



UNIVERSIDADE  
FEDERAL DE  
SERGIPE



DEPARTAMENTO  
DE COMPUTAÇÃO

# Introdução

## Arquitetura de Computadores

Bruno Prado

Departamento de Computação / UFS

# Introdução

- ▶ O que é um computador?
  - ▶ Sistema natural ou artificial que é capaz de resolver problemas dinamicamente

# Introdução

- ▶ O que é um computador?
  - ▶ Sistema natural ou artificial que é capaz de resolver problemas dinamicamente
  - ▶ Atende a um conjunto de regras

# Introdução

- ▶ O que é um computador?
  - ▶ Sistema natural ou artificial que é capaz de resolver problemas dinamicamente
  - ▶ Atende a um conjunto de regras
  - ▶ Executa um conjunto de passos ou algoritmo

# Introdução

- ▶ Período de 1642 a 1945
  - ▶ Blaise Pascal
    - ▶ Criou a primeira máquina de calcular (1642)
    - ▶ Utilizava engrenagens e era movida por manivela
    - ▶ Operações de soma e subtração

# Introdução

- ▶ Período de 1642 a 1945
  - ▶ Blaise Pascal
    - ▶ Criou a primeira máquina de calcular (1642)
    - ▶ Utilizava engrenagens e era movida por manivela
    - ▶ Operações de soma e subtração
  - ▶ Baron Gottfried Wilhelm von Leibniz
    - ▶ Aprimorou a máquina para realizar operações de multiplicação e divisão (1646 - 1716)

# Introdução

- ▶ Período de 1642 a 1945
  - ▶ Blaise Pascal
    - ▶ Criou a primeira máquina de calcular (1642)
    - ▶ Utilizava engrenagens e era movida por manivela
    - ▶ Operações de soma e subtração
  - ▶ Baron Gottfried Wilhelm von Leibniz
    - ▶ Aprimorou a máquina para realizar operações de multiplicação e divisão (1646 - 1716)
  - ▶ Charles Babbage
    - ▶ Construiu uma máquina mecânica capaz de executar algoritmos de propósito geral (1792 - 1871)
    - ▶ Programação em cartões perfurados

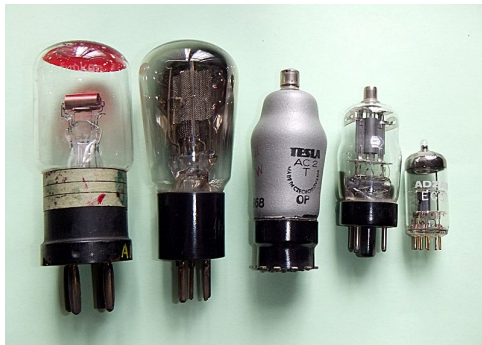
# Introdução

- ▶ Período de 1945 a 1956
  - ▶ *Universal Automatic Computer* (UNIVAC)
  - ▶ Programas em cartões perfurados
  - ▶ Tecnologia de válvulas eletrônicas



# Introdução

- ▶ Válvulas eletrônicas
  - ▶ Gás de alta ou baixa pressão
  - ▶ Funcionamento mecânico (aquecimento)



# Introdução

- ▶ UNIVAC
  - ▶ Painel de controle



# Introdução

- ▶ Período de 1956 a 1964
  - ▶ Busca de redução de custos
  - ▶ Maior necessidade de armazenamento e robustez
  - ▶ Entradas e saídas grandes

# Introdução

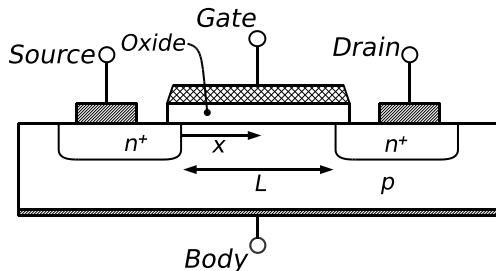
- ▶ Período de 1956 a 1964
  - ▶ Busca de redução de custos
  - ▶ Maior necessidade de armazenamento e robustez
  - ▶ Entradas e saídas grandes
- ▶ Substituição dos cartões perfurados
  - ▶ Circuitos elétricos
  - ▶ Memória magnética
  - ▶ Não volátil

# Introdução

- ▶ Período de 1965 até os dias atuais
  - ▶ Transistores + Circuitos Integrados
  - ▶ Redução de custo, potência e área utilizada
  - ▶ Maior confiabilidade

