



# Arquitetura e Organização de computadores

---

ENGENHARIA DA COMPUTAÇÃO – UFC/SOBRAL

Prof. Danilo Alves  
danilo.alves@alu.ufc.br

# Apresentação

---

- Qual o objetivo do Curso?
- Para que serve?
  - Entender o hardware de um sistema computacional
  - Entender o funcionamento dos vários módulos que compõem um sistema computacional
  - Desenvolver uma visão crítica sobre os requisitos de desempenho associados a um sistema computacional
- E isso em minha vida?



# Motivação

---

- Hardware para engenharia da computação
- Funcionamento interno do computador



# Conteúdo programático

---

- Introdução a organização de computadores;
- Organização, arquitetura e componentes funcionais (hardware e software) de computadores;
- Sistemas de numeração e codificação;
- Aritmética computacional;
- Processadores, memórias e dispositivos E/S;
- Conceitos de lógica digital, chips de CPU e barramentos;
- Instruções, exemplos de microarquitetura;
- Nível de linguagem de montagem



# Formas de Ensino

---

- Aulas teóricas expositivas e práticas
- Pesquisas individuais e em grupo
- Discussões, atividades e listas de exercícios



# Avaliação

---

- 2 avaliações discursivas e objetivas
- Projeto final
- Atividades ao longo das aulas
- Listas de Exercícios
  
- $NTF = (AP1 + AP2 + PF) / 3$
- Listas ajudarão na prova
- Arredondamento  $\text{nota} \geq 6.9$  ou  $\text{nota} \geq 3.9$



# Bibliografia

---

TANENBAUM, A. S. Organização Estruturada de Computadores. Editora LTC, 5 ed, Rio de Janeiro, 2007.

STALLINGS, W. Arquitetura e Organização de Computadores. Editora Prentice Hall, 5 edição, 2002.



# Introdução

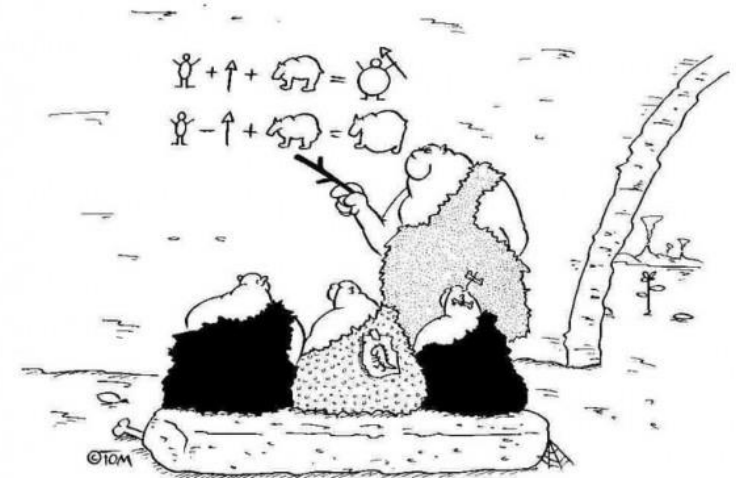
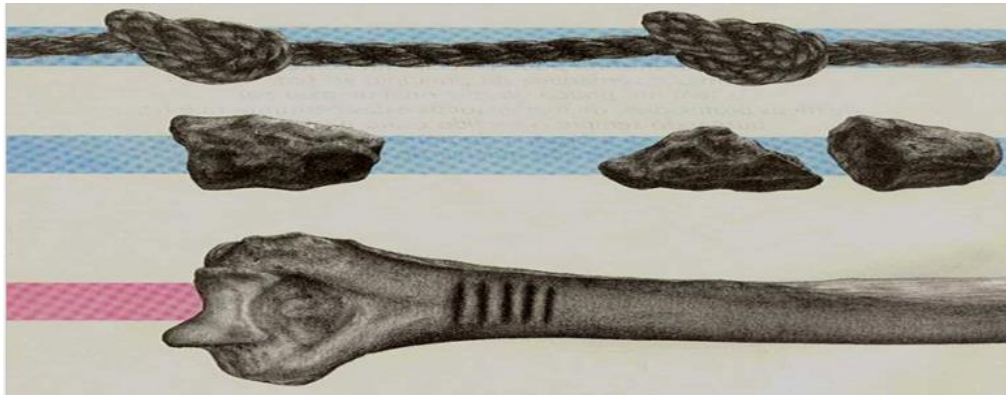
---





# Introdução

- Contar e calcular são atividades que remontam aos primórdios da humanidade
- O computador surgiu com a necessidade humana de fazer contas



# Introdução

---

- Computar é uma palavra derivada do latim (*computare*), que significa calcular
  - Computar (séc. XIX): fazer cálculos (contas) e resolver problemas com números
  - Termo moderno (1944): matéria do jornal London Times
- Computador digital



# Introdução

---

- O que é um computador digital?

“Máquina digital que pode resolver problemas para as pessoas, executando instruções que lhe são dadas”.

- Características de sistemas computacionais:

- Processamento
- Memória (armazenamento)
- Comunicação

- Ex: Celular



# Introdução

---

- Como comunicar com o computador?
- Programa
- Linguagem de máquina
- Entendimento da linguagem de máquina
- Estrutura do Computador em abstrações



# Introdução

---

- Linguagens L1 e L0
- Tradução
  - “Transcrever todo o programa na linguagem de máquina e executar o novo programa”
- Interpretação
  - “Executar cada instrução por vez, verificando a equivalência com a linguagem de máquina”
- Máquinas Virtuais



# Tradução

- Executar L1 em L0

- Primeiro substituir cada instrução nele por uma sequência equivalente de instruções em L0
- O programa resultante consiste totalmente em instruções L0
- O computador, então, executa o novo programa L0 em vez do antigo programa L1
- Todo programa em L0 é carregado em memória e é executado.
- Programa pode ser traduzido uma única vez e executado várias vezes.

```
#include <stdio.h>

int main() {
    unsigned long int n, i, a, b, c;

    printf("Informe com um numero natural entre 2 e 40: ");
    scanf("%lu", &n);

    printf("Os %lu primeiros numeros da sequencia de Fibonacci  
são:\n", n);
    a=0; b=1;
    for (i=1; i<=n; i++) {
        printf("%lu ", a);
        c=a+b; a=b; b=c;
    }
    printf("\n\n");
}
```

[illegible]

