

No sistema numérico decimal, há dez dígitos, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Cada dígito representa um múltiplo de uma potência de dez de acordo com a posição ocupada pelo dígito (dizemos que se trata de um sistema posicional).

Exemplo

$$1321,7 = 1 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 1 \times 10^0 + 7 \times 10^{-1}$$

Com n dígitos, é possível representar até 10^n números. Por exemplo, com 3 dígitos podemos representar os $10^3 = 1000$ números inteiros que vão de 0 até 999.

Sistema numérico binário

No sistema numérico binário, há apenas dois dígitos 0, 1. Em um número representado por esse sistema, cada dígito representa, de acordo com sua posição, um múltiplo de uma potência de dois.

Exemplos

Rep. binária	Significado	Rep. decimal
0	0×2^0	0
1	1×2^0	1
10	$1 \times 2^1 + 0 \times 2^0$	2
11	$1 \times 2^1 + 1 \times 2^0$	3
101	$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$	6
11 010	$1 \times 2^4 + 1 \times 2^3 + \dots + 0 \times 2^0$	26

- O sistema binário moderno foi desenvolvido por Liebeniz em 1679.
- Porém, sistema semelhantes já haviam sido desenvolvidos séculos antes na Índia e na China.

O que é um bit.

Bit é uma abreviação de *binary digit* ou “dígito binário”. E representa um dígito de um número binário.

Qualquer informação capaz de ser descrita simbolicamente também pode ser descrita através de números binários, desde que se faça a correspondência entre cada símbolo e um número binário.

Entre as primeiras aplicações de números binários e sua correspondência simbólica, estão os cartões perfurados, usados em máquinas de tabular dados inventadas por Herman Hollerith (1860-1929), e a fita perfurada.

Quantidade de números diferentes que podem ser armazenados em

1 bit: 2;

2 bits: 4;

8 bits (1 byte): 256;

10 bits: 1024;

20 bits: 1048576.0

Em 1854, George Booles, criou uma representação do sistema lógico, empregando um sistema binário. Nesse sistema, o número 1 representaria o valor lógico “Verdadeiro” e o número 0, “Falso”. Booles desenvolveu uma álgebra para esse sistema definindo três operações básicas.

Operações básicas.

- 1 Conjunção ou “E lógico”: $x \wedge y = 1$ apenas quando $x = y = 1$ e $x \wedge y = 0$ caso contrário.
- 2 Dijunção ou “OU lógico”: $x \vee y = 0$ caso $x = y = 0$ e $x \vee y = 1$ caso contrário.
- 3 Negação ou complementação: $\neg 1 = 0$ e $\neg 0 = 1$.

Operações derivadas

- 1 Implicação material: $x \rightarrow y = (\neg x \vee y)$.
- 2 Ou exclusivo: $x \oplus y = (x \vee y) \wedge \neg(x \wedge y)$.
- 3 Identidade: $x \equiv y = \neg(x \oplus y)$.

Ao final do século XIX e início do século XVIII, vários pesquisadores perceberam que um bit poderia ser representado por um circuito eletrônico. Um circuito fechado representaria 1 e um circuito aberto representaria 0. Com o uso de relês, seria possível construir circuitos lógicos que realizariam as operações booleanas, usando eletricidade.

Soma de dois bits através das operações booleanas

Sejam dois bits x e y . Sua soma é representada por um número binário de dois dígitos sendo o mais à direita dado por

