

DCC007 – Organização de Computadores II

Aula 1 – Introdução ao Curso + Evolução

Prof. Omar Paranaiba Vilela Neto



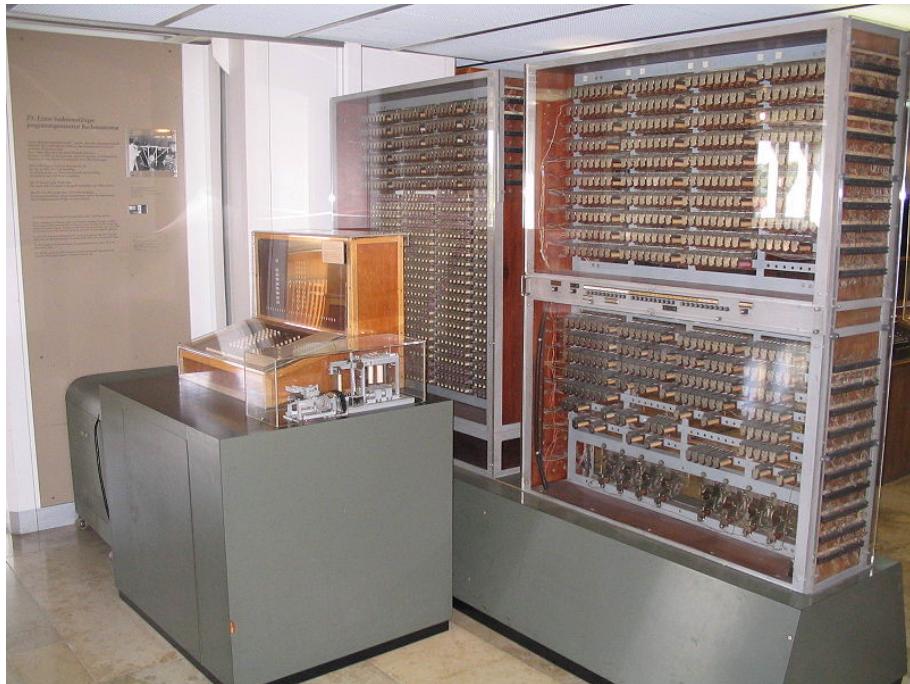
História

- .1906 – Semicondutores usados para detecção de sinais de rádio
- .1925 – Conceito de **FET*** patenteado por J. Lilienfeld
- .1941 – Z3 desenvolvido por Konrad Zuse – primeiro computador
- .1946 – ENIAC – primeiro computador eletrônico
- .1947 – **Transistor “Inventado”**
 - Bardeen, Brattain and Shockley, AT&T, Prêmio Nobel em 1956
- .1958 – **Circuito integrado**
 - Kilby & Noyce (falecido em 1990)
 - Kilby – Prêmio Nobel em 2000
- .1960 - **MOSFET** manufaturado e patenteado
- .1963 – Invenção da lógica **CMOS**
 - Resistores são substituídos pelo transistores

* Transistor de Efeito de Campo

História

•Zuse Z3 – First computer* (1941)

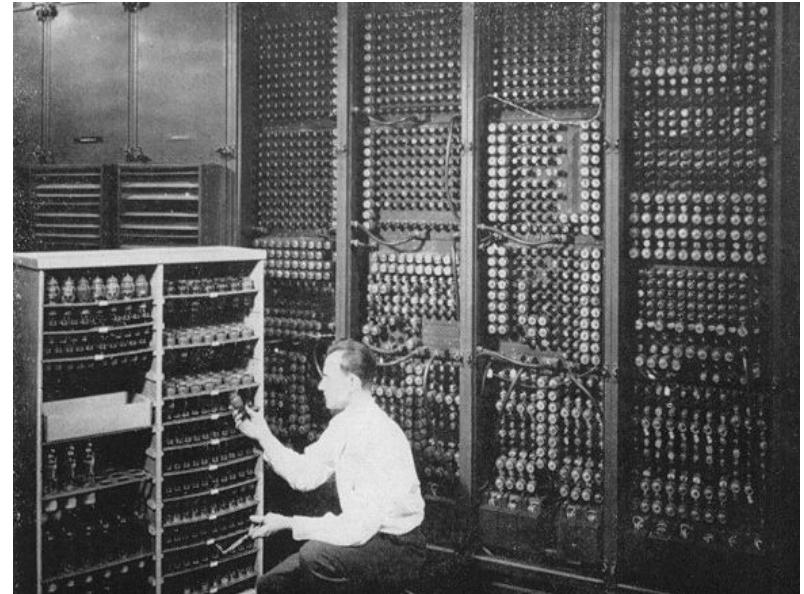
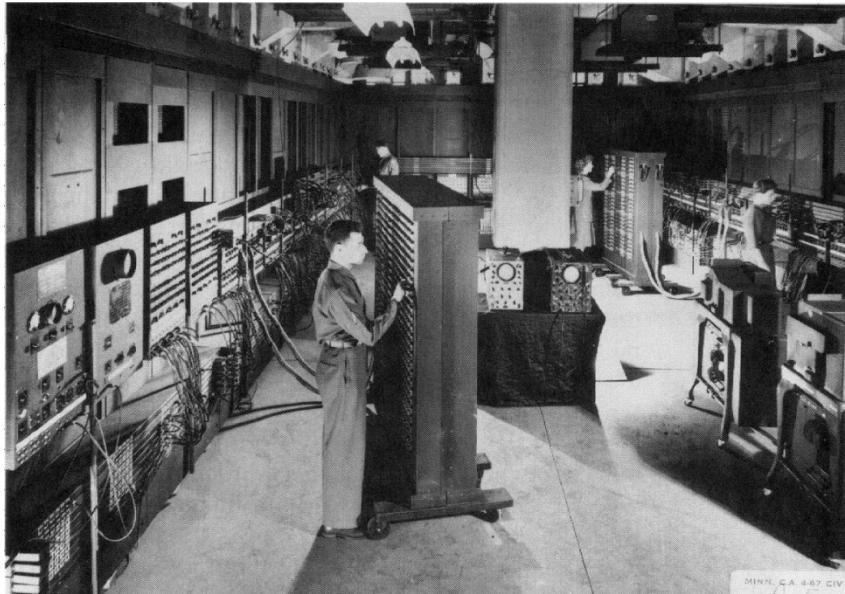


