

DCC006 – Organização de Computadores I

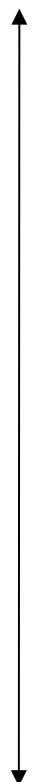
Aula 1 – Introdução

Prof. Omar Paranaiba Vilela Neto



O que é Organização/Arquitetura de Computadores?

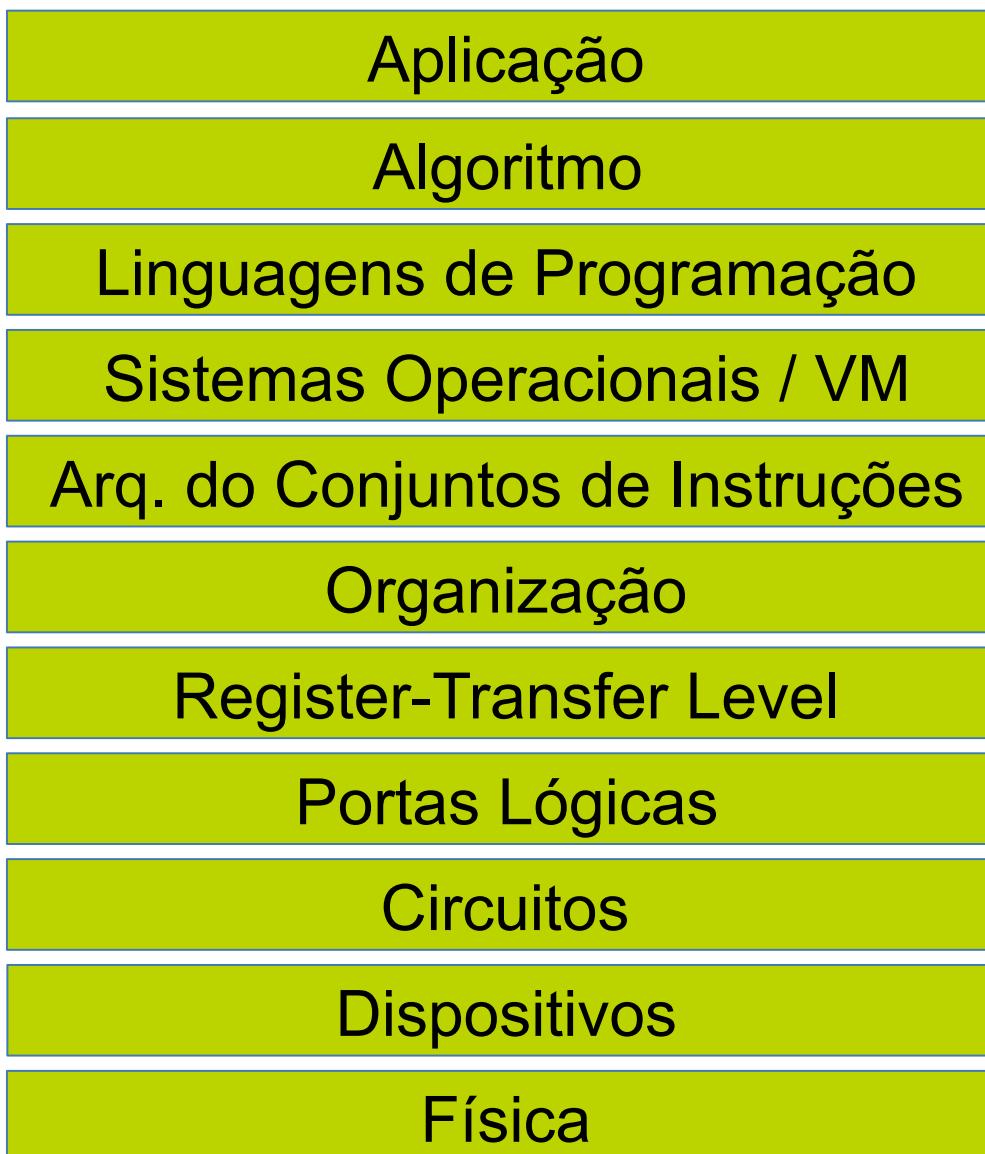
Aplicação



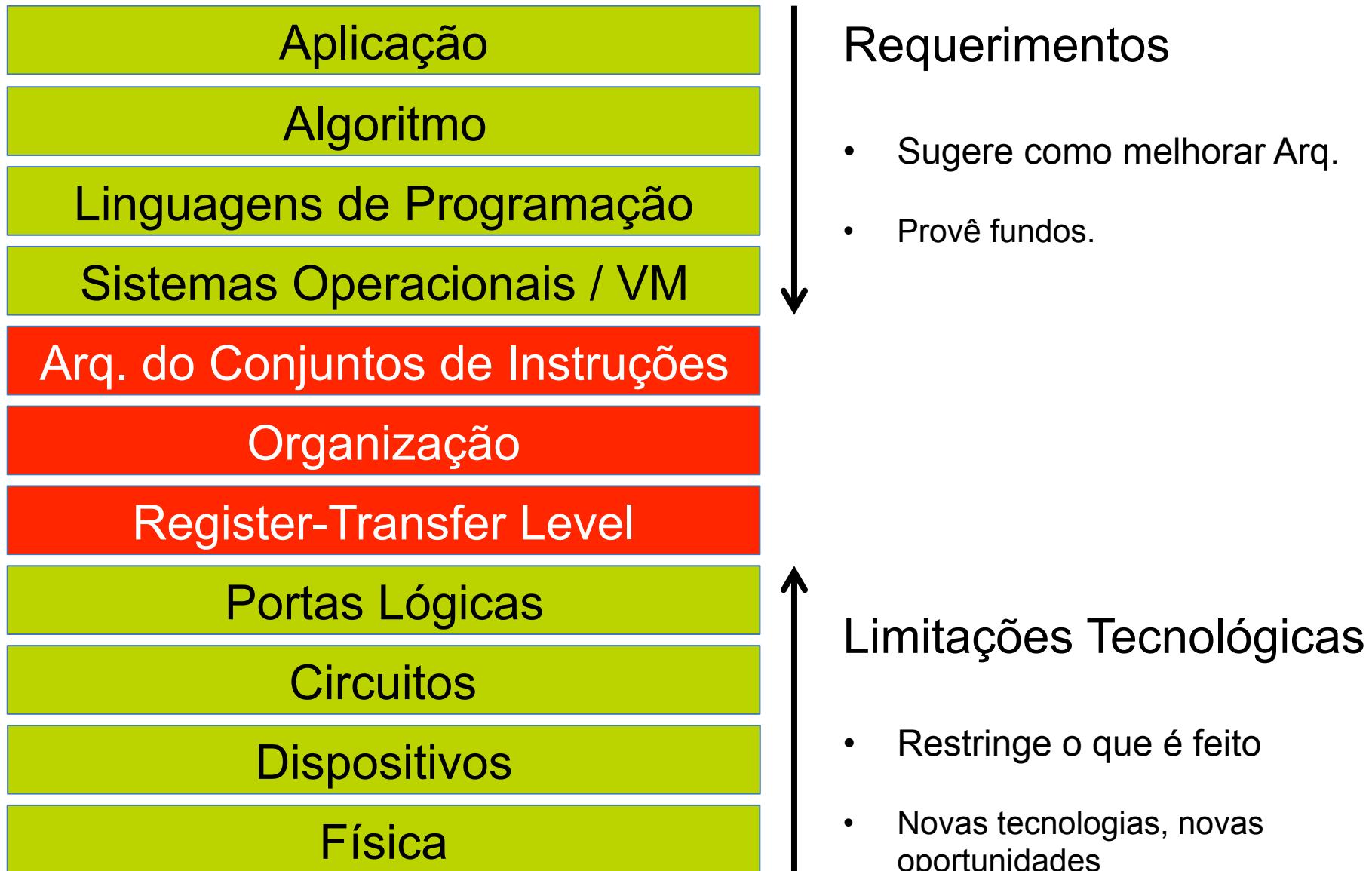
GAP

Física

O que é Organização/Arquitetura de Computadores?



O que é Organização/Arquitetura de Computadores?