$$x \oplus y$$

e, o mais a esquerda, por

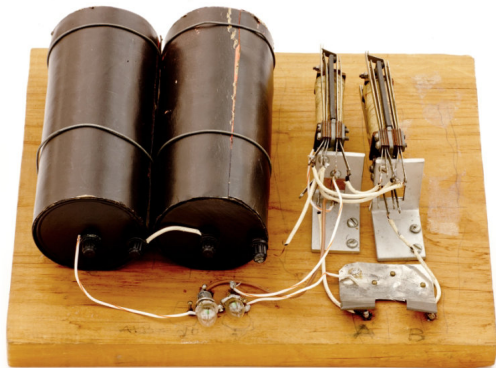
$$x \wedge y.$$

Exemplos

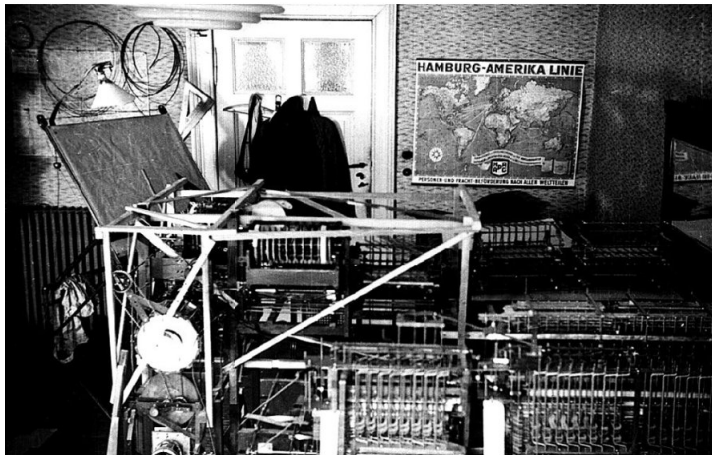
$1 + 1$ é um número binário com o dígito à direita dado por $1 \oplus 1 = 0$ e o segundo dígito dado por $1 \wedge 1 = 1$, isto é, o número binário 10.

$1 + 0$ é um número binário com o primeiro dígito da direita para a esquerda dado por $1 \oplus 0 = 1$ e o segundo dígito dado por $1 \wedge 0 = 0$. Assim, $1 + 0$ é o número binário 01 = 1.

George Stibitz: circuito para adicionar dois bits

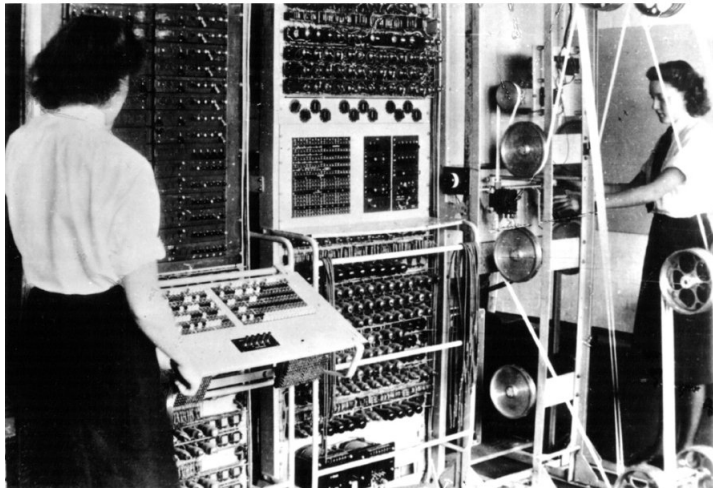


Z1 – primeiro computador eletro-mecânico programável (Alemanha, 1936).



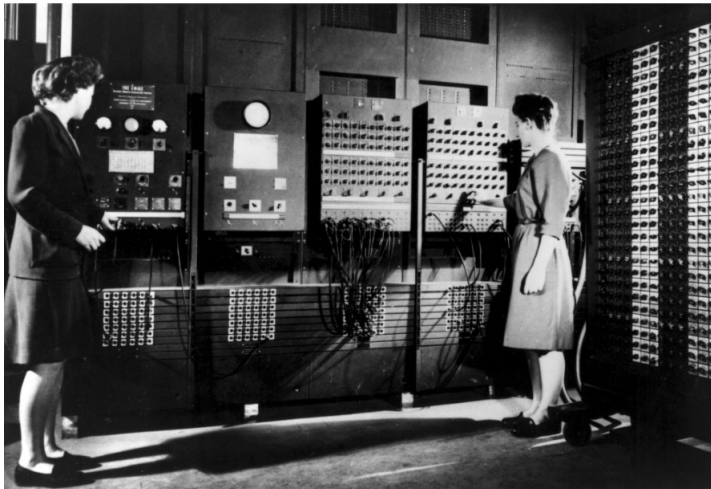
Colossus (Inglaterra, 1944)

Computador eletrônico usado na quebra do sistema de código alemão.