- .Primeira máquina de computação que foi programável e completamente automatizada
- .2.000 relés
- .Frequência de relógio: 5,3 Hz
- .Tamanho dos words: 22 bits
- .Programado através de cartões perfurados
- .Adição, multiplicação, divisão, raiz quadrada

•* Eleito na “1st International Conference on the History of Computing” em Paderborn, Germany, 1998

História

- ENIAC – Primeiro computador eletrônico (1946)



.Integrador numérico e computador

.Na “Moore School of Electrical Engineering”, Universidade de Pensilvânia

.17.468 válvulas termiônicas, 7.200 diodos (+ ≈80k resistores & capacitores)

.5 Milhões ligações soldadas a mão

História

- 5 megabytes HD da IBM – Pesava mais de 1 tonelada (1956)

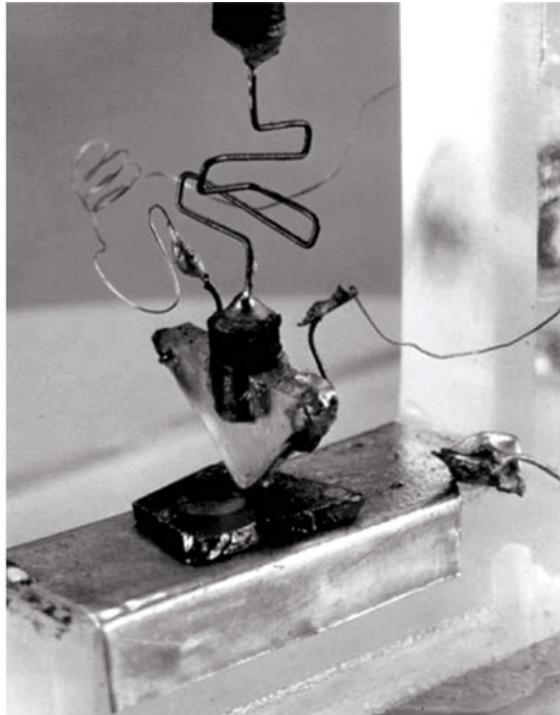


História

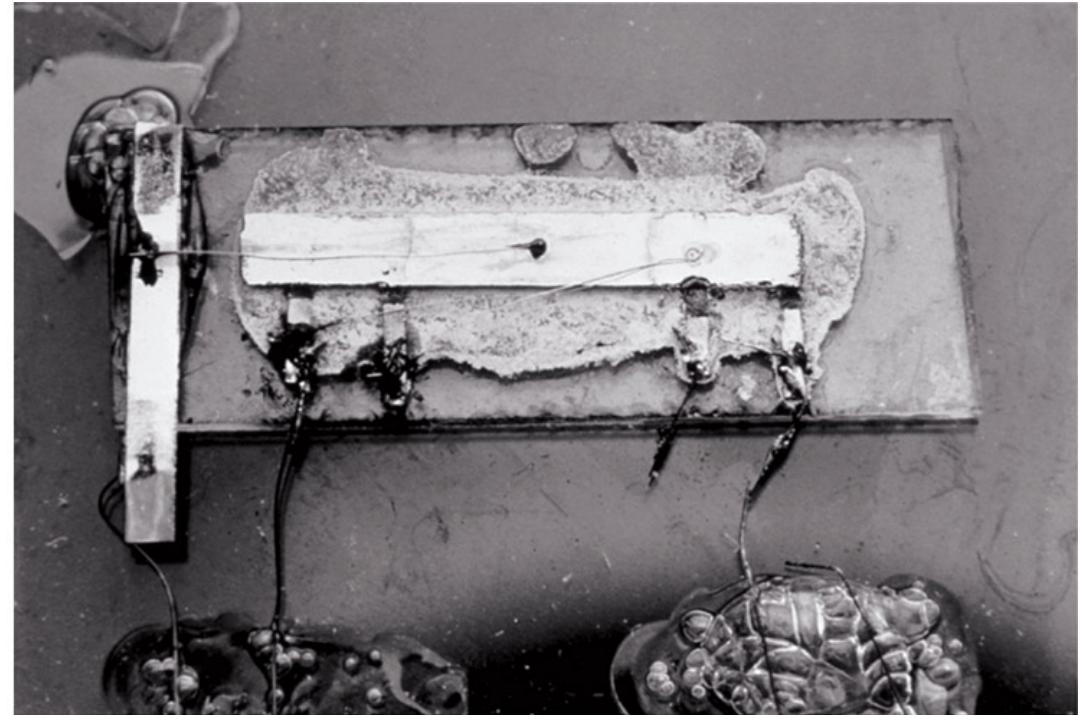
- Válvulas termiônicas no ENIAC



História



(a)



(b)

a) Primeiro transistor (1947, Bardeen & Brattain, Bell labs)

b) Primeiro circuito integrado (1958, Kilby, AT&T)

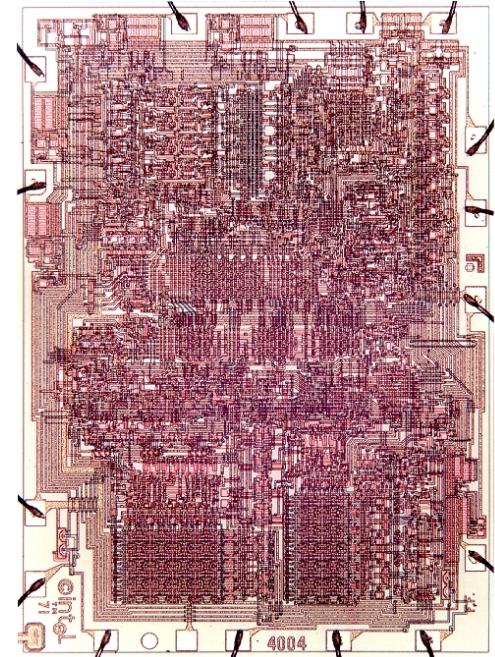
História

.1969:

- BUSICON (Japão) pediu 12 chips (funções diferentes, personalizados) para uma certa empresa chamada INTEL
- A INTEL na ocasião, não tinha recursos suficientes para esses chips → decisão de fazer um só chip com todas as funções desejadas

.1971:

- A INTEL fabrica o primeiro processador: o 4004 (4 Bit, *add e sub*)
- 4004: Primeiro chip que unifica várias funções
- 4004: Dispositivo básico das primeiras calculadoras eletrônicas portáteis
- 4004: Sem muito uso (pelas suas próprias limitações) mas foi um marco importante na história dos processadores