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

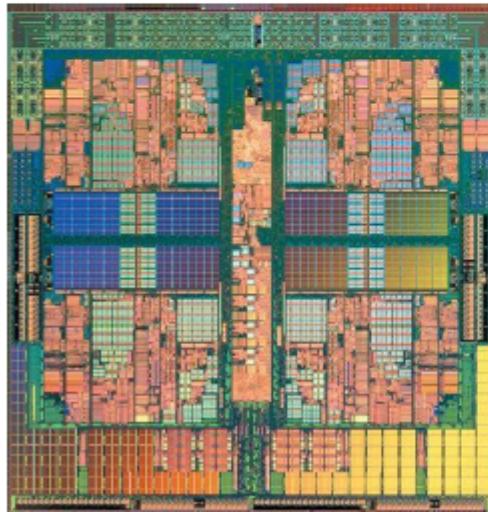


Image Credit: AMD

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

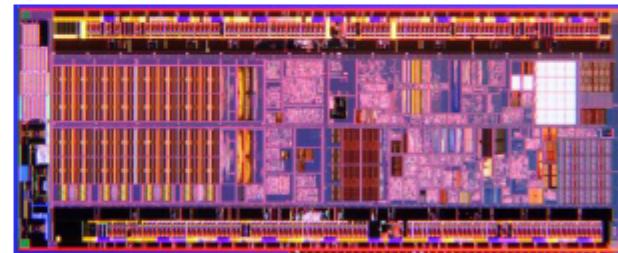


Image Credit: Intel

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

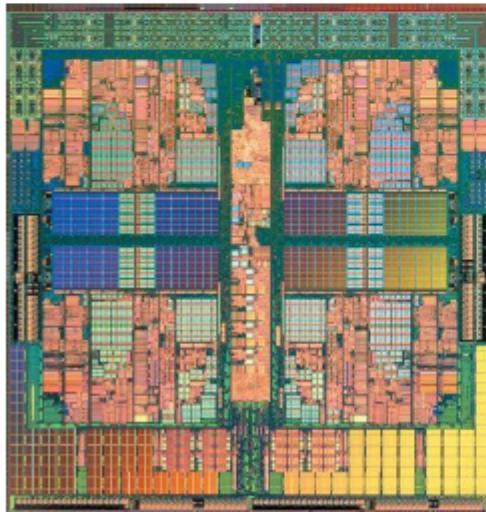


Image Credit: AMD

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

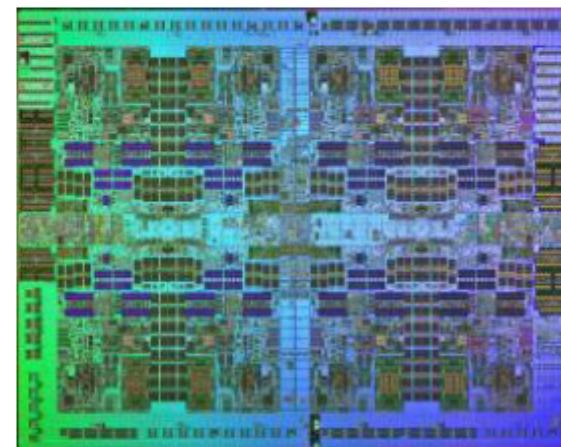


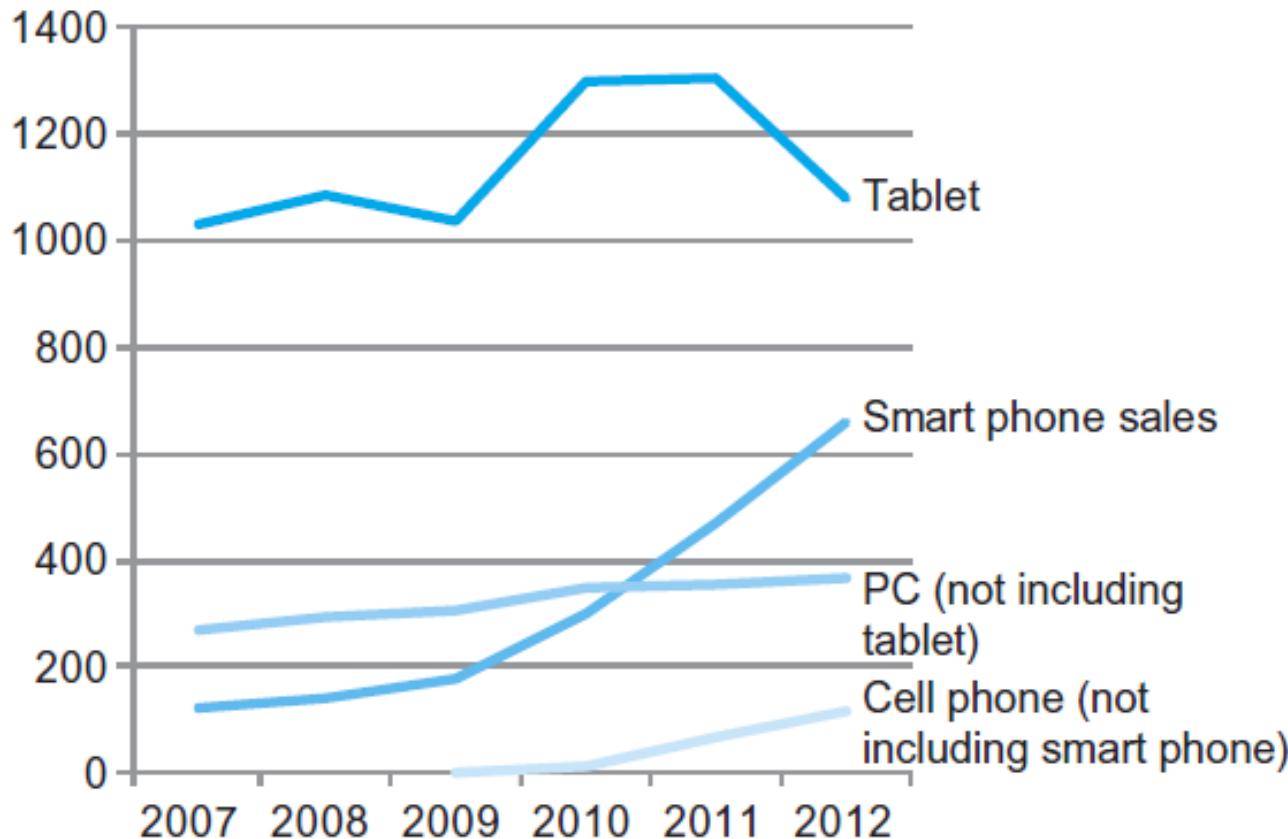
Image Credit: IBM
Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

Tipos de Máquinas

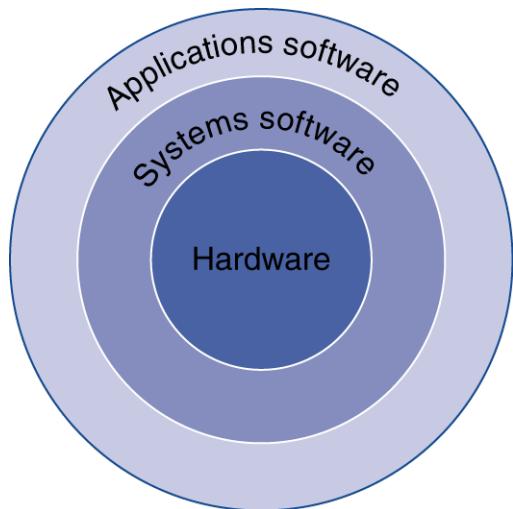
- Dispositivo Pessoal Móvel (PMD)
- Desktop / Laptops
- Servidores
- Clusters/escala wharehouse
- Sistemas embarcados

Tipos de Máquinas

A era pós-PC



O Que tem por baixo do seu programa?