# Interpretação

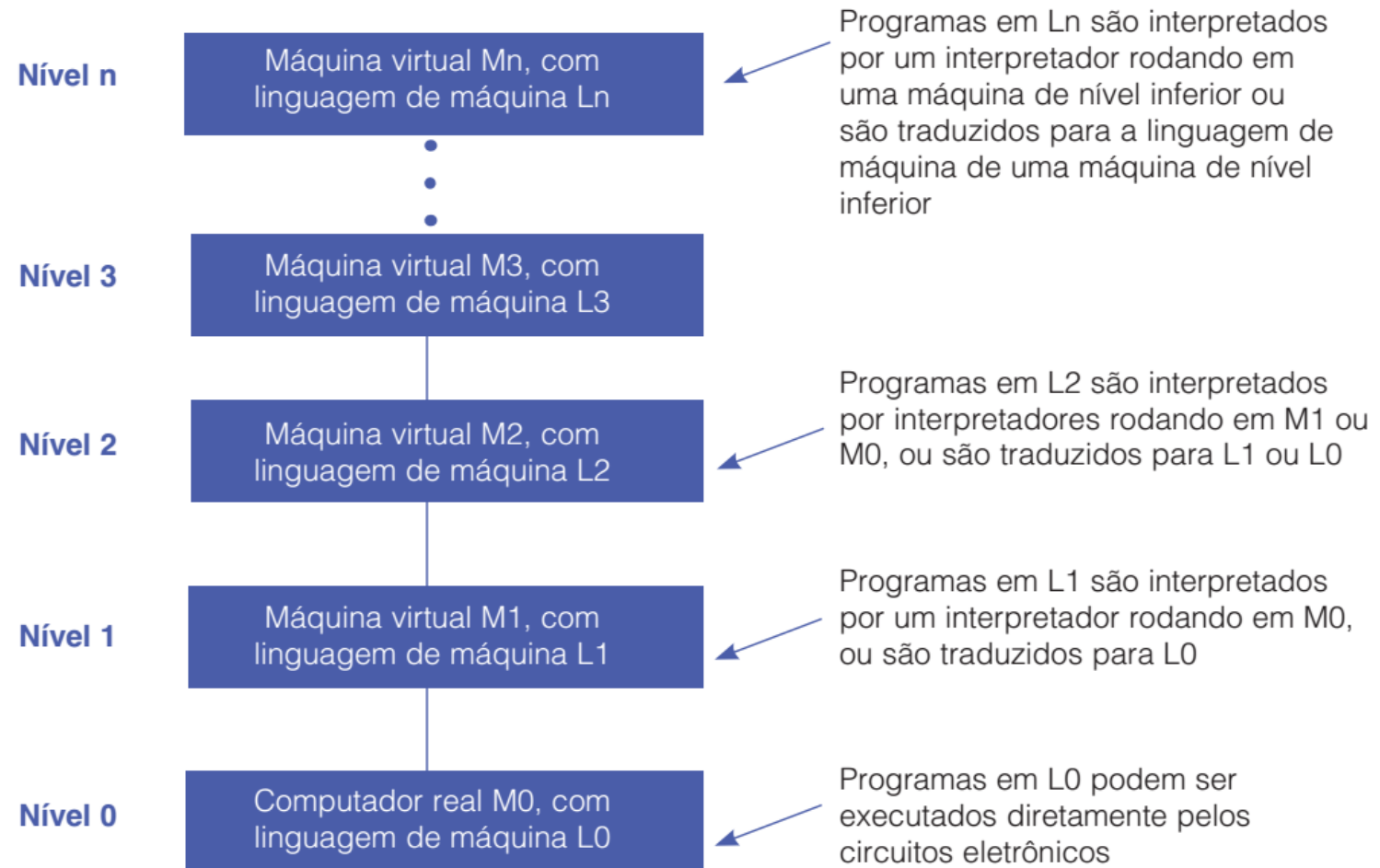
---

- Executar L1 em L0
  - Escrever um programa em L0 que considere os programas em L1 como dados de entrada
  - Examina cada instrução de entrada L1 por sua vez, executando diretamente a sequência equivalente de instruções L0
  - Cada instrução de L1, transformada para L0, é carregada na memória e executada
  - Programa deve ser novamente interpretado para ser executado.
- Essa técnica não requer que se gere um novo programa em L0