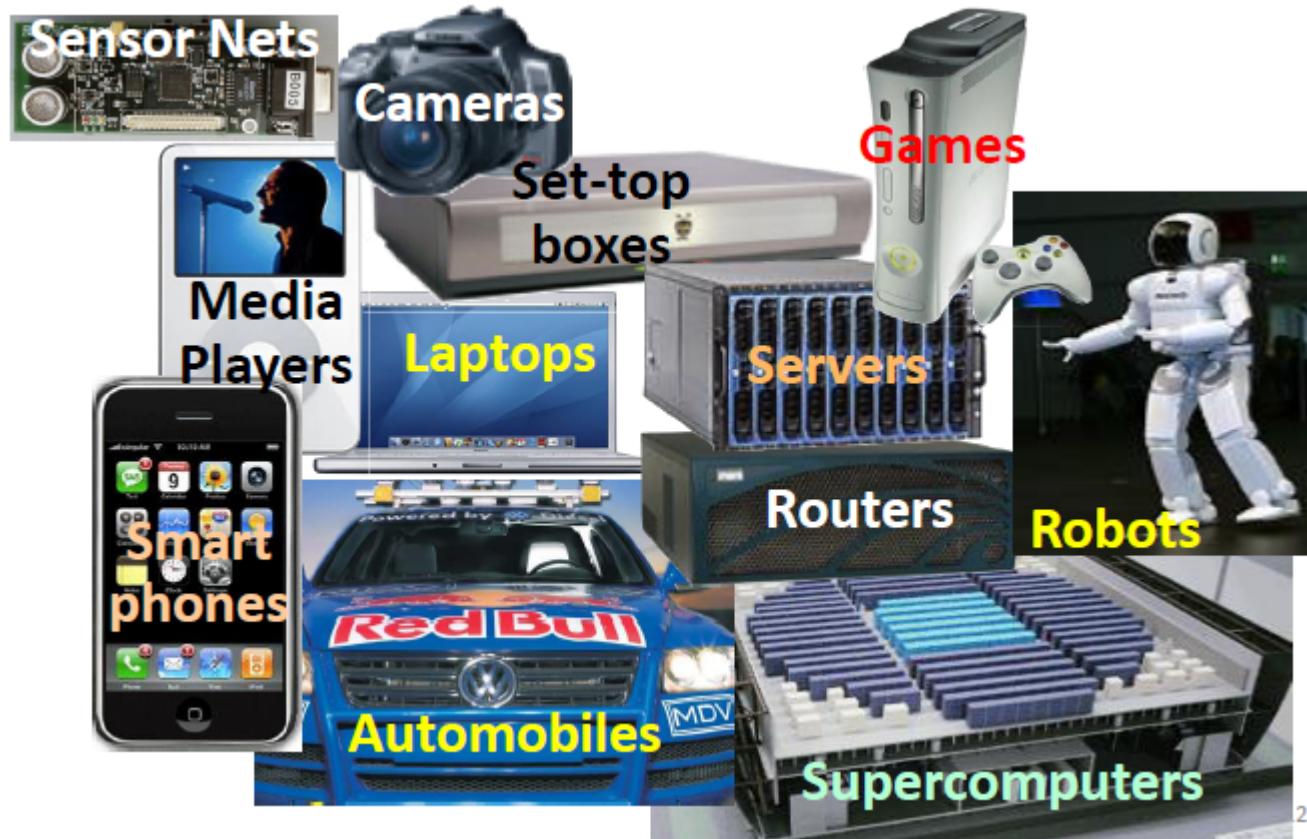


www.MyNikko.com 2006

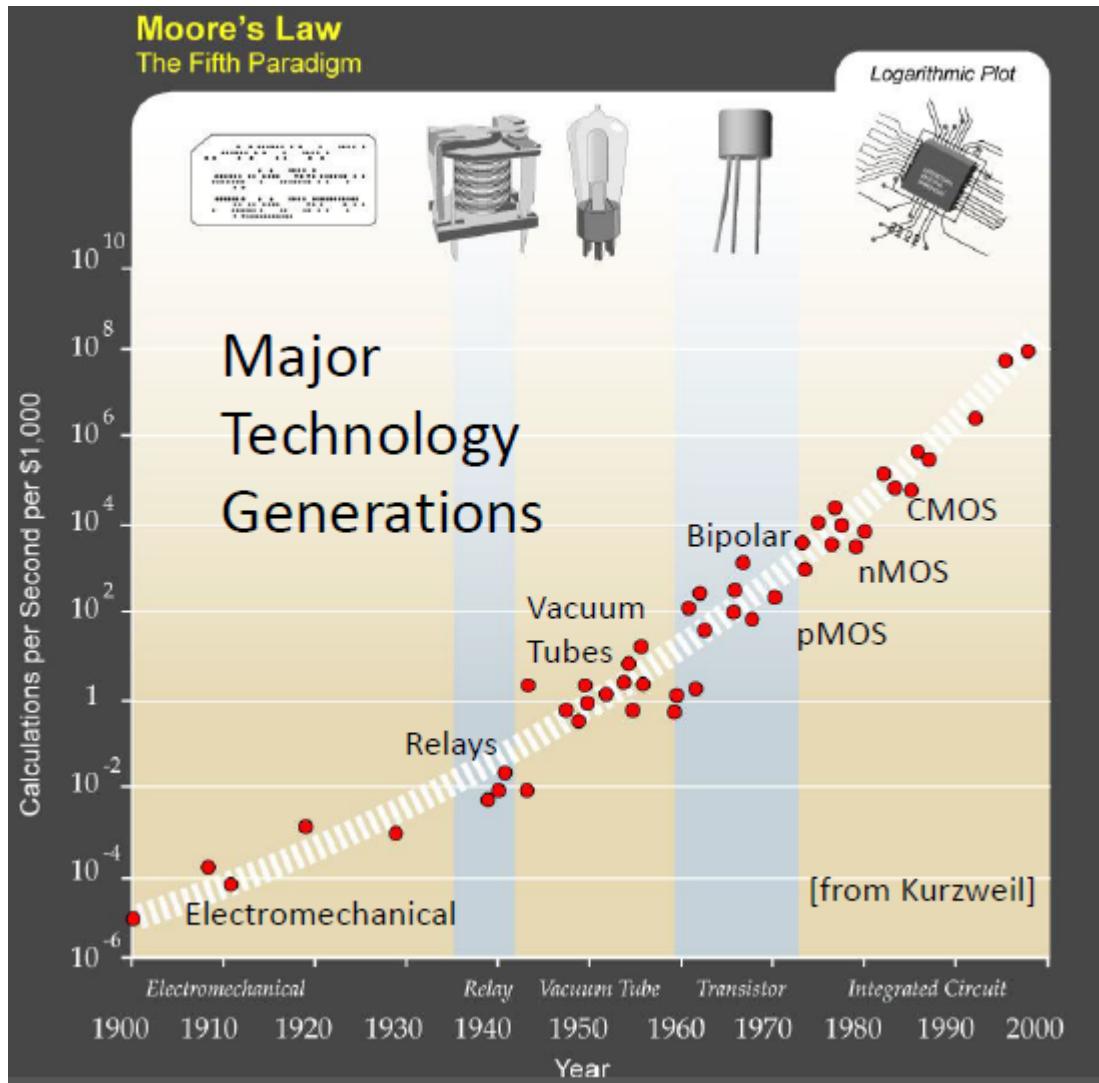
•INTEL© - 4004

História

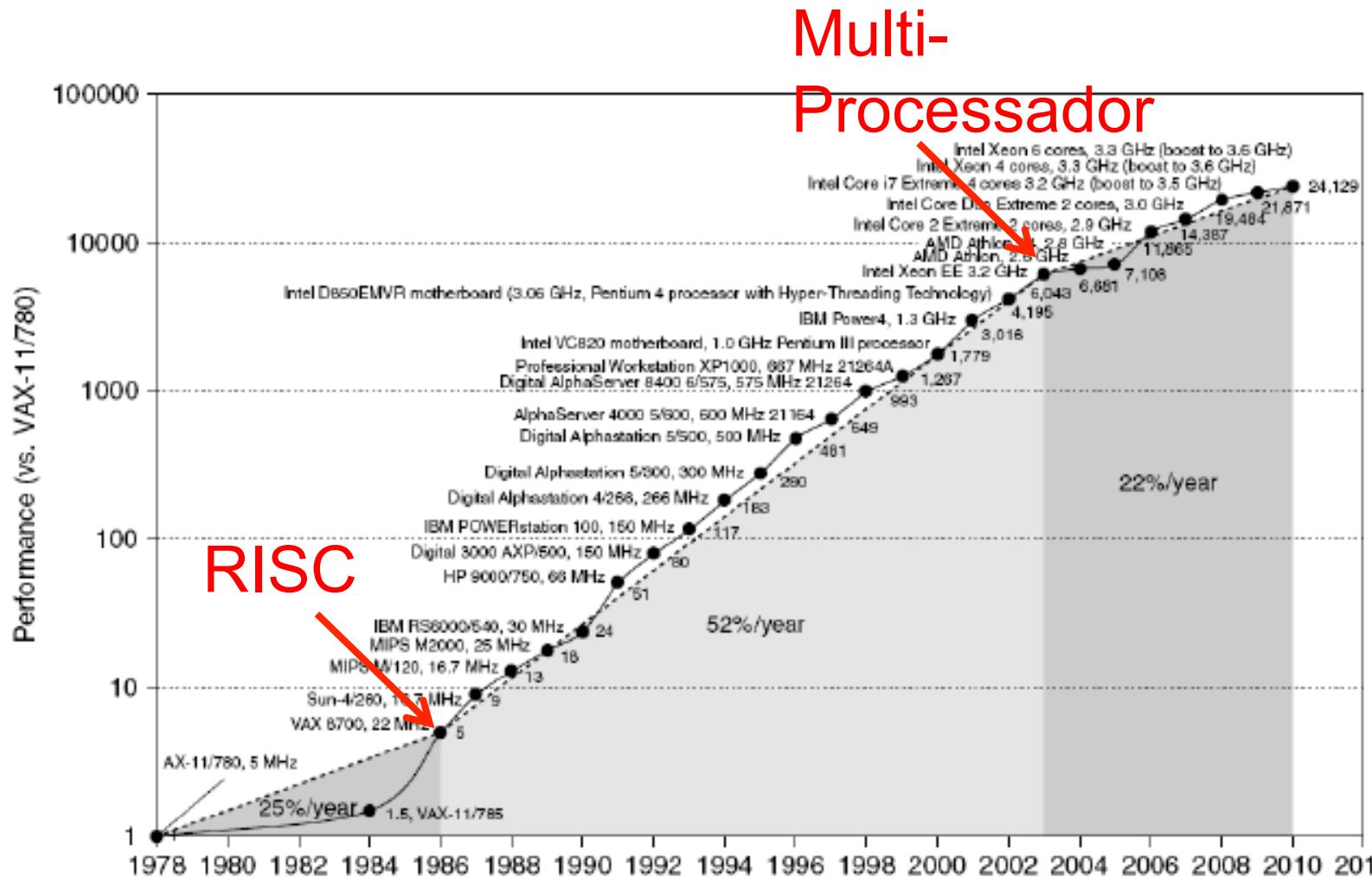
Computadores Atuais



Lei de Moore



Desempenho do Processador Sequencial



Processadores

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

Processadores



The stories of the greatest and most influential microchips in history—and the people who built them

<http://spectrum.ieee.org/static/chip-hall-of-fame>

Arquitetura x Organização

■ **Arquitetura:** atributos do computador visíveis ao programador

- Conjunto de instruções;
- Número de bits utilizados para representação de dados;
- Mecanismos de E/S;
- Técnicas de endereçamento.