Software de Aplicação

- Escrito em linguagem de alto nível

Software de Sistema

- Compilador: traduz código HLL para código de máquina
- Sistema Operacional: código de serviço
 - Manipula input/output
 - Gerencia memória e armazenamento
 - Agenda tarefas & compartilha recursos

Hardware

- Processador, memoria, controladores de I/O

Níveis de códigos de programas

Linguagem de alto nível

- Nível de abstração próximo do domínio do problema
 - Focado em produtividade e portabilidade

Linguagem Assembly

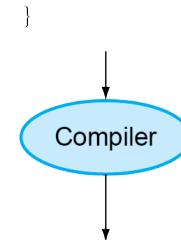
- Representação textual de instruções

Linguagem de Hardware

- Dígitos binários (bits)
 - Codifica instruções e dados

High-level
language
program
(in C)

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

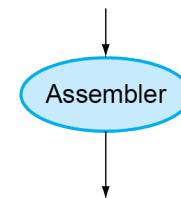


Assembly language program (for RISC-V)

```

swap:
    slli x6, x11, 3
    add  x6, x10, x6
    ld   x5, 0(x6)
    ld   x7, 8(x6)
    sd   x7, 0(x6)
    sd   x5, 8(x6)
    jalr x0, 0(x1)

```



Binary machine
language
program
(for RISC-V)

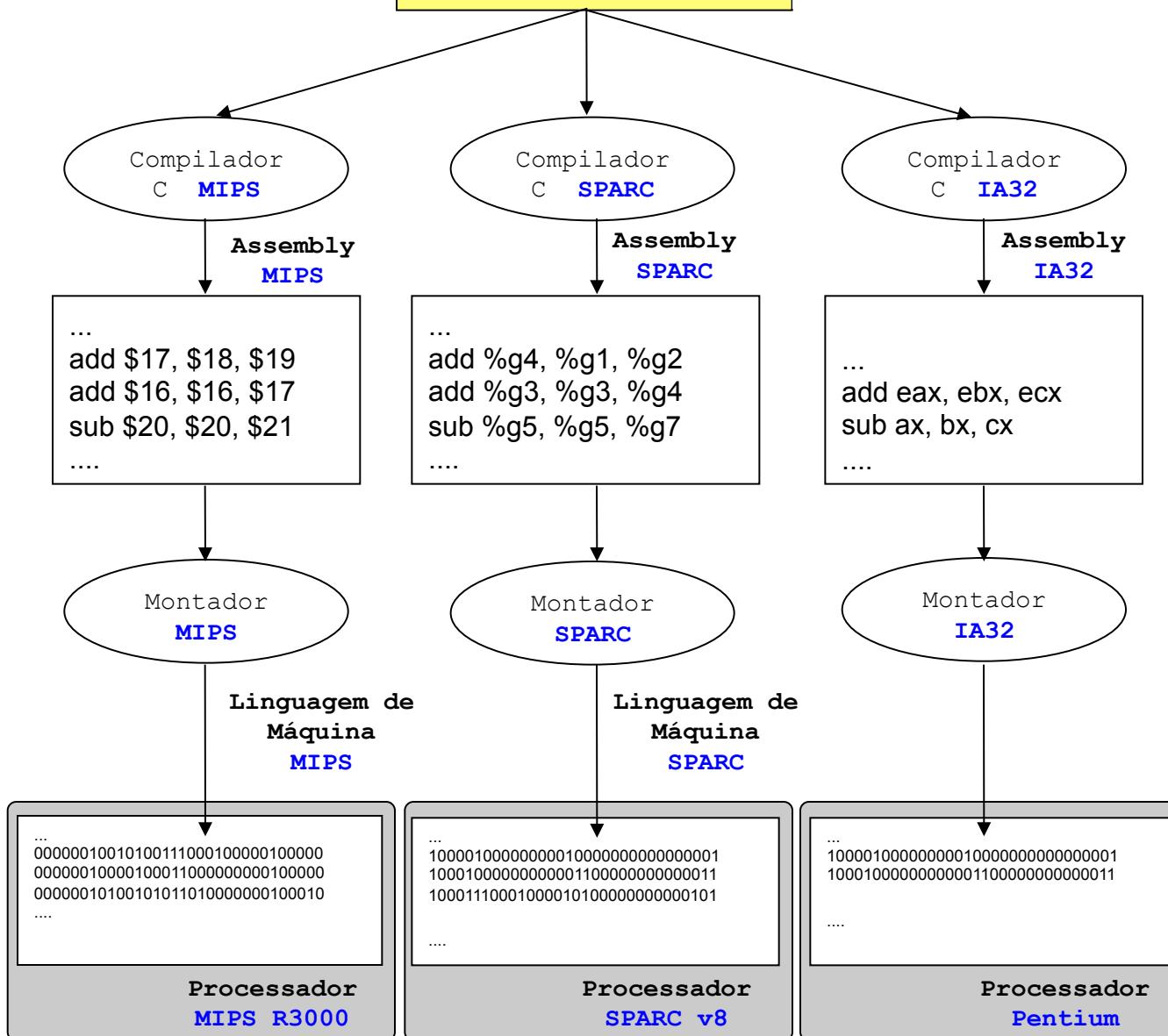
```
000000000001101011001001100010011  
00000000011001010000001100110011  
00000000000000110011001010000011  
00000000100000110011001110000011  
00000000011100110011000000100011  
00000000010100110011010000100011  
00000000000000001000000001100111
```

```

for (i=0; i<10; i++) {
    m=m+j+c;
    j=j-i;
}

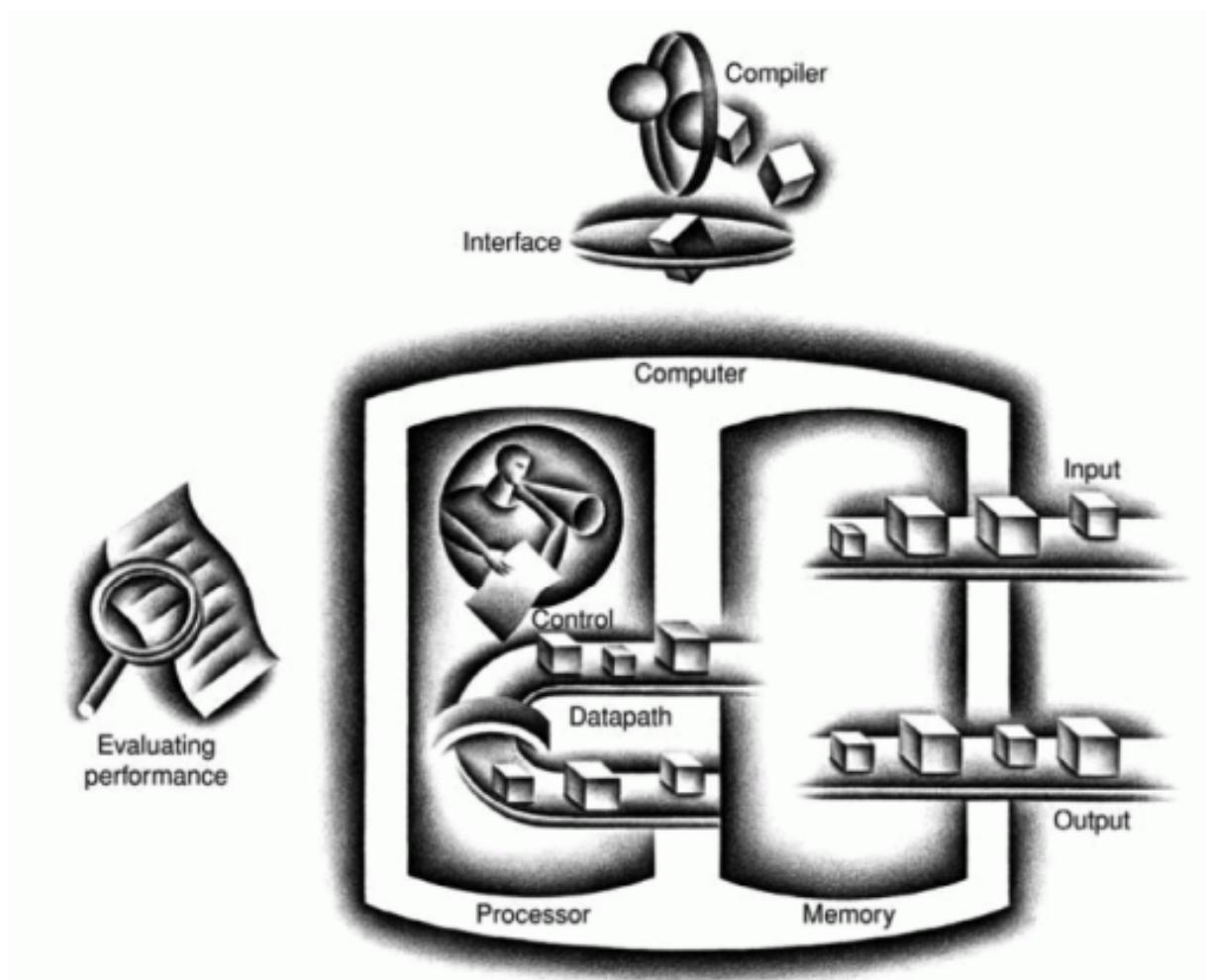
```

