



Olá, Professor!  
Peço que realize a **avaliação do conteúdo** com o intuito de manter o nosso material sempre atualizado.

Avalie este  
conteúdo! 🚀 ✨



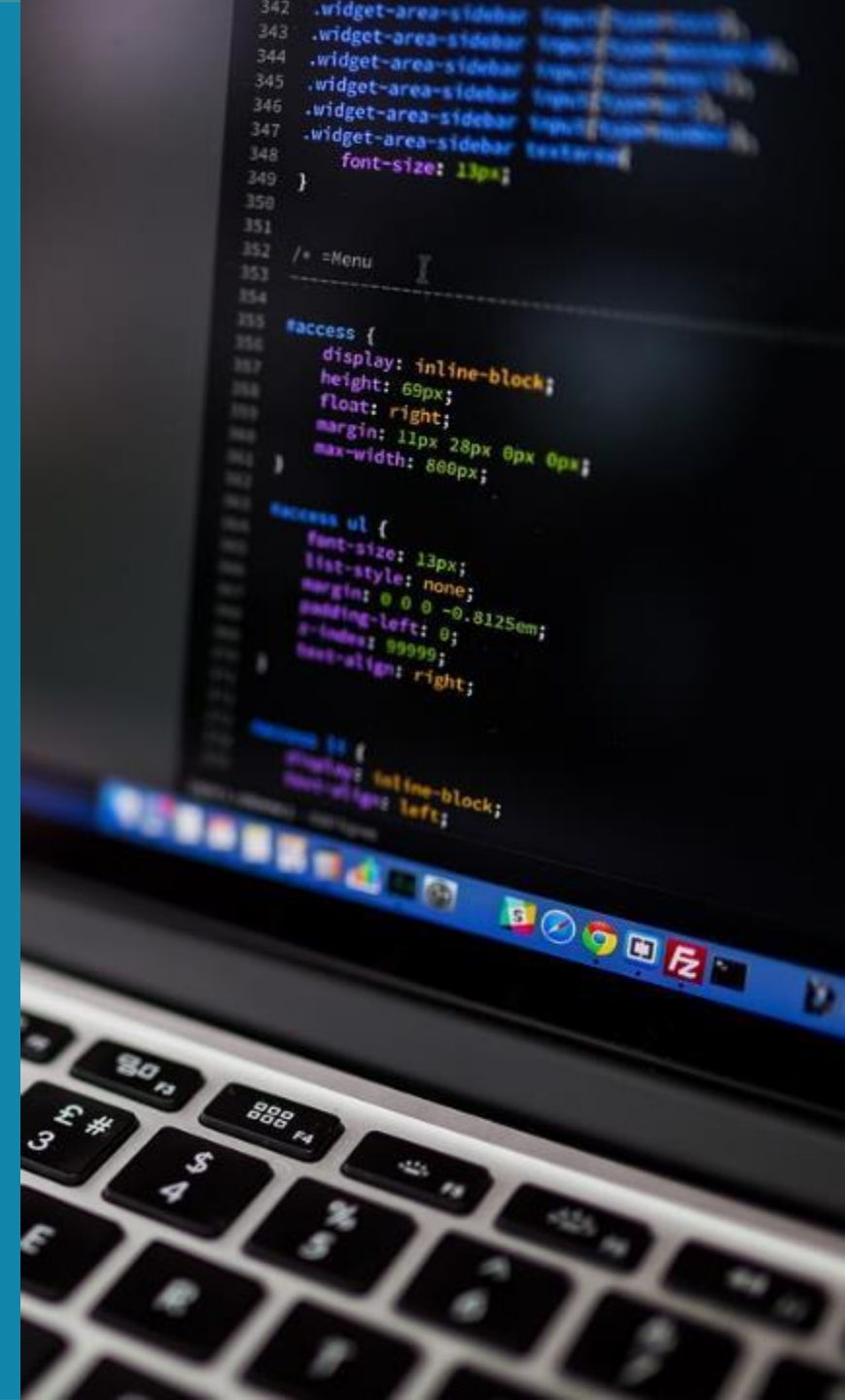
<https://bit.ly/451BDqS>



# Arquitetura de computadores



# Base Computacional



# — Desenvolvimento do computador em quatro gerações

## Primeira geração: válvulas termiônicas

- Durante a Segunda Guerra Mundial, nos EUA, desenvolveram o ENIAC, o primeiro computador eletrônico.
- ENIAC era um integrador numérico eletrônico impressionante para sua época.

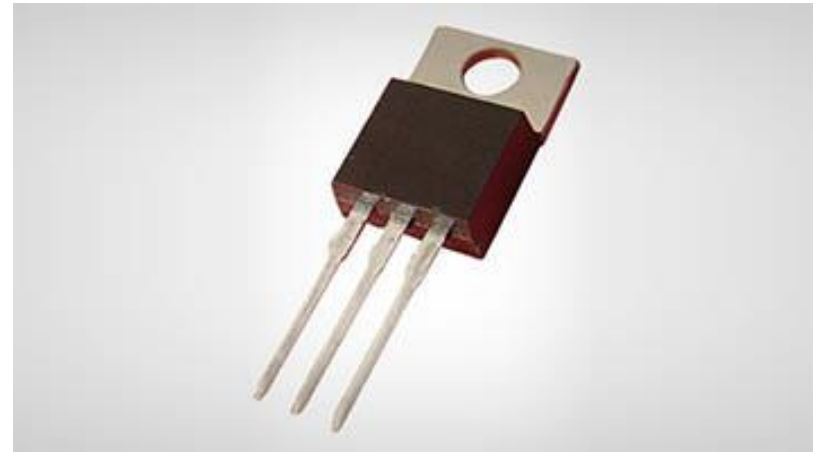


- **Componentes:** 170.000 válvulas termiônicas.
- **Peso:** Cerca de 30 toneladas.
- **Espaço utilizado:** Sala de 150m<sup>2</sup>.
- **Capacidade de processamento (número de cálculos por segundo):** 1 bilhão de vezes menor que a dos celulares usados hoje em dia.

# — Desenvolvimento do computador em quatro gerações

## Segunda geração: transistores

- Primeiros transistores eram pequenos e eficientes, ocupando poucos milímetros.
- Consumiam menos energia que válvulas, possibilitando redução de tamanho em rádios, eletrônicos e computadores.





# — Desenvolvimento do computador em quatro gerações

## Terceira geração: circuitos integrados

- Década de 1960: avanço com circuitos integrados (CI), pastilhas de silício com circuitos miniaturizados.
- Popularmente conhecidos como chips de computador.
- Com o uso de transistores e CI, os computadores ficaram menores e cada vez mais baratos.



# — Desenvolvimento do computador em quatro gerações

## Terceira geração: circuitos integrados

Em 1970, houve a eclosão dos computadores pessoais. Duas famosas empresas do setor, aliás, surgiram nesse período:



### **Microsoft Corporation**

Maior faturamento em software, conhecida pelo Windows e Office. Fundada em 1975 por Bill Gates e Paul Allen nos EUA.



### **Apple Inc.**

Em 1976, vendeu 200 unidades do Apple I; sucesso com o Apple II no ano seguinte. Em 1980, a Apple abriu seu capital na Bolsa de Nova York.

# — Desenvolvimento do computador em quatro gerações

## Quarta geração: microprocessadores

- Década de 1980: Proliferação de PCs potentes, baratos e redes locais.
- Surgimento da internet. Aparecimento de videogames especializados em gráficos e interação.
- Fim do século XX: Computadores pequenos e poderosos integrados em automóveis, aviões e videogames.
- Laptops comuns nas casas. Integração com televisões e celulares cresce nos anos 2010, especialmente por meio de smartphones e smart TVs.





# — Computação no cotidiano

Atualmente, celulares são verdadeiros computadores portáteis conectados à rede móvel.

Usamos computadores para:



**Comunicação**



**Meios de transporte**



**Transações bancárias  
e comerciais**

## — Computação no cotidiano

Atualmente, até o dinheiro não é guardado mais em cofres. Os saldos bancários são armazenados digitalmente nos servidores dos bancos. Se todos os correntistas de um banco solicitassem retirar inteiramente o dinheiro guardado nele, não haveria cédulas suficientes no cofre para atendê-los.

- Em 2008, proposto o coin, sistema independente de bancos, usando blockchain.
- Blockchain permite transações descentralizadas, auditáveis e verificáveis online.



## — Tendências

Em 2019, cinco das seis maiores empresas globais eram de Ciência da Computação.



Microsoft



Apple



Amazon



Alphabet



Facebook

# — Hardware e Software

## Hardware (HW)

Componentes físicos, ou seja, o que pode ser visto e tocado.



## Software (SW)

Programas executados no computador.

# — Hardware e Software

## Conceitos

Propagação dos computadores ocorre devido a funções genéricas (hardware) e programas úteis (software).

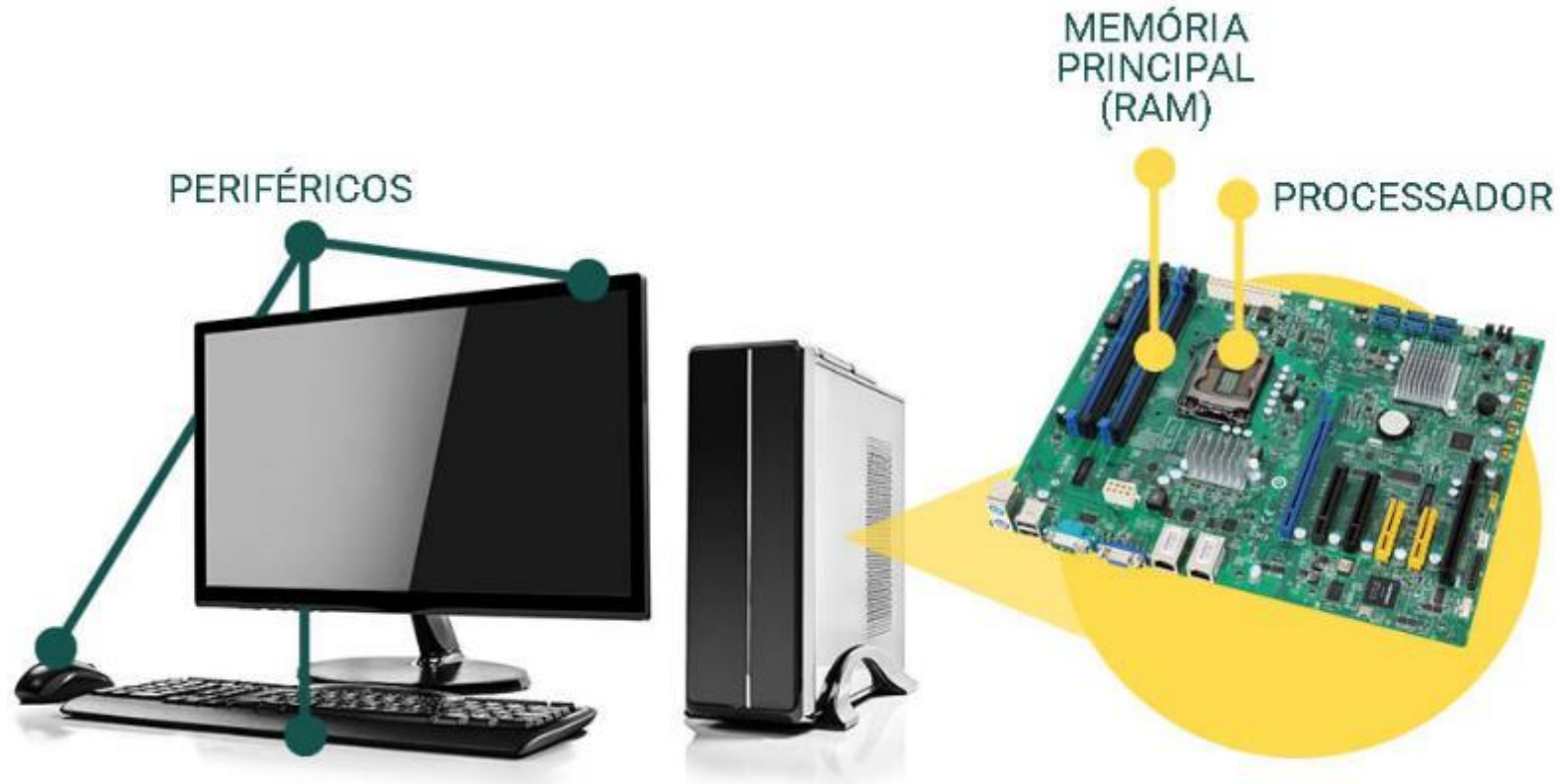
Hardware forma a base para a utilidade de sistemas computacionais.

- Assim como um automóvel tem potenciais, seu uso é definido pelo motorista.
- Da mesma forma, o software em um computador determina suas funções e usos específicos.
- Motorista precisa de engate para usar reboque; similarmente, computador precisa do hardware adequado para executar programas.





# — Principais componentes de hardware dos computadores



# — Principais componentes de hardware dos computadores

## Processador

- Processador, ou CPU, é o cérebro do computador, executando instruções sequencialmente.
- Unidade lógica e aritmética realiza operações fundamentais, como adição e subtração.
- Uma das principais características de um processador é a velocidade com que consegue executar instruções. Isso depende diretamente da frequência do clock.



# — Principais componentes de hardware dos computadores

## Processador

- Overclocking: aumentar a velocidade do clock do processador para melhorar o desempenho.
- Aumento de clock continuou até limitações físicas.
- Solução: múltiplos núcleos em um único chip, simulando vários processadores.



**Dual core (dois núcleos)**



**Quad core (quatro núcleos)**

# — Principais componentes de hardware dos computadores

## Processador

Além do clock e do número de núcleos, outra característica importante do processador é a sua memória cache.



**1**

Funciona como uma pequena parte da memória principal dentro do próprio chip do processador.



**2**

As informações são lidas ou escritas na memória cache com muito mais velocidade que nos pentes de memória.



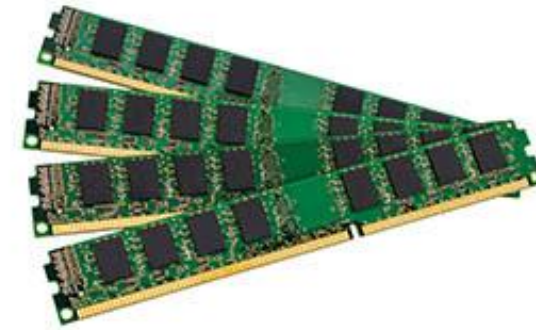
**3**

Uma boa quantidade dela consegue acelerar a velocidade do processador.

# — Principais componentes de hardware dos computadores

## Memória principal

- Memória principal armazena dados e programas para execução no processador.
- Composta por células, cada uma armazenando um byte, semelhante a caixas de correio com endereços únicos.
- É chamada de RAM (em seu formato mais comum) por permitir o acesso a qualquer endereço em qualquer ordem.





# — Principais componentes de hardware dos computadores

## Memória principal

Principais características de uma memória RAM:

### **Capacidade de armazenamento**

Se possui 4 GB (quatro gigabytes) de armazenamento, a memória RAM conta com 4 bilhões dessas células, podendo armazenar até 32 bilhões de bits de dados.

### **Velocidade de comunicação com o barramento**

Trata-se da velocidade com que a memória consegue transferir os dados para o processador.

# — Principais componentes de hardware dos computadores

## Placa-mãe

Consiste em um circuito elétrico impresso e uma série de componentes conectados nela.

- Placa-mãe conecta processador, memória e periféricos através de barramentos.
- Integração avançada inclui periféricos, antes encaixados, como placas de vídeo e redes.



# — Principais componentes de hardware dos computadores

## Periféricos

- Componentes conectados à parte central são periféricos, essenciais para sistemas computacionais.
- Primeiro computador usava lâmpadas como saída e cartões perfurados como entrada, destacando a evolução dos dispositivos.

# — Principais componentes de hardware dos computadores

## Dispositivos de entrada

- **Tela *touchscreen*:** permite a seleção de elementos sem precisar de um *mouse*.
- **Teclado:** segue um padrão já difundido das máquinas de escrever, facilitando seu uso e sua aceitação.
- ***Mouse*:** abre novos rumos por permitir a indicação de pontos na tela e a seleção deles, o que gerou um grande impacto no desenvolvimento de interfaces gráficas.
- **Microfone:** capta o áudio.
- **Câmera:** capta áudio com vídeo.
- **Placa de rede:** recebe os dados transmitidos pela rede.

# — Principais componentes de hardware dos computadores

## Dispositivos de saída

- **Sistema de vídeo:** Composto, geralmente, por uma placa de vídeo e um monitor ou uma tela.
- **Alto-falantes ou caixas de som:** Emite sons e sinais sonoros.
- **Impressora:** Imprime documentos elaborados no computador.
- **Placa de rede:** Envia os dados pela rede.



## — Principais componentes de hardware dos computadores



A memória principal, sendo volátil, perde informações ao desligar o computador. Existe uma solução para esse problema?

### **Resposta**

sim. Precisamos de um sistema de armazenamento persistente que não perca as informações após esse desligamento. Para isso, são usadas as memórias secundárias.

# — Principais componentes de hardware dos computadores

Principais características das memórias secundárias:

## **Capacidade de armazenamento**

Normalmente medida em GB (gigabytes ou bilhões de bytes) ou TB (terabytes ou trilhões de bytes).

## **Velocidade de transmissão**

Depende do barramento que o liga à placa-mãe.

# — Software

## **Softwares finalísticos ou de aplicação**

Geralmente, são rodados de forma consciente nos computadores, entregando as funcionalidades desejadas por seu usuário.

## **Softwares de sistema**

Permitem que os finalísticos rodem em muitas máquinas com hardwares diversificados. Os softwares de sistema incluem os drivers dos dispositivos instalados no computador, ou seja, programas que controlam como se acessa e comanda determinado periférico.

## — Tendências

- Avanços em hardware e software impulsionam casos de sucesso.
- Circuitos Integrados, microprocessadores, placas de vídeo, mouse, touchscreen, USB são marcos evolutivos.
- Transformações significativas na forma de uso dos computadores.

## — Tendências



### **Microsoft**

DOS e Windows.



### **Apple**

PCs, iPods e iPads.



### **Amazon**

Sistema de vendas on-line com grande qualidade de serviço e sem lojas físicas.



### **Alphabet**

Mecanismo de busca na internet (Google).



### **Facebook**

Redes sociais.



## — O sistema operacional

- O sistema operacional (OS) é um dos principais – e o mais conhecido – softwares de sistema.
- Responsável por conhecer o hardware instalado no computador, ele possui diversas funções importantes.

# — Funções do sistema operacional

Algumas funções do OS:

Servir como uma camada de abstração entre o hardware e a aplicação do usuário.

Cuidar da alocação do armazenamento e da memória principal. Ao pedir para executar um programa, o OS deve alocar espaço na memória tanto para ele quanto para os dados que ele precisará manipular.

Nos computadores atuais, diversos programas executam simultaneamente. Por uma solicitação do usuário, alguns o fazem em primeiro plano, enquanto outros rodam em segundo (ou em background).

# — Funções do sistema operacional

Algumas funções do OS:

Informar quando e quais programas ganham acesso ao uso do processador.

Escolher quando determinados programas devem passar o uso do processador para outro programa. Este subsistema do OS é chamado de escalonador. Como os processadores atuais são normalmente compostos por muitos núcleos, para decidir a alocação de seu tempo a cada programa, o escalonador precisa levar em conta:

- Todos os núcleos disponíveis.
- Características de cada programa a ser executado.

# — Funções do sistema operacional

O Sistema Operacional é um programa intermediário que: gerencia os recursos de hardware do computador, fornece acesso a eles para os demais programas.

- Sistema operacional, peça essencial, carrega após a inicialização.
- Antes, o BIOS (Basic Input/Output System) é carregado, armazenado em memória não volátil na placa-mãe.



# — Histórico

## Anos 1970

Surge o Unix, um OS de uso geral muito difundido por ter sido distribuído gratuitamente para o setor acadêmico e o público nos Estados Unidos.

Como ele não possuía uma interface gráfica, todos os seus comandos eram dados via console a partir de um texto.

The image shows the word "UNIX" in a bold, green, sans-serif font. A registered trademark symbol (®) is located at the top right of the letter "X".

# — Histórico

## Anos 1980

Microsoft vende MS-DOS para computadores da IBM. Sistema ganha destaque, alcançando projeção significativa na computação.

Microsoft: Windows destaca-se com novo OS. Permite execução de programas em janelas, alcançando 90%90% do mercado

Apple: Apple lança OS com interface gráfica e ícones para representar programas.



# — Histórico

## Século XX

Vemos o nascimento de OS para os dispositivos móveis (mobiles) que surgiam

- smartphones;
- tablets;
- smart TVs.



Apple lança o iOS exclusivo para seus dispositivos.

Google e parceiros desenvolvem o Android, desbancando a Microsoft no mercado mobile.

Microsoft, apesar do Windows Mobile, desiste da competição.



## — Tendências

- Sistemas operacionais cruciais para o funcionamento eficiente de sistemas computacionais.
- Com a integração em celulares e tablets, destaque para o desenvolvimento e melhoria dos OS para mobile.
- Principais sistemas incluem iOS, Android e outros.

# — Origem da internet

## Década de 1960

- Computadores consolidados, surgimento da ARPANET.
- Departamento de Defesa dos EUA inicia estudos para rede de comunicação global.
- Objetivo: permitir colaboração remota.
- Evolução dos sistemas em rede permite interação com diversas aplicações.
- Transformação de uma realidade isolada para interconectada e interativa.



# — A internet

## Rede de redes



**1960**

Surgiu a ideia de conectar computadores espalhados pelos centros acadêmicos norte-americanos.



**1970**

ARPANET criada em 1970, conectando pessoas nas duas costas dos EUA.  
Primeira rede a integrar a internet, facilitando colaboração e pesquisa.

# — A internet

## Rede de redes

### **Comunicação entre computadores**

Lançar meios de comunicação (como fios de cobre ou enlaces de rádio) entre os dois computadores.

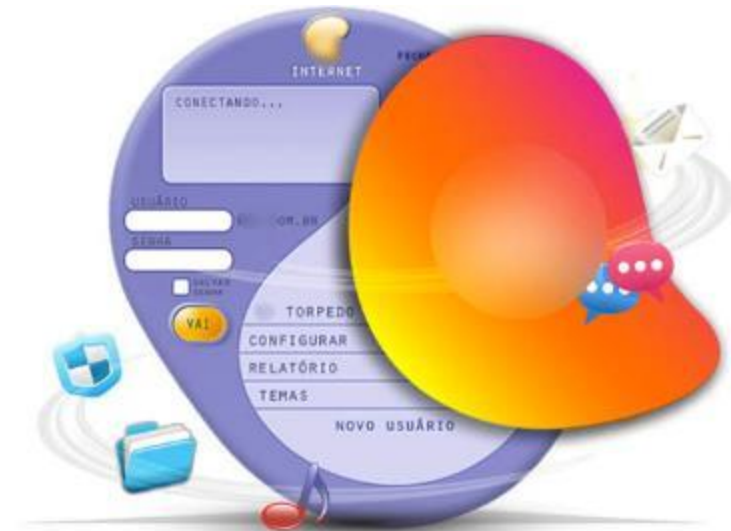
### **Protocolos de comunicação**

Criar protocolos para que ambos pudessem entender como falar entre si, pois eles precisam de um hardware para enviar e receber dados em rede.

