm. The several dozen partially rounded chips at the boundaries of the wafer are useless; they are included because it's easier to create the masks used to pattern the silicon. This die uses a 32-nanometer technology, which means that the smallest features are approximately 32 nm in size, although they are typically somewhat smaller than the actual feature size, which refers to the size of the transistors as “drawn” versus the final manufactured size.

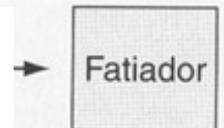
Obtida por observação



# Fabricação de um chip



Lingote de Silício  
Monocristalino  
99.999999999% puro

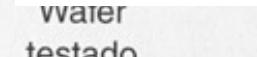


Wafers  
virgens

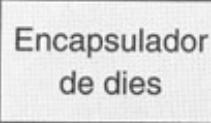
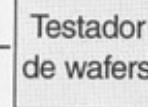
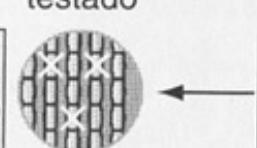
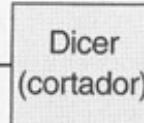


20 a 40 passos  
de processamento

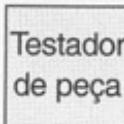
Wafers com padrões



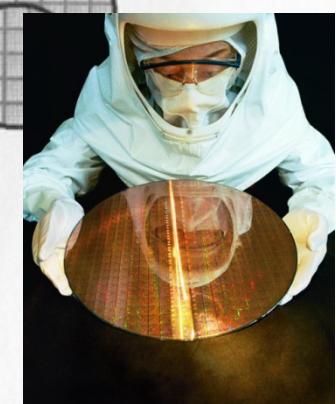
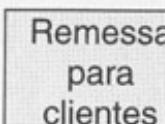
Dies testados