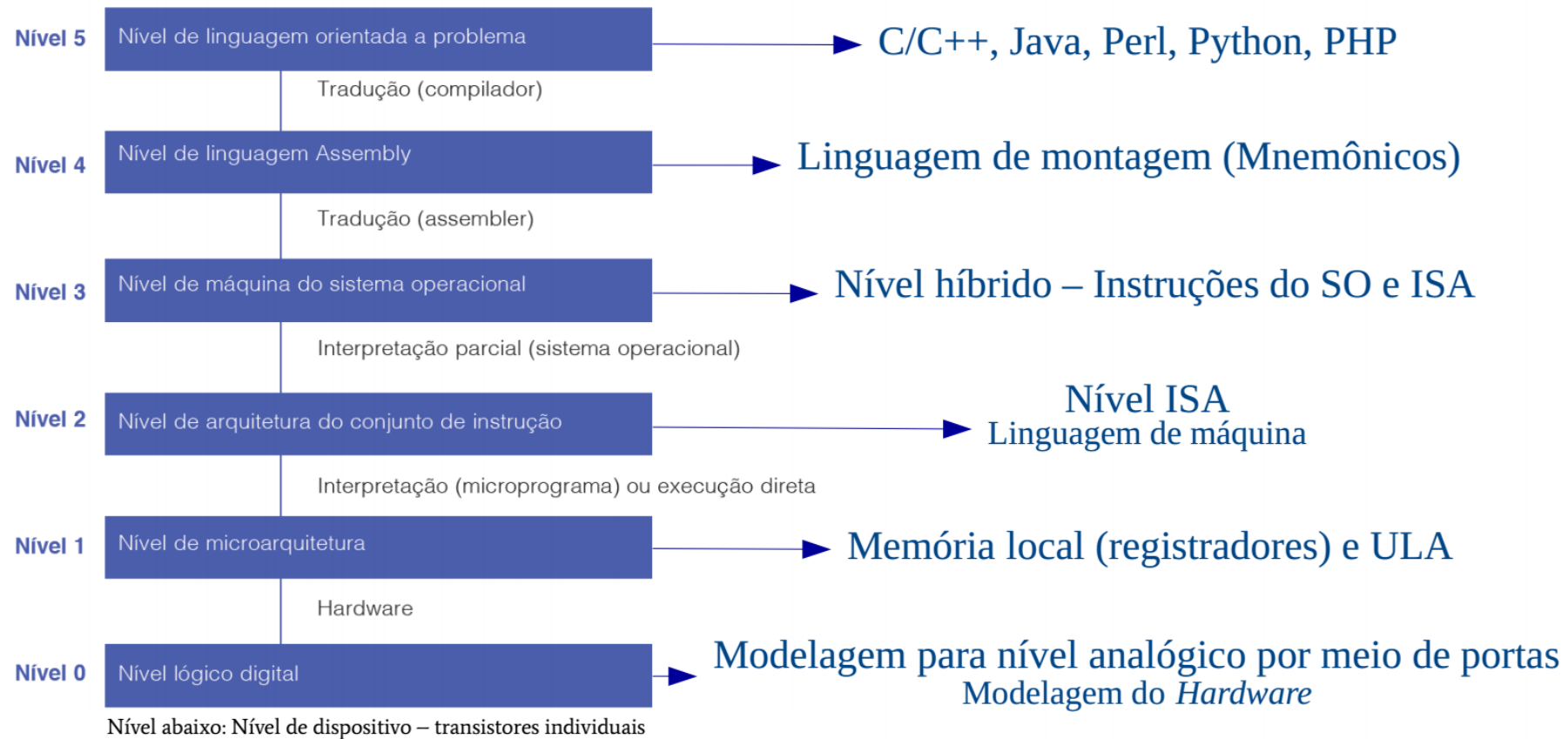
# Introdução

## ■ Máquina multinível;





# Máquinas multiníveis



# Máquinas multiníveis

---

- Nível de dispositivo:
  - Campo da engenharia elétrica, circuitos, transistores...
- Nível 0 - Lógica digital:
  - Nível mais baixo da estrutura.
  - Objetos de interesse são conhecidos como portas lógicas.
  - Cada porta lógica possui 1 ou mais entradas digitais (aceitam 0 ou 1) e calculam funções lógicas simples sobre essas entradas. Exemplos: AND, OR, XOR, ...
  - Portas lógicas são combinadas para formar memórias de um bit → registradores → o processador - principal dispositivo do computador.



# Máquinas multiníveis

---

- Nível 1 – Microarquitetura:
  - Uma memória local (8 a 32 registradores) e a ULA (Unidade Aritmética Lógica) que realiza operações aritméticas muito simples.
  - Registradores - conectados a ULA formando o caminho dos dados.
  - Operações são controladas por um microprograma ou diretamente por hardware.
  - Microprograma - interpretador para as instruções do Nível 2. Busca, decodifica e executa as instruções, uma a uma, usando o caminho de dados para a realização desta tarefa.



# Máquinas multiníveis

---

- Nível 2 - Nível da Arquitetura do Conjunto de Instruções:
  - Nível ISA - Instruction Set Architecture.
  - Definida pelo fabricante e dependente da arquitetura da máquina.
  - Fabricantes disponibilizam “Manual de Referência da Linguagem de Máquina” ou “Princípios de operação do Computador Modelo XYZW” (ou algo similar).
  - Manuais descrevem como as instruções são executadas interpretativamente pelo microprograma ou como são executadas diretamente pelo hardware.
  - Essas informações são necessárias para os desenvolvedores de sistemas operacionais.



# Máquinas multiníveis

---

- Nível 3 - Nível do Sistema Operacional:
  - Instruções da linguagem deste nível também podem conter instruções do nível ISA.
  - Suporta uma organização diferente de memória.
  - Suporta capacidade de rodar 2 ou mais programas simultaneamente.
  - Suporta sistemas de comandos ou de janelas (windows).
  - Programadores deste nível, e também dos níveis mais baixos, são conhecidos como programadores de sistema.
  - Os programadores dos níveis mais altos que este são chamados programadores de aplicação.



# Máquinas multiníveis

---

- Nível 4: Nível da linguagem do montador ou de montagem (Assembly language)
  - Linguagem de montagem: forma simbólica de representação das linguagens do nível mais baixo.
  - Programas nessa linguagem são, inicialmente, traduzidos para as linguagens dos níveis 1, 2 e 3 e depois interpretados pela máquina virtual apropriada ou pela própria máquina real.
  - Programa que realiza a tradução - montador.



# Máquinas multiníveis

---

- Nível 5 - Nível das linguagens orientadas para solução dos problemas:
  - Conhecidas como linguagens de alto nível. Exemplos: Basic, C, Pascal, Java, LISP, ....
  - Programas são geralmente traduzidos para os níveis 3 e 4 por compiladores.
  - Alguns são interpretados: Exemplo: programas em Java.



# Máquinas multiníveis

---

- Fronteira entre o hardware e o software
  - “Hardware é apenas um software petrificado [Karen Panetta]
  - Qualquer operação executada por software também pode ser embutida diretamente no hardware e qualquer operação implementada em hardware também pode ser simulada via software
- Implementação entre hardware e software
- Permite ignorar detalhes irrelevantes de níveis mais baixos