# — A internet

## Internet discada

- Empresas de telefonia fornecem conexão à internet via modem discado.
- Modem conectava como uma chamada telefônica, atingindo taxas de 14400 bps.



# — A internet

## Endereço IP e roteador

- Barreira para interligar máquinas: endereçamento único.
- Solução: Endereço IP, sequência de quatro números de 0 a 255 (8 bits cada).
- Pacotes (ou datagramas) enviados entre remetente e destino via redes.
- Roteadores atuam como agências de correio, escolhendo rotas para os pacotes.
- Pacotes passam por roteadores até alcançar o destino com a mensagem.

# — A internet

## E-mail e páginas disponíveis na rede

- Conexão de muitos usuários em rede trouxe desafios técnicos e oportunidades.
- Estimulou o desenvolvimento de novos mercados e tecnologias.
- E-mails inicialmente baseados no sistema de correios da ARPANET.
- Atualmente, correios tradicionais lidam principalmente com encomendas, enquanto a maioria das correspondências é eletrônica.
- Desenvolvimento da World Wide Web (WWW) e navegadores define o que a maioria entende por internet.
- Conceito de páginas interligadas transforma a experiência online.



## — A internet

### Páginas de pesquisa, chats e programas de troca de mensagens

- Google destaca-se como vitorioso na organização da busca na World Wide Web.
- Supera concorrentes na eficácia de indexar e apresentar informações online.
- Atualmente, é possível buscar qualquer página ou tópico na web. Isso tornou a Google uma das empresas mais ricas do mundo.
- Com a popularização da internet, comunicação por e-mail tornou-se lenta.
- Surgiram servidores de chat, permitindo interações em tempo real para contornar a demora do e-mail.

# — A internet

## Interação entre usuários

- Interatividade inicial em LANs impulsionou LAN houses para jogos em rede.
- Evolução: empresas adotam jogos online, expandindo a interação pela internet.

# — A internet

## Banda larga, streaming e aplicativos

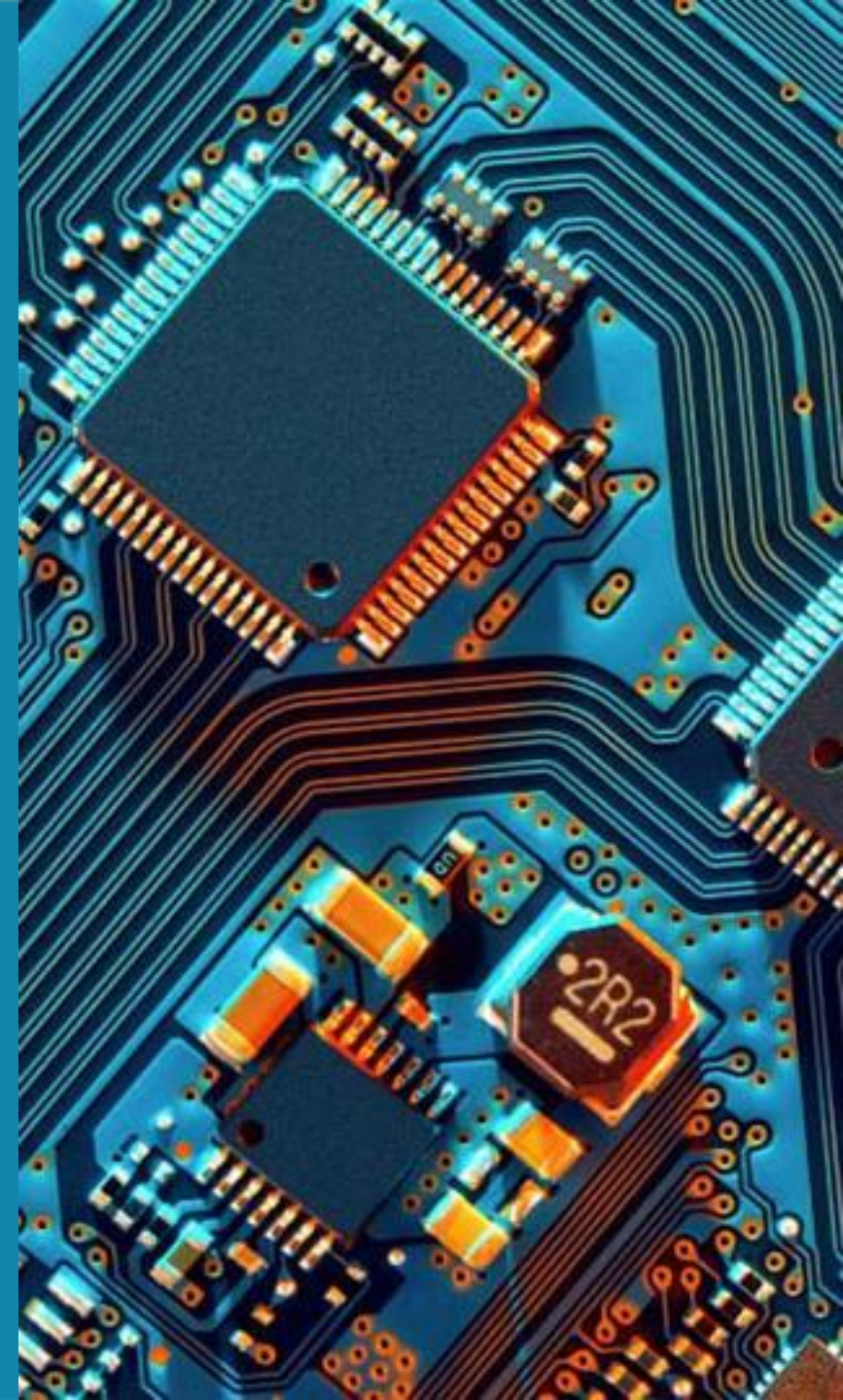
- Desenvolvimento de tecnologias, como fibra ótica, amplia banda de internet doméstica.
- Surgem serviços de música e vídeo online.
- Televisão gradualmente substituída por serviços de streaming.

# — A internet

## Tendências

- Conexão em rede transforma o uso de computadores, impactando hábitos sociais.
- Tendência atual é a conectividade plena, com aplicativos projetados para funcionar online.
- Cientistas e engenheiros buscam rede global para acesso à internet.
- Projeto Starlink, liderado por Elon Musk e SpaceX, visa alcançar conectividade mundial.

# Componentes de hardware



# — Sistemas de computação

## **Implementação de computadores**

Se relaciona, em geral, com a abordagem de aspectos que são desnecessários ao programador.



## **Arquitetura de computadores**

e relaciona, em geral, ao tratamento de pontos que são de interesse do programador, a saber, conjunto de instruções do processador, tamanho da palavra, modos de endereçamento de instruções, entre outros.

A definição de computação pode ser entendida como a realização de cálculos de forma ordenada ou apenas a manipulação de valores.

## — Sistemas de computação



Como você definiria um sistema de computação?

### **Resposta**

Um sistema de computação é um conjunto de partes coordenadas que concorrem para a realização do objetivo de computar (dados).

# — Sistemas de computação



## Dados

Constituem um conjunto de fatos em estado bruto a partir dos quais conclusões podem ser tiradas.



## Informação

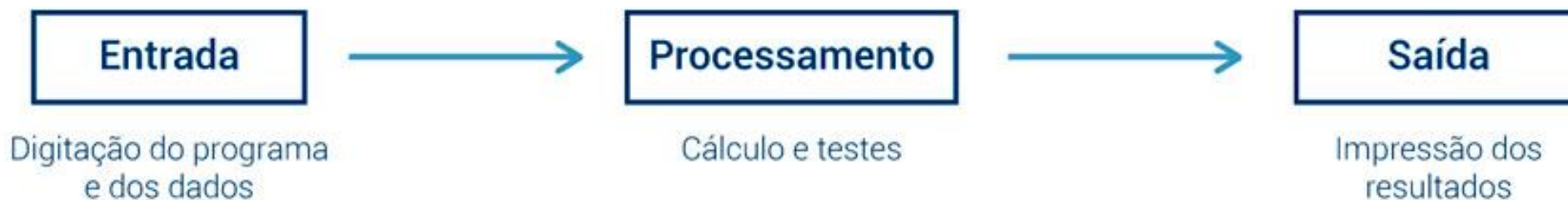
É a inteligência e o conhecimento derivados dos dados



# — Sistemas de computação

## Processamento de dados

Consiste em uma série de atividades **ordenadamente realizadas** (receita de bolo), com o objetivo de produzir um arranjo determinado de **informações** a partir de outras obtidas inicialmente.



# — Linguagem de programação

## Sistemas, dados e informação

Um algoritmo pode ser formalizado em comandos de uma linguagem de programação, entendida pelo sistema de computação.

Por exemplo, um algoritmo para soma de 100 números (1 a 100) está exemplificado, a seguir - exemplo adaptado de Monteiro (2007, p. 9):



→

### **Passo 1**

Escrever e guardar  $N = 0$  e  $SOMA = 0$

→

### **Passo 2**

Ler número da entrada

# — Linguagem de programação

## Sistemas, dados e informação

→

### **Passo 3**

Somar valor do número ao de SOMA e guardar resultado como SOMA

→

### **Passo 4**

Somar 1 ao valor de N e guardar resultado como novo N

→

### **Passo 5**

Se valor de N for menor que 100 então passar para item 2

# — Linguagem de programação

## Sistemas, dados e informação



### **Passo 6**

Se não for, imprimir valor de SOMA



### **Passo 7**

Parar

# — Linguagem de programação

## Tipos de linguagens

### Linguagem de baixo nível de abstração

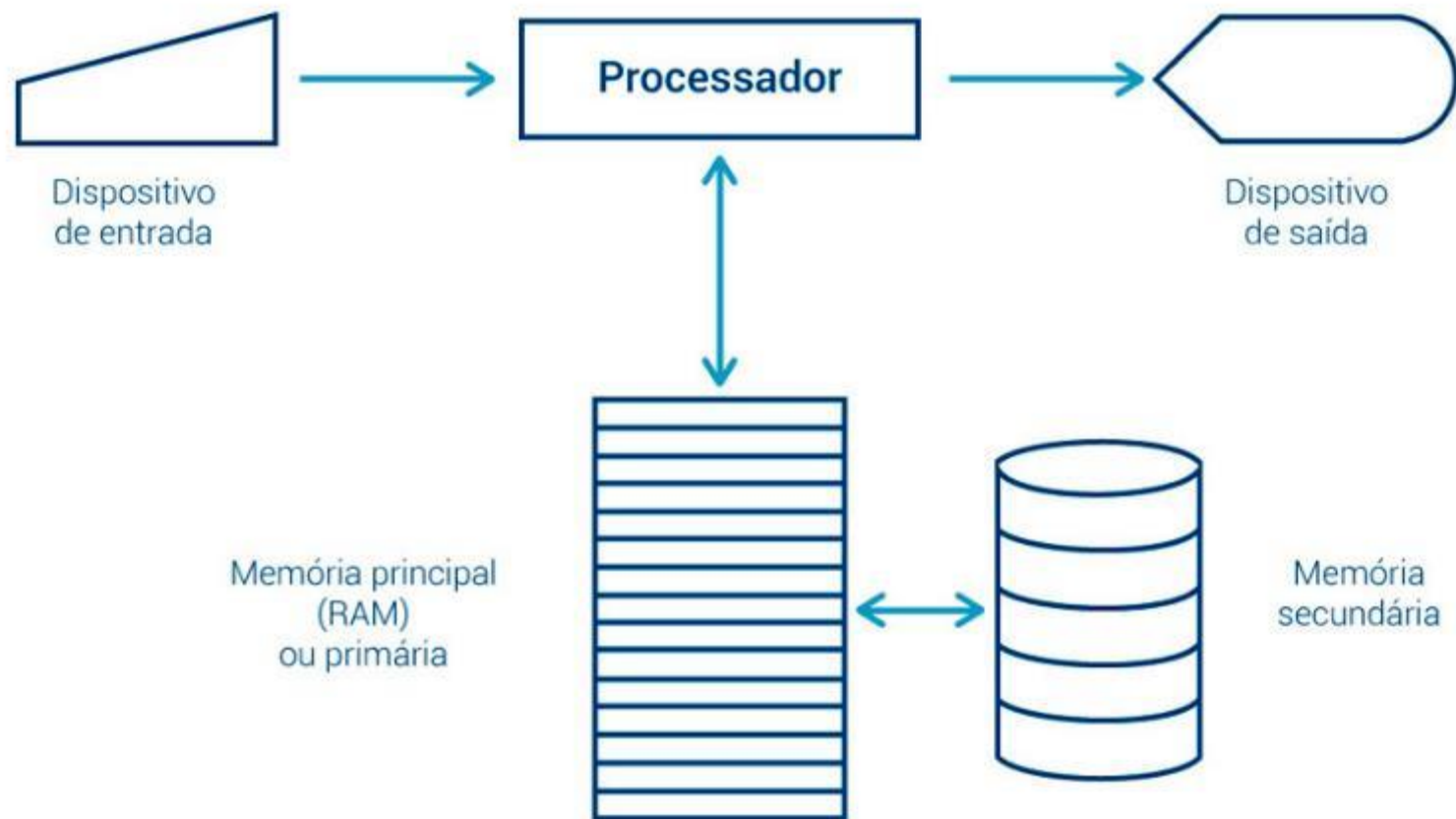
Usa os chamados mnemônicos ao invés de bits. Está relacionada diretamente à arquitetura do processador. Pode ser conhecida como linguagem de montagem e Assembly.



### Linguagem de alto nível de abstração

Possui um nível de abstração relativamente elevado, mais afastado da linguagem de montagem e mais próximo à linguagem humana.

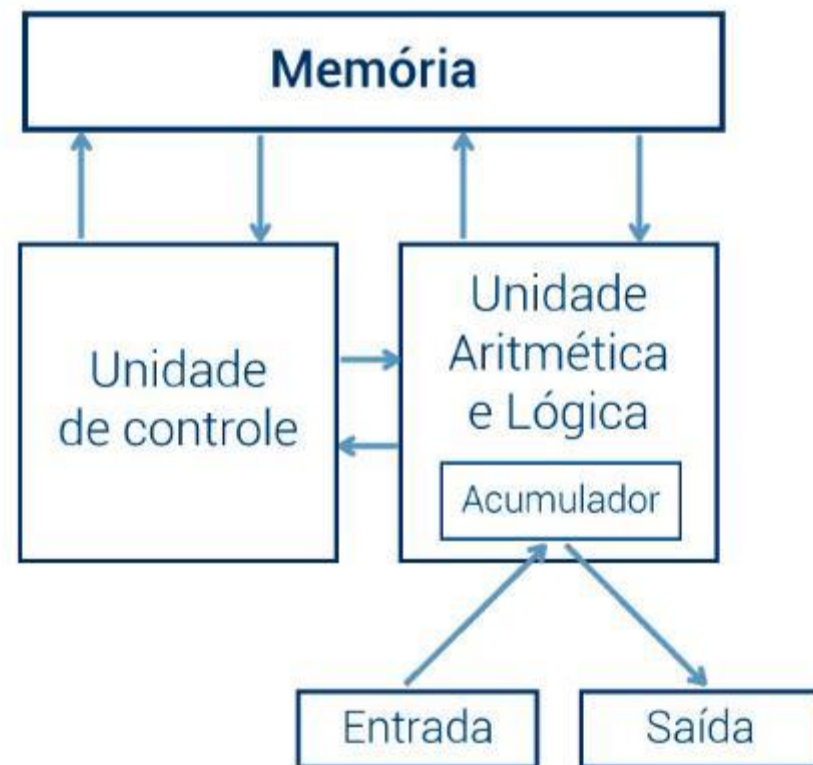
## — Organização de um sistema de computação



# — Organização de um sistema de computação

A **Arquitetura de John von Neumann** foi concebida a partir de 1946, precursora da arquitetura que conhecemos hoje. Ela possibilita a uma máquina digital **armazenar seus programas** no mesmo **espaço de memória** que os **dados**, permitindo, assim, a manipulação de tais programas.

A arquitetura da imagem é um projeto modelo de um computador digital de programa armazenado que utiliza uma unidade de processamento (CPU) e uma de armazenamento (memória) para comportar, respectivamente, instruções e dados.



# — Organização de um sistema de computação

## Barramento

Fundamentalmente, todo sistema de computação (computador) é organizado (funcionalmente) em três grandes módulos ou subsistemas:

**Processador**

**Memória**

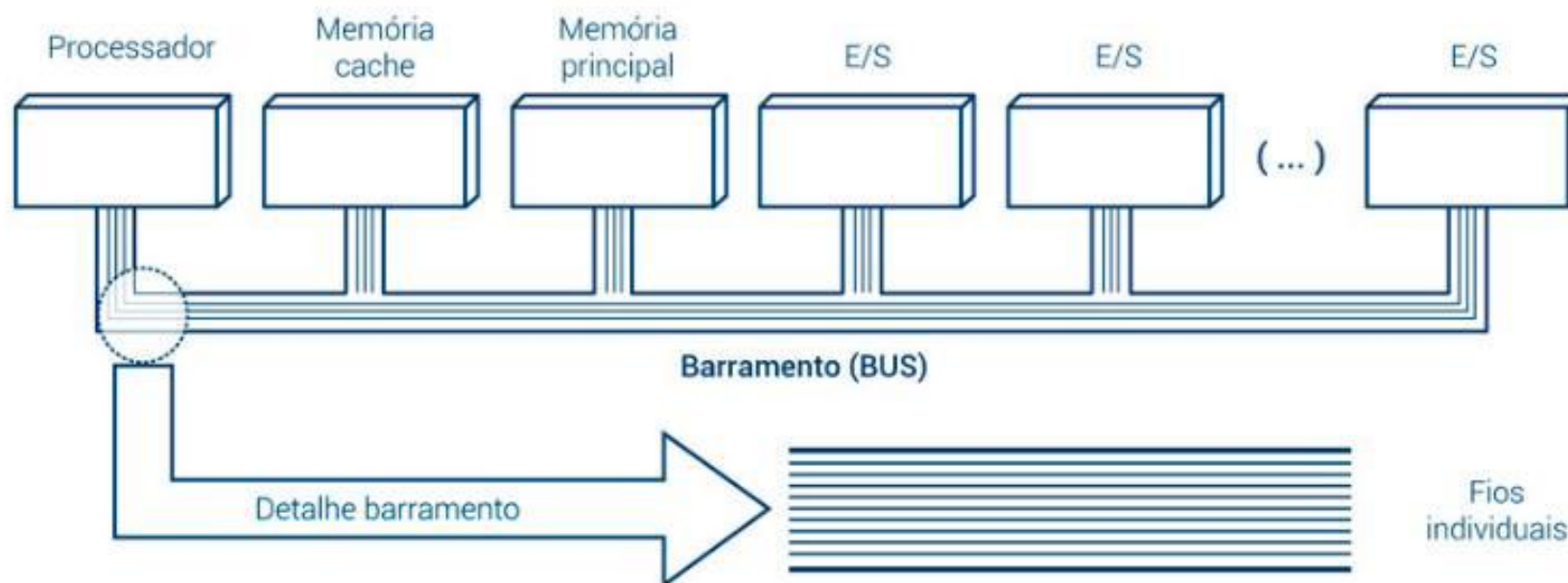
**Entrada/saída (E/S)**



# — Organização de um sistema de computação

## Barramento

Como se trata de componentes eletrônicos, a comunicação e o controle entre eles realiza-se por sinais elétricos que percorrem fios. Estes fios são chamados, em conjunto, de barramento.



# — Organização de um sistema de computação

## Barramento



Você sabe definir qual é a função do barramento?

### **Resposta**

Barramento é um conjunto de fios que têm por função transportar sinais de informação e sinais de controle e comunicação entre os componentes interligados.

# — Organização de um sistema de computação

## Barramento

### **Barramentos de dados (BD)**

São bidirecionais, transportam bits de dados entre o processador e outro componente, vice-versa.

### **Barramentos de endereços (BE)**

São unidirecionais, transportam bits de um endereço de acesso de memória ou de um dispositivo de E/S, do processador para o controlador do barramento.

### **Barramentos de controle (BC)**

Possuem fios que enviam sinais específicos de controle e comunicação durante uma determinada operação.

# — Organização de um sistema de computação

## Barramento

A soma dos fios do BC, do BD e do BE é igual ao total de pinos do processador ou total de furos do soquete, ou seja: **Totalpinos = BD + BE + BC.**

Em uma operação de transferência ou acesso (seja para leitura ou para escrita, exemplificado na (figura A), o barramento é único, embora dividido em grupos de fios que realizam funções diferentes (figura B):

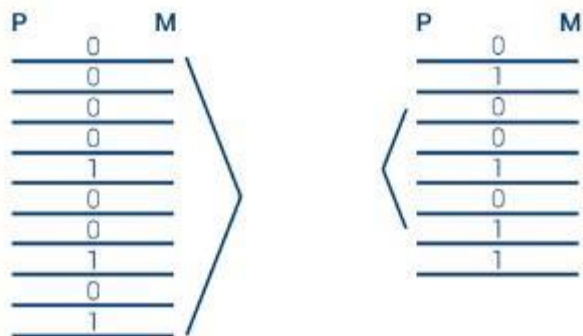


Figura A: operação de acesso do processador à memória principal.