# Introdução

- ▶ Transistor
  - ▶ Semicondutor
  - ▶ Estado sólido



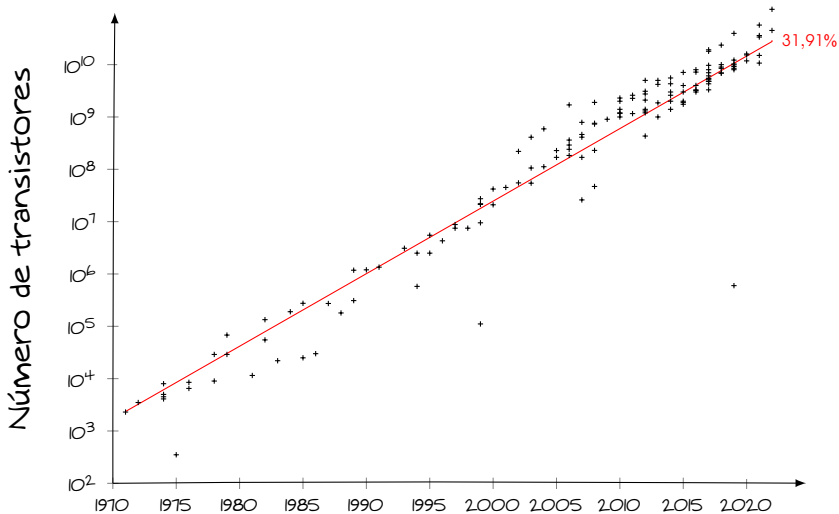
# Introdução

- ▶ Computador em um chip
  - ▶ Intel 8080



# Introdução

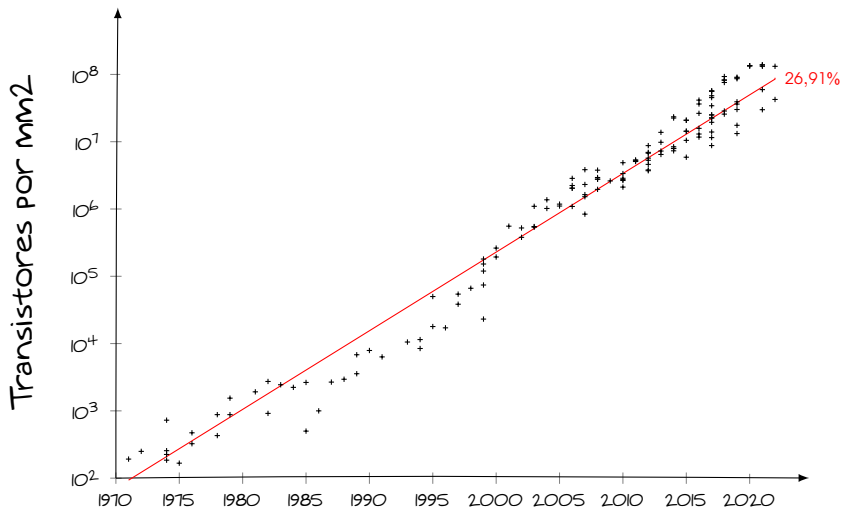
- ▶ Lei de Moore
  - ▶ Quantidade de transistores duplica a cada 30 meses





# Introdução

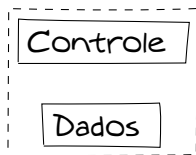
- ▶ Lei de Moore
  - ▶ Densidade da tecnologia duplica a cada 36 meses



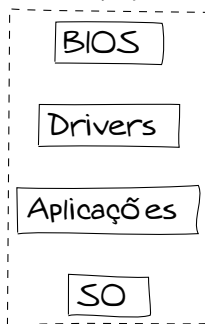
# Introdução

- ▶ Visão abstrata de um computador
  - ▶ Hardware × Software

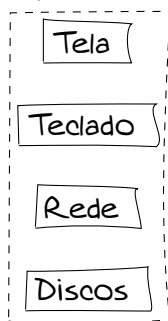
## Processador



## Memória



## Entrada/Saída



# Hardware

- ▶ Tecnologia digital
  - ▶ Codificação binária
    - ▶ Falso ou nível lógico baixo (0)
    - ▶ Verdadeiro ou nível lógico alto (1)