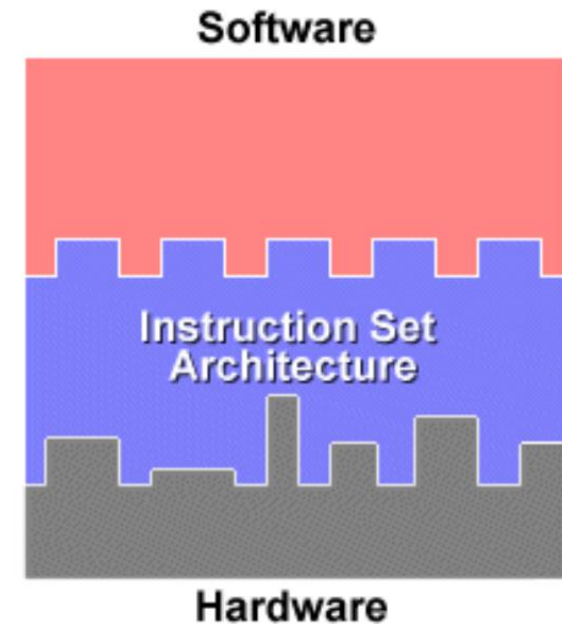




# Evolução de máquinas multiníveis

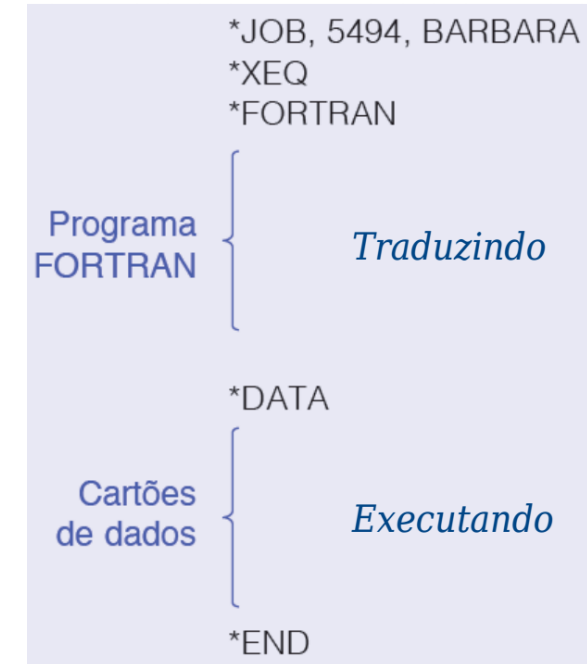
---

- Primeiros computadores digitais (1940)
  - Somente nível lógico e ISA;
- Inventar um Microprograma para executar programas em nível ISA por interpretação (1960 – 1970)



# Evolução de máquinas multiníveis

- Invenção do Sistema Operacional
  - Parecer outro nível
  - Chamadas do sistema operacional
- Esse programa é mantido no computador o tempo todo
- Adicionando funcionalidades de hardware por microprogramação
- Microprogramação se tornou pesada
- Eliminar microprogramação



# Computadores mecânicos

---

- 1623 – Wilhelm Schickard (1592–1635) criou a primeira máquina (mecânica) de calcular
  - Primeira calculadora (4 operações)
  - Johannes Kepler



# Computadores mecânicos

---

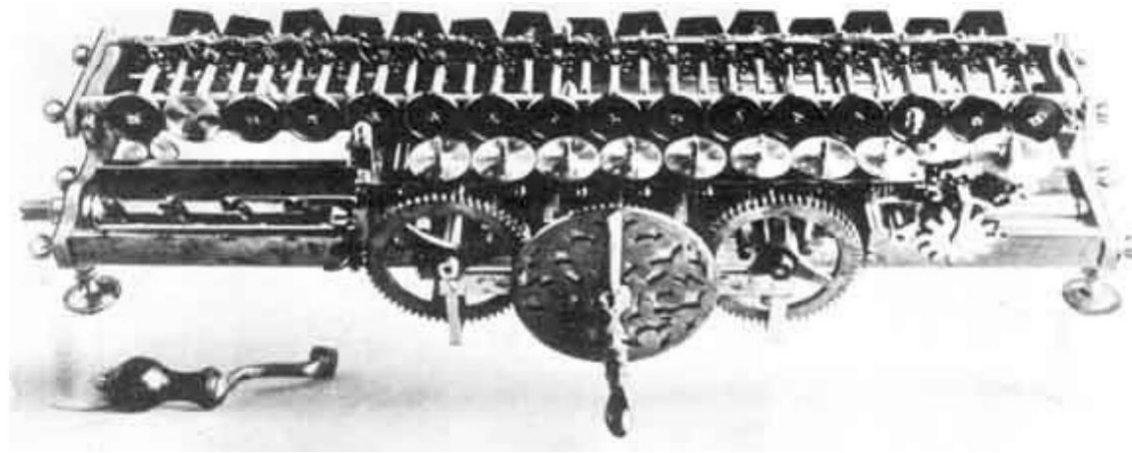
- O reconhecimento de construir uma máquina de calcular operacional foi do cientista francês Blaise Pascal (1623–1662) – Pascalina (1642)
  - Duas operações
  - Máquina com 6 rodas dentadas de 0 – 9



# Computadores mecânicos

---

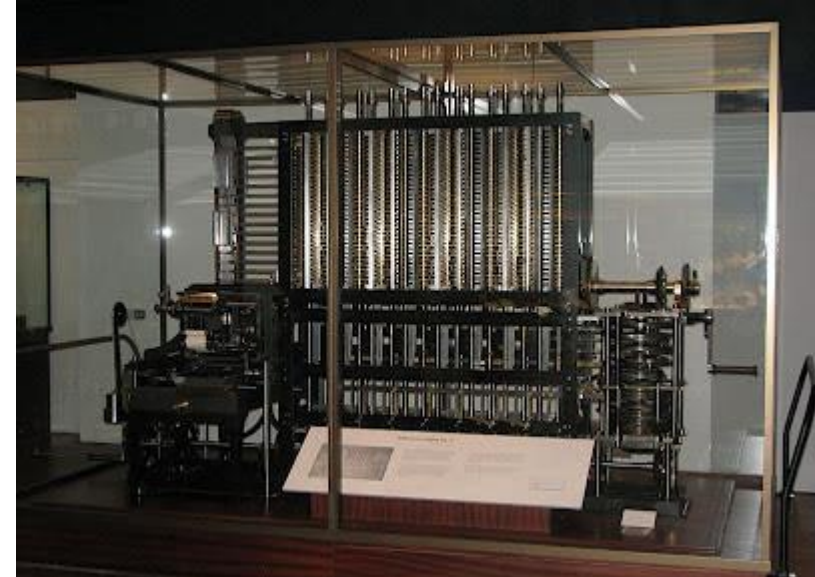
- Gottfried Leibniz (1646–1716) criou, em 1673 a primeira calculadora com as quatro operações e a raiz quadrada
- Cálculos astronômicos
- Sem uso comercial



# Computadores mecânicos

---

- Em 1834, Charles Babbage (1792 – 1871) desenvolveu a Máquina diferencial
  - Método de saída de dados: perfuração de resultados em chapa de cobre com uma punção de aço
  - Projetada para executar um único algoritmo, o método de diferenças finitas que usava polinômios





# Computadores mecânicos

- Após, Charles Babbage (1792–1871) desenvolveu a Máquina analítica
  - Uso geral – podia executar vários algoritmos (programável)
  - Logo, havia a necessidade de realiza a criação de softwares
- Primeira programadora: Ada Lovelace

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 et seq.)