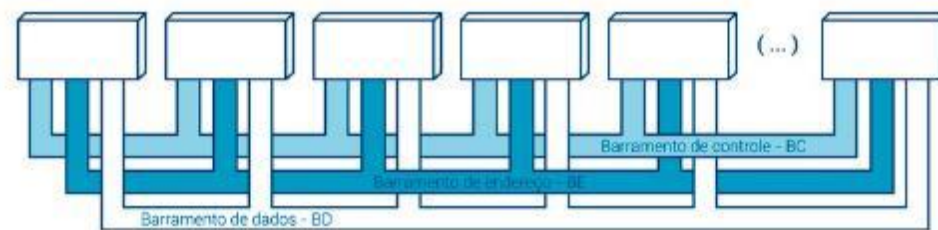
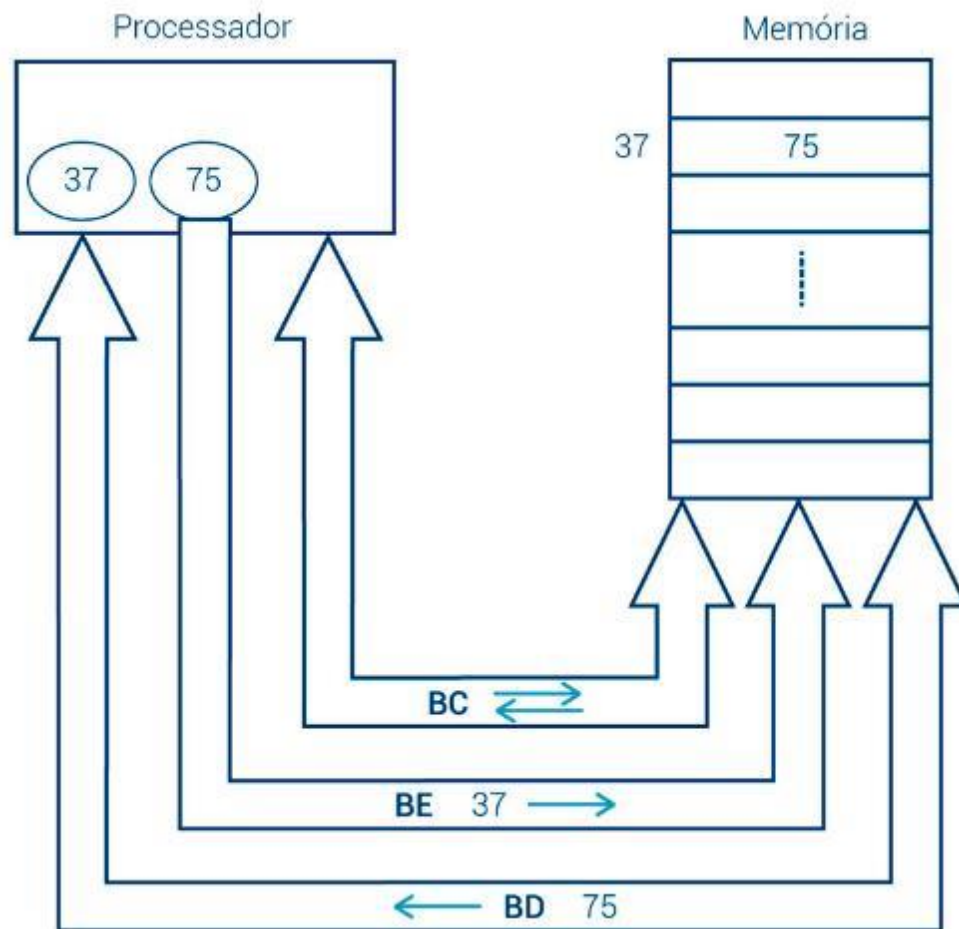


Figura B: Divisão do barramento em grupos de fios com funções diferentes.

# — Organização de um sistema de computação

## Barramento



# — Organização de um sistema de computação

## Processadores

Os **processadores** são projetados com a capacidade de realizarem diretamente (pelo hardware) pequenas e simples (primitivas) **operações**.

Executar operações aritméticas com 2 números (somar, subtrair, multiplicar, dividir)

Mover um número (dado) de um local para outro

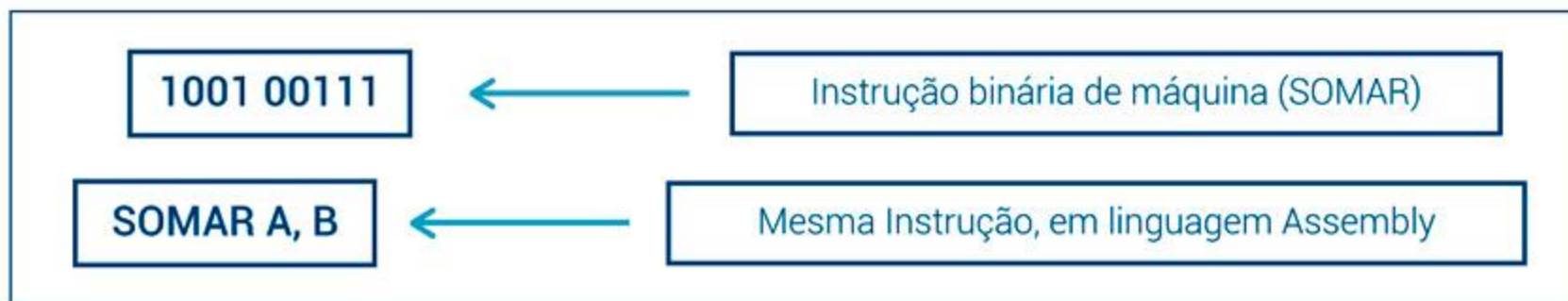
Mover um número (dado) de dispositivo de entrada ou de saída

Desviar a sequência de controle

# — Organização de um sistema de computação

## Processadores

A execução de um comando em linguagem de alto nível (por exemplo, Pascal), como  $X = A + B$  requer, primeiro, sua conversão para instruções de máquina e, em seguida, sua execução propriamente dita (figura), ou seja, somar o valor indicado por A com o valor indicado por B e armazenar o resultado no local indicado por A.



# — Organização de um sistema de computação

## Processadores

Os processadores, então:

### **Interpretam**

O que fazer (qual a operação – no exemplo anterior, a operação era SOMAR).

### **Executam**

A operação (como fazer – algoritmo para completar a operação propriamente dita).



# — Organização de um sistema de computação

## Processadores

Uma instrução de máquina consiste no conjunto de bits que identifica uma determinada operação primitiva a ser realizada diretamente pelo hardware, por exemplo, 1001 00111 00001.

Podemos citar como exemplos de operações primitivas:

- Operações aritméticas– Somar, subtrair, multiplicar e dividir;
- Operações lógicas– AND, OR, XOR;
- Operações de entrada e saída de dados;
- Operações de desvio de controle;
- Operações de movimentação de dados.

# — Organização de um sistema de computação

## Ciclo de instruções

### Conjunto de instruções

São todas as possíveis instruções que podem ser interpretadas e executadas por um processador. Por exemplo, o Intel 8080 tinha 78 instruções de máquina, o Pentium 4 tinha 247;

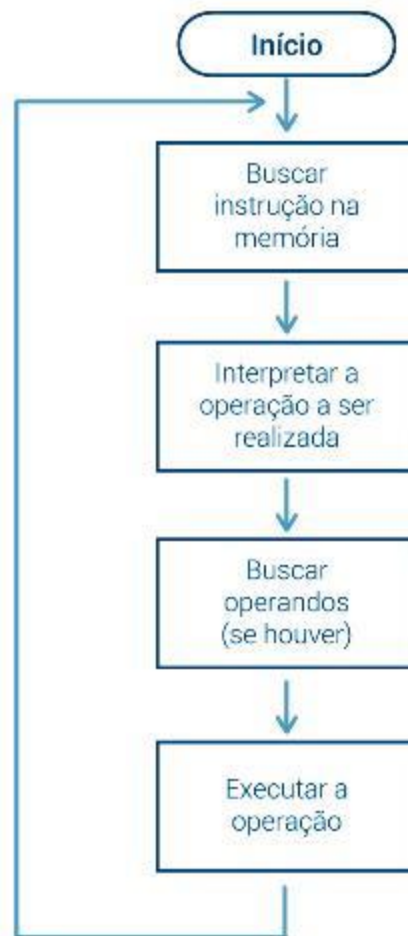


### Ciclo de instruções

É um conjunto de instruções de máquina sequencialmente organizadas para a execução de um programa.

# — Organização de um sistema de computação

## Ciclo de instruções



# — Organização de um sistema de computação

## Ciclo de instruções

O formato básico de uma instrução de máquina é constituído de duas partes. Vamos conferir!

**Código de operação (C.Op.):** Identificação da operação a ser realizada.

**Operando(s) (Op.):** Pode ter 1, 2 ou 3.



# — Tendências

- Progresso tecnológico em computação é notável desde a sua criação.
- Telefones celulares atuais superam desempenho de computadores de 1993.
- Avanços em tecnologia e design impulsionam rápido progresso.
- Reflexão necessária sobre conhecimentos para liderar na área de computação no futuro.



## — Subsistemas de processamento

### **Função processamento**

Responsável pelo processamento dos dados.

### **Função controle**

É a parte funcional que realiza as atividades de buscar a instrução; interpretar as ações; gerar os sinais de controle para ativar as atividades requeridas (dentro ou fora do processador).

# — Memória

## Tipos de memória

**Registradores**

**Memória cache**

**Memória principal– MP**

**Memória secundária**

**Memória virtual**

# — Memória

## Registradores

### **Registradores de dados**

Armazenam os dados que serão processados pelas unidades de cálculo, separados em unidades para números inteiros e números de ponto flutuante.

### **Registrador de dados de memória - RDM (Memory Buffer Register - MBR)**

Para transferências externas de dados.

### **Registrador de endereço - REM (Memory Address Register - MAR)**

Para transferências externas de endereços de memória.



# — Memória

## Registradores

### **Contador de instrução ou contador de programa - CI (Program Counter PC)**

Para buscar a próxima instrução.

### **Registrador de instrução - RI (Instruction Register - IR)**

Armazena instrução.

### **Segmentos**

Para armazenar endereços de Segmentos [apontam para determinados segmentos (programa, dados, pilha, etc.)].

### **Flags**

Podem ser usadas para indicar o resultado de certas instruções.

# — Memória

## Memória cache

- Memória cache fica entre a memória principal e o processador.
- Armazena dados em uso para transferência rápida ao processador.
- Contribui para aumentar a velocidade de acesso aos dados.
- Melhora o desempenho do processador ao reduzir o tempo de espera por dados na memória.

# — Memória



Por que muitas memórias e não apenas uma?

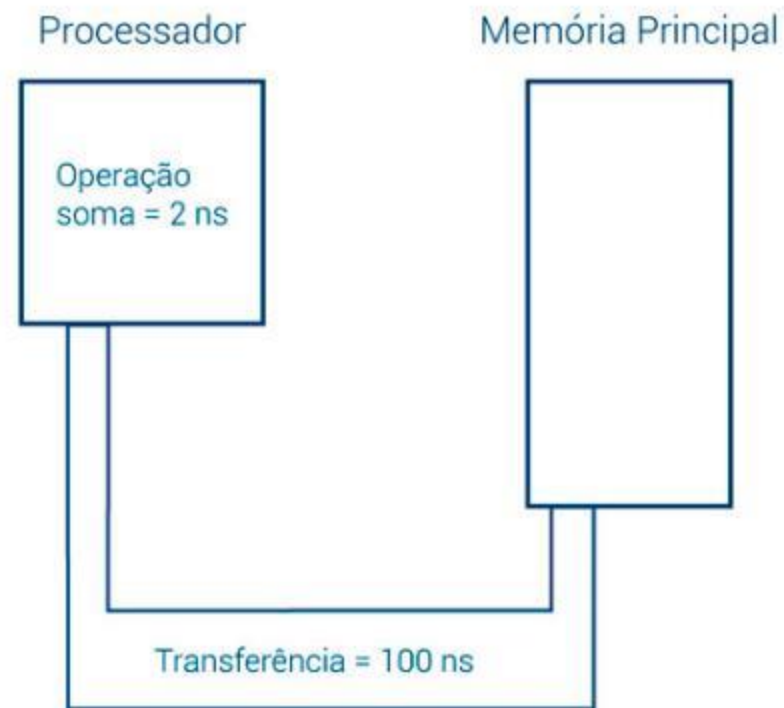
## **Resposta**

O ideal seria haver apenas uma memória nos computadores, com os seguintes requisitos: Tempo de acesso muito curto (semelhante ao do processador), grande capacidade de armazenamento, armazenamento permanente (não volátil) e baixo custo. Como, em termos práticos, hoje ainda não é possível a existência desse tipo de memória, faz-se necessária a adoção de soluções mais viáveis para compensar a diferença de velocidades que existe entre o processador e a MP.

Page 10 of 10

A imagem ao lado ilustra a comparação de velocidades que existem entre o processador e a MP (apenas processador e MP).

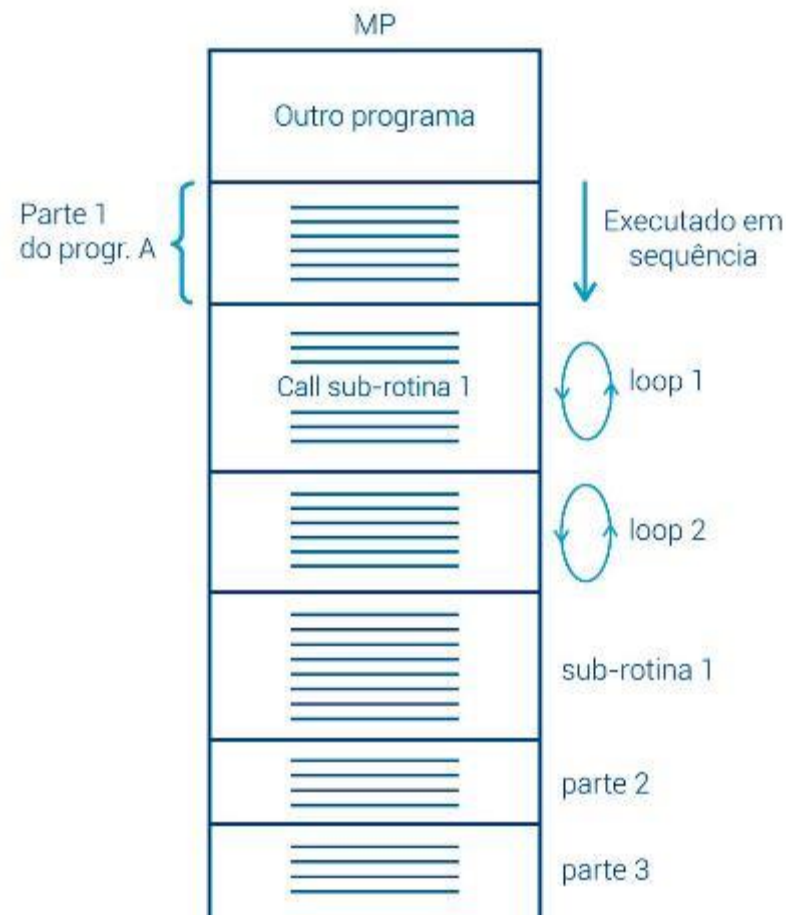
Há uma grande diferença de tempo entre a transferência da MP para o processador e este usar os dados. No exemplo, o processador gasta 2 nanosegundos para somar e espera 100 nanosegundos para receber novos dados.



# — Memória

## Princípio da localidade

- É um princípio de programação que determina o modo como as instruções são executadas (em sequência, durante certo tempo).
- Os programas são organizados de modo que as linhas de código costumam ser executadas em sequência.
- Apenas em alguns momentos a sequência é interrompida e o processo desvia da sequência, sendo esta retomada em seguida.



# — Memória

## Princípio da localidade

### Localidade espacial

Sempre que o processador realiza um acesso a um endereço de memória, é provável que o próximo acesso seja ao endereço contíguo seguinte.



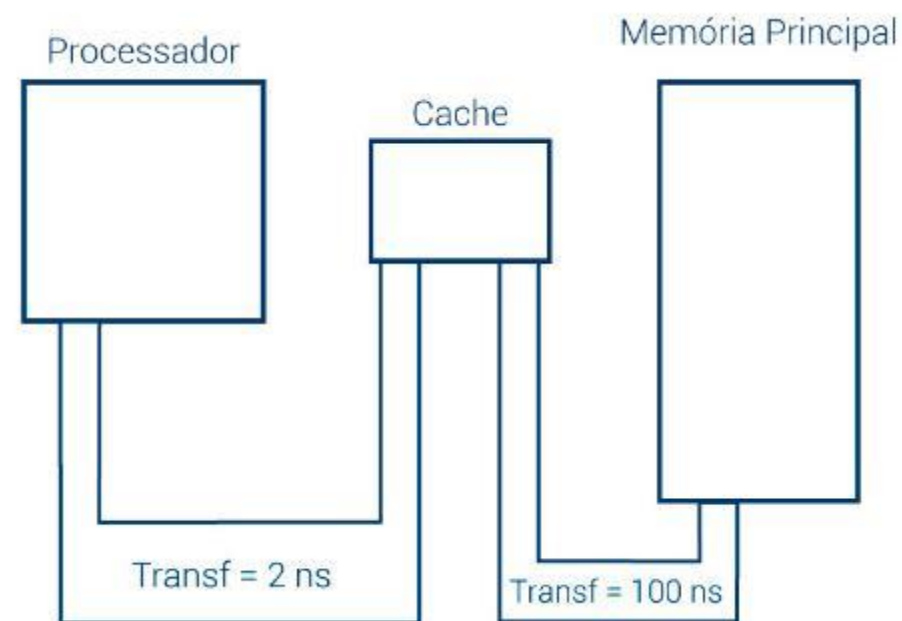
### Localidade temporal

Sempre que o processador realiza um acesso a um endereço de memória, é provável que, em curto tempo, ele acesse novamente o mesmo endereço.

# — Memória

## Princípio da localidade

- Memória cache atua como intermediária entre a MP e o processador.
- Princípio da localidade espacial reduz o tempo de espera por dados.
- Permite acesso mais rápido aos dados em uso.
- Melhora a eficiência do processador ao reduzir o tempo de espera por dados na MP.



# — Memória

## Memória principal (MP)

- MP armazena programas e dados para acesso pelo processador.
- Memórias antigas usavam acesso sequencial.
- Endereço de acesso relativo ao endereço inicial.



# — Memória

## Memória principal (MP)

A memória principal (RAM) permite a realização de duas operações:

### **Escrita (armazenar)**

O dado anteriormente armazenado é apagado.

### **Leitura (recuperar)**

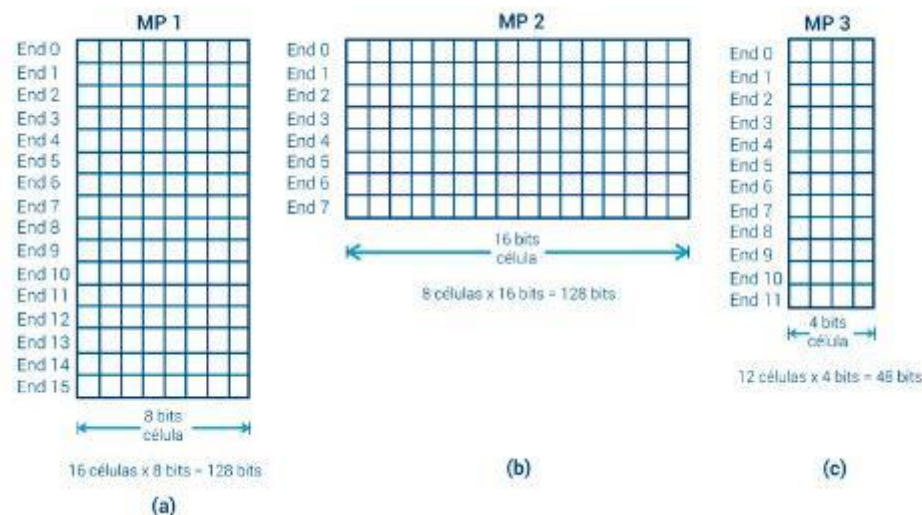
Normalmente se recupera uma cópia do dado.

# — Memória

## Memória principal (MP)

A memória é organizada como um conjunto de N partes iguais, com cada parte possuindo um conteúdo fixo de M bits.

O valor de M depende do tipo de memória. Usualmente é 8 bits (1 Byte) nas memórias RAM, mas existem valores maiores para outras memórias.



# — Memória

## Memória principal (MP)

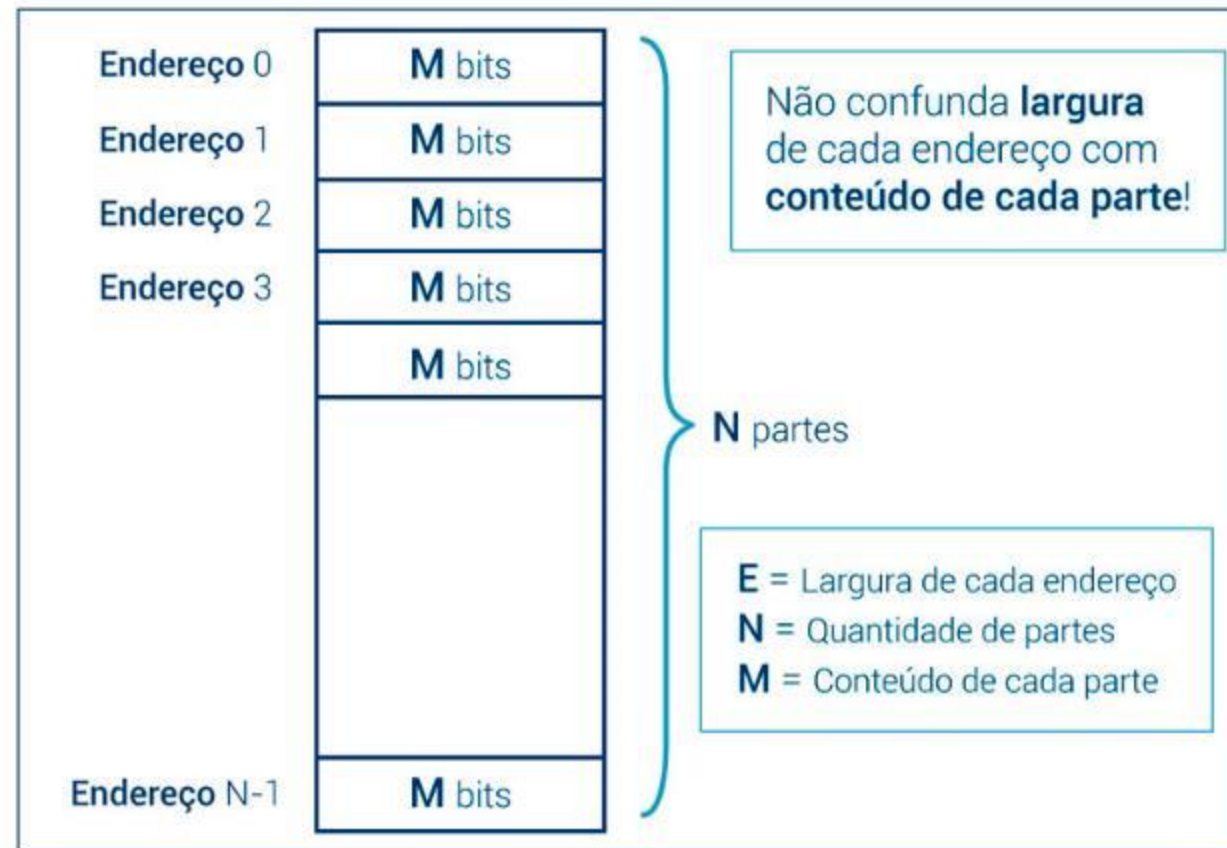
Cada parte (chama-se célula ou, em alguns casos, palavra nas memórias RAM, linha nas memórias cache, setor nos HDs etc.) é identificada por um número, chamado endereço. Todos os  $N$  endereços têm mesma largura de endereço, de  $E$  bits.

Uma memória com  $N$  partes também possui  $N$  endereços. O cálculo de  $N$  no endereçamento pode ser realizado da seguinte maneira:

$$2^E = N$$

# — Memória

## Memória principal (MP)



# — Memória

## Memória principal (MP)

As memórias eletrônicas que empregam o acesso aleatório podem ser fabricadas para permitir duas aplicações:

**Para leitura e escrita**

Read/Write – R/W

**Somente para leitura**

Read Only Memory – ROM

# — Memória

## Memória principal (MP)

As memórias RAM são constituídas de dois tipos:

### **SRAM (Static Random-Access Memory)**

- Cada bit é constituído de 5 a 7 transístores;
- Não requer recarregamento, sendo, por isso, mais rápidas, mas ocupam mais espaço e são mais caras;
- Usadas como memória cache.