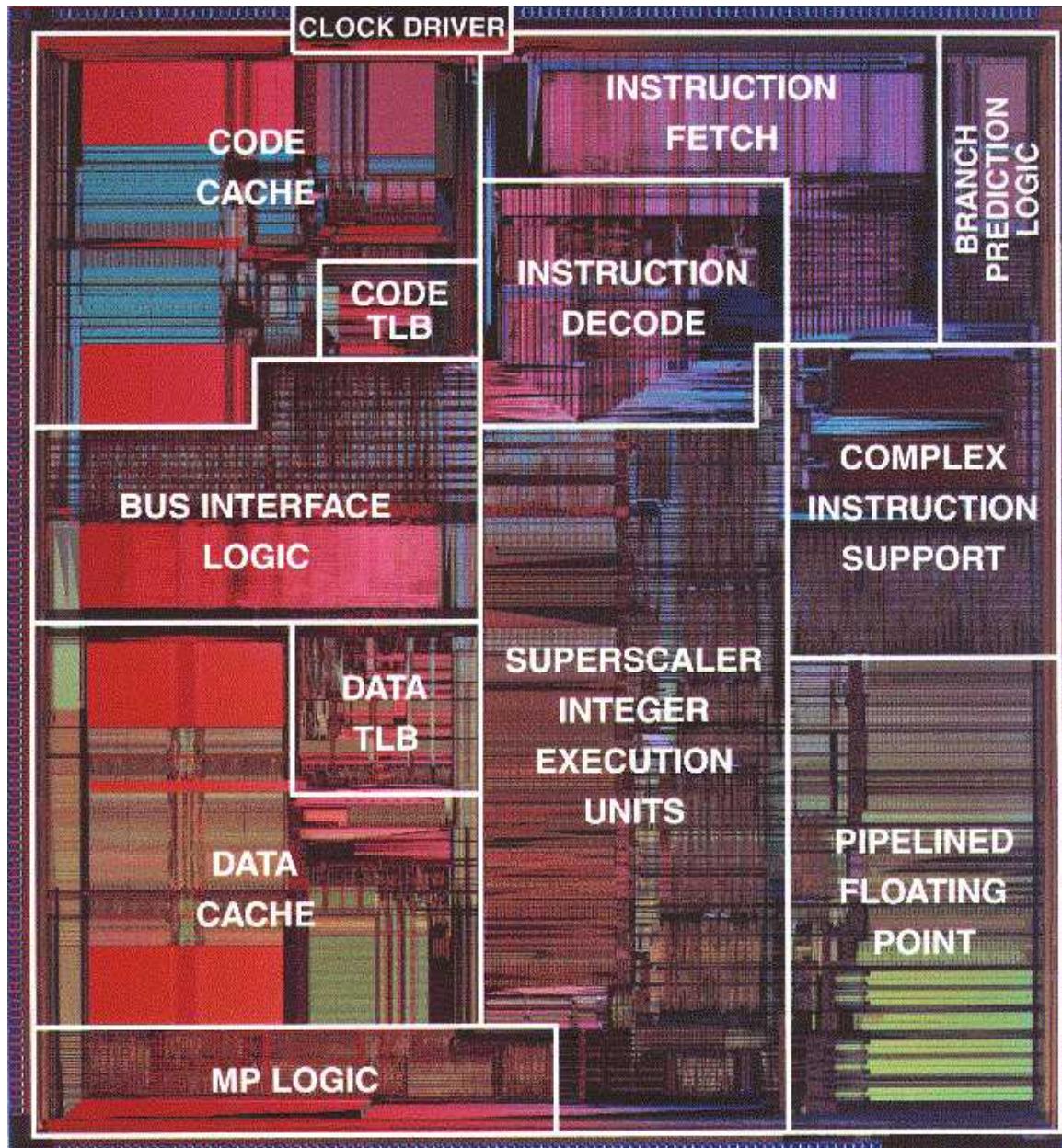
Dies encapsulados



Dies encapsulados  
e testados



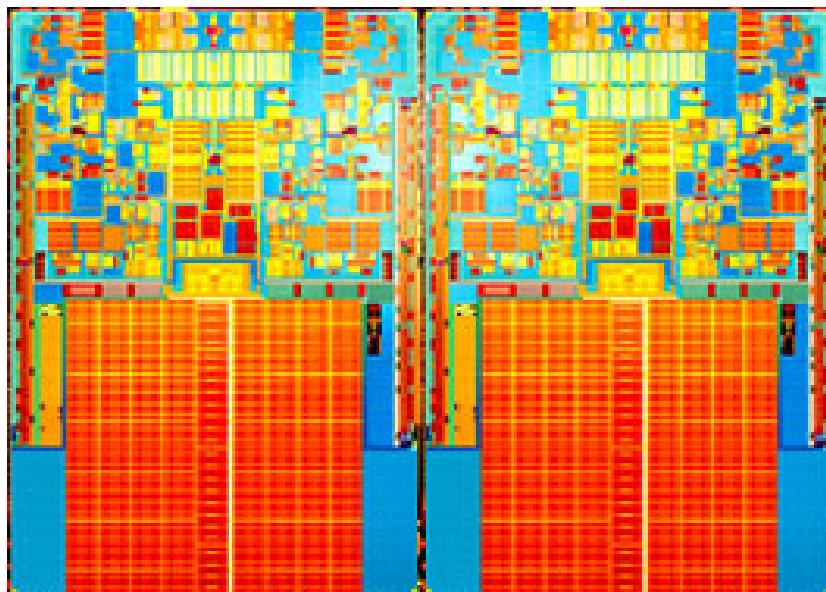
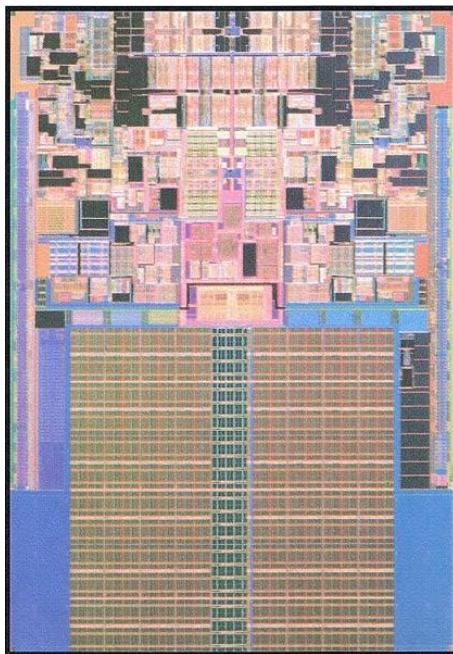
**FIGURA 1.14 Processo de fabricação de um chip.** Após ser fatiado de um lingote de silício, os wafers virgens passam por 20 a 40 passos para criar wafers com padrões (veja a Figura 1.15). Esses wafers com padrões são testados com um testador de wafers e é criado um mapa das partes boas. Depois, os 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.