Number of Operations.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data.										Working Variables.										Result Variables.		
					$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$v_9$	$v_{10}$	$v_{11}$	$v_{12}$	$v_{13}$	$v_{14}$	$v_{15}$	$v_{16}$	$v_{17}$	$v_{18}$	$v_{19}$	$v_{20}$	$v_{21}$	$v_{22}$	
1	$\times v_1 \times v_2$	$v_3 = v_1 v_2$		$2x$	...	2	x	2x	2x	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2	$- v_1 - v_2$	$v_4 = v_1 + v_2$		$2x-1$	...	1	...	2x-1	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3	$+ v_1 + v_2$	$v_5 = v_1 + v_2$		$2x+1$	...	1	...	2x+1	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4	$- v_1 - v_2$	$v_6 = v_1 + v_2$		$2x-1$	...	...	...	0	0	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5	$+ v_1 + v_2$	$v_7 = v_1 + v_2$		$2x+1$	...	2	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6	$- v_1 - v_2$	$v_8 = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7	$+ v_1 + v_2$	$v_9 = v_1 + v_2$		$2x+1$	...	1	...	x	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8	$- v_1 - v_2$	$v_{10} = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9	$+ v_1 + v_2$	$v_{11} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10	$- v_1 - v_2$	$v_{12} = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11	$+ v_1 + v_2$	$v_{13} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
12	$- v_1 - v_2$	$v_{14} = v_1 + v_2$		$2x-1$	...	1	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13	$+ v_1 + v_2$	$v_{15} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14	$- v_1 - v_2$	$v_{16} = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15	$+ v_1 + v_2$	$v_{17} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
16	$- v_1 - v_2$	$v_{18} = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
17	$+ v_1 + v_2$	$v_{19} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18	$- v_1 - v_2$	$v_{20} = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19	$+ v_1 + v_2$	$v_{21} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20	$- v_1 - v_2$	$v_{22} = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
21	$+ v_1 + v_2$	$v_{23} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22	$- v_1 - v_2$	$v_{24} = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
23	$+ v_1 + v_2$	$v_{25} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
24	$- v_1 - v_2$	$v_{26} = v_1 + v_2$		$2x-1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
25	$+ v_1 + v_2$	$v_{27} = v_1 + v_2$		$2x+1$	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Here follows a repetition of Operations thirteen to twenty-three.



# A primeira geração - Válvulas

---

- Em 1930, Konrad Zuse (1910–1995) construiu os primeiros computadores eletromecânicos, com relés
- Efetuavam cálculos e exibiam o resultado em uma fita perfurada
- Máquinas destruída pelo bombardeio da guerra





# A primeira geração - Válvulas

---

- (1944) – Howard Aiken – Mark I
  - Eletromecânico
  - Utilizado por Neumann no projeto Manhattan
  - Poderia utilizar 72 números, cada um de 23
  - Tempo de instrução: 30 segundos



# A primeira geração - Válvulas

---

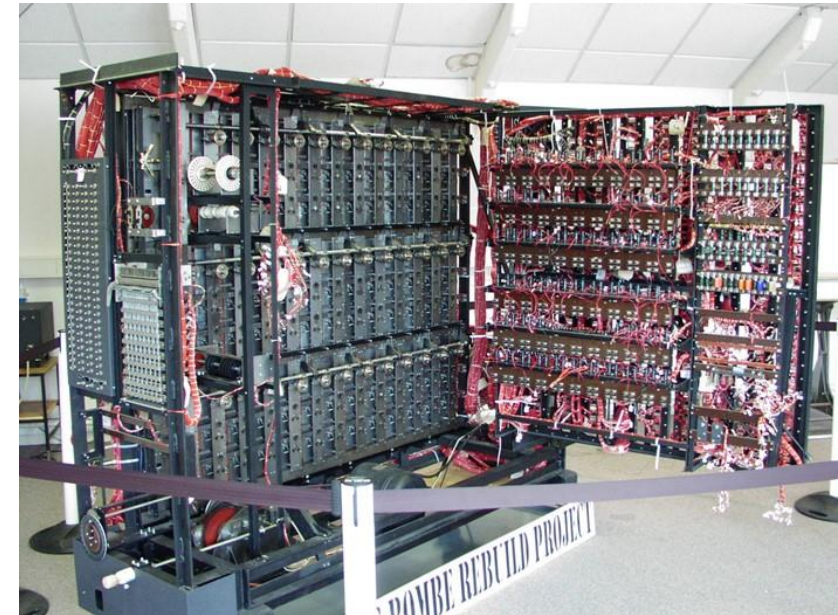
- A motivação para a criação do computador eletrônico foi a Segunda Guerra Mundial
  - Durante a fase inicial do conflito, submarinos alemães atingiam navios britânicos
  - Instruções partiam de Berlim até os submarinos via rádio, que podiam ser interceptadas
  - Problema: mensagens eram codificadas através da máquina Enigma
    - Máquina eletromecânica de criptografia com rotores, utilizada tanto para criar como para decifrar códigos



# A primeira geração - Válvulas

---

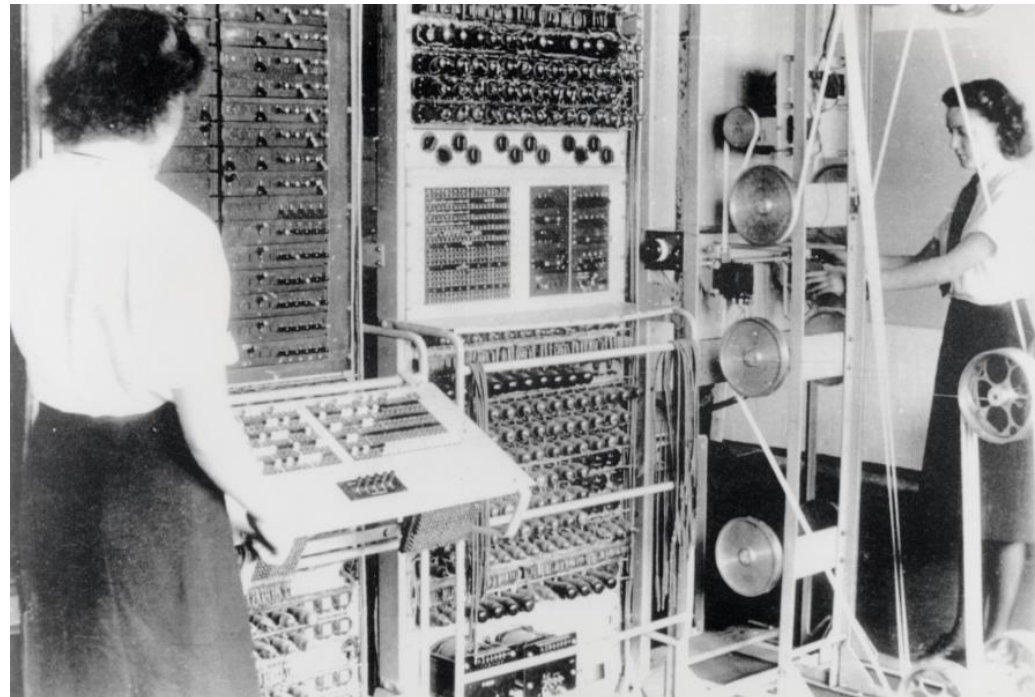
- Alan Turing (1912–1954)
  - Máquina de Turing – formalismo matemático para criação de algoritmos
  - Criou a máquina que decifrou os códigos dos alemães