### **DRAM (Dynamic Random-Access Memory)**

- Cada bit é constituído por 1 capacitor e 1 transístor;
- O capacitor serve para representar o valor do bit e o transístor para ser usado nas leituras/escritas;
- Como o capacitor se descarrega, é preciso recarregar periodicamente (sinal de refresh – gasta tempo);
- Usadas como memória principal.

# — Memória

## Memória principal (MP)

As memórias dinâmicas podem ser de dois tipos, vejamos a seguir:

### **Memórias dinâmicas assíncronas**

Não são sincronizadas com o processador, por exemplo, Dynamic RAM (DRAM), Fast Page Mode (FPM), Extended Data Out DRAM (EDO), Burst Extended Data Out DRAM (BEDO).

### **Memórias Dinâmicas Síncronas**

Sincronizadas com o processador, evitam que o processador espere os dados, por exemplo, Synchronous DRAM (SDRAM), Double Data Rate (DDR), Double Data Rate 2 (DDR2).

# — Memória

## Memória principal (MP)

Hoje em dia, é comum o uso de memórias DDR SDRAM, pois as memórias **Single Data Rate** (SDRAM) só transferem dados na subida do sinal de clock; Já as memórias **Double Data Rate** (DDR-SDRAM) transferem dados na subida e na descida do sinal de clock, dobrando a taxa de transferência de dados (data rate); Assim, uma **memória DDR-SDRAM** operando num clock de 100MHz (real) consegue desempenho equivalente a 200MHz (efetivo).

Também existe a classificação quanto ao tipo de encapsulamento das memórias (formatos dos módulos):

### **SIMM (Single In Line Memory Module)**

O contato elétrico de um lado é igual ao do outro lado.

### **DIMM (Dual In Line Memory Module)**

Os contatos dos dois lados são independentes.



# — Memória

## Memória secundária

Objetiva o armazenamento persistente (permanente) aos programas de usuário e seus dados.

O emprego de diferentes tecnologias para compor os diferentes tipos de memórias da hierarquia pode ser feito através de parâmetros para análise, tais como:

### **Tempo de acesso**

Também conhecido como tempo de acesso para leitura ou tempo de leitura.

### **Ciclo de memória**

É outro parâmetro (apenas para memórias eletrônicas), indica o tempo entre 2 operações sucessivas de leitura ou escrita.

# — Memória

## Memória secundária

**Capacidade**

**Volatilidade**

**Custo**

**Tecnologia de fabricação**

Memórias de semicondutores,  
memórias de meio magnético,  
memória de meio óptico.

**Temporariedade**

Permanente, transitório

# — Memória

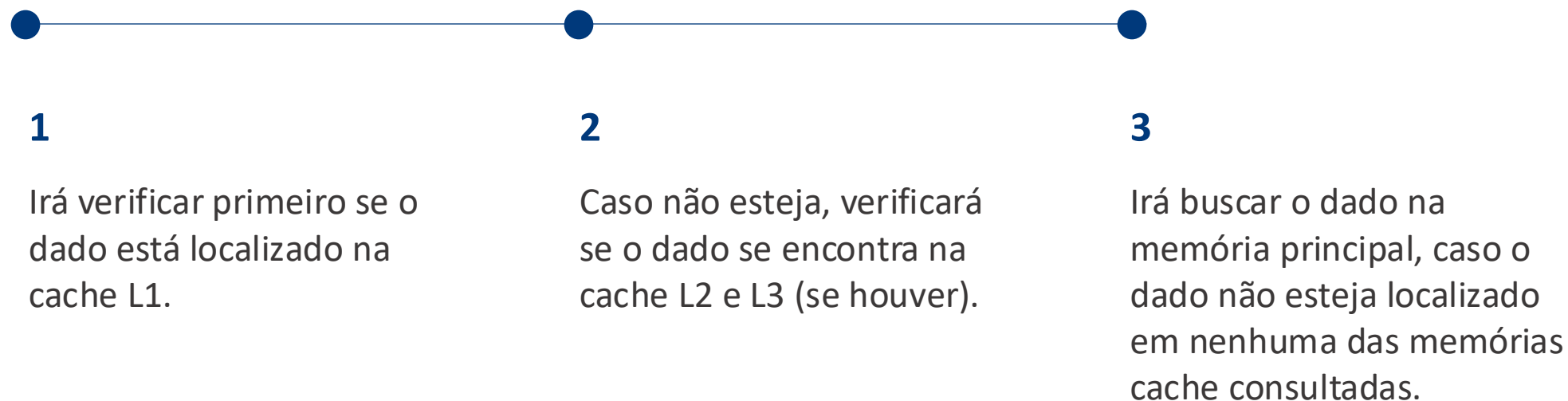
## Memória secundária

	Internamente no processador			Na placa-mãe	Externa
	Registrador(es)	Cache L1	Cache L2	Memória principal (RAM)	Memória secundária (pen drives, discos etc.)
Tecnologia de fabricação	Eletrônica	Eletrônica	Eletrônica	Eletrônica	Várias tecnologias (ótica, magnética, eletrônica)
Volatilidade	Volátil	Volátil	Volátil	Volátil (parte é não volátil – ROM)	Não volátil
Tempo de acesso	Ex.: 1 a 2ns	Ex.: 2 a 6ns	Ex.: 2 a 6ns	Ex.: 5 a 10ns	Ex.: de 8ns até alguns segundos
Capacidade	32 ou 64 bits	8KB, 256KB, ...	2MB, 8MB, ...	4GB, 8GB, ...	Depende da mídia

# — Memória

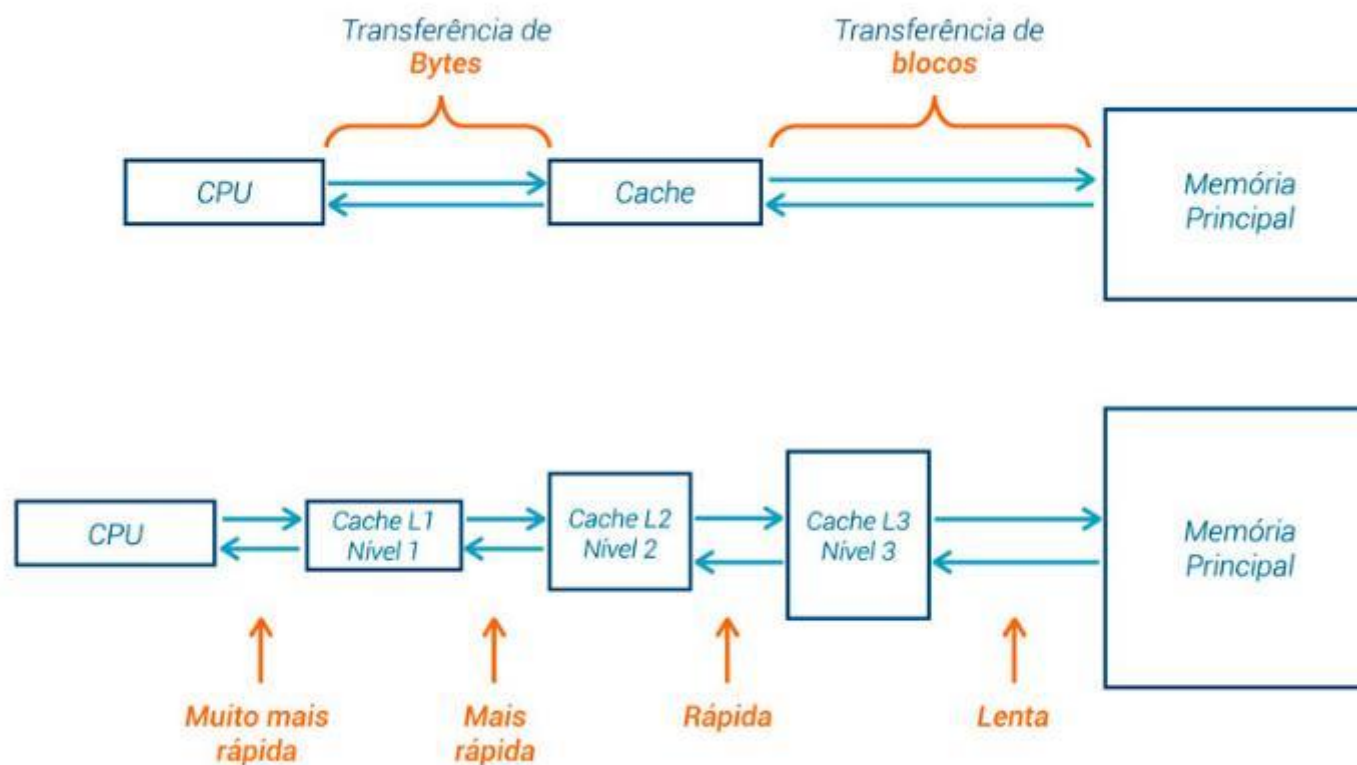
## Memória secundária

A sequência de transferência de dados realizada entre o processador e as memórias em um sistema computacional é hierárquica. Em uma operação de leitura, o processador:



# — Memória

## Memória secundária



# — Subsistema de entrada e saída (E/S)

## Objetivos e funções

- Subsistema E/S conecta o mundo exterior ao processador/memória.
- Dispositivos de E/S são chamados periféricos.
- Funciona como interface entre o mundo exterior e o processador.

São funções do subsistema de E/S:

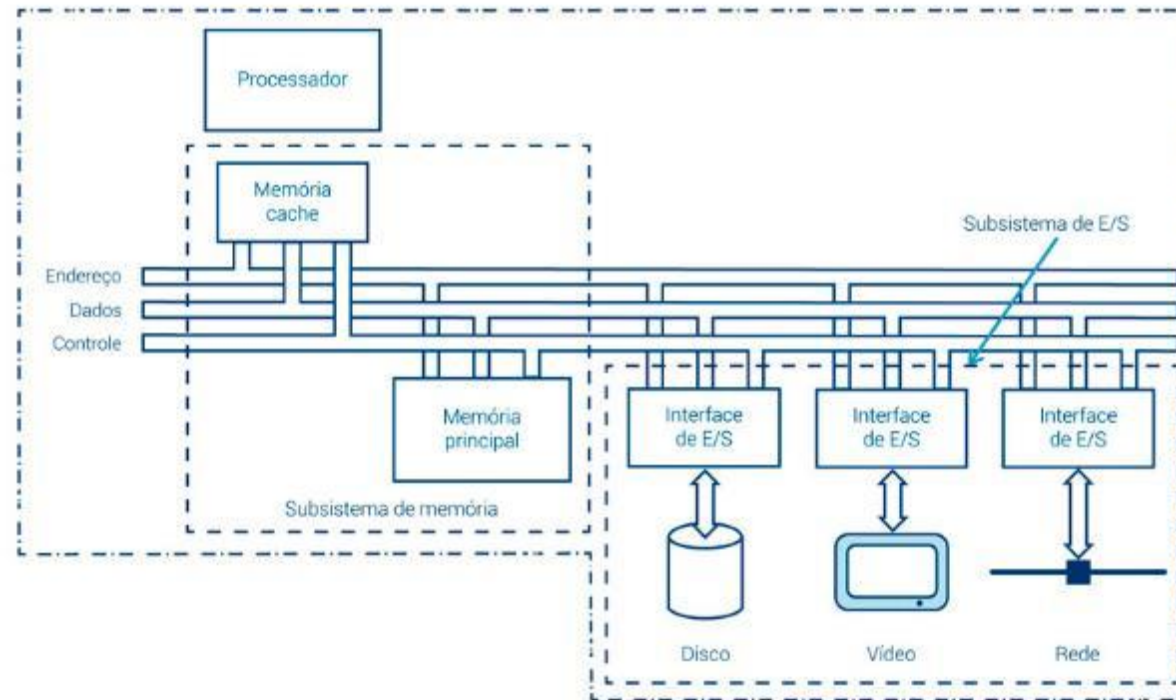
Receber ou enviar  
informações do/para o meio  
exterior..

Converter as informações (de  
entrada ou de saída) em uma  
forma inteligível para a  
máquina (se estiver  
recebendo) ou para o  
operador (estar enviando).

# — Subsistema de entrada e saída (E/S)

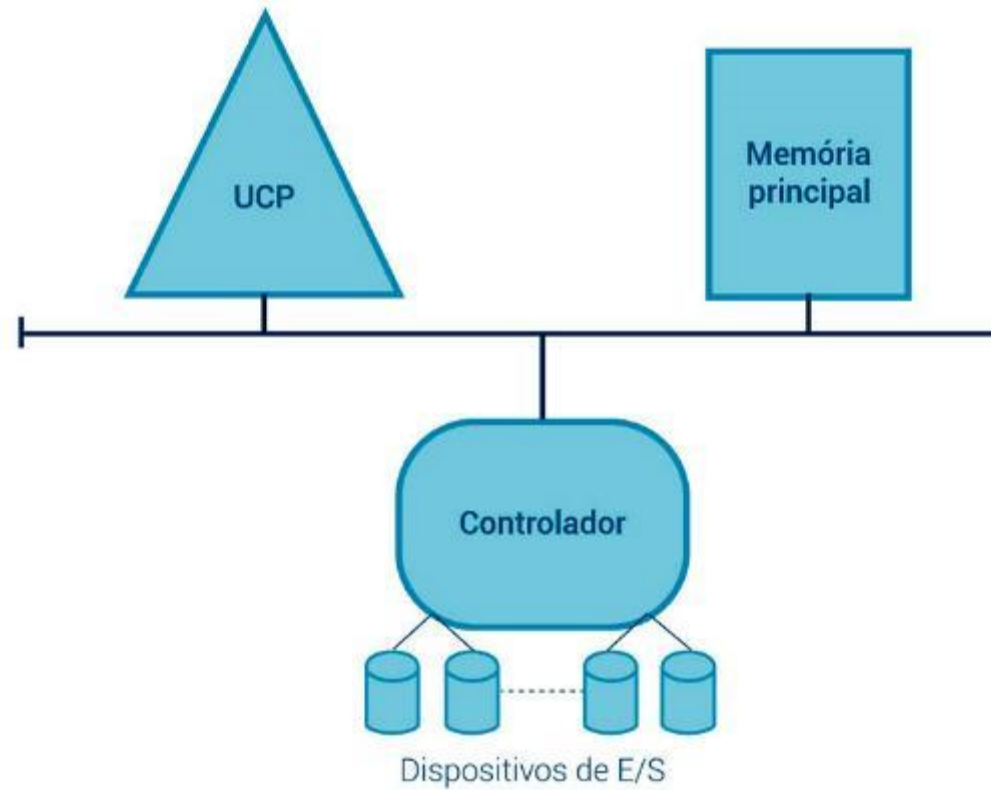
## Dispositivo e interface

Todo componente de E/S é constituído de 2 partes: O **dispositivo** propriamente dito; um componente denominado **interface**.



## — Subsistema de entrada e saída (E/S)

Dispositivo e interface

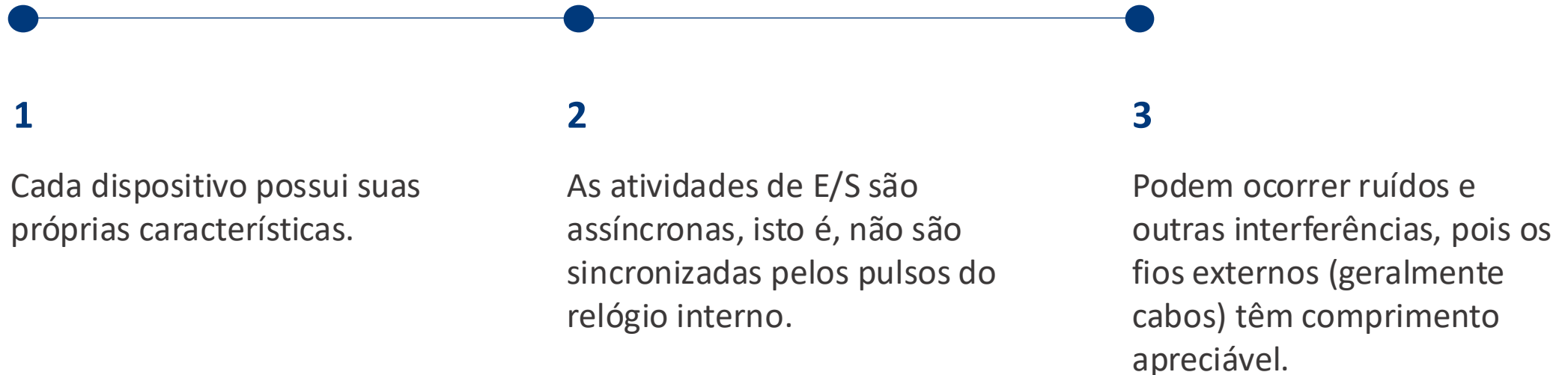




# — Subsistema de entrada e saída (E/S)

## Dispositivo e interface

A necessidade do emprego de interfaces tem origem em diversos fatores:



# — Subsistema de entrada e saída (E/S)

## Dispositivo e interface



# — Subsistema de entrada e saída (E/S)

## Dispositivo e interface

Exemplos de **dispositivos** ou **periféricos** são:

### Dispositivos de entrada

Teclado, mouse, mesa digitalizadora, scanner etc.

Exemplo: O teclado possui o seguinte processo de funcionamento:

- Detecção do pressionamento de uma tecla;
- Confirmação do pressionamento;
- Geração do código de identificação da tecla;
- Sinal de interrupção (grosso modo, incluir o pedido de processamento da tecla no meio de outras execuções que estão sendo realizadas);
- O programa de controle (BIOS) processa o significado daquela tecla (por exemplo, um caractere), e envia o resultado para a aplicação que está em execução; utilizará o resultado desse processamento.

# — Subsistema de entrada e saída (E/S)

## Dispositivo e interface

### **Dispositivos de saída**

Impressora, caixa de som, monitor.

Exemplo: A impressora pode ser dos tipos:

- Impacto (esfera, matricial);
- Sem impacto (jato de tinta, laser, sublimação de tinta).

### **Dispositivos de entrada e saída**

Disco magnético, pen drive, SD Card, disco SSD etc).

# — Subsistema de entrada e saída (E/S)

## Disco magnético

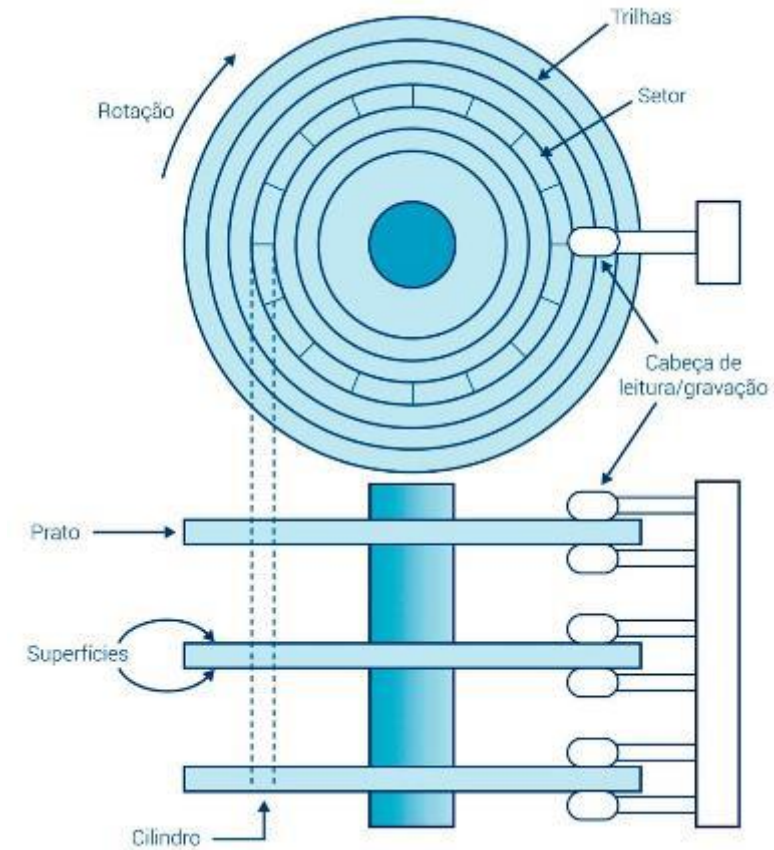
- É um prato não magnético revestido com material magnetizável.
- Dados gravados/recuperados por cabeça condutora.
- Operação com cabeçote parado e prato girando.



# — Subsistema de entrada e saída (E/S)

## Disco magnético

- Algumas unidades de disco acomodam vários pratos empilhados.
- Cabeça móvel para leitura/gravação em cada superfície.
- Cabeças fixadas mecanicamente, movendo-se juntas.
- Conjunto de faixas na mesma posição no prato chamado cilindro.



# — Subsistema de entrada e saída (E/S)

## Disco magnético

- Comunicação inicial processador-periférico por instruções de E/S.
- Dependência forte entre processador e dispositivos de E/S.
- Surgimento de controlador/interface para agir independentemente.
- Técnicas: E/S controlada por programa e por interrupção.
- Introdução da técnica DMA (Direct Access Memory) para transferência eficiente de grandes volumes de dados.

# — Subsistema de entrada e saída (E/S)

## Driver de dispositivo

Cada dispositivo de E/S ligado ao computador precisa de algum código específico do dispositivo para controlá-los. A esse código dá-se o nome de driver de dispositivo.

### Exemplo A

Um driver de disco deve saber sobre setores, trilhas, cilindros, cabeçotes, movimento do braço etc.

### Exemplo B

Um driver de mouse deve aceitar informações dizendo o quanto se moveu e qual botão foi pressionado.



# — Subsistema de entrada e saída (E/S)

## Driver de dispositivo

Os dispositivos podem transmitir dados em grupos de bits (paralela) ou bit por bit em série (ou serial):

### Transmissão paralela

Na transmissão paralela, um grupo de bits é transmitido de cada vez, cada um sendo enviado por uma linha separada de transmissão.



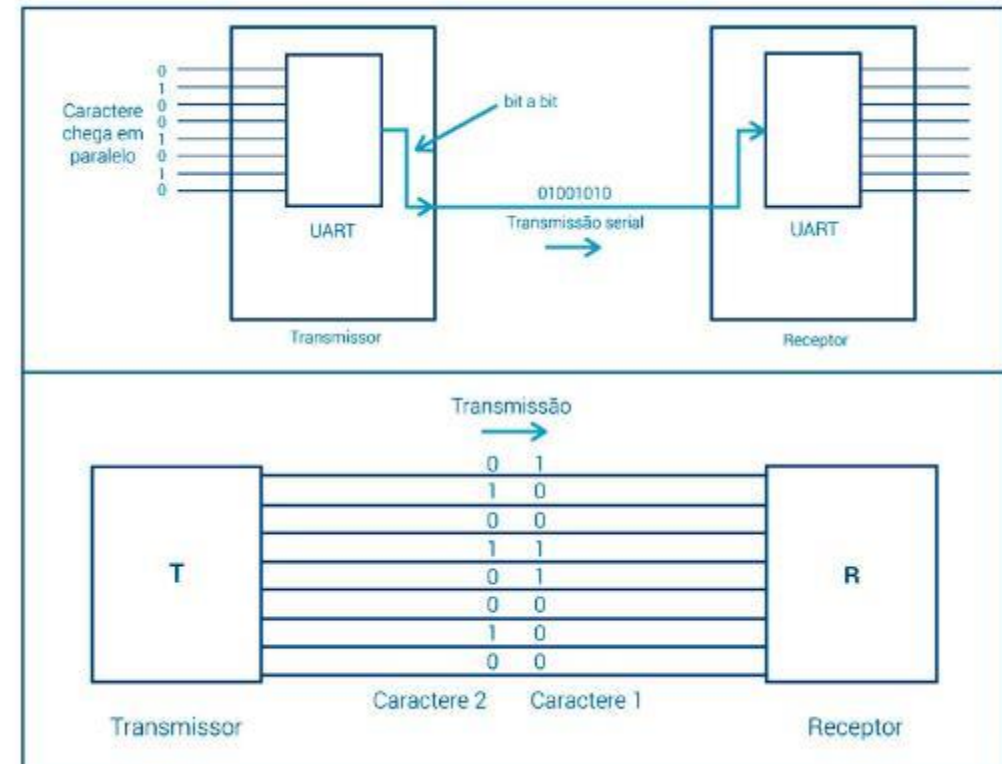
### Transmissão serial

Na transmissão serial, o periférico é conectado ao dispositivo controlador por uma única linha de transmissão de dados, um bit de cada vez.