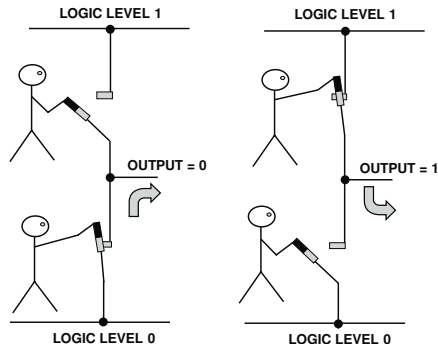
# Hardware

- ▶ Tecnologia digital
  - ▶ Codificação binária
    - ▶ Falso ou nível lógico baixo (0)
    - ▶ Verdadeiro ou nível lógico alto (1)
  - ▶ Controle por chaves eletrônicas
    - ▶ Relê
    - ▶ Transistor
    - ▶ Válvula

# Hardware

- ▶ Tecnologia digital
  - ▶ Codificação binária
    - ▶ Falso ou nível lógico baixo (0)
    - ▶ Verdadeiro ou nível lógico alto (1)
  - ▶ Controle por chaves eletrônicas
    - ▶ Relê
    - ▶ Transistor
    - ▶ Válvula
  - ▶ Lógica booleana
    - ▶ Portas lógicas (AND, OR e NOT)

- ▶ Tecnologia CMOS
  - ▶ *Complementary Metal Oxide Semiconductor*

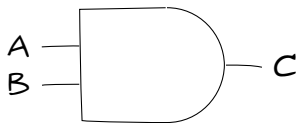


# Hardware

- ▶ Controle do fluxo elétrico
  - ▶ Permitir ou impedir a passagem dos elétrons
  - ▶ Analogia com fluxo de água
    - ▶ Fio  $\equiv$  Mangueira
    - ▶ Transistor  $\equiv$  Registro

# Hardware

## ► Porta lógica AND

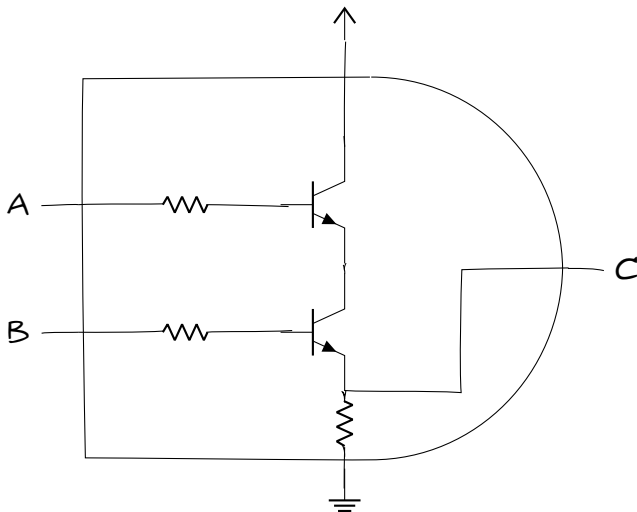


A	B	C
0	0	0
0	1	0
1	0	0
1	1	1



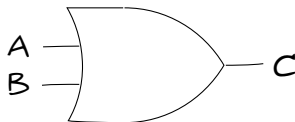
# Hardware

## ► Porta lógica AND



# Hardware

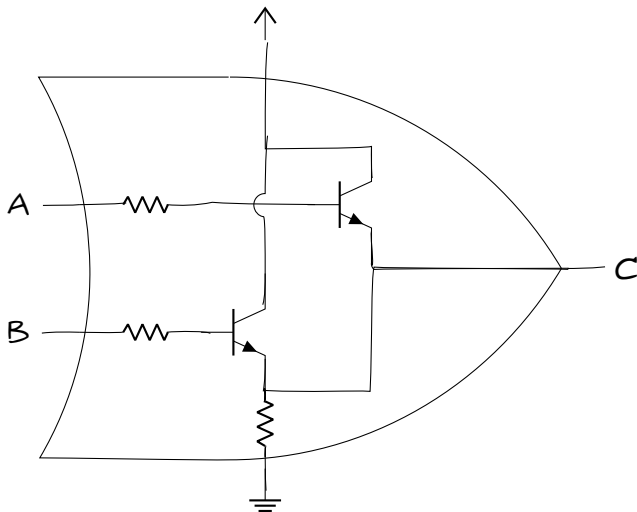
## ► Porta lógica OR



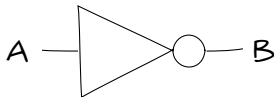
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