# A primeira geração - Válvulas

---

- (1942) Colossus – primeiro computador eletrônico programável, projetado com a ajuda de Alan Turing





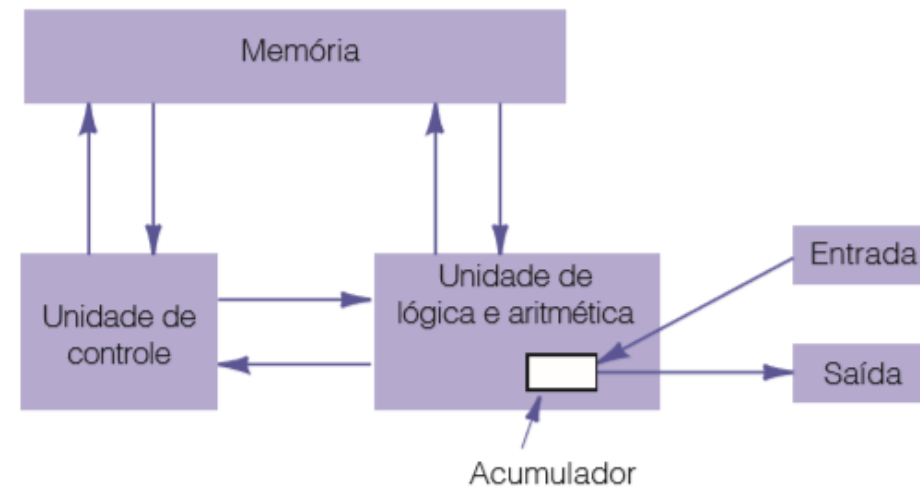
# A primeira geração - Válvulas

- Em 1943, Mauchley e seu aluno de pós-graduação, J. Presper Eckert, passaram a construir um computador eletrônico, ao qual deram o nome de ENIAC
  - Eletronic Numerical Integrator And Computer – Integrador e Computador Numérico Eletrônico
  - 18 mil válvulas; 1500 relés; 30 toneladas; 140kw
  - 20 registradores, cada um com capacidade para um decimal de 10 algarismos
  - Primeiro computador digital eletrônico de grande escala
  - Eckert e Mauchley criaram o EDVAC



# A primeira geração - Válvulas

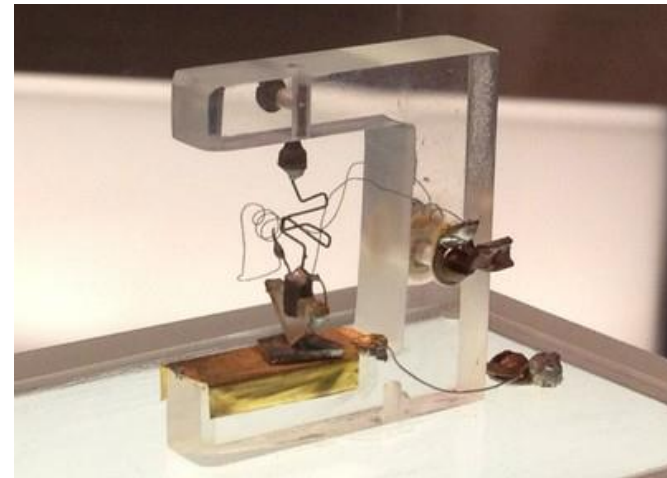
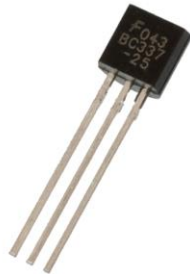
- John von Neumann, no institute of Advanced Studies de Princeton construiu sua própria versão do EDVAC, a máquina IAS.
- Representar o programa digitalmente na memória, junto com os dados
- Trocou a aritmética decimal serial pela binária paralela
- Máquina de von Neumann



# A segunda geração - Transistores

---

- O transistor foi inventado no Bell Labs em 1948 por John Bardeen, Walter Brattain e William Shockley, pelo qual receberam o **Prêmio Nobel de física de 1956**
- Em dez anos, o transistor revolucionou os computadores e, ao final da década de 1950, **computadores de válvula estavam obsoletos**
- O primeiro computador transistorizado foi construído no Lincoln Laboratory do MIT, o **TX-0 – Transistorized e Xperimental computer 0**



# A segunda geração - Transistores

---

- (1961) PDP-1 – primeiro microcomputador
  - 4096 palavras de 18 bits
  - Podia executar 200 mil instruções/segundo
  - US\$ 120 mil
- Primeiro videogame
  - Implementação do jogo SpaceWar
- (1965) PDP-8
  - 12 bits
  - US\$ 16 mil
  - 50 mil unidades vendidas





# A terceira geração – Circuitos integrados

---

- A invenção do CI permitiu empacotar dezenas de transistores em um único chip, permitindo a construção de computadores menores, mais baratos e mais rápidos
- (1964) IBM 360 – Primeira linha de produtos projetada como uma família
  - Permitiu a emulação de outros computadores
  - Clientes podiam continuar a executar seus antigos programas binários sem modificação durante a conversão para a 360
    - Multiprogramação – vários programas na memória ao mesmo tempo
    - Imenso espaço de endereçamento
  - $2^{24} = 16.777.216$  bytes
- (1970) PDP-11
  - Sucessor de 16 bits do PDP-8



# A terceira geração – Circuitos integrados

---



# A quarta geração – Integração

---

- Possibilidade de colocar milhões de transistores em um único chip, levando a computadores menores e mais rápidos
- Com a chegada do microcomputador, cada departamento podia comprar seu próprio computador
- Os preços caíram tanto que se tornou viável um único indivíduo possuir um computador
- Começava a era do computador pessoal
- Kits com Intel 8080, sistema SO CP/M
- Apple e Apple II



# A quarta geração – Integração

---

- A IBM entra no mercado de PC com o 8088 da Intel e cria o IBM Personal Computer.
- Publicação dos projetos em um livro, com objetivo de placas de expansão.
- Empresas que não utilizavam chips da Intel: Commodore, Apple e Atari
- O IBM PC rodava sobre o MS-DOS, com o OS/2.
- Microsoft desenvolveu o Windows e rompeu com a IBM