HLL
(Linguagem de nível alta)



Componentes de um Computador

The BIG Picture



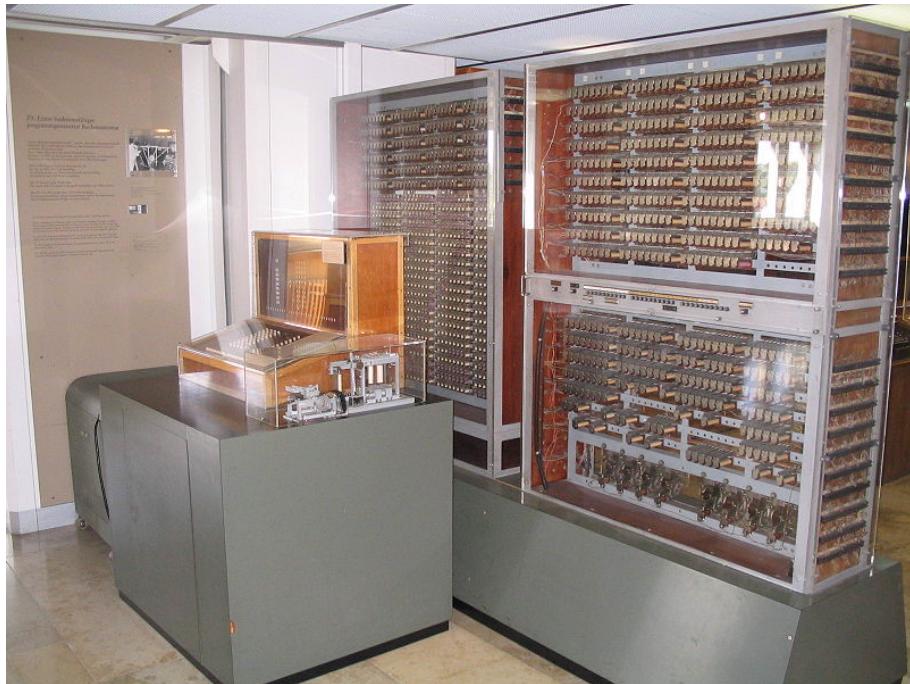
História

- .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)