■ Ex. **Existe** uma instrução de multiplicação?

Arquitetura x Organização

- **Organização:** como atributos são implementados
 - Sinais de controle;
 - Interfaces com periféricos;
 - Tecnologia de memória.
- Ex. A instrução de multiplicação é **implementada** por um hardware dedicado ou por repetição de soma?

Arquitetura x Organização

- Arquitetura única provê compatibilidade de código
- Organização difere em cada implementação
- Arquitetura básica do **x86 da Intel** compartilha mesma arquitetura básica
- Família **IBM 370** compartilha mesma arquitetura básica
- Migração dos Mac's de **680x0** para **Power PC** teve fase de transição de emulação de código 680x0 em hardware e software

Mesma Arquitetura / Diferente Organização

AMD Phenom X4

- X86 Instruction Set
- Quad Core
- 125W
- Decode 3 Instructions/Cycle/Core
- 64KB L1 I Cache, 64KB L1 D Cache
- 512KB L2 Cache
- Out-of-order
- 2.6GHz

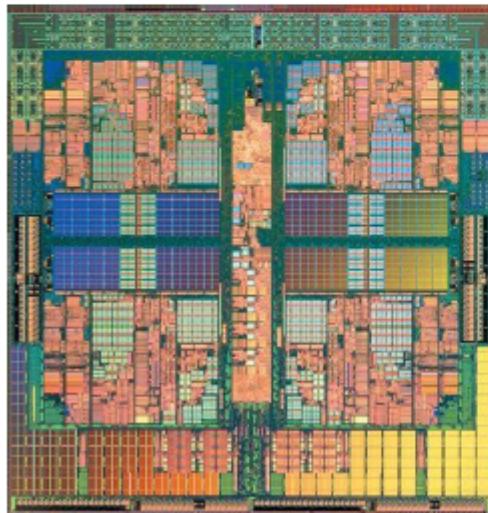


Image Credit: AMD

Intel Atom

- X86 Instruction Set
- Single Core
- 2W
- Decode 2 Instructions/Cycle/Core
- 32KB L1 I Cache, 24KB L1 D Cache
- 512KB L2 Cache
- In-order
- 1.6GHz

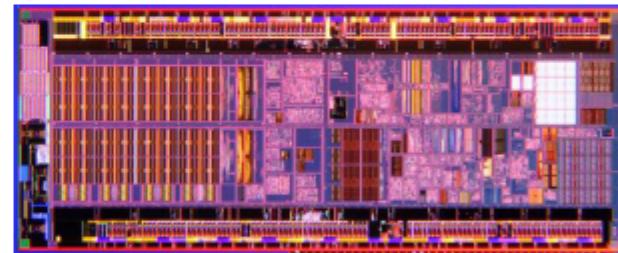


Image Credit: Intel

Diferente Arquitetura / Diferente Organização

AMD Phenom X4

- X86 Instruction Set
- Quad Core
- 125W
- Decode 3 Instructions/Cycle/Core
- 64KB L1 I Cache, 64KB L1 D Cache
- 512KB L2 Cache
- Out-of-order
- 2.6GHz

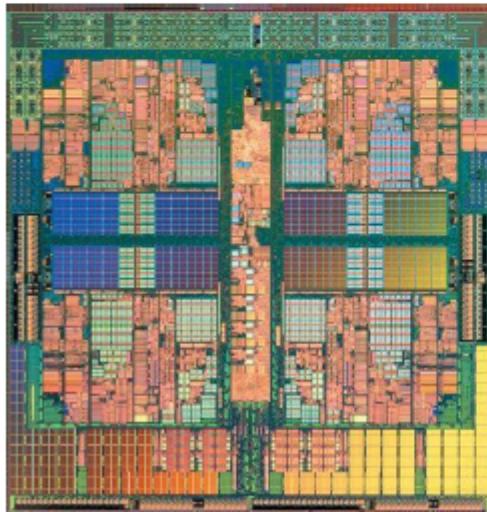


Image Credit: AMD

IBM POWER7

- Power Instruction Set
- Eight Core
- 200W
- Decode 6 Instructions/Cycle/Core
- 32KB L1 I Cache, 32KB L1 D Cache
- 256KB L2 Cache
- Out-of-order
- 4.25GHz