# A quarta geração – Integração

---

- (1981) Osborne 1 – Primeiro computador portátil
  - Pesava 11 kg
    - “arrastávelo, na verdade
- (1982) Compaq lança o clone portátil do IBM PC



# A quarta geração – Integração

---

- (1983) Apple Lisa – Primeiro PC com GUI
- (1984) Apple Macintosh – Sucessor do Lisa, mais barato
- (1985) Intel 386 – Primeiro ancestral de 32 bits da linha Pentium



# A quinta geração – computadores de baixa potência e invisíveis

---

- (1992) IBM – Primeiro smartphone
- (1993) Apple Newton – Primeiro computador palmtop – PDA
- A quinta geração está mais para uma mudança de paradigma do que uma nova arquitetura específica





# A sexta geração

---

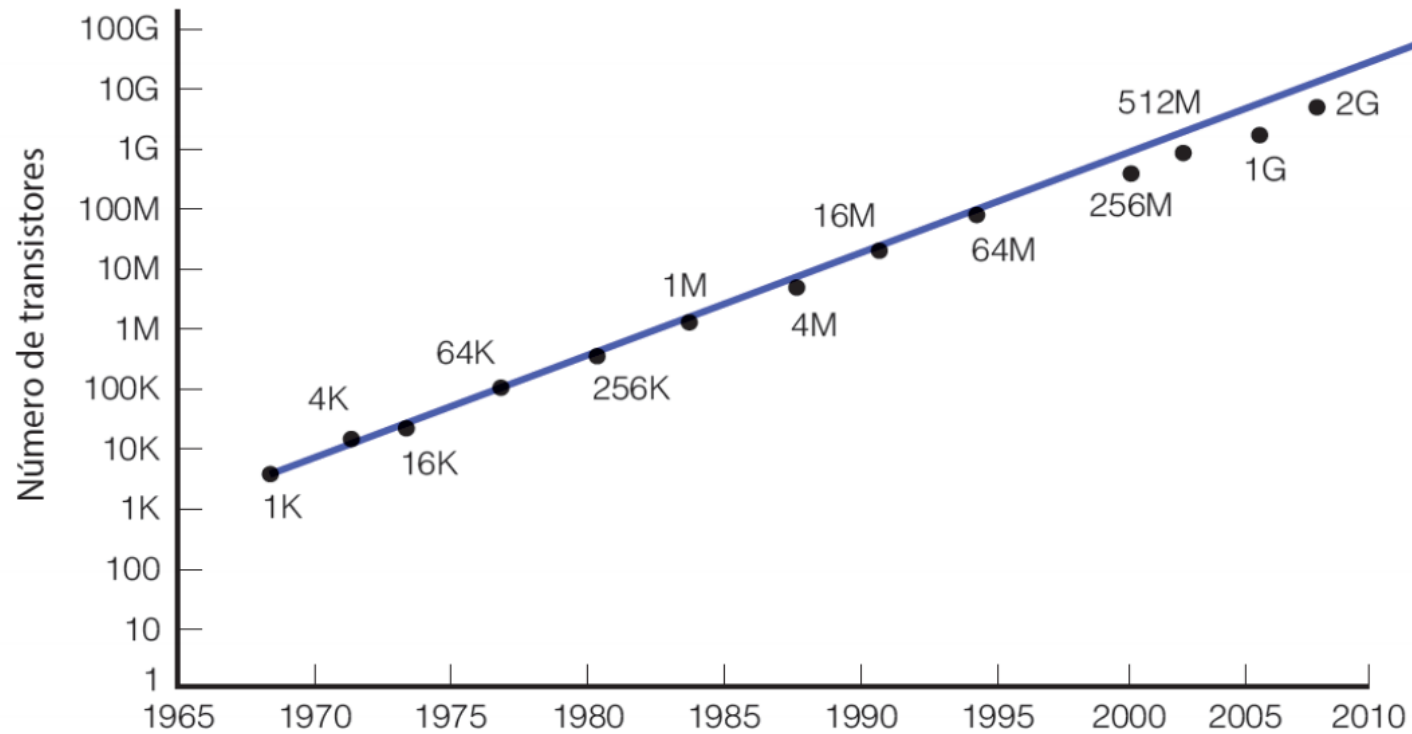
- O que seria?
  - 2009 – A empresa D-Wave lança o primeiro computador quântico comercial
    - 64 mil operações ao mesmo tempo
    - Custo superior a US\$ 10 mi





# Lei de moore

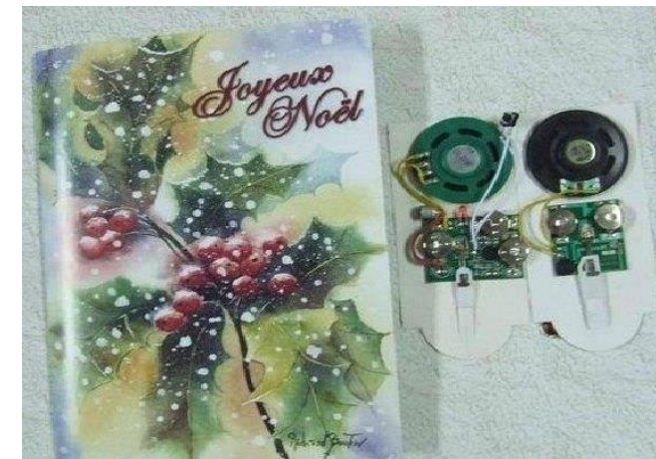
---



# Tipos de computadores

---

- Computadores descartáveis
  - O chip RFIId (Radio Frequency Identification)
  - 0,5 mm de espessura
  - Minúsculo transpônder de rádio
  - 128 bits para dados ou informação
  - Ausência de bateria



# Tipos de computadores

---

- Microcontroladores
  - Gerenciam os dispositivos e manipulam a interface de usuário
  - Processador, memória e capacidade de E/S
  - Maioria dos casos software incorporado de fábrica
  - Necessidade de tempo real
  - Projetos com restrições



# Tipos de computadores

---

- Computadores móveis e de jogos
  - Computadores normais com recursos gráficos especiais e capacidade de som.
  - Software limitado e com pouca possibilidade de extensão
- Sony PlayStation 3
  - CPU multicore de 3,2 GHz, 512 MB de RAM, um chip gráfico Nvidia de 550 MHz e um player Blu-ray
- Microsoft xbox 360
  - CPU triple core PowerPC da iBM de 3,2 GHz, 512 MB de RAM, um chip gráfico ATi de 500 MHz, um DvD player e um disco rígido