# 486DX2



Core2Duo  
Conroe(65nm)



Core2 Quad Yorkfield: 2x Penryn (45nm)

**CPU-Z**

- [CPU](#) | [Caches](#) | [Mainboard](#) | [Memory](#) | [SPD](#) | [Graphics](#) | [About](#)

**Processor**

Name	Intel Core 2 Duo E6750		
Code Name	Conroe	Brand ID	
Package	Socket 775 LGA		
Technology	65 nm	Core Voltage	1.296 V

Specification: Intel(R) Core(TM)2 Duo CPU E6750 @ 2.66GHz

Family	6	Model	F	Stepping	B
Ext. Family	6	Ext. Model	F	Revision	G0

Instructions: MMX, SSE, SSE2, SSE3, SSSE3, EM64T

**Clocks (Core #0)**

Core Speed	2666.8 MHz
Multiplier	x 8.0
Bus Speed	333.4 MHz
Rated FSB	1333.4 MHz

**Cache**

L1 Data	2 x 32 KBytes	8-way
L1 Inst.	2 x 32 KBytes	8-way
Level 2	4096 KBytes	16-way
Level 3		

Selection: Processor #1    Cores: 2    Threads: 2

**CPU-Z**

- [CPU](#) | [Cache](#) | [Mainboard](#) | [Memory](#) | [SPD](#) | [About](#)

**Processor**

Name	Intel Core 2 Quad		
Code Name	Yorkfield	Brand ID	
Package	Socket 775 LGA		
Technology	45 nm	Core Voltage	1.232 V

Specification: Intel(R) Core(TM)2 Extreme CPU X9650 @ 3.00GHz (ES)

Family	6	Model	7	Stepping	6
Ext. Family	6	Ext. Model	17	Revision	

Instructions: MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, EM64T

**Clocks (Core#0)**

Core Speed	3015.3 MHz
Multiplier	x 9.0 ( 6 - 9 )
Bus Speed	335.0 MHz
Rated FSB	1340.1 MHz