# — Subsistema de entrada e saída (E/S)

## Driver de dispositivo

- Transmissão em série é mais econômica e comum atualmente.
- Transmissão paralela causava problemas de deslizamento em altas taxas.
- Atualmente, USB, SATA substituem transmissões paralelas.
- Uso interno em processador e placas-mãe persiste para ligações internas.



## — Tendências

- Futuros processadores terão núcleos especializados para cálculos específicos.
- Foco na melhoria do custo de desempenho energético.
- Avanços notáveis em todas as camadas da pirâmide de hierarquia de memória.
- Estratégias como a especialização são aplicadas para otimização de tarefas.

## — Tendências

A memória cache possui velocidades e tamanhos cada vez maiores, que auxiliam no desempenho dos processadores.

As memórias DDR SDRAM possuem latência (atraso que existe para que o dado seja acessado pelo processador na memória) cada vez menor e maior capacidade de armazenamento, sendo utilizadas inclusive em placas gráficas de alto desempenho.

Os dispositivos de memória secundária, principalmente discos, estão sendo fabricados com novas tecnologias diferentes da tecnologia dos antigos discos rígidos, por exemplo, Solid State Disks (SSDs), o que confere uma maior velocidade de acesso aos dados.

# — Sistema operacional

## Definição

- O único programa executado em modo núcleo (kernel), possuindo acesso completo ao hardware e execução de qualquer instrução possível.
- Um programa de controle que comanda a execução dos programas do usuário e as operações dos dispositivos de E/S.
- Um gerenciador de recursos (hardware) que dirige e aloca as partes de todo um sistema complexo.

# — Sistema operacional

## Definição

O computador pode ser compreendido como uma máquina de níveis ou máquina de camadas, onde existem inicialmente duas camadas:

### Hardware

É a camada física, refere-se aos componentes físicos, tais como placas, fios, componentes eletrônicos.

### Software

É a camada lógica, refere-se a todas as abstrações do sistema de computação, tais como aplicativos e seus dados.

# — Sistema operacional

## Definição

A partir da imagem ao lado, podemos dizer que:

- Os usuários são as pessoas, máquinas, outros computadores.
- Os aplicativos definem as maneiras como os recursos são usados, para resolver os problemas de computação dos usuários.
- O sistema operacional controla e coordena o uso do hardware entre os vários programas de aplicação, para os diversos usuários.
- O hardware fornece recursos básicos de computação CPU, memória, dispositivos de E/S.



# — Sistema operacional

## Tipos de sistemas operacionais





# — Sistema operacional

## Tipos de sistemas operacionais

### **Sistemas monoprogramáveis/monotarefas**

O processador, memória e periféricos permanecem dedicados exclusivamente à execução de somente um programa. Um exemplo é o MS-DOS.

### **Sistemas multiprogramáveis/multitarefas**

Os recursos computacionais são compartilhados entre os diversos usuários e aplicações. Exemplos são Windows, Linux, macOS.

### **Sistemas com múltiplos processadores**

Possuem duas ou mais UCPs (CPUs) interligadas e trabalhando em conjunto. Sistemas como Linux e Windows oferecem esse suporte.

## — Sistema operacional

Sistemas multiprogramáveis / multitarefas



# — Sistema operacional

## Tipos de sistemas operacionais

### Sistemas batch

Processam tarefas de rotina sem a presença interativa do usuário. Exemplo: Processamento de apólices de companhia de seguro; relatório de vendas de uma cadeia de lojas.

### Sistemas de tempo compartilhado

Permitem que múltiplos usuários remotos executem suas tarefas simultaneamente no computador. Exemplo: Realização de consultas a um banco de dados.

### Sistemas em tempo real

Possuem o tempo como parâmetro fundamental. Exemplo: Linha de montagem de um carro.

# — Sistema operacional

## Tipos de sistemas operacionais

### **Sistemas fortemente acoplados**

Existem vários processadores compartilhando uma única memória física e dispositivos de E/S sendo gerenciados por apenas um SO.



### **Sistemas fracamente acoplados**

Existem dois ou mais sistemas computacionais conectados através de linhas de comunicação.

# — Sistema operacional

## Tipos de sistemas operacionais

- Sistemas pessoais e estações de trabalho agora têm múltiplos processadores.
- Redes de computadores exemplificam sistemas fracamente acoplados.
- Em sistemas distribuídos, terminais na rede são tratados como um sistema único.
- Clusters consistem em servidores interligados, como na web.

## — Tendências

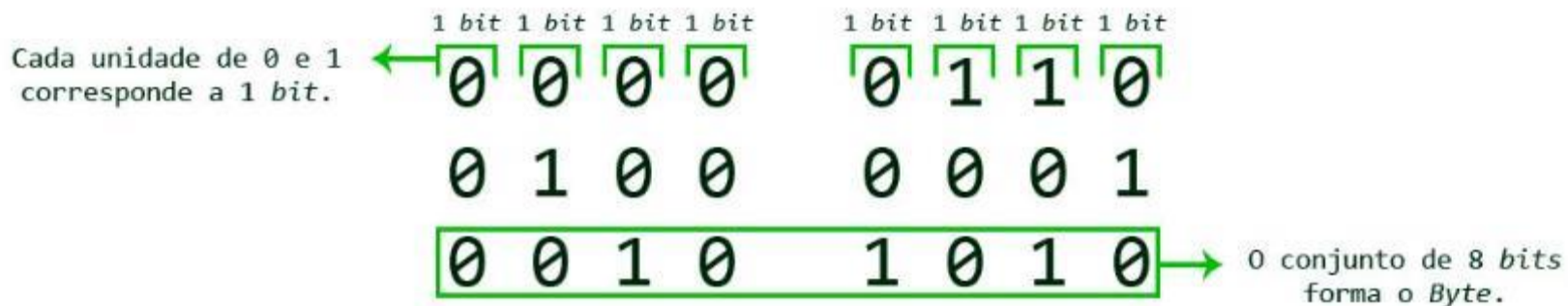
- Sistemas operacionais são virtualizados e acessíveis na nuvem.
- Virtualização de servidores reproduz um computador inteiro em software.
- Contêineres compartilham um kernel do SO, tornando-os menores e mais rápidos.
- Contêineres executam apenas o aplicativo e suas dependências.

# Representação de dados



# — Unidades de informação

- Dados eletrônicos são armazenados e movidos por voltagem ou corrente.
- Representação binária com valores 0 e 1 é fundamental.
- Agrupamento de bits em conjuntos de oito forma um byte.





# — Unidades de informação

A regra geral para a conversão é:

## De bit para byte

Dividimos o valor por 8.

## De byte para bit

Multiplicamos o valor por 8.

- Um conjunto ordenado de bytes, que representa uma informação útil para os computadores, constitui uma **palavra**.
- Um conjunto estruturado de palavras forma um **registro**.
- Um conjunto organizado de registros forma um **arquivo**.
- Um conjunto organizado de arquivos forma um banco de **dados**.

# — Unidades de informação

## Múltiplos e submúltiplos

Unidade	Valor em potência de 2	Valor unitário	Valor em potência de 10	Valor unitário
<i>1K</i> (quilo)	$2^{10}$	<i>1024</i>	$10^3$	<i>1.000</i>
<i>1M</i> (mega)	$2^{20}$	<i>1.048.576</i>	$10^6$	<i>1.000.000</i>
<i>1G</i> (giga)	$2^{30}$	<i>1.073.741.824</i>	$10^9$	<i>1.000.000.000</i>
<i>1T</i> (tera)	$2^{40}$	<i>1.099.511.627.776</i>	$10^{12}$	<i>1.000.000.000.000</i>

# — Unidades de informação

## Múltiplos e submúltiplos

Os valores unitários podem ser obtidos a partir da notação dos múltiplos da grandeza em:

### Potência de 2

Expressam os valores (em decimal) que, de fato, são manipulados pelo computador em binário.

### Potência de 10

Exprimem os valores (em decimal) equivalentes àqueles manipulados pelo computador em binário, embora empreguem os múltiplos no sistema decimal comumente utilizado no nosso dia a dia.

# — Unidades de informação

## Múltiplos e submúltiplos

Suponha que você queira mudar o prefixo do valor a seguir:

*52,9 GB  $\Rightarrow$  TB ?*

Usando como referência os valores mostrados na tabela anterior, veja uma maneira de realizar isso:

# — Unidades de informação

## Múltiplos e submúltiplos



**1**

Escrever os prefixos em sua sequência (B, KB, MB etc.).



**2**

Colocar o valor (no caso, 52,9) embaixo da unidade com o prefixo dado. Esse será seu ponto de partida ou ponto de referência relacionado ao novo valor obtido.

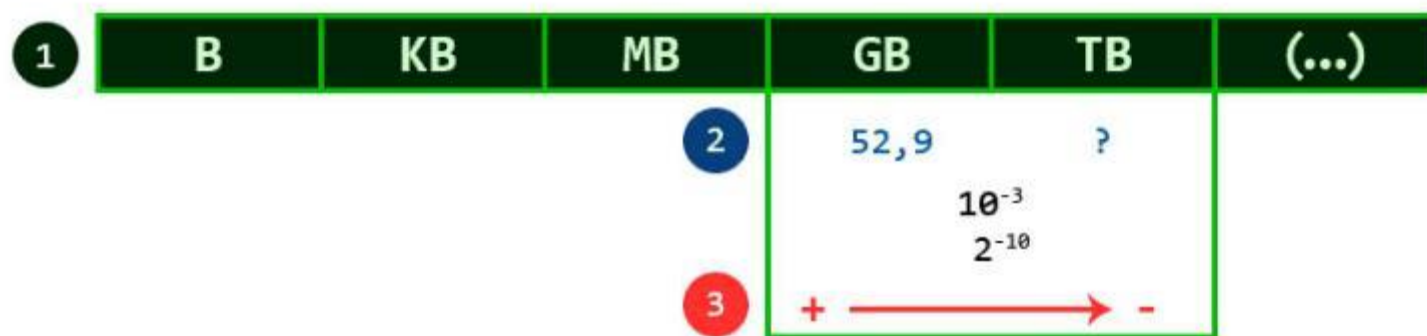


**3**

Desenhar uma seta no sentido do prefixo pretendido. Do lado esquerdo dela, fica o operador matemático "+"; do direito, o sinal "-". O sinal resultante fica ao lado do valor do expoente da base utilizada.

# — Unidades de informação

## Múltiplos e submúltiplos



### Potência de 2

$$52,9 \times 10^{-3} = 0,0529 \text{ TB}$$

Isso ocorrerá se considerarmos os múltiplos em potência de 10.

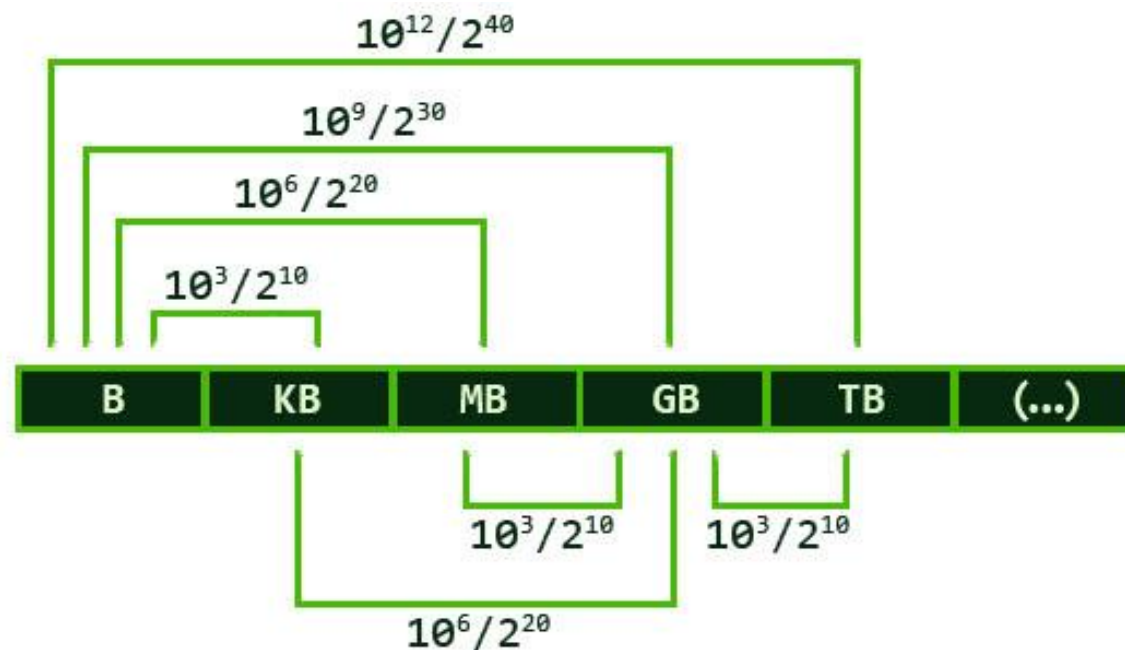
### Potência de 10

$$52,9 \times 2^{-10} \text{ TB} \sim 0,05166$$

# — Unidades de informação

## Múltiplos e submúltiplos

Entre dois prefixos de unidades de medida vizinhos tipicamente utilizados na computação, o expoente da base decimal varia, em geral, de 3 em 3 (ou de 10 em 10, se a base for binária).



# — Unidades de informação

## Múltiplos e submúltiplos

Suponhamos que você deseje mudar o prefixo do valor a seguir:


*143 GB  $\Rightarrow$  KB ?*



# — Unidades de informação

## Múltiplos e submúltiplos

B	KB	MB	GB	TB	(...)
	?		143		
		$10^6$ $2^{20}$			



### Potência de 2

$$143 \times 2^{20} = 149.946.368 \text{ KB}$$

### Potência de 10

$$143 \times 10^6 = 143.000.000 \text{ KB}$$

# — Representação das informações no computador

Para cada tipo de dado, serão amplamente utilizados os seguintes métodos padrão de codificação:



## **Padrão ASCII**

O padrão ASCII pode associar um número binário de 7 bits a cada um dos 128 caracteres distintos possíveis.



## **Formato de arquivo MP3**

O formato de arquivo MP3 especifica o modo de codificar um arquivo de áudio como uma sequência de 0s e 1s.



## **Formato de imagem .png**

O formato de imagem .png especifica os pixels das imagens digitais como uma sequência de 0s e 1s.

# — Sistemas de numeração

## Bases de numeração

### **Decimal**

Base numérica mais utilizada no nosso cotidiano.

### **Binária**

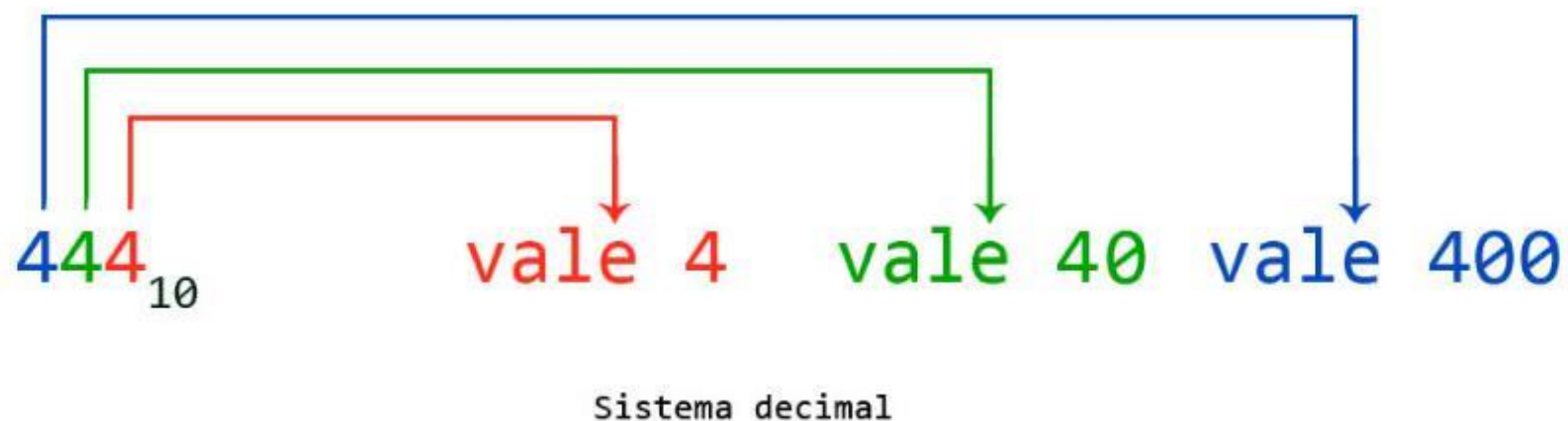
Base usada pelos computadores no processamento de dados.

### **Octal e hexadecimal**

Múltiplos da base binária.

# — Sistemas de numeração

## Posicional



Uma base é a quantidade de símbolos de um sistema posicional. Seu número indica o total de símbolos dela.

# — Sistemas de numeração

## Posicional

$$36_{10} = 3 * 10^1 + 6 * 10^0$$

10 é a base  
1 é a posição

10 é a base  
0 é a posição

Todo número cresce de valor da direita para a esquerda

# — Sistemas de numeração

## Posicional

### **Da direita para a esquerda**

Cresce a partir do valor 0 (seguido do 1 até o último algarismo válido).

### **Retorna a 0**

Quando a contagem chega ao último algarismo válido de uma posição, ela retorna a 0 e cresce uma unidade para a esquerda.

## Não posicional

Todo algarismo tem valor fixo independentemente de sua posição no número.

# — Operações aritméticas com números inteiros em qualquer base

## Adição

- Soma de parcelas feita algarismo por algarismo.
- Se a soma ultrapassar a base, subtrai-se o excesso e passa-se 1.
- Unidade passada ("vai 1") para a posição à esquerda, repetindo o processo.

# — Operações aritméticas com números inteiros em qualquer base

## Adição

$$\begin{array}{r} 11 \\ 574 \\ + 893 \\ \hline 1467 \end{array}$$

### Na base 10

Base numérica mais utilizada no nosso cotidiano.

$$\begin{array}{r} \text{"vai 1"} \leftarrow \boxed{11} \\ 1101 \\ + 1110 \\ \hline 11011 \end{array}$$

### Na base 2

Na base binária,  $1 + 1 = 0$ , com o "vai 1" = 10  
Logo, veja na imagem:

$$\begin{array}{r} 111 \\ CD94 \\ + 8E77 \\ \hline 15C0B \end{array}$$

### Na base 16

Antes de abordarmos as regras para cálculos com números hexadecimais, é importante sabermos a equivalência dos números decimais e seus respectivos hexadecimais.



# — Operações aritméticas com números inteiros em qualquer base

## Adição

Uma maneira de pensar a adição em qualquer base é:



**1**

Somar os números na posição na base 10.



**2**

Diminuir o número obtido nessa base pelo valor dela se ele passar do seu último algarismo.



**3**

Não se esquecer do “vai 1”.

# — Operações aritméticas com números inteiros em qualquer base

## Subtração

- Subtração algarismo por algarismo, da direita para a esquerda.
- Se o minuendo for maior que o subtraendo, faz-se a subtração normalmente.
- Se o minuendo for menor que o subtraendo, "pede-se emprestado" da próxima posição à esquerda.
- Processo continua até a última posição.

### **Minuendo maior que subtraendo**

Realiza a operação e registra sua diferença embaixo.

### **Minuendo menor que subtraendo**

É subtraída uma unidade do algarismo à esquerda (trata-se do famoso “pedir 1 emprestado”).

# — Operações aritméticas com números inteiros em qualquer base

## Subtração

- Subtração algarismo por algarismo, da direita para a esquerda.
- Se o minuendo for maior que o subtraendo, faz-se a subtração normalmente.
- Se o minuendo for menor que o subtraendo, "pede-se emprestado" da próxima posição à esquerda.
- Processo continua até a última posição.

# — Operações aritméticas com números inteiros em qualquer base

## Subtração

$$\begin{array}{r} \phantom{0}12 \\ 3\cancel{2}10 \\ 4\cancel{3}\cancel{0}7 \\ - 2734 \\ \hline 1573 \end{array}$$

Na base 10

$$\begin{array}{r} \phantom{0}11\phantom{0}29\phantom{0}8\phantom{0}20 \\ \phantom{0}C\phantom{0}D\phantom{0}9\phantom{0}4 \\ - 8E77 \\ \hline 3F1D \end{array}$$

(B<sub>16</sub>) ← (11) (29) (8) (20) → (1D<sub>16</sub>)  
→ (14<sub>16</sub>)

Na base 16

Na base 2

0 – 1 = 1 (e "pede emprestado"  
do dígito seguinte);  
1 – 1 = 0;  
1 – 0 = 1;  
0 – 1 = 0.

# — Operações aritméticas com números inteiros em qualquer base

## Subtração

- O zero desta coluna vale 2.
- Todos os zeros à esquerda até o primeiro "número 1" valem 1.
- Este "primeiro número 1" vale 0 (zero).
- Se acontecer novamente  $0 - 1$ , o processo se repetirá.

$$\begin{array}{r} \phantom{0} 1 \phantom{0} 1 \phantom{0} 2 \\ 1 \phantom{0} 0 \phantom{0} 0 \phantom{0} 1 \\ - 0 \phantom{0} 1 \phantom{0} 1 \phantom{0} 1 \phantom{0} 0 \\ \hline 0 \phantom{0} 0 \phantom{0} 0 \phantom{0} 1 \phantom{0} 1 \end{array}$$

# — Operações aritméticas com números inteiros em qualquer base

## Computação útil

“

Podemos realmente fazer computação útil com 4.096 bits de memória? [...] O avanço da tecnologia minimizou essas restrições. [...] Em um nível prático, você verá que é bastante viável desenvolver implementações em linguagem de máquina que realizam todos os tipos de tarefas.