# Tipos de computadores

---

- Computador pessoal
  - Abrange os modelos de desktop e notebook
  - A placa de circuito impresso está no coração de cada computador pessoal



# Tipos de computadores

---

- Servidores e Mainframes
  - Servidores possuem sistemas dedicados
    - Têm gigabytes de memória, centenas de gigabytes de espaço de disco rígido e capacidade para trabalho em rede de alta velocidade
    - Clusters consistem em sistemas padrão do tipo servidor, conectados por redes de gigabits/s
  - Os mainframes têm mais capacidade de E/S e costumam ser equipados com vastas coleções de discos que contêm milhares de gigabytes de dados





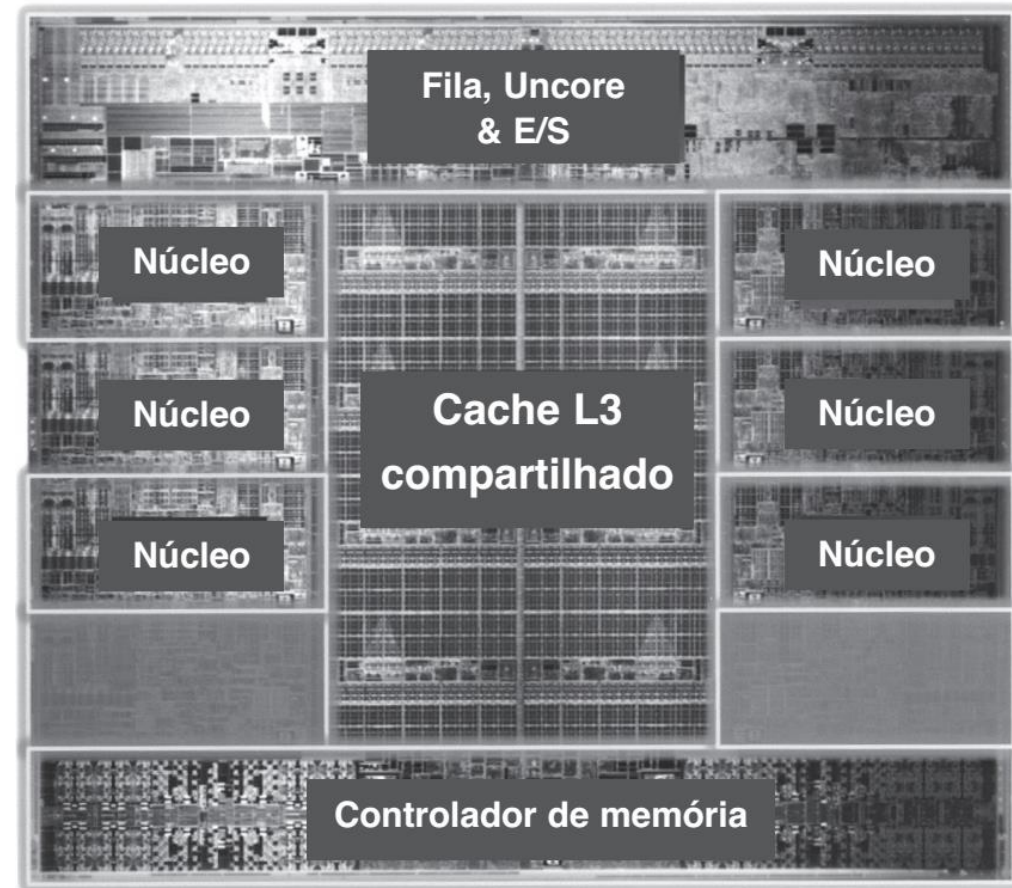
# Arquitetura x86

Chip	Data	MHz	Trans.	Memória	Notas
4004	4/1971	0,108	2.300	640	Primeiro microprocessador em um chip
8008	4/1972	0,108	3.500	16 KB	Primeiro microprocessador de 8 bits
8080	4/1974	2	6.000	64 KB	Primeira CPU de uso geral em um chip
8086	6/1978	5–10	29.000	1 MB	Primeira CPU de 16 bits em um chip
8088	6/1979	5–8	29.000	1 MB	Usada no IBM PC
80286	2/1982	8–12	134.000	16 MB	Com proteção de memória
80386	10/1985	16–33	275.000	4 GB	Primeira CPU de 32 bits
80486	4/1989	25–100	1,2M	4 GB	Memória cache de 8 KB embutida
Pentium	3/1993	60–233	3,1M	4 GB	Dois <i>pipelines</i> ; modelos posteriores tinham MMX
Pentium Pro	3/1995	150–200	5,5M	4 GB	Dois níveis de cache embutidos
Pentium II	5/1997	233–450	7,5M	4 GB	Pentium Pro mais instruções MMX
Pentium III	2/1999	650–1.400	9,5M	4 GB	Instruções SSE para gráficos em 3D
Pentium 4	11/2000	1.300–3.800	42M	4 GB	<i>Hyperthreading</i> ; mais instruções SSE
Core Duo	1/2006	1.600–3.200	152M	2 GB	Dual cores em um único substrato
Core	7/2006	1.200–3.200	410M	64 GB	Arquitetura <i>quad</i> core de 64 bits
Core i7	1/2011	1.100–3.300	1.160M	24 GB	Processador gráfico integrado



# Arquitetura x86

- Core i7, possui 2,27 bilhões de transistores





# Arquitetura ARM

---

- No início da década de 1980, a empresa Acorn Computer começou a trabalhar em uma segunda máquina com a esperança de competir com o recém-lançado IBM PC
- Eles decidiram montar sua própria CPU para o projeto e o chamaram de Acorn RISC Machine (ou ARM)
- A primeira arquitetura ARM (denominada ARM2) apareceu no computador pessoal Acorn Archimedes
- A Apple fez contato com a Acorn para desenvolver um processador ARM para seu próximo projeto Apple Newton
- Arquitetura muito utilizada em smartphones e tablets



# Arquitetura AVR

---

- A arquitetura AVR é usada em sistemas embutidos de muito baixo nível
- Alf-Egil Bogen e Vegard Wollan
- Junto com diversos periféricos adicionais, cada classe de processador AVR inclui alguns recursos de memória adicionais.
- Os microcontroladores possuem em geral três tipos de memória na placa:
  - Flash, EEPROM e RAM.
- Arquitetura utilizada pelas placas Arduino