# Hardware

## ► Porta lógica OR



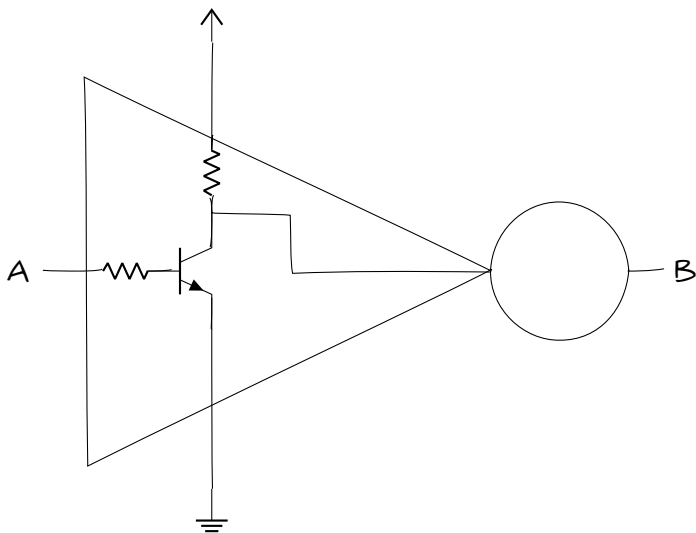
## ► Porta lógica NOT



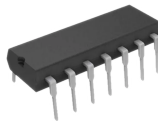
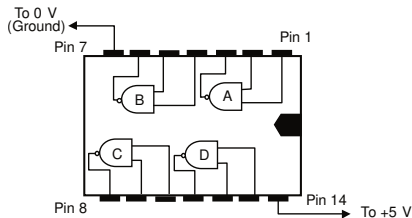
A	B
0	1
1	0

# Hardware

## ► Porta lógica NOT



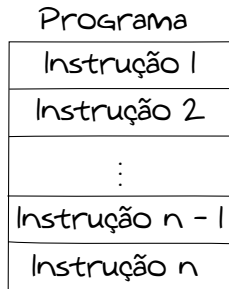
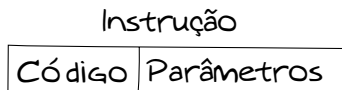
- ▶ Portas lógicas
  - ▶ São os blocos básicos utilizados para a construção de módulos mais complexos



# Exemplo

- ▶ Considerando as portas lógicas AND, OR e NOT, implemente a lógica de controle de um sistema de cancela de estacionamento
  - ▶ Sensor de presença de veículo
    - ▶ Ausente (0)
    - ▶ Presente (1)
  - ▶ Horário de funcionamento
    - ▶ Fechado (0)
    - ▶ Aberto (1)
  - ▶ Ações executadas pelo sistema
    - ▶ Abaixar a cancela (0)
    - ▶ Levantar a cancela (1)

- ▶ Unidade Central de Processamento (CPU)
  - ▶ Cada processador possui seu repertório de instruções
  - ▶ Executa um programa armazenado na memória





# Hardware

- ▶ Fatos interessantes
  - ▶ Escala de nanômetros (1 nanômetro =  $10^{-9}$  metro)

# Hardware

- ▶ Fatos interessantes
  - ▶ Escala de nanômetros (1 nanômetro =  $10^{-9}$  metro)
  - ▶ Tamanho do átomo de silício (0,2 nm)

- ▶ Fatos interessantes
  - ▶ Escala de nanômetros (1 nanômetro =  $10^{-9}$  metro)
  - ▶ Tamanho do átomo de silício (0,2 nm)
  - ▶ Transistor em produção (5 nm)

- ▶ Fatos interessantes
  - ▶ Escala de nanômetros (1 nanômetro =  $10^{-9}$  metro)
  - ▶ Tamanho do átomo de silício (0,2 nm)
  - ▶ Transistor em produção (5 nm)
  - ▶ No diâmetro do fio de cabelo humano (90.000 nm) caberiam mais de 18.000 transistores

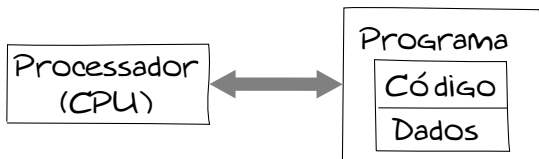
## ► Fatos interessantes

- Escala de nanômetros (1 nanômetro =  $10^{-9}$  metro)
- Tamanho do átomo de silício (0,2 nm)
- Transistor em produção (5 nm)
- No diâmetro do fio de cabelo humano (90.000 nm) caberiam mais de 18.000 transistores
- Seria necessário aumentar o chip para o tamanho de uma casa para começar a ver alguma coisa