(SEDGWICK; WAYNE, 2017, p. 930, tradução nossa)

## — Conversão entre sistemas de numeração

Base 2	Base 8	Base 10	Base 16
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11

# — Conversão entre sistemas de numeração

## De uma base X para outra Y



### Passo 1

Vamos converter 234 da base 6 para um valor equivalente na base 8. O número a ser convertido é expresso em N produtos, em que N é igual à quantidade algarismo do número. Assim, temos:

$$234_6 = 2 \times 6^2 + 3 \times 6^1 + 4 \times 6^0$$



### Passo 2

Calcularemos, agora, os produtos usando a aritmética da base 10. Desse modo, o resultado será expresso em valores decimais. Continuando a realizar o cálculo, teremos:

$$(2 \times 36) + 18 + 4 = 94_{10}$$



# — Conversão entre sistemas de numeração

## De uma base X para outra Y



### Passo 3

Dessa vez, o resultado (decimal 94) será convertido para a base desejada (8). Esse processo é o inverso do anterior, afinal, o inverso da multiplicação é a divisão. Dividimos o valor (94) pela base desejada (8). O resto obtido é o primeiro algarismo do número desejado (aquele mais à direita):

$$94 \div 8 = 11 \text{ e resto} = 6$$



### Passo 4

O quociente obtido na divisão é novamente dividido por 8. A seguir, o novo resto é acrescentado à esquerda do primeiro algarismo:

$$11 \div 8 = 1 \text{ e resto} = 3$$

# — Conversão entre sistemas de numeração

De uma base X para outra Y



## **Passo 5**

O número a ser obtido na base 8 é: 36.



## **Passo 6**

Novamente, faremos a divisão até obtermos um quociente 0 (zero):

$$1 \div 8 = 0 \text{ e resto} = 1$$



## **Passo 7**

O número obtido, portanto, é: 1368.

# — Conversão entre sistemas de numeração

## Casos especiais

$$2^2 = 4$$

$$2^1 = 2$$

$$2^0 = 1$$

Toda vez que o dígito 1 for inserido em uma posição, ela corresponderá ao valor relativo da posição (escrito em decimal). Em outras palavras, toda vez que você colocar:

**1**

“Liga” o bit na posição.

**0**

“Desliga” o bit na posição.

O valor final em decimal será a soma de todos os valores (escritos em decimal) relativos das posições em que estiverem “ligados”.

# — Conversão entre sistemas de numeração

## Casos especiais

$2^0 = 1$	$2^7 = 128$
$2^1 = 2$	$2^8 = 256$
$2^2 = 4$	$2^9 = 512$
$2^3 = 8$	$2^{10} = 1024$
$2^4 = 16$	$2^{11} = 2048$
$2^5 = 32$	$2^{12} = 4096$
$2^6 = 64$	(...)

# — Lógica booleana

- Valores binários seguem lógica booleana.
- Verdadeiro/Falso, Sim/Não, 0/1 representam a mesma lógica.
- Escolha da referência não altera o conceito.
- Lógica booleana é fundamental para a infraestrutura computacional.
- Conexão íntima entre lógica booleana e circuitos computacionais.
- Sedgewick e Wayne (2017) perguntam: "Como os circuitos calculam?".
- Compreender a representação binária é crucial nesse contexto.

# — Tabelas de representação de dados

## Caractere

Representa símbolos (não numéricos). Modo primário de introduzir dados no computador. Serve para escrever um texto em algum idioma.

## Lógico

Representa verdadeiro ou falso.

## Numérico

Representa os números.

# — Tabelas de representação de dados

## ASCII

Cada caractere tem sete bits. O ASCII possui um total de 128 caracteres (2<sup>7</sup>) que podem ser expressos em hexadecimal.

Os códigos 0 a 1F, por sua vez, não são impressos, pois ambos são caracteres de controle.

### Letras maiúsculas

A B C D E F G H I J K L M N O P  
Q R S T U V W X Y Z

### Letras minúsculas

a b c d e f g h i j k l m n o p q r  
s t u v w x y z

### Sinais de pontuação

\* ~ . , ; " ' + = ! ? @ / # ( ) { } [ ]

### Símbolos matemáticos

+ × ÷ − =

# — Tabelas de representação de dados

## ASCII

Hex	Nome	Significado	Hex	Nome	Significado
0	NUL	Null	10	DLE	Data Link Scope
1	SOH	Start Of Heading	11	DC1	Device Control 1
2	STX	Start Of Text	12	DC2	Device Control 2
3	ETX	End Of Text	13	DC3	Device Control 3
4	EOT	End Of Transmission	14	DC4	Device Control 4
5	ENQ	Enquiry	15	NAK	Negative Acknowledgement
6	ACK	ACKnowledgement	16	SYN	SYNchronous Idle
7	BEL	BELl	17	ETB	End Of Transmission Block
8	BS	BackSpace	18	CAN	CANcel
9	HT	Horizontal Tab	19	EM	End Of Medium
A	LF	Line Feed	1*	SUB	SUBstitute
B	VT	Vertical Tab	1B	ESC	ESCape
C	FF	Form Feed	1C	FS	File Separator
D	CR	Carriage Return	1D	GS	Group Separator
E	SO	Shift Out	1E	RS	Record Separator
F	SI	Shift In	1F	US	Unit Separator



# — Tabelas de representação de dados

## Unicode

Padrão internacional no qual cada caractere possui um único valor: 16 bits. Com isso, podemos ter um total de 65.536 símbolos (2<sup>16</sup>).

### Símbolos monetários

£ ¥ ₣ ₤ € \$

### Símbolos matemáticos

+ \* / - =

### Formas geométricas

■ □ △ ○ ◆ ▭

### Emojis



# — Tabelas de representação de dados

## Evolução do caracteres

- Dispositivos modernos suportam muitos caracteres especiais.
- Evolução trouxe padronizações e maior compatibilidade.
- Incompatibilidades prejudicavam usabilidade e acessibilidade.
- Priorizar compatibilidade é essencial em funções como programação web.

# Lógica digital



## — Portas lógicas e lógica booleana

Operações complexas em computadores são combinações de operações aritméticas e lógicas.

Somar bits

Complementar bits

Mover bits

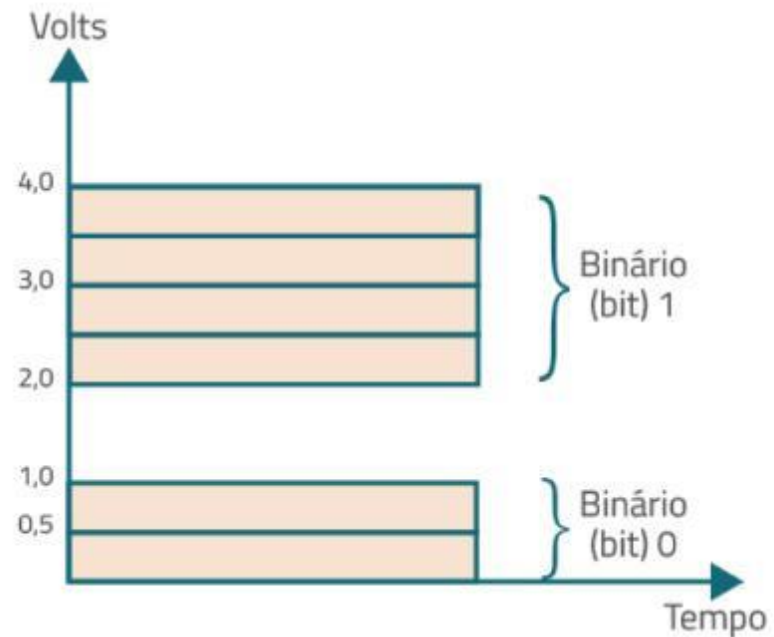
Comparar bits

- Operações lógicas são realizadas por circuitos eletrônicos chamados circuitos lógicos ou portas lógicas.
- Esses circuitos operam com valores binários (1 e 0) e podem tomar decisões inteligentes.

# — Portas lógicas e lógica booleana

Conversão de tensão em circuitos: valores baixos são 0, e valores altos são 1.

A análise visa transformar sinais analógicos em representações digitais (0 e 1) para operações lógicas.



## — Portas lógicas e lógica booleana

Perceba que, com a tensão baixa (bit 0), o equipamento estará desligado, já, com a tensão alta (bit 1), ele estará ligado.

BIT	AÇÃO	CONDIÇÃO	TENSÃO	RESULTADO
<b>1</b>	LIGADO	VERDADEIRO	ALTO	SIM
<b>0</b>	DESLIGADO	FALSO	BAIXO	NÃO

# — Portas lógicas e lógica booleana

- Claude Shannon, em 1938, propôs a aplicação da álgebra booleana em projetos de circuitos com comutadores.
- A álgebra booleana é eficaz na análise e desenvolvimento de circuitos digitais eletrônicos.

## **Análise**

A função de um circuito digital é descrita de acordo com a análise de um modo simplificado

## **Projeto**

A lógica booleana é utilizada para que seja desenvolvida uma implementação simplificada desta função, ao especificar uma determinada função de um circuito.

## — Tabela-verdade

- Descreve a relação entre as entradas e saídas de um circuito lógico.
- Representa todas as combinações possíveis das variáveis de entrada e seus resultados correspondentes (0 ou 1).
- Os valores 0 e 1 indicam FALSO e VERDADEIRO, respectivamente.





## — Tabela-verdade

Entradas		Saída
A	B	X
0	0	1
0	1	0
1	0	1
1	1	0

# — Operadores e portas lógicas básicas

## Portas lógicas

- Componente de hardware com entradas que produz uma saída com base na lógica do circuito.
- Operadores booleanos básicos: AND, OR e NOT são funções lógicas fundamentais.

# — Operadores e portas lógicas básicas

## O operador e a porta OR (OU)

$$X = A + B$$

Para a variável **A** será igual a:

- **0** se a porta estiver **fechada**.
- **1** se a porta estiver **aberta**.

Para a variável **B** será igual a:

- **0** se o interruptor estiver **desativado**.
- **1** se o interruptor estiver **ativado**.

Para a variável **X** será igual a:

- **0** para a lâmpada **apagada**.
- **1** para a lâmpada **acesa**.

## — Operadores e portas lógicas básicas

### O operador e a porta OR (OU)

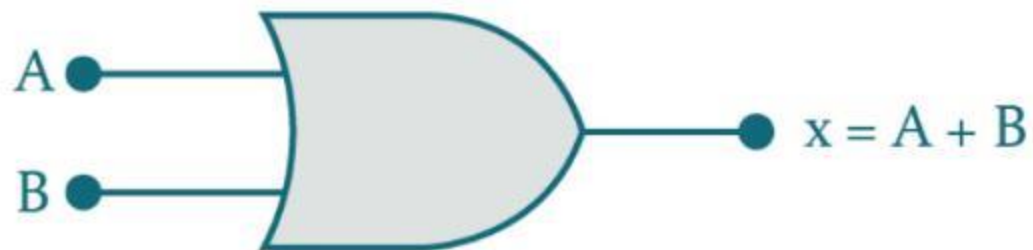
A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Ao analisar a tabela-verdade, chegaremos à conclusão de que a lâmpada estará apagada (valor igual a 0, FALSO) se — e somente se — tanto o interruptor quanto a porta possuírem o valor de entrada igual a FALSO (igual a 0) e, para as demais combinações, a lâmpada estará acesa (igual a 1, VERDADEIRO).

## — Operadores e portas lógicas básicas

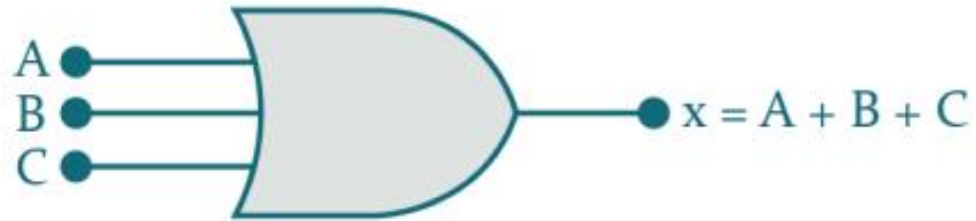
### O operador e a porta OR (OU)

Nos circuitos digitais, uma porta OR é um circuito que tem duas ou mais entradas e a sua saída é igual à combinação das entradas através da operação OR, como ilustrado na seguinte imagem:



## — Operadores e portas lógicas básicas

O operador e a porta OR (OU)



A	B	C	$x = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

# — Operadores e portas lógicas básicas

## O operador e a porta AND (E)

$$X = A \cdot B$$

Para a variável **A** será igual a:

- **0** se a porta estiver **aberta**.
- **1** se a porta estiver **fechada**.

Para a variável **B** será igual a:

- **0** se o botão estiver **desativado**.
- **1** se o botão estiver **ativado**.

Para a variável **X** será igual a:

- **0** para o motor **em repouso**.
- **1** para o motor **acionado**.

## — Operadores e portas lógicas básicas

### O operador e a porta AND (E)

A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

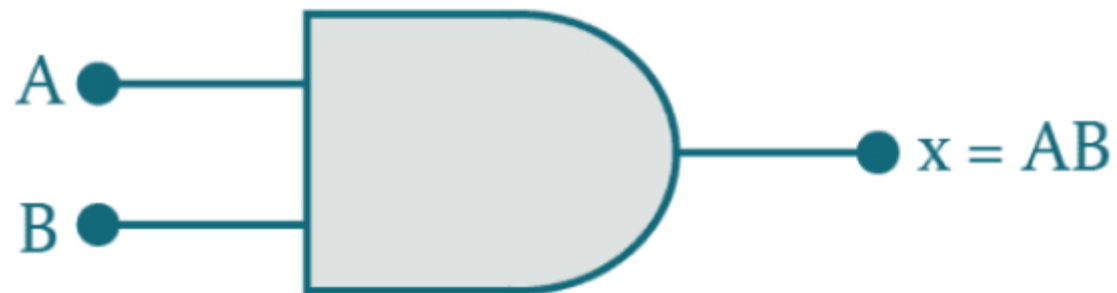
Ao analisar a tabela-verdade, chegaremos à conclusão de que o motor será acionado (valor igual a 1, verdadeiro) se — e somente se — tanto o botão quanto o sensor da porta possuírem o valor igual a verdadeiro (igual a 1) e, para as demais combinações, o motor estará desligado (igual a 0).



## — Operadores e portas lógicas básicas

### O operador e a porta AND (E)

Nos circuitos digitais, uma porta AND é um circuito que tem duas ou mais entradas e a sua saída é igual à combinação das entradas através da operação AND, conforme ilustrado na imagem a seguir:



## — Operadores e portas lógicas básicas

O operador e a porta AND (E)



A	B	C	$x = ABC$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

## — Operadores e portas lógicas básicas

O operador e a porta NOT (NÃO)

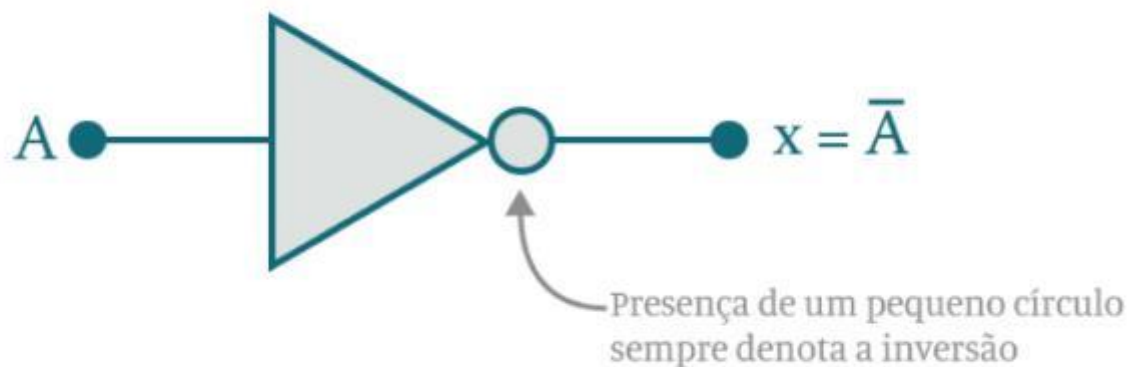
$$X = \bar{A}$$

A	$x = \bar{A}$
0	1
1	0

# — Operadores e portas lógicas básicas

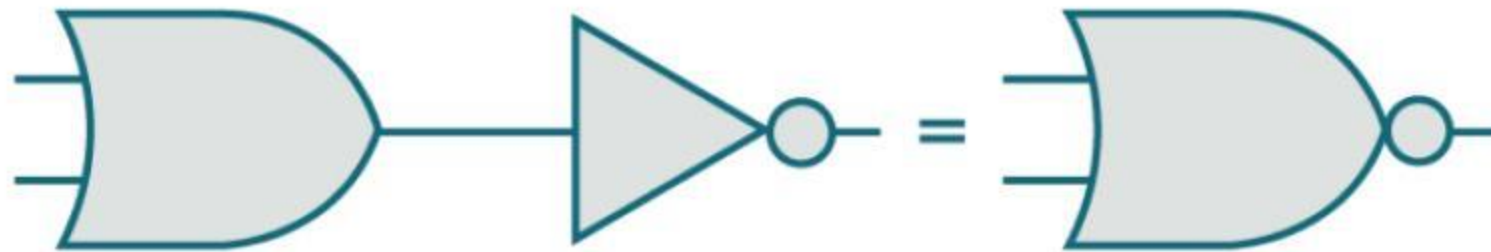
## O operador e a porta NOT (NÃO)

Nos circuitos digitais, uma porta NOT é um circuito que tem uma entrada, e a sua saída, a negação, é indicada por um pequeno círculo, como mostrado a seguir:



## — Outras portas lógicas fundamentais

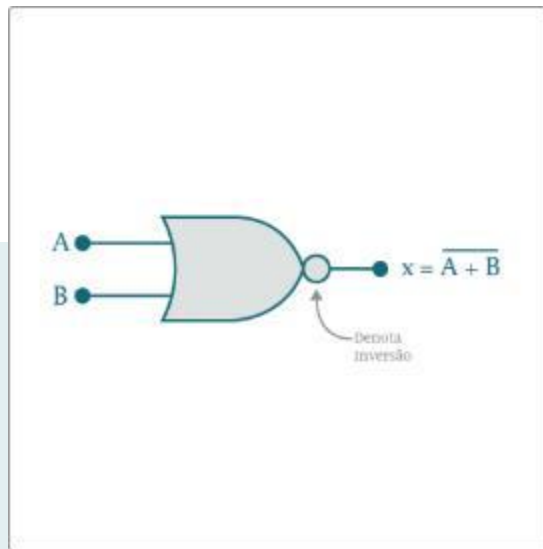
A porta NOR (Não OU)



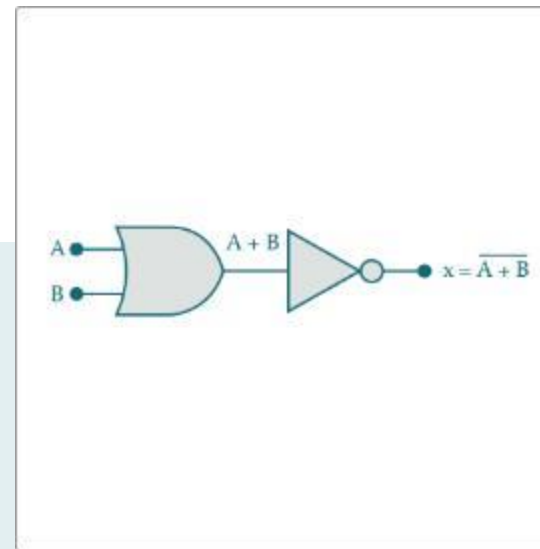
$$X = \overline{A + B}$$

## — Outras portas lógicas fundamentais

### A porta NOR (Não OU)



**Símbolo de uma Porta NOR.**



**A porta NOR ()'NOT OR' 'NÃO-OU').**

## — Outras portas lógicas fundamentais

### A porta NOR (Não OU)

		OR	NOR
A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Ao analisar as combinações possíveis com duas entradas, levando em consideração os valores para as variáveis de entrada A e B, somente produzirá uma saída 1 (VERDADEIRO) se — e somente se — todas as entradas sejam 0 (FALSO) e, para as demais condições, produzirá como resultado na saída igual a 0 (FALSO).

# — Outras portas lógicas fundamentais

## A porta NOR (Não OU)

### Variável A, B e X

Nós podemos representar a aplicação desta função como: a lâmpada poderá estar apagada em duas situações distintas, se a porta do veículo estiver aberta e o interruptor da lâmpada for acionado. Neste cenário, vamos representar cada uma dessas possibilidades: a variável A representará a abertura da porta, a variável B representará o interruptor e a variável X representará o estado da lâmpada; se está acesa ou apagada.



### Igual a 0 e igual a 1

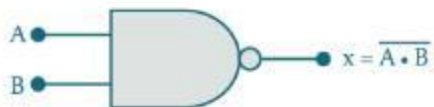
Ao analisarmos as combinações possíveis, levando em consideração os valores para a variável A, será igual a 0 se a porta estiver fechada e será igual a 1 se a porta estiver aberta. Em relação à variável B, temos o valor igual a 0 para o interruptor ativado e 1 para o interruptor desativado e, por fim, a variável X possuirá o valor 0 para a lâmpada apagada e 1 para a lâmpada acesa.



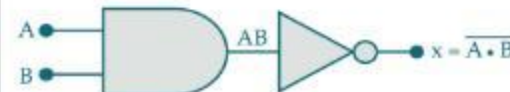
## — Outras portas lógicas fundamentais

### A porta NAND (Não E)

$$X = \overline{A \cdot B}$$



Símbolo de uma porta NAND.



A porta NAND ('NOT AND' 'NÃO-E').

## — Outras portas lógicas fundamentais

### A porta NAND (Não E)

		AND	NAND
A	B	$AB$	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Ao analisar as combinações possíveis, levando em consideração os valores para as variáveis de entrada A e B, somente produzirá uma saída 0 (FALSO) se — e somente se — todas as entradas forem 1 (VERDADEIRO) e, para as demais condições, produzirá como resultado na saída igual a 1 (VERDADEIRO).

## — Outras portas lógicas fundamentais

A porta XOR (Ou exclusivo)

A	B	$x = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



$$X = A \oplus B$$

## — Outras portas lógicas fundamentais

A porta XNOR (coincidência)

$$X = A \odot B$$

A	B	$x = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1



## — Expressões lógicas

- Definidas com sinais de entrada (variáveis binárias) e conectivos lógicos, resultando em um sinal de saída.
- Conectivos lógicos: símbolos que representam operações lógicas (como AND, OR) e podem incluir parênteses.

$$X = A + \bar{B} \cdot C$$

## — Avaliação de uma expressão lógica

Na avaliação de uma expressão lógica, uma ordem de precedência deverá ser seguida da mesma forma que é considerada em uma expressão aritmética, de acordo com o definido a seguir:

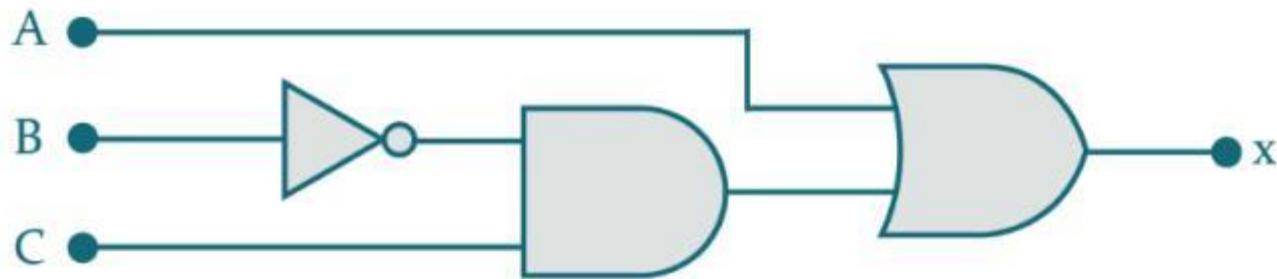


Lembre-se de que os conteúdos entre parênteses devem ser executados primeiro.