**Cache**

L1 Data	4 x 32 KBytes
L1 Inst.	4 x 32 KBytes
Level 2	2 x 6144 KBytes
Level 3	

Selection: Processor #1    Cores: 4    Threads: 4

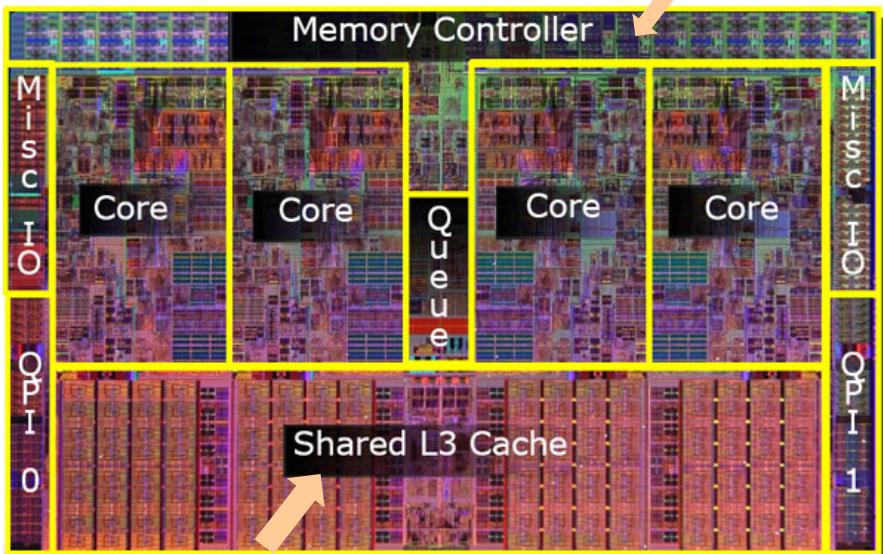
Version 1.41

**CPU-Z**

OK



## Core i7



**CPU-Z**

CPU | Cache | Mainboard | Memory | SPD | About

**Processor**

Name	Intel Core i7 920
Code Name	Bloomfield
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 Stepping: 4  
 Ext. Family: 6 Ext. Model: 1A Revision: C0/C1  
 Instructions: MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T

**Clocks (Core #0)**

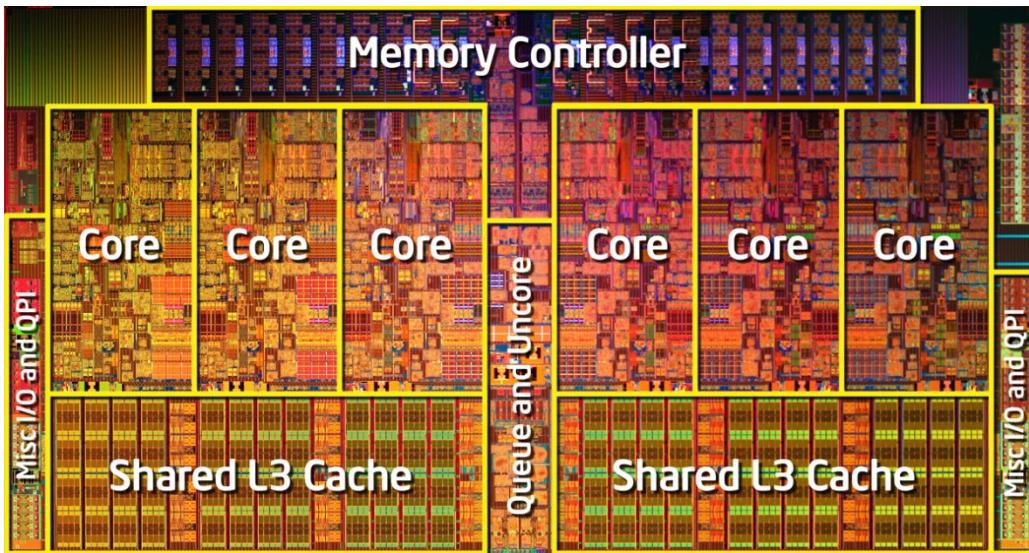
Core Speed	2659.5 MHz
Multiplier	x 20.0
Bus Speed	133.6 MHz
QPI Link	3207.2 MHz

**Cache**

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

Selection: Processor #1 Cores: 4 Threads: 8

Version 1.50



**CPU-Z**

CPU | Caches | Mainboard | Memory | SPD | Graphics | About

**Processor**

Name	Intel Core i7 Extreme 980X
Code Name	Gulftown
Package	Socket 1366 LGA
Technology	32 nm
Core Voltage	1.184 V

Specification: Intel(R) Core(TM) i7 CPU X 980 @ 3.33GHz  
 Family: 6 Model: C Stepping: 2  
 Ext. Family: 6 Ext. Model: 2C Revision: B1  
 Instructions: MMX, SSE (1, 2, 3, 3S, 4.1, 4.2), EM64T, VT-x, AES