ENIAC – Electronic Numerical Integrator And Computer (USA, 1945)

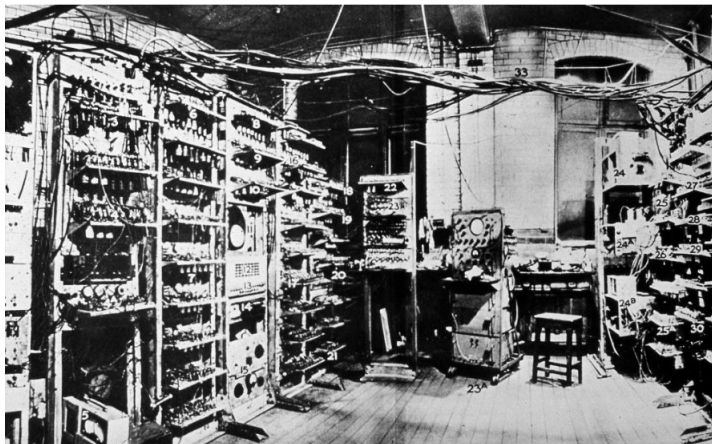
Computador eletrônico programável.



- O primeiros computadores possuíam uma capacidade muito pequena de acumular dados.
- Essa capacidade era empregada para guardar resultados de operações intermediárias.
- Toda a programação do ENIAC era feita através de mudanças de fios e interruptores.
- Os dados eram entrados de modo similar.

Manchester Small Scale Experimental Machine (SSEM), 1949

Primeiro computador com memória para programa



Arquitetura de Von Neumann

- CPU (Central Processing Unit): responsável por interpretar instruções e realizar operações numéricas e lógicas. Contém dois componentes:
 - 1 CU (Control Unit) responsável por extrair e interpretar instruções da memória e
 - 2 ALU (Arithmetic Logic Unit) responsável por realizar as operações lógicas e numéricas quando assim instruída pela CU.
- Memória armazena dados e instruções para a CPU.
- A CPU troca dados e instruções com a memória através do chamado data bus.

Gargalo do Von Neumann:

A capacidade e a velocidade de processamento da CPU tem crescido mais rapidamente que a capacidade do data bus. A velocidade de processamento efetiva é limitada pela capacidade de troca entre a CPU e a memória.

Primeiros computadores: A programação era feita em painéis através do acionamento de interruptores e conexão de cabos e poderia durar semanas.

1950's Sistemas operacionais do tipo “batch” .

- Permitiam que programas lidos, usualmente a partir de cartões perfurados, fosse executados um a um de acordo com um critério de prioridade pré-definido.
- Ofereciam programas auxiliares, chamados livrarias, que executavam tarefas comuns e rotineiras, tais como imprimir um texto, realizar cálculos específicos, etc.

- 1960's
- Sistemas com compartilhamento de tempo. Programas executados “simultaneamente” por diferentes usuários.
 - Desenvolvimento do UNICS, posteriormente UNIX. (AT&T)
- 1970's
- Apple DOS – sistema operacional para o Apple.
- 1980's
- PC-DOS, MS-DOS, MAC OS, Amiga OS, Windows 1.0
 - GNU Project – início do software aberto.
- 1990's
- Linux, Windows 95, FreeBSD.

Funções de um sistema operacional

Acesso de usuários a recursos: gerenciar acesso de diferentes usuários ao recurso do computador via critérios de prioridade.

Acesso dos programas a recursos:

- Gerenciamento de memória: um programa não pode alterar a memória empregada por outro programa ou pelo sistema operacional.
- Gerenciamento de acesso à CPU.

Acesso de programas a programas.

Interface software hardware: virtualização e drivers.

Interface com o usuário:

- Interface gráfica.
- Console.