## — Avaliação de uma expressão lógica

Seguindo o exemplo,  $X = A + \bar{B} \cdot C$ , lê-se:

$$X = A \text{ OR } (\text{NOT } B) \text{ AND } C$$



# — Equivalência de funções lógicas

## Conceito

Duas funções lógicas são equivalentes se — e somente se — **para a mesma entrada, produzirem iguais valores de saída**, isto é, quando duas funções lógicas possuírem o mesmo resultado na sua tabela-verdade, esses circuitos serão considerados como equivalentes.



# — Propriedades da álgebra de Boole

Álgebra de Boole simplifica expressões lógicas: Regras da álgebra de Boole ajudam na análise, equivalência e simplificação de expressões lógicas em dispositivos digitais. Reduz custos e complexidade.

PROPRIEDADE	VERSÃO OR	VERSÃO AND
1. IDENTIDADE	$X + 0 = X$	$X \cdot 1 = X$
2. ELEMENTO NULO	$X + 1 = 1$	$X \cdot 0 = 0$
3. EQUIVALÊNCIA	$X + X = X$	$X \cdot X = X$
4. COMPLEMENTO	$X + \overline{X} = 1$	$X \cdot \overline{X} = 0$
5. INVOLUÇÃO	$\overline{\overline{X}} = X$	$\overline{\overline{X}} = X$
6. COMUTATIVA	$X + Y = Y + X$	$X \cdot Y = Y \cdot X$
7. ASSOCIATIVA	$(X + Y) + Z = X + (Y + Z)$	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$
8. DISTRIBUTIVA	$X + Y \cdot Z = (X + Y) \cdot (X + Z)$	$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
9. ABSORÇÃO 1	$X + X \cdot Y = X$	$X \cdot (X + Y) = X$
10. ABSORÇÃO 2	$X + \overline{X} \cdot Y = X + Y$	$X \cdot (\overline{X} + Y) = X \cdot Y$
11. CONSENSUS	$X \cdot Y + \overline{X} \cdot Z + Y \cdot Z = X \cdot Y + \overline{X} \cdot Z$	$(X + Y) \cdot (\overline{X} + Z) \cdot (Y + Z) = (X + Y) \cdot (\overline{X} + Z)$
12. DE MORGAN	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$

Tabela das regras básicas da álgebra booleana.  
Stalling, 2017.

## — Propriedades da álgebra de Boole

Agora observe a tabela das propriedades da função Exclusive or (XOR):

A	B	$A \oplus B$	$\overline{A \oplus B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Soma dos minitermos:

$$A \oplus B = \bar{A} \cdot B + A \cdot \bar{B}$$

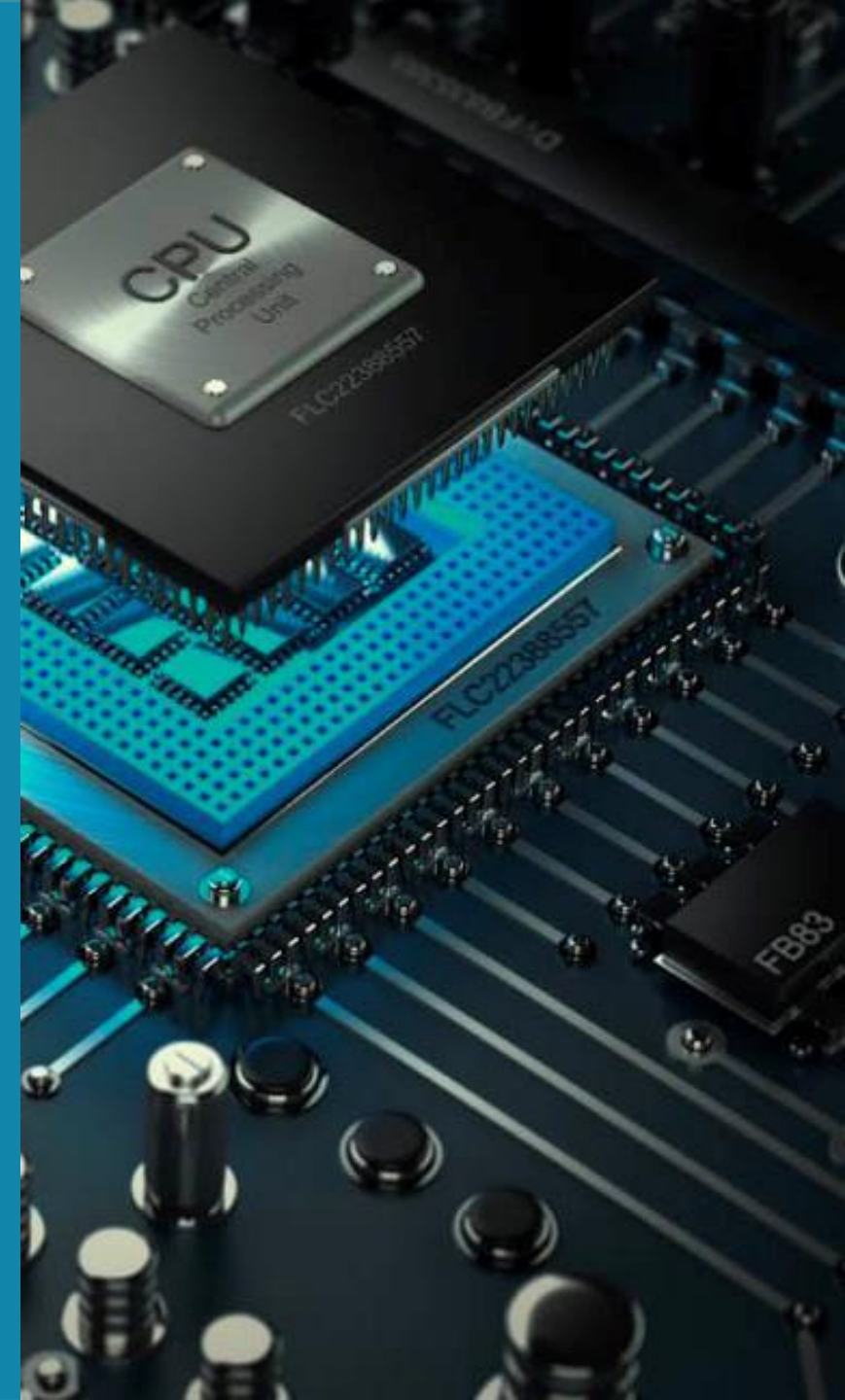
$$\overline{A \oplus B} = A \cdot B + \bar{A} \cdot \bar{B}$$

Outras propriedades:

$$A \oplus A = 0$$

$$A \oplus \bar{A} = 1$$

# Processamento em paralelo



# — Evolução dos computadores

## Questões de desempenho



Algumas aplicações que, atualmente, possuem uma grande capacidade de processamento em sistemas baseados em microprocessadores:

- Processamento de imagem
- Renderização tridimensional
- Reconhecimento de linguagem
- Videoconferência
- Modelagem e simulação

# — Evolução dos computadores

## Questões de desempenho



Qual seria o resultado se determinada tarefa fosse realizada por várias pessoas ao invés de apenas uma?

### **Resposta**

As respostas poderiam ser várias, inclusive a possibilidade de não conclusão dessa tarefa. Mas, se olhássemos a distribuição das ações que compõem a tarefa de forma organizada, buscando controlar a sua eficiência, poderíamos obter as seguintes respostas:

- A tarefa será realizada no mesmo tempo.
- O tempo de conclusão da tarefa será reduzido.

# — Paralelismo em nível de instruções e processadores superescalares

## Superescalar e pipeline

### Superescalar

Nesta organização, as instruções comuns (aritméticas de inteiros e pontos flutuantes), a carga do valor da memória em um registrador, o armazenamento do valor do registrador na memória e os desvios condicionais poderão ser iniciados e executados de forma independente.



### Pipeline

Nesta técnica, o caminho executável de uma instrução é dividido em estágios discretos, permitindo ao processador efetuar várias instruções simultaneamente, contanto que apenas uma delas ocupe cada estágio durante um ciclo de relógio.

# — Paralelismo em nível de instruções e processadores superescalares

## Superescalar e pipeline

Veja a comparação da execução de instruções sem e com pipeline.

	TEMPO →					
	1	2	3	4	5	6
INSTRUÇÃO 1						
INSTRUÇÃO 2						
INSTRUÇÃO 3						
INSTRUÇÃO 4						
INSTRUÇÃO 5						



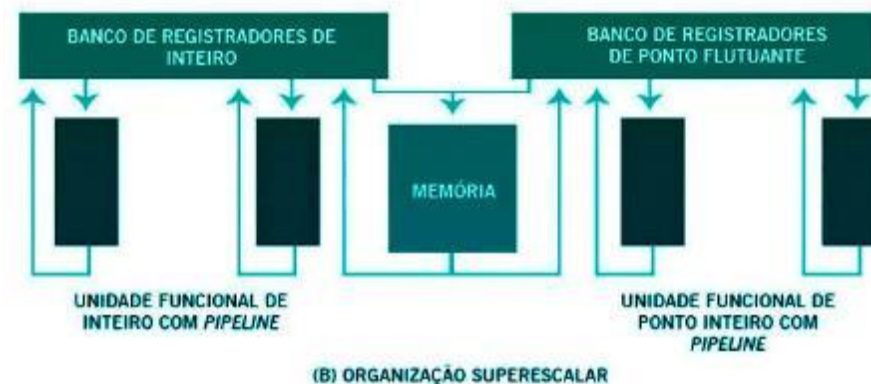
	TEMPO →					
	1	2	3	4	5	6
INSTRUÇÃO 1						
INSTRUÇÃO 2						
INSTRUÇÃO 3						
INSTRUÇÃO 4						
INSTRUÇÃO 5						



# — Paralelismo em nível de instruções e processadores superescalares

## Superescalar e pipeline

- VLIW e superescalar: Emitir e executar múltiplas instruções simultaneamente.
- Hyper-Threading Intel: Cria dois processadores virtuais, aumentando paralelismo.
- Processadores vetoriais: Executam operações em conjuntos de dados, dependendo de pipelines e alta velocidade.
- Processadores matriciais: Contêm muitas unidades de processamento, eficientes para grandes conjuntos de dados.





# — Paralelismo em nível de instruções e processadores superescalares

## Pipelines profundos

- Pipelines profundos: Realizam várias instruções simultaneamente, proporcionando paralelismo.
- Paralelismo: Múltiplas instruções em diferentes estágios do pipeline simultaneamente.
- Organização escalar tradicional: Uma unidade funcional para operações inteiras e outra para operações de ponto flutuante em um único pipeline.

# — Paralelismo em nível de instruções e processadores superescalares

## Superescalar X superpipeline

- Superpipeline: Múltiplos estágios do pipeline executam tarefas que demandam menos da metade de um ciclo de clock.
- Aumento de desempenho: Dobrar a velocidade do clock interno ao executar duas tarefas em um ciclo de clock externo.

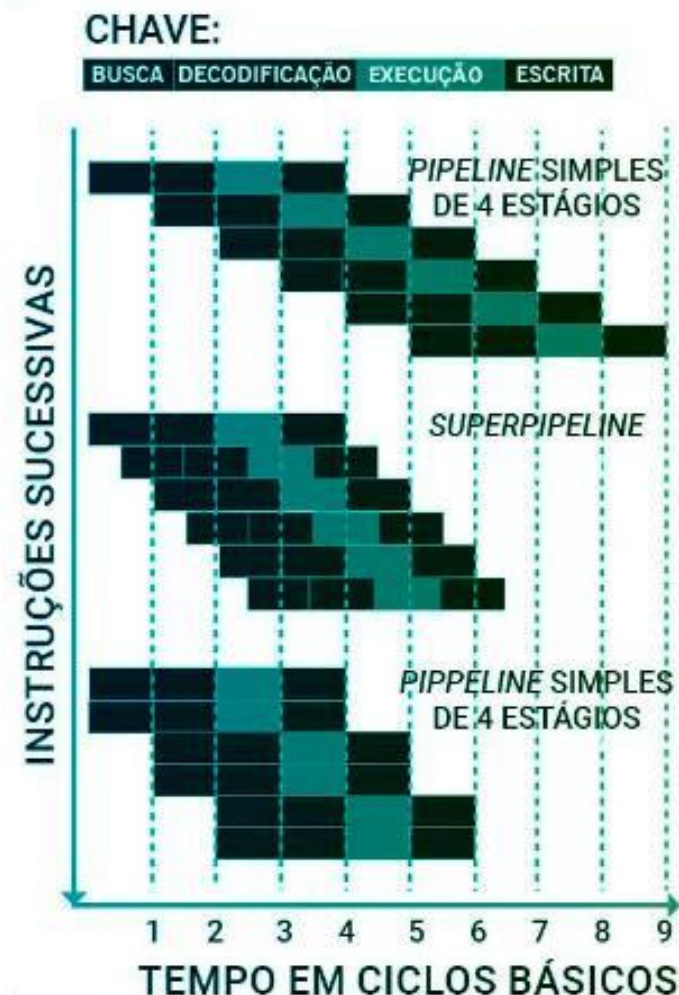


MIPS R4000.

# — Paralelismo em nível de instruções e processadores superescalares

## Superescalar X superpipeline

- No esquema a seguir, podemos visualizar uma comparação entre as organizações superpipeline e superescalar, tendo um pipeline comum na parte superior do diagrama que será utilizado apenas como referência:
- Pipeline: Inicia e executa uma instrução por ciclo de clock, com quatro estágios.
- Superpipeline: Executa dois estágios de pipeline por ciclo de clock, dividindo funções em duas partes não sobrepostas.



# — Múltiplos processadores

## Arquitetura clássica

- Os processadores executam programas com instruções de máquina em sequência porque cada instrução traz uma sequência de operações.
- Análise detida revela operações paralelas em nível de micro-operações.
- Pipeline de instruções: Sobreposição de leitura e execução.
- Redução de custo impulsiona arquiteturas mais rápidas e robustas.
- Soluções paralelizadas atendem demandas eficientemente.

# — Múltiplos processadores

## Tipos de processadores paralelos – taxonomia de Flynn

- Taxonomia de Flynn (1966) classifica computadores em 4 categorias.
- Baseada em tipos de fluxos: instruções e dados.
- Múltiplos processadores visam paralelismo crescente.
- Esquema pioneiro para entender configurações paralelas.

# — Múltiplos processadores

## Tipos de processadores paralelos – taxonomia de Flynn

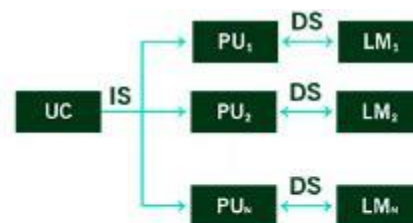
		Nº FLUXOS DE INSTRUÇÃO	
		ÚNICO	MÚLTIPLOS
Nº FLUXO DE DADOS	ÚNICO	<u>SISD (Single Instruction Single Data)</u> Monoprocessador	<u>MISD (Multiple Instruction Single Data)</u> Máquina Teórica
	MÚLTIPLOS	<u>SIMD (Single Instruction Multiple Data)</u> Máquinas Vetoriais	<u>MIMD (Multiple Instruction Multiple Data)</u> SMP, Cluster, NUMA

# — Múltiplos processadores

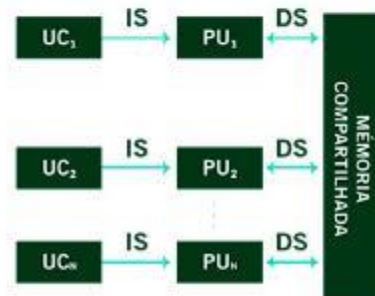
## Tipos de processadores paralelos – taxonomia de Flynn



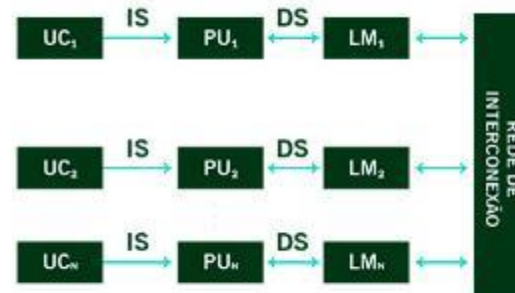
(A) SISD



(B) SIMD (COM MEMÓRIA DISTRIBUÍDA)



(C) MIMD (COM MEMÓRIA COMPARTILHADA)



(D) MIMD (COM MEMÓRIA DISTRIBUÍDA)

UC = UNIDADE DE CONTROLE  
IS = FLUXO DE INSTRUÇÕES  
PU = UNIDADE DE PROCESSAMENTO  
DS = FLUXO DE DADOS  
MU = UNIDADE DE MEMÓRIA  
LM = MEMÓRIA LOCAL

SISD = INSTRUÇÃO ÚNICA, ÚNICO FLUXO DE DADO  
SIMD = INSTRUÇÃO ÚNICA, MÚLTIPLOS FLUXOS DE DADOS  
MIMD = INSTRUÇÃO INSTRUÇÕES, MÚLTIPLOS FLUXOS DE DADOS

# — Múltiplos processadores

## Multiprocessadores simétricos

- Computadores antigos tinham 1 processador de uso geral.
- Evolução tecnológica permitiu redução de custo e aumento de desempenho.
- Surgimento de sistemas SMP (Symmetric Multiprocessing).
- SMP relacionado à arquitetura de hardware e comportamento de sistemas operacionais.



# — Múltiplos processadores

## Multiprocessadores simétricos

Um sistema SMP pode ser definido como uma arquitetura independente com as seguintes características:

Há dois ou mais processadores semelhantes (capacidades que podem ser comparadas).

Os processadores compartilham a mesma memória principal e os dispositivos de E/S, além de serem interconectados por barramentos ou algum outro esquema de conexão que permita a equalização do tempo de resposta para cada processador.

Todos os processadores desempenham as mesmas funções. O termo simétrico é derivado dessa afirmação.

Todo o sistema é controlado pelo sistema operacional integrado que fornece a integração entre os processadores, seja para os seus programas, arquivos, dados ou suas tarefas.

# — Coerência de cache

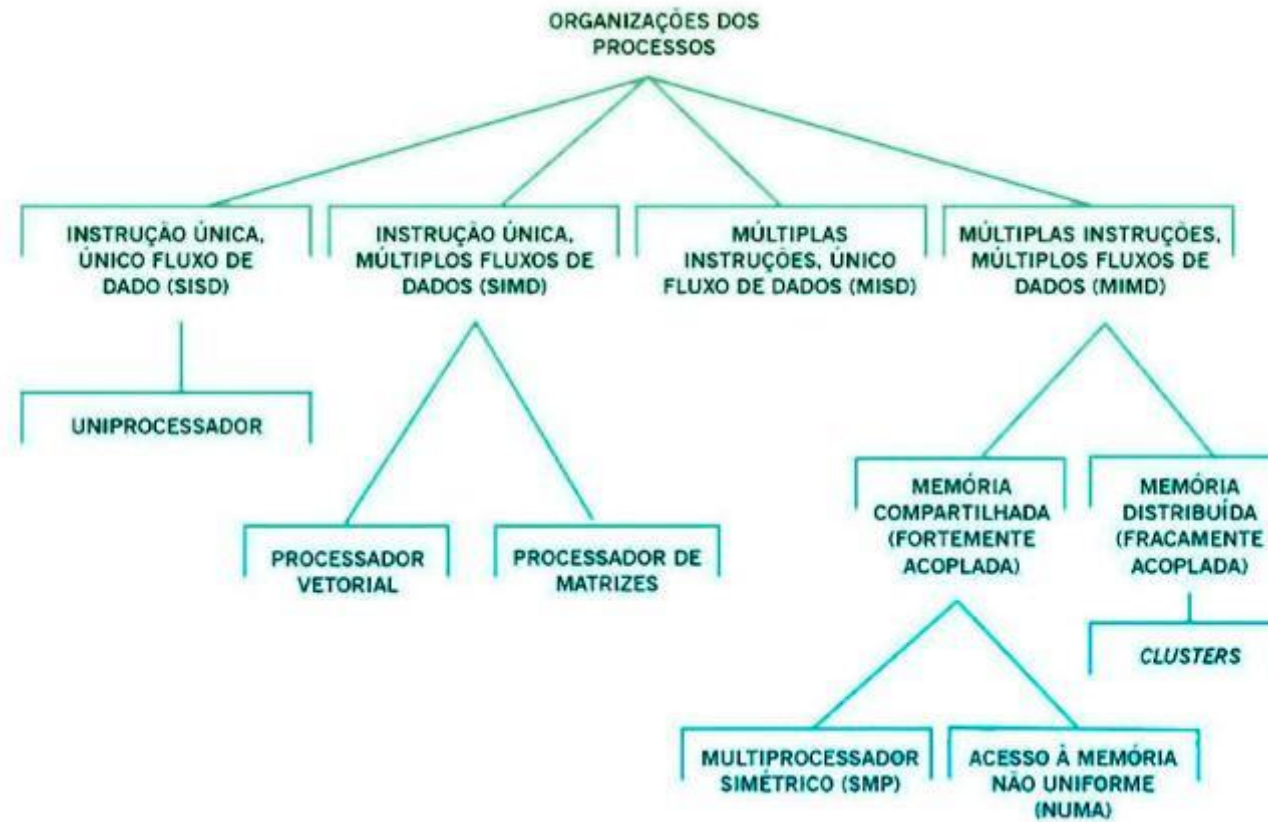
## Coerência de cache UMA

- Protocolos de coerência de cache para UMA são simples.
- Barramento rápido e caches relativamente pequenos facilitam implementação.
- Atualização de dados requer escuta do barramento.
- Escuta de barramento gera tráfego adicional.

## NUMA com cache coerente (CC-NUMA)

- CC-NUMA: Cache Coherent Non-Uniform Memory Access.
- Cada endereço associado a um nó nativo.
- Falha de cache: contato com nó associado ao endereço.
- Item limpo: despachado para cache do requisitante.
- Item sujo: despachado para nó com cópia suja e requisitante.
- Modificação: requisitada via nó nativo, notifica modificação a outros nós.
- Protocolo eficiente, mas pode ter mau desempenho se muitos acessos vindos de nós remotos.

## — Coerência de cache



## — Computadores multicore

- Multicore: Combinação de duas ou mais unidades de processamento em uma pastilha de silício.
- Cada core tem componentes independentes (registradores, ULA, pipeline, UC).
- Inclui caches L1 de dados e instruções, bem como caches L2 e L3.
- Alguns processadores multicore também incluem memória e controladores de periféricos.