**Clocks (Core #0)**

Core Speed	3347.7 MHz
Multiplier	x 25.0
Bus Speed	133.9 MHz
QPI Link	3213.8 MHz

**Cache**

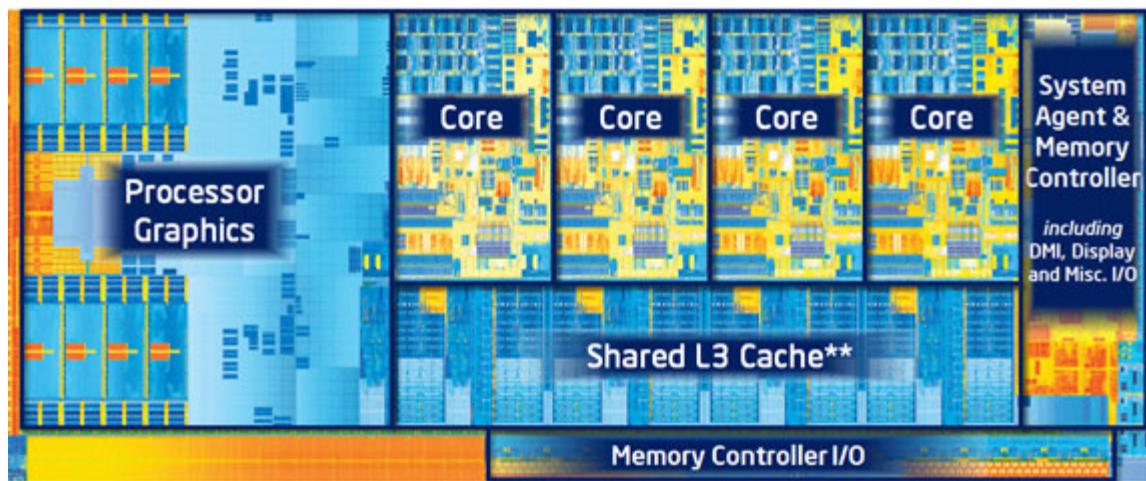
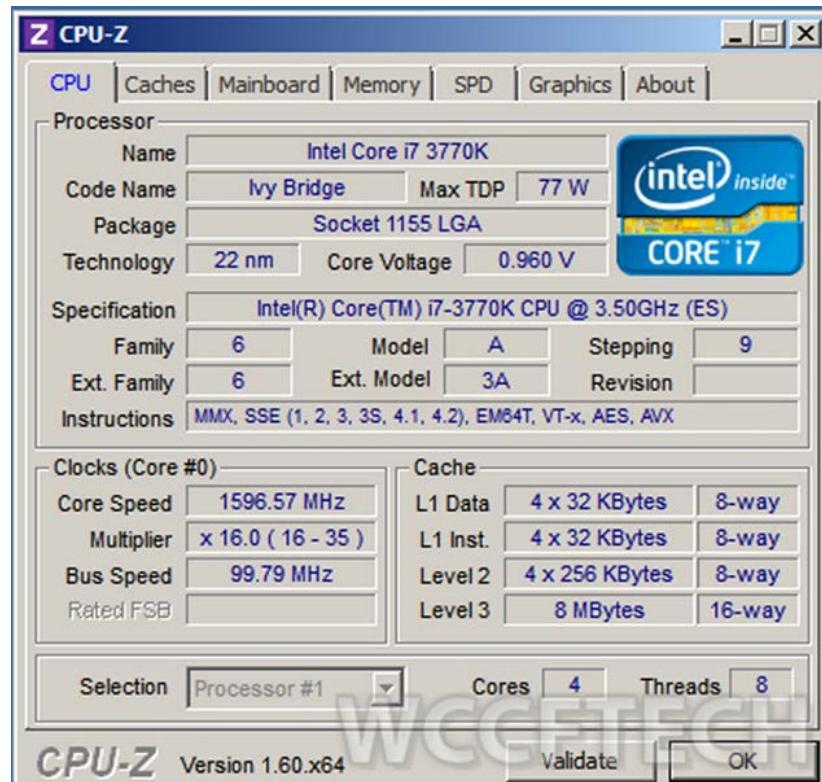
L1 Data	6 x 32 KBytes
L1 Inst.	6 x 32 KBytes
Level 2	6 x 256 KBytes
Level 3	12 MBytes