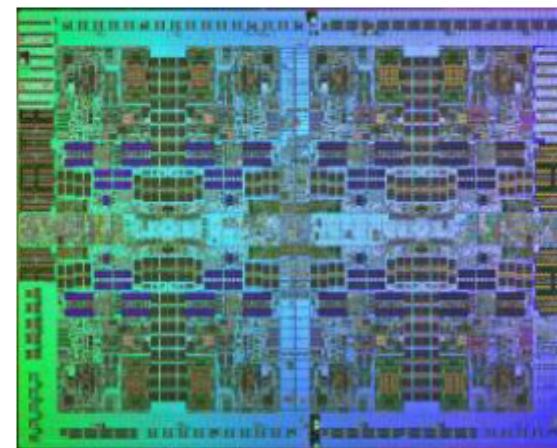


Image Credit: IBM
Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

Tipos de Máquinas

- Dispositivo Pessoal Móvel (PMD)
- Desktop / Laptops
- Servidores
- Clusters/escala wharehouse
- Sistemas embarcados

Tipos de Máquinas

■ Área de aplicação

- Propósito específico (e.g., aceleradores) / propósito genérico
- Científico (intenso em FP) / Comercial
- Computação embutida

■ Nível de compatibilidade de Software

- Compatibilidade de código objeto/binário (custo HW vs. SW, x86)
- Linguagem de máquina (modificações no código objeto/binário são possíveis no projeto da arquitetura)
- Linguagens de programação (por que não?)

Tipos de Máquinas

■ Requisitos do sistema operacional

- Tamanho do espaço de endereçamento (Address Space)
- Gerenciamento de memória e proteção
- Trocas de contexto
- Interrupções e Traps

■ Padrões: inovação vs. competição

- Ponto flutuante (IEEE 754)
- Barramentos de I/O (PCI, SCSI, PCMCIA)
- Sistemas operacionais (UNIX, iOS, Windows)
- Redes (Ethernet, Infiniband)
- Sistemas operacionais / Linguagens de programação ...

Metodologia de Projeto

Como melhorar o desempenho de computadores?

O que devemos priorizar?

Metodologia de Projeto

Princípio Básico

Torne rápido o mais comum !!!

Favoreça o mais frequente em
relação ao caso pouco
frequente !!!

Metodologia de Projeto

Lei de Amdahl

Speedup devido à melhoria E:

$$\text{Speedup}(E) = \frac{\text{ExTime sem E}}{\text{ExTime com E}} = \frac{\text{Desempenho com E}}{\text{Desempenho sem E}}$$



Suponha que melhoria E acelere porção F da tarefa por fator S, e que restante da tarefa permanece sem alteração, então

$$\text{ExTime}(E) =$$

$$\text{Speedup}(E) =$$

Metodologia de Projeto

Lei de Amdahl

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

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

Em última análise, desempenho de qualquer sistema será limitada por porção que não é melhorada...

Métricas de Desempenho

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa			
Compilador			
Conj. Instrs.			
Organização			
Tecnologia			

Métricas de Desempenho

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador			
Conj. Instrs.			
Organização			
Tecnologia			

Métricas de Desempenho

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador	X	X	
Conj. Instrs.			
Organização			
Tecnologia			

Métricas de Desempenho

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador	X	X	
Conj. Instrs.	X	X	
Organização			
Tecnologia			

Métricas de Desempenho

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador	X	X	
Conj. Instrs.	X	X	
Organização	X	X	
Tecnologia			

Métricas de Desempenho

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador	X	X	
Conj. Instrs.	X	X	
Organização	X	X	
Tecnologia			X

SPEC

SPEC2006 benchmark description	Benchmark name by SPEC generation			
	SPEC2006	SPEC2000	SPEC95	SPEC92
GNU C compiler				gcc
Interpreted string processing		perl		espresso
Combinatorial optimization	mcf			ll
Block-sorting compression	bzip2		compress	eqntott
Go game (AI)	go	vortex	go	
Video compression	h264avc	gzip	jpeg	
Games/path finding	astar	eon	m88kslm	
Search gene sequence	hmmer	twolf		
Quantum computer simulation	lbquantum	vortex		
Discrete event simulation library	omnetpp	vpr		
Chess game (AI)	sjeng	crafty		
XML parsing	xalancbmk	parser		
CFD/blast waves	bwaves			fpppp
Numerical relativity	cactusADM			tomcatv
Finite element code	calculix			doduc
Differential equation solver framework	dealII			nasa7
Quantum chemistry	gamess			splice
EM solver (freq/time domain)	GemsFDTD			matrix300
Scalable molecular dynamics (~NAMD)	gromacs		swim	
Lattice Boltzman method (fluid/air flow)	lbm		hydro2d	
Large edde simulation/turbulent CFD	LESile3d		su2cor	
Lattice quantum chromodynamics	mlc	wupwlse	wave5	
Molecular dynamics	namd	apply		
Image ray tracing	povray	galgel		
Spare linear algebra	soplex	mesa		
Speech recognition	sphinx3	art		
Quantum chemlstry/object oriented	tonto	equake		
Weather research and forecasting	wrf	facerec		
Magneto hydrodynamics (astrophysics)	zeusmp	ammp		
		lucas		
		fma3d		
		slxtrack		