## — Desempenho do hardware

- Evolução dos processadores: Inicial foco no aumento do paralelismo em nível de instruções.
- Soluções como pipeline, superescalar e SMT buscaram maximizar desempenho.
- Limitações e desafios surgiram, como aumento de consumo de energia e calor.
- Regra de Pollack: Aumento de desempenho proporcional à raiz quadrada do aumento da complexidade.
- Multiplicidade de cores permite ganhos quase lineares, especialmente com software otimizado.
- Integração de pastilhas amplia espaço para caches, reduzindo o consumo de energia.

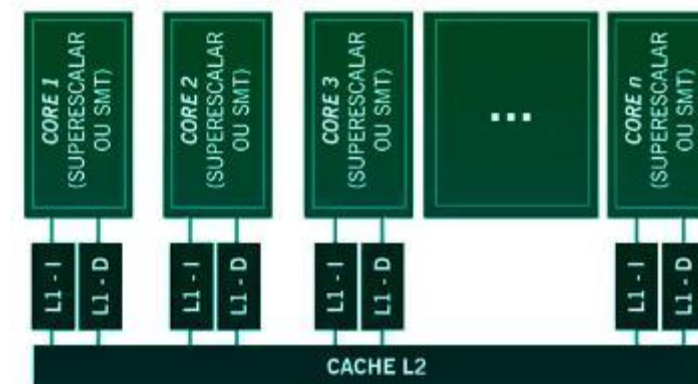
## — Desempenho do hardware



Superescalar



Multithreading simultâneo



Multicore

## — Desempenho do software

- Desempenho em softwares pode ser medido considerando o tempo para partes serial e paralelizável, incluindo sobrecarga de escalonamento.
- Mesmo com parte do código sendo serial, sistemas multicore podem ser vantajosos comparados a soluções de um único core.
- Softwares de servidores, lidando com muitas transações em paralelo, podem eficientemente aproveitar estruturas multicore.
- A habilidade de dimensionar o rendimento em função do número de cores é uma vantagem em softwares de uso geral para servidores.

# — Desempenho do software

## **Aplicações multithread nativas (paralelismo em nível de thread)**

São caracterizadas por possuírem um pequeno número de processos com alto nível de paralelização.

## **Aplicações com múltiplos processos (paralelismo em nível de processo)**

São caracterizadas pela presença de muitos processos de thread única (threads podem ser consideradas como fluxos de um processo).

## **Aplicações Java**

Aceitam threads de uma maneira natural, isto é, a Máquina Virtual Java (JVM) é um processo multithread que permite o escalonamento e o gerenciamento de memória para aplicações Java.

## **Aplicações com múltiplas instâncias (paralelismo em nível de aplicação)**

Mesmo que uma aplicação individual não possa ser dimensionada para obter vantagem em um número grande de threads, ainda é possível que se beneficie com a execução de várias instâncias da aplicação em paralelo.



# — Organização multicore

As principais variáveis em uma organização multicore são:

**Número de cores processadores no chip**

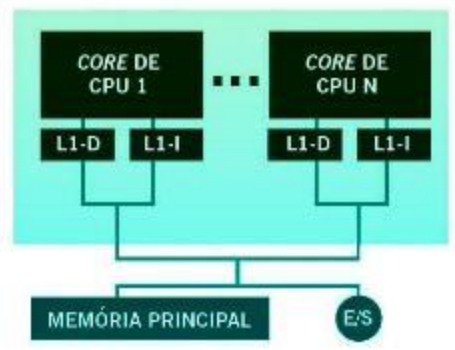
**Número de níveis da memória cache**

**Quantidade de memória cache compartilhada**

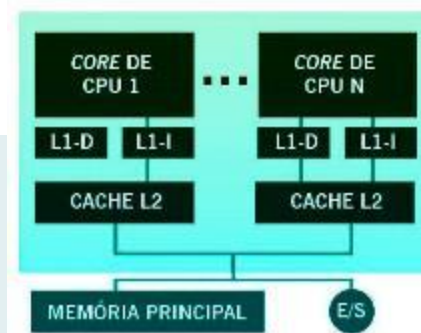
**Emprego do multithreading simultâneo (SMT)**

# — Organização multicore

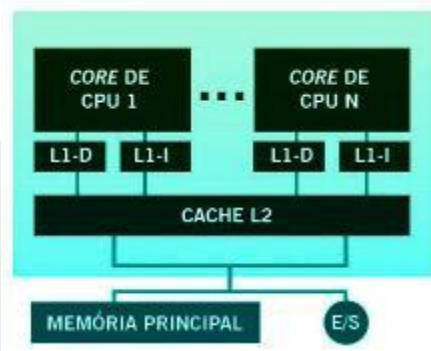
## Níveis de cache e cache compartilhada



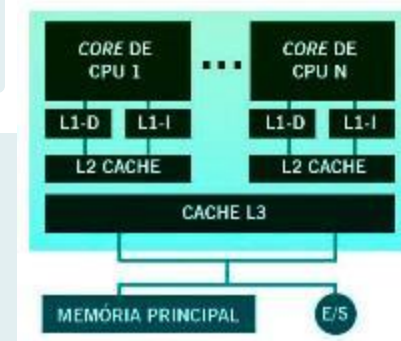
Cache L1 dedicada



Cache L2 dedicada



Cache L2 compartilhada



Cache L3 compartilhada

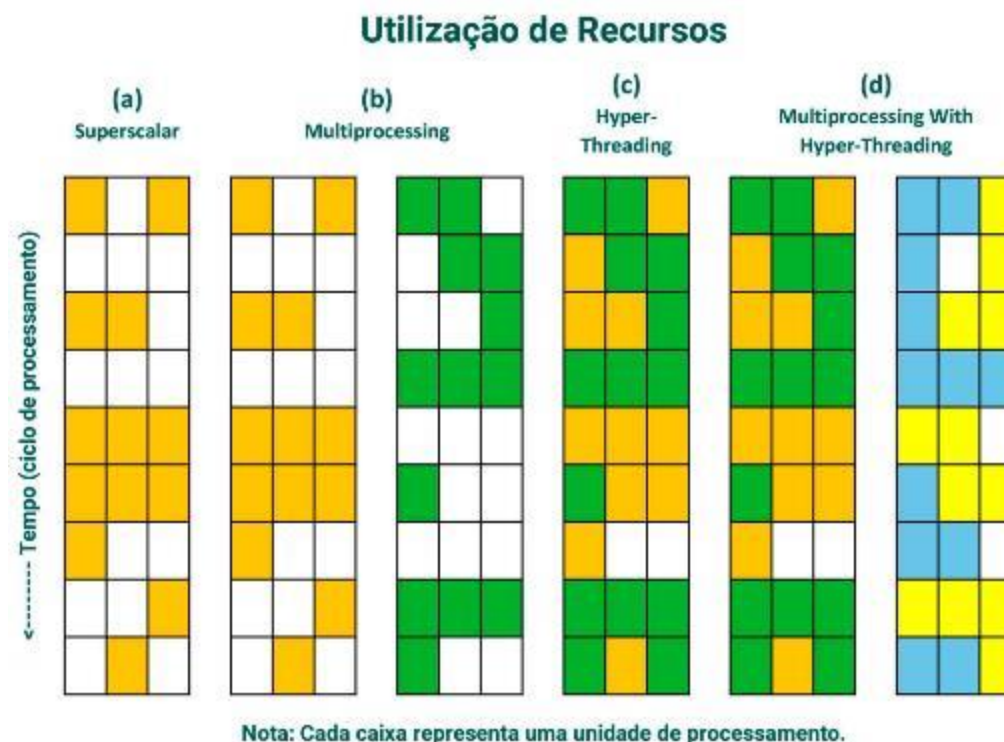
# — Organização multicore

## Multithreading simultâneo



# — Organização multicore

## Multithreading simultâneo



Exemplo de execução de threads com: (a) um processador escalar; (b) dois processadores; (c) um processador habilitado com hyper-threading; (d) dois processadores habilitados com hyper-threading.

# — Organização multicore

## Multithreading simultâneo

### **Superescalar**

No primeiro exemplo, com um único processador superescalar – o processador está em uso, mas cerca de metade do tempo do processador permanece sem uso.

### **Multiprocessing**

No multiprocessamento, é possível verificar um sistema de CPU dupla trabalhando em dois threads separados. No entanto, novamente cerca de 50% do tempo de ambas as CPUs permanecem sem uso.

# — Organização multicore

## Multithreading simultâneo

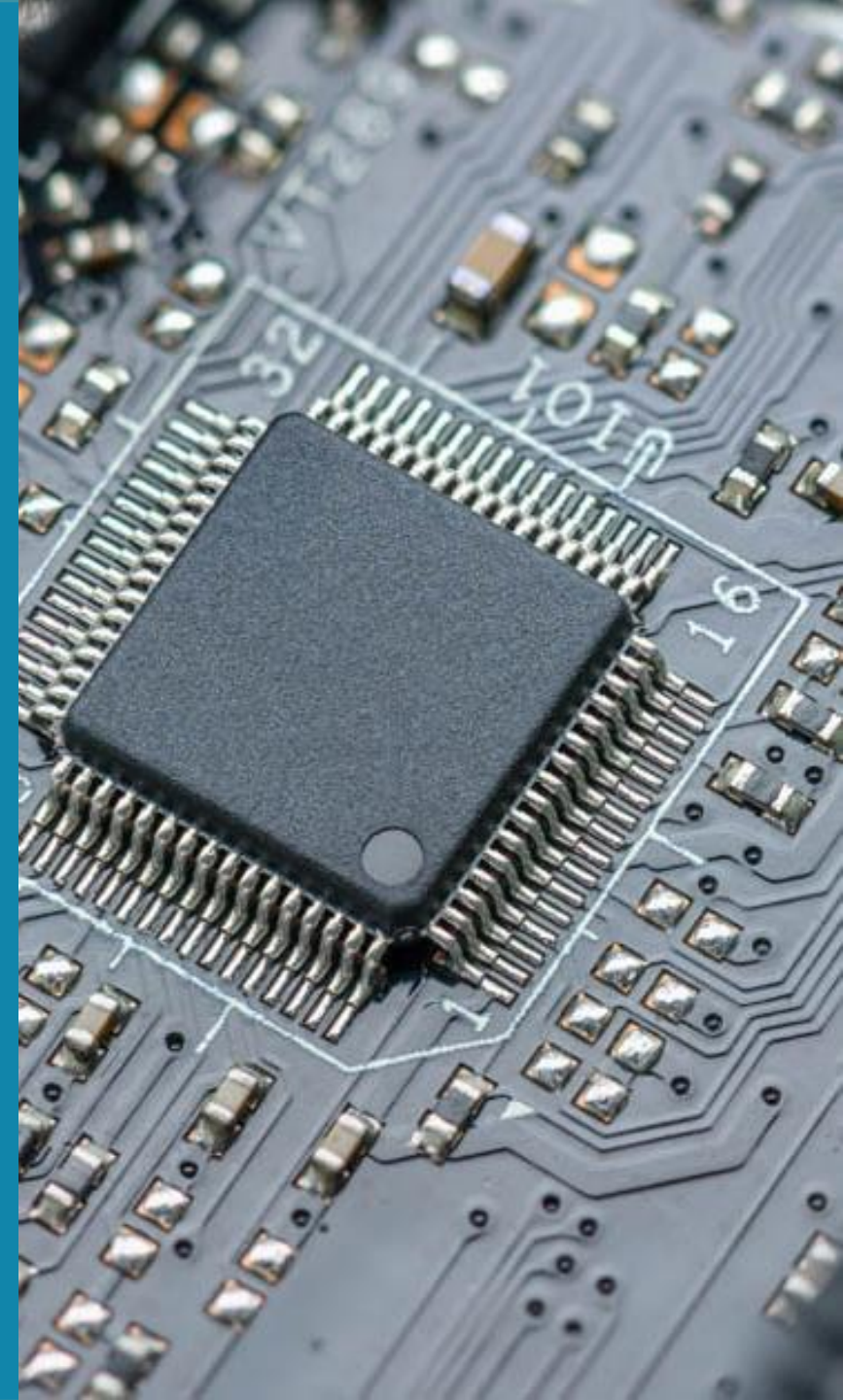
### **Hyper-threading**

No terceiro caso, um único processador está habilitado para hyper-threading, os dois threads estão sendo computados simultaneamente e a eficiência da CPU aumentou de cerca de 50% para mais de 90%.

### **Multiprocessing with hyper-threading**

No último exemplo, há dois processadores habilitados para hyper-threading que podem funcionar em quatro threads independentes ao mesmo tempo. Novamente, a eficiência da CPU é de cerca de 90%. Nesse caso, teríamos quatro processadores lógicos e dois processadores físicos.

# Arquitetura CISC X RISC



# — Arquitetura CISC

## O conceito da arquitetura

- Projeta arquiteturas com extenso conjunto de instruções para diversas funcionalidades.
- Instruções variam em complexidade, frequentemente envolvendo acesso à memória.
- Complexidade na Unidade de Controle é compensada por eficiência em acessos à memória e soluções específicas.





# — Arquitetura CISC

## Origem

- Surgiu como evolução dos processadores, retroativamente chamados de CISC após o conceito de RISC.
- A evolução tecnológica permitiu a resolução de problemas mais complexos.
- Novas instruções específicas foram adicionadas para lidar com problemas mais complexos.

# — Arquitetura CISC

## Características

Processador	Ano	Tamanho da instrução	Quantidade de instruções	Tamanho do registrador	Endereçamento
IBM 370	1970	2 a 6 bytes	208	32 bits	R-R; R-M; M-M
VAX11	1978	2 a 57 bytes	303	32 bits	R-R; R-M; M-M
Intel 8008	1972	1 a 3 bytes	49	8 bits	R-R; R-M; M-M
Intel 286	1982	2 a 5 bytes	175	16 bits	R-R; R-M; M-M
Intel 386	1985	2 a 16 bytes	312	32 bits	R-R; R-M; M-M

# — Arquitetura CISC

## Múltiplo endereçamento

### R-R

Para instruções que usam registradores como entrada e saída.

### R-M

Quando um dos elementos (operandos ou resultado) deve ser buscado/escrito na memória e ao menos um em registrador.

### M-M

Para instruções em que os operandos e o resultado estão na memória.

# — Arquitetura CISC

## Múltiplo endereçamento

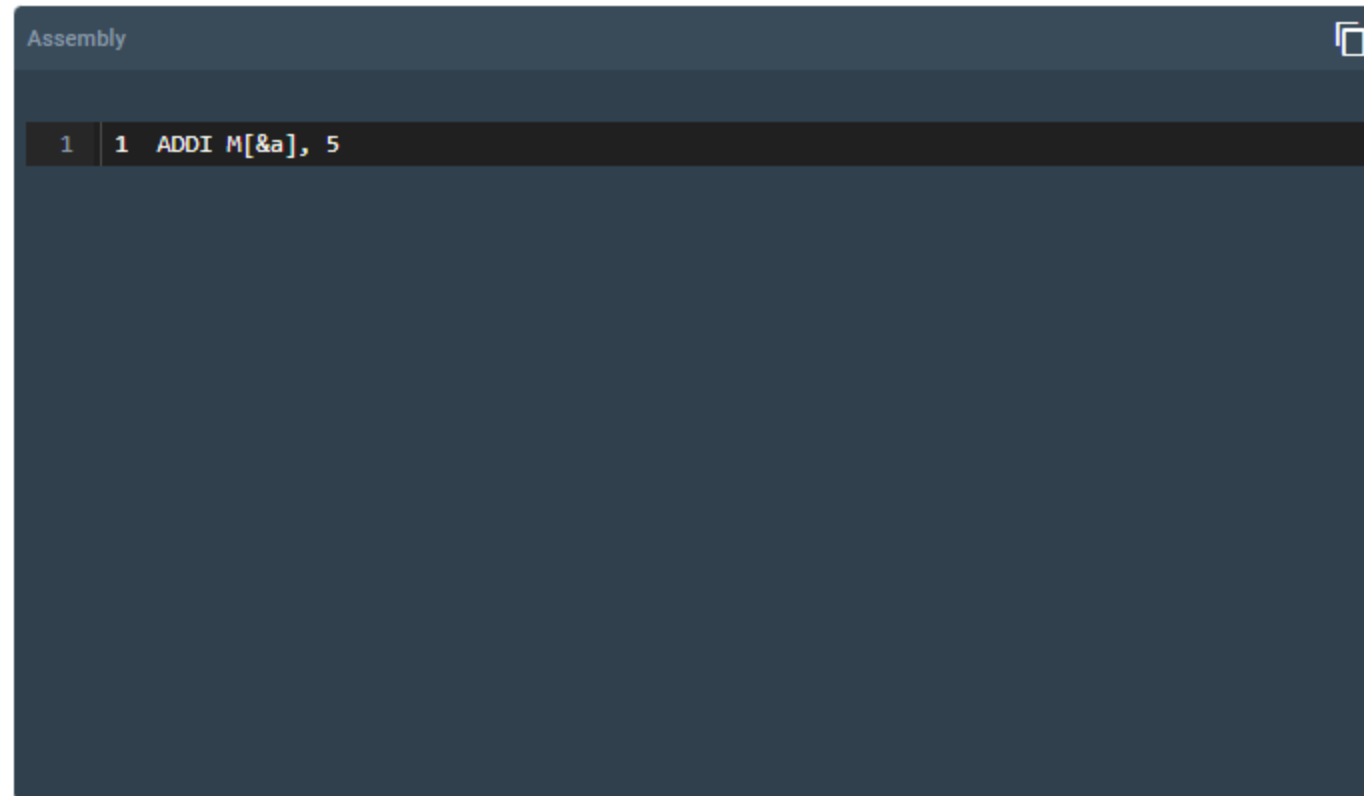
- CISC realiza operações complexas envolvendo busca, operação e armazenamento direto na memória principal.
- Código de máquina é mais simples de gerar pelo compilador devido à relação direta entre código em alto nível e instruções da arquitetura.

```
Pseudocódigo De Alto Nível

1
2      1  int a = 3
3      2  a = a + 5
```

# — Arquitetura CISC

Múltiplo endereçamento



The image shows a screenshot of an assembly editor window. The window has a title bar that says "Assembly" and a copy icon in the top right corner. The main area of the window is dark gray and contains a single line of assembly code. The code is displayed in a table-like format with three columns: a line number, a label, and the instruction itself. The first column contains the number "1", the second column contains the label "1", and the third column contains the instruction "ADDI M[&a], 5".

1	1	ADDI M[&a], 5
---	---	---------------

# — Arquitetura RISC

## O conceito da arquitetura

### Instruções:

- Poucas instruções genéricas.
- Operações mais complexas são montadas a partir dessas instruções.
- Instruções operam apenas sobre os registradores, exceto para acesso à memória.

### Unidade de Controle:

- Simples para decodificar as instruções.
- Mais espaço disponível para registradores.



# — Arquitetura RISC

## Origem

- RISC surgiu nos anos 80, retroativamente chamando os anteriores de CISC.
- Tentativa de resolver deficiências dos processadores tradicionais.
- Programas pouco usavam operações complexas devido à otimização de código.
- Múltiplos endereçamentos causavam variação nos ciclos de instrução, e U.C. grande limitava registradores.

# — Arquitetura RISC

## Premissas e características

A nova abordagem proposta possuía algumas premissas que geraram certas características na arquitetura resultante:

### Quantidade de Instruções

**Premissa:** quantidade restrita de instruções, com as quais era possível montar as outras.

A quantidade reduzida de instruções diminui o tamanho e a complexidade da Unidade de Controle para decodificação da instrução. Com isso, sobra mais espaço para registradores no processador. Enquanto os processadores CISC costumam ter até 8 registradores, é comum que os processadores RISC tenham mais de 32, chegando até a algumas centenas.



# — Arquitetura RISC

## Premissas e características

### Tempo de Execução

**Premissa:** as instruções devem ser executadas com duração próxima, facilitando o pipeline.

A premissa de execução com duração próxima serve para facilitar a previsibilidade do processamento de cada instrução. A ideia é que cada etapa da instrução consiga ser executada em um ciclo de máquina (CLK). Como todas as instruções operam usando apenas os rápidos registradores, isso é possível. As exceções são as instruções LOAD e STORE.

# — Arquitetura RISC

## Premissas e características

### Operação das Instruções

**Premissa:** as instruções devem operar sobre registradores, exceto em algumas específicas para busca e gravação de dados na memória (LOAD e STORE). As instruções LOAD e STORE servem para acessar a memória.

Com a operação sobre os registradores, o pipeline executa de forma próxima ao ideal (1 etapa por ciclo), exceto pelos acessos à memória das instruções LOAD e STORE, que demandam um tempo maior de espera.

# — Arquitetura RISC

## Pipeline

Considere as etapas vistas anteriormente: Buscar Instrução (BI), Decodificar Instrução (DI), Execução (EXE), Acesso à Memória (AM) e WriteBack (WB).

Exemplo de pipeline com modelo RISC com 3 instruções executadas durante 10 pulsos de clock:

Pulso de CLK (tempo →)		1	2	3	4	5	6	7	8	9	10
I N S T R U C A O	LOAD Reg1, M[&a]	BI	DI	EXE	AM	AM	WB				
	ADDI Reg1, Reg1, 5		BI	DI			EXE	AM	WB		
	STORE Reg1, M[&a]			BI			DI	EXE	AM	AM	WB

# — Arquitetura RISC

## Convertendo o código

Como podemos realizar essa operação no modelo RISC, onde não há operações que façam adição de elementos diretamente da memória (terceira premissa de um processador RISC)?

```
Pseudocódigo De Alto Nível
```

1				
2				1 int a = 3
3				2 a = a + 5

# — Arquitetura RISC

## Convertendo o código

Para isso, precisaremos desmembrar a operação complexa em várias simples:

**1**

Buscar o valor da variável *a* na memória (LOAD).

**2**

Realizar a adição imediata de 5.

**3**

Salvar o valor resultante em um registrador (ADDI).

**4**

Guardar o novo valor na variável *a* em memória (STORE).

## — Arquitetura RISC

### Convertendo o código

Na abordagem CISC esse código usava apenas uma instrução – “ADDI M[&a], 5”.

Essa sequência de operações é exatamente a mesma do exemplo do pipeline dado anteriormente:



# — Arquitetura RISC

## Convertendo o código

- Pipeline de RISC tem operações demoradas (10 ciclos) comparado ao CISC (7 ciclos).
- LOAD e STORE são custosas e atrasam o pipeline, mas não são frequentemente usadas.
- Vários registradores permitem que variáveis fiquem nos registradores até serem necessárias.
- Reduzir conjunto de instruções permite Unidade de Controle menor, sem microprograma.

# Exemplo

Veja algumas arquiteturas que usam a abordagem RISC:

Processador	Ano	Endereçamento	Registradores	Tamanho de Instrução
MIPS	1981	R-R	04 a 32	4 bytes
ARM/A32	1983	R-R	15	4 bytes
SPARC	1985	R-R	32	4 bytes
PowerPC	1990	R-R	32	4 bytes
		R-R		4 bytes

Tamanho fixo de instrução: Para simplificar a decodificação.

Operações com endereçamento: Registrador-Registrador.



## — Tendências

- Abordagem CISC evoluiu naturalmente; termo surgiu retroativamente após a chegada do RISC.
- Atualmente, processadores híbridos combinam instruções complexas (CISC) e otimizadas (RISC).
- Mistura busca as melhores características de ambas as abordagens para otimizar o desempenho.