Selection: Processor #1 Cores: 6 Threads: 12

CPU-Z Version 1.54 Validate OK



## Core i7 de 3<sup>a</sup> geração





# Core i7 de 4<sup>a</sup> geração

Maior eficiência energética

**CPU-Z**

**CPU** | Caches | Mainboard | Memory | SPD | Graphics | About

**Processor**

Name	Intel Core i7 4770K		
Code Name	Haswell	Max TDP	84 W
Package	Socket 1150 LGA		
Technology	22 nm	Core VID	1.114 V

**Specification**

Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz (ES)			
Family	6	Model	C
Ext. Family	6	Ext. Model	3C
Revision	C0		

**Instructions**

- MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x, AES, AVX, AVX2, FMA3

**Clocks (Core #0)**

Core Speed	3900.12 MHz
Multiplier	x 39.0 ( 8 - 39 )
Bus Speed	100.00 MHz
Rated FSB	

**Cache**

L1 Data	4 x 32 KBytes	8-way
L1 Inst.	4 x 32 KBytes	8-way
Level 2	4 x 256 KBytes	8-way
Level 3	8 MBytes	16-way

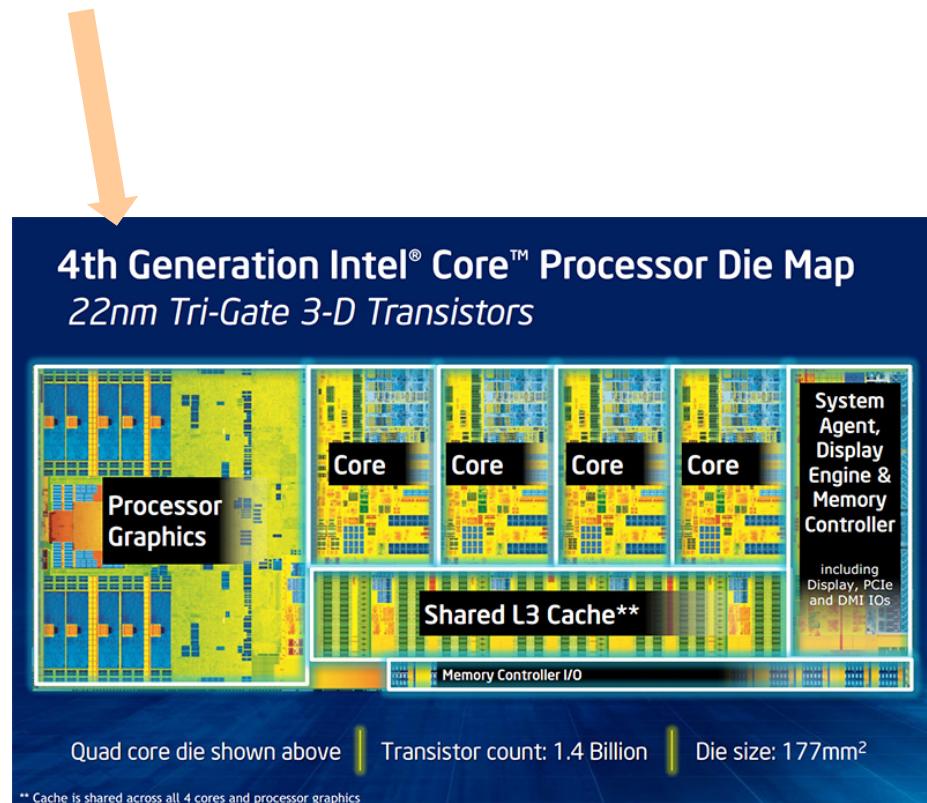
**Selection**

Processor #1
--------------

**Cores** 4    **Threads** 8

**Validate**    **OK**

**CPU-Z** Version 1.64.1.x64





# Core i7 de 6<sup>a</sup> geração

Mudanças pouco significativas em relação à 5<sup>a</sup>

**CPU-Z**

**CPU** | Caches | Mainboard | Memory | SPD | Graphics | About |

**Processor**

Name	Intel Core i7 6700K		
Code Name	Skylake	Max TDP	95.0 W
Package	Socket 1151 LGA		
Technology	14 nm	Core Voltage	1.328 V

**Specification**

Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz (ES)			
Family	6	Model	E
Ext. Family	6	Ext. Model	5E
Revision	R0		