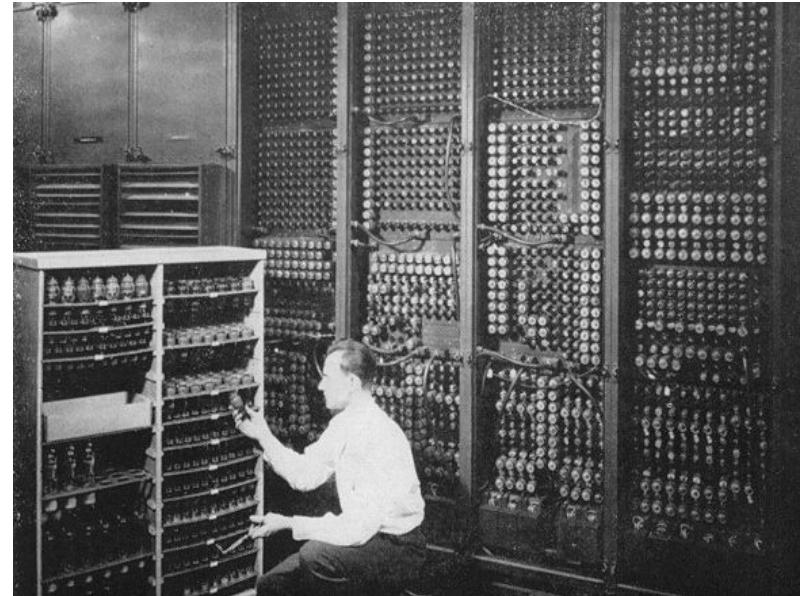
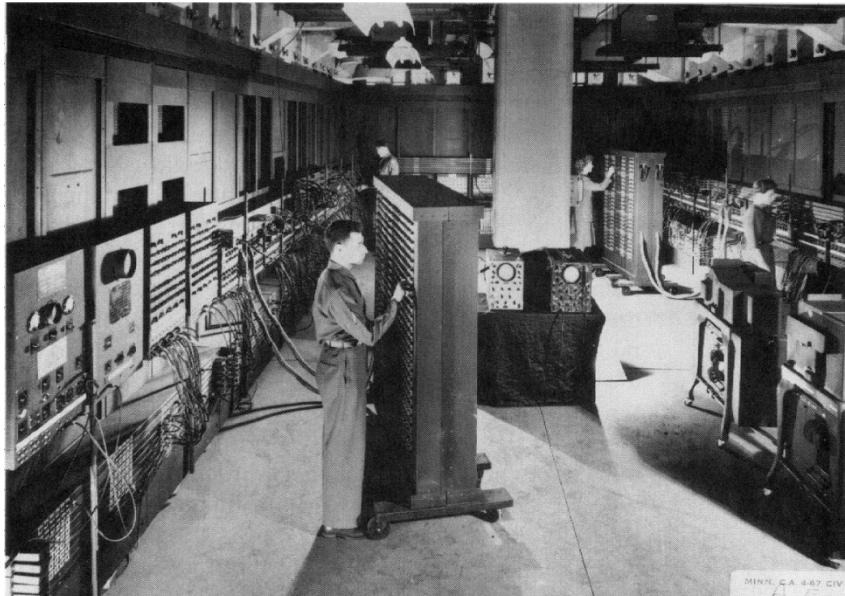


- .Primeira máquina de computação que foi programável e completamente automatizada
- .2.000 relés
- .Frequência de relógio: 5,3 Hz
- .Tamanho dos words: 22 bits
- .Programado através de cartões perfurados
- .Adição, multiplicação, divisão, raiz quadrada

•* 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)