RISC vs. CISC

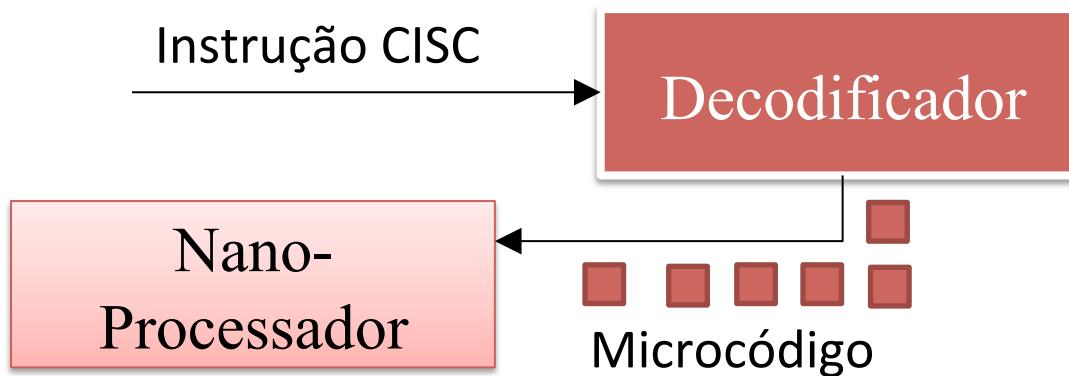
CISC

Complex Instruction Set Computer (Computador com um Conjunto Complexo de Instruções)

Caracterizam-se por:

- Conjunto alargado de instruções
- Instruções complexas
- Instruções altamente especializadas
- Existência de vários formatos de instruções
- Suporte de vários modos de endereçamento
- Suporte para operandos em memória
- Baseado na microprogramação
- Exemplos: 80x86 de Intel, 680x0 de Motorola

CISC - Microprogramação



- Cada instrução CISC separado em: **instrução de máquina**, **tipo de endereçamento** e **endereços, registradores**
- Seguinte: Envio de instruções pequenas (**microcódigo**) para **Nano-processador** (processador no processador)
- Execução de uma instrução CISC demora **vários ciclos** de clock

CISC – Vantagens

- Programação de código de máquina mais fácil
- Código executável pequeno → menos memória necessário
- Instruções memória-à-memória (carregar e armazenar dados com mesma instrução) → menos registradores necessários

**Ótimo para os primeiros computadores
(memória, registradores caros)**

CISC – Desvantagens

- Aumento de **complexidade** de **processadores novos** por causa da inclusão das instruções velhas
- Muitas **instruções especiais** - **menos usadas**
- Execução de varias instruções complexas mais lento do que execução da sequencia equivalente
- **Alta complexidade**
- *Pipelining* muito difícil → frequência de clock reduzido
- Tratamento de eventos externos (*Interrupts*) mais difícil
- Execução de instruções simples demora mais do que necessário

Menos aplicável para computadores atuais

RISC

Reduced Instruction Set Computer (Computador com um Conjunto Reduzido de Instruções)

- Menor quantidade de instruções (Intel 80486 com 200 instruções versus SPARC com 50 instruções)
- Instruções mais simples
- Largura fixa de cada instrução
- Cada instrução demora um ciclo de clock (ou menos)
- Exemplos: MIPS, SPARC, Apple iPhone (Samsung ARM1176JZF), Processadores novos do Intel (parcialmente)

RISC - Vantagens

- Menos transistores (área) para implementar lógica
- Instruções para acesso à memória (armazenar/ e carregar dados) são separados
- Complexidade baixa
- Menos sujeito às falhas
- Tempo de decodificação reduzido

RISC – Desvantagens

- Mais registradores necessários
- Compilação de código de máquina mais complicado
- Código mais complexo / maior

RISC vs. CISC

- CISC:
 - Redução do número de **instruções** por **programa**
 - Aumento do número de **ciclos** por **instrução**
- RISC:
 - Redução do número de **ciclos** por **instrução**
 - Aumento de número de **instruções** por **programa**

RISC vs. CISC - Hoje

Fronteiras indistintas

- Processadores **RISC** atuais **usam técnicas CISC** (por exemplo, mais instruções, instruções mais complexos)
- Processadores **CISC** atuais **usam técnicas RISC** (p.e., um ciclo de clock por instrução – ou menos, menos instruções)
- **Técnicas avançadas** (p.e., *pipelining, branch prediction*) aplicadas em processadores **RISC e CISC**
- **Outros fatores** podem ser **mais importante** (p.e., Cache)
- Mas: **Sistemas embutidos só com processadores RISC**
- Área (CISC grande demais)
- Consumo de energia / dissipação de calor