## ► Fatos interessantes

- Escala de nanômetros (1 nanômetro =  $10^{-9}$  metro)
- Tamanho do átomo de silício (0,2 nm)
- Transistor em produção (5 nm)
- No diâmetro do fio de cabelo humano (90.000 nm) caberiam mais de 18.000 transistores
- Seria necessário aumentar o chip para o tamanho de uma casa para começar a ver alguma coisa
- Um ser humano levaria cerca de 5.000 anos para chavear o mesmo número de vezes que um transistor é capaz de fazer em apenas 1 segundo

- ▶ Codificação binária
  - ▶ Armazenado na memória principal
    - ▶ Sequência de instruções (código)
    - ▶ Dados e parâmetros de operações (dados)

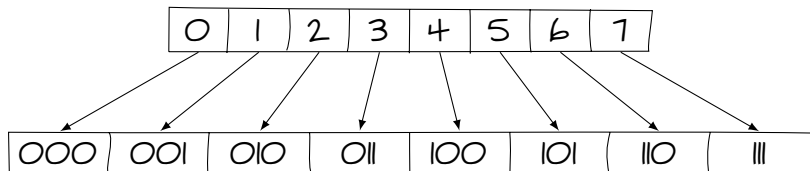


- ▶ Codificação binária

- ▶ Base 2

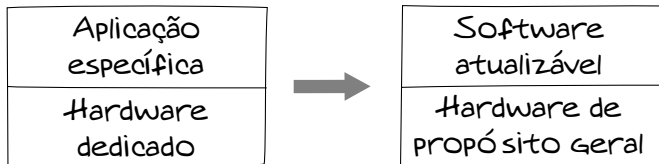
- ▶ Número de bits é  $2^n - 1$

- ▶  $n = 3$  bits representa de 0 até  $2^3 - 1 = 7$





- ▶ Contexto de surgimento do software
  - ▶ Hardware é complexo e caro de ser construído
  - ▶ Necessidade de flexibilidade: software



- ▶ Linguagem de máquina
  - ▶ Codificação binária
  - ▶ Instrução de desvio codificada em 0s e 1s

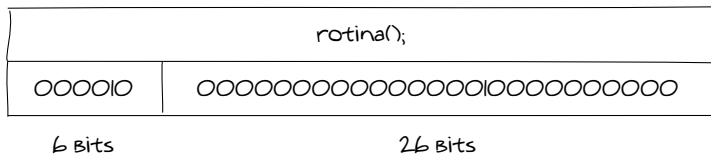
Código	Endereço de desvio
2	1024
000010	000000000000000000010000000000
6 Bits	26 Bits

- ▶ Linguagem de montagem (*assembly*)
  - ▶ Operações mnemônicas
  - ▶ Instrução de desvio traduzida pelo montador

Mnemônico	Endereço de desvio
Bun	1024
000010	000000000000000000100000000000
6 Bits	26 Bits

- ▶ Linguagem de programação
  - ▶ Descrição de alto nível
  - ▶ Desvio de fluxo de execução gerado pelo compilador

Procedimento em C



# Exemplo

- ▶ Considerando um conjunto de 4 passos para realizar operações matemáticas
  1. Adição ( $C = A + B$ )
  2. Subtração ( $D = D - C$ )
  3. Multiplicação ( $E = C \times D$ )
  4. Repete passo 1
- ▶ Defina uma codificação para as instruções utilizadas neste programa, com código de operação e seus respectivos parâmetros
- ▶ Quantos bits tem sua arquitetura? 8, 16 ou 32 bits?

## ► Linguagem de programação C

```
1 // E/S padrão
2 #include <stdio.h>
3 // Função principal
4 int main() {
5     // Imprimindo mensagem no terminal
6     printf("Hello_World!\n");
7     // Retornando com sucesso
8     return 0;
9 }
```

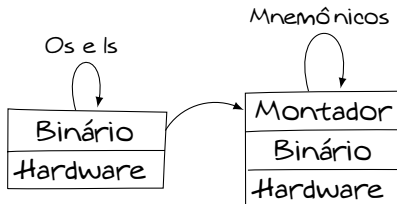
- ▶ Dilema do ovo e da galinha
  - ▶ Como criar um programa se não existe o programa que irá convertê-lo em linguagem de máquina?

- ▶ Dilema do ovo e da galinha
  - ▶ Como criar um programa se não existe o programa que irá convertê-lo em linguagem de máquina?

Passos incrementais com  
a infraestrutura disponível



- ▶ Processo de *bootstrapping*
  - ▶ Linguagem de máquina (binário)
  - ▶ Construção de montadores (*assemblers*)



- ▶ Processo de *bootstrapping*
  - ▶ Linguagem de montagem (*assembly*)
  - ▶ Construção de montadores e compiladores