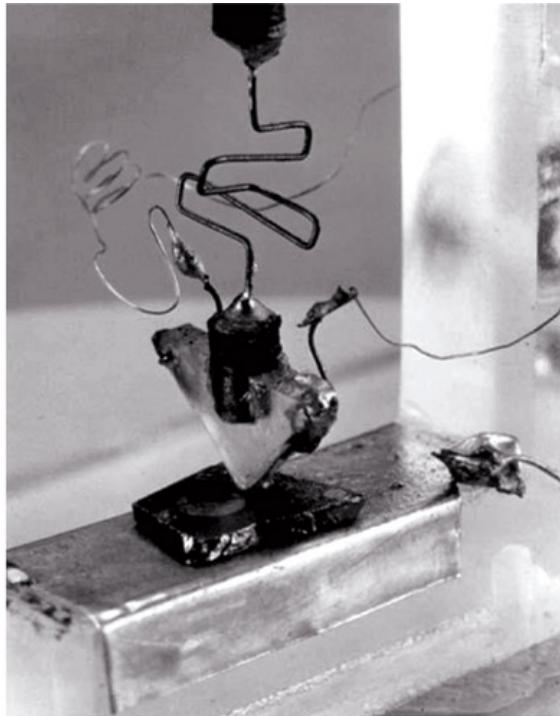


História

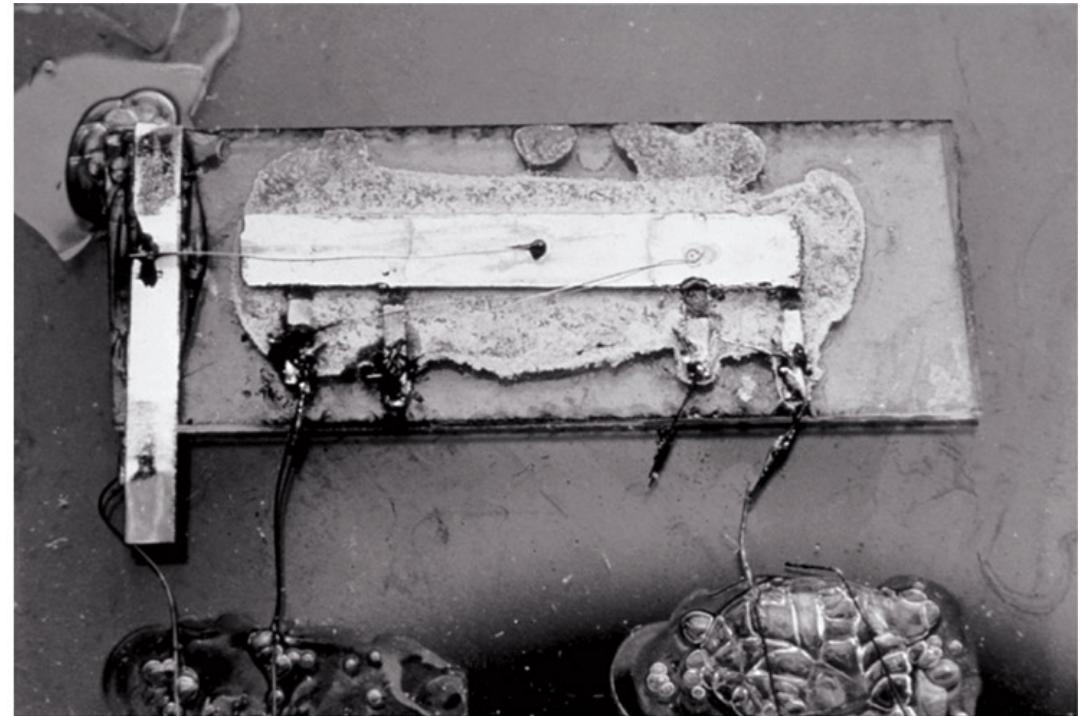
- Válvulas termiônicas no ENIAC



História



(a)



(b)