**Instructions**

MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x, AES, AVX, AVX2, FMA3, TSX

**Clocks (Core #0)**

Core Speed	4213.31 MHz
Multiplier	x 42.0 (8 - 42)
Bus Speed	100.24 MHz
Rated FSB	

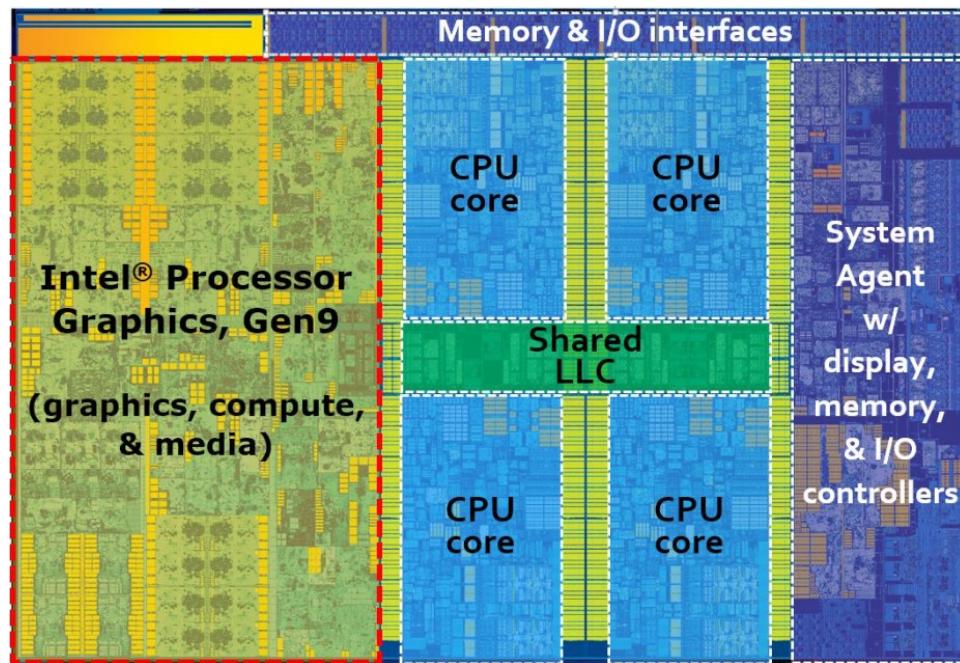
**Cache**

L1 Data	4 x 32 KBytes	8-way
L1 Inst.	4 x 32 KBytes	8-way
Level 2	4 x 256 KBytes	4-way
Level 3	8 MBytes	16-way

**Selection**

Processor #1 | Cores: 4 | Threads: 8

**CPU-Z** Ver. 1.72.1.x64 | Tools | Validate | OK



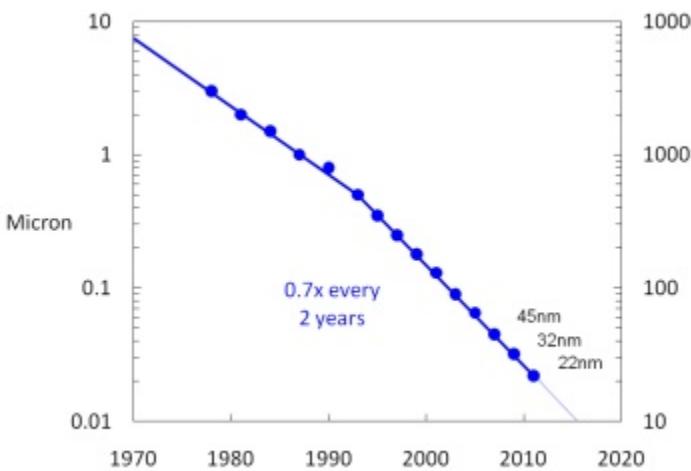




# Lei de Moore (1965)



Gordon Moore (1929~)  
Co-fundador da Intel



A capacidade de integração de transistores dobra a cada 18 a 24 meses