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

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

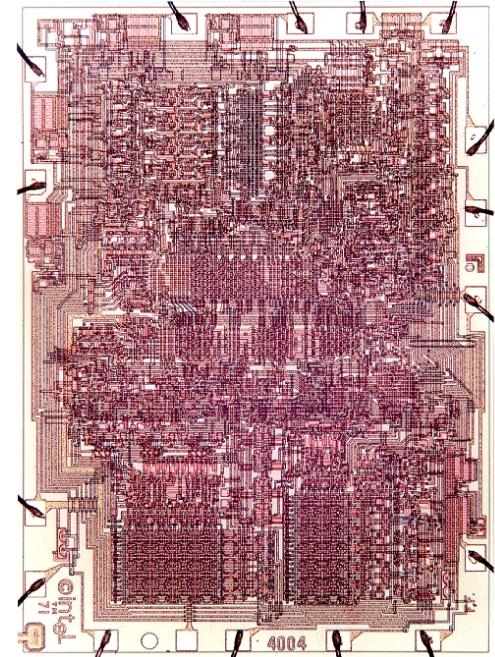
História

.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

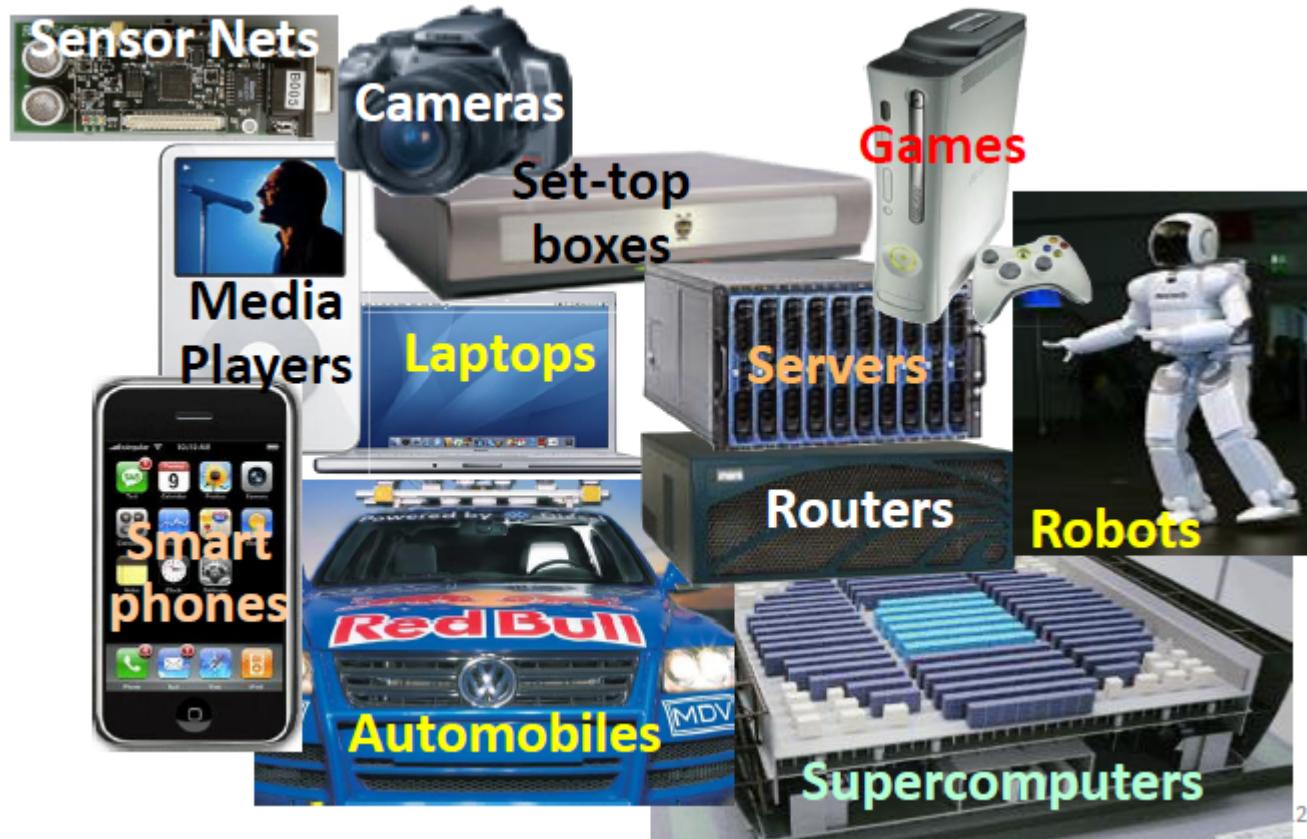


www.MyNikko.com 2006

•INTEL© - 4004

História

Computadores Atuais



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>

RISC vs. CISC

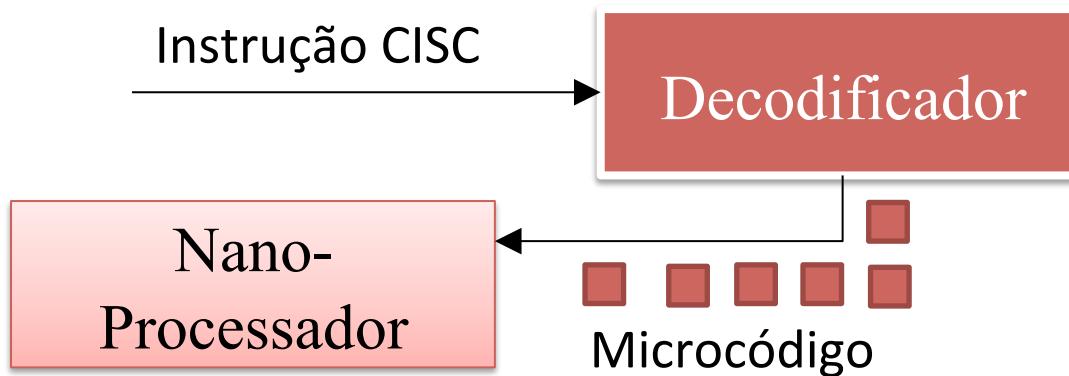
CISC

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

Informações do Curso

Tudo que vocês querem saber
na PRIMEIRA AULA !

Avaliação

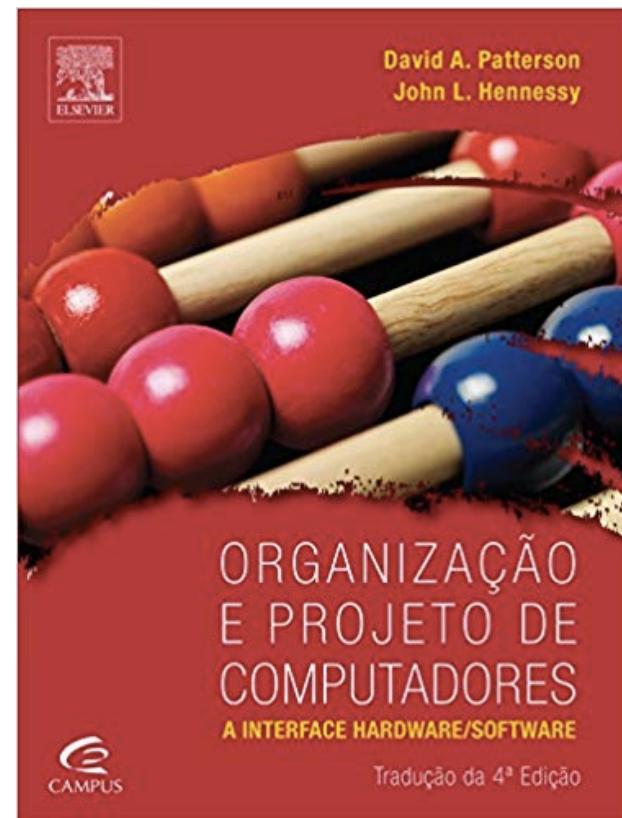
Calendário tentativa

- Prova 1: 25 pontos – 16/09/2019
 - Prova 2: 25 pontos – 30/10/2019
 - Prova 3: 25 pontos – 27/11/2019
-
- Trabalho: 25 pontos
 - Seminários
 - Trabalhos práticos
-
- Notas dos trabalhos só valerão caso nota acumulada das provas for maior que 20 pontos

Referência



Referência



Motivação

É preciso saber **como** a máquina
funciona para tirar proveito de
todo o seu **potencial**.

Bom Curso !!!