Algoritmos e Estrutura de Dados I

Aula 04

Introdução a Computação – Parte 3

Prof. Dr. Dilermando Piva Jr

1º Semestre - CDN

PROBLEMA DO MUNDO REAL

Uma empresa de e-commerce está enfrentando desafios com seu sistema de controle de estoque.

O sistema atual está causando atrasos nas atualizações de estoque, inconsistências nos registros e lentidão nas operações.

Isso resultou em pedidos equivocados, clientes insatisfeitos e perda de receita.

A empresa deseja melhorar a eficiência do seu sistema de controle de estoque para atender melhor às demandas do mercado.

Problema: Otimização de Desempenho do Sistema de Controle de Estoque de uma Empresa de E-Commerce

DETALHAMENTO

1. **Análise de Hardware:**
   - **Avaliar o hardware atual usado no sistema de controle de estoque.**
   - **Identificar possíveis gargalos de desempenho.**
   - **Propor melhorias ou atualizações no hardware para otimizar o processamento de dados.**
2. **Arquitetura de Software:**
   - **Analisar a arquitetura de software do sistema.**
   - **Identificar e corrigir possíveis problemas de eficiência no código.**
   - **Propor melhorias na estrutura do software para otimizar a gestão de estoque.**
3. **Sistema Computacional:**
   - **Examinar a interação entre hardware e software no contexto do sistema de controle de estoque.**
   - **Aplicar conceitos de RISC e CISC para escolher uma arquitetura mais adequada.**
   - **Utilizar álgebra de Boole para otimizar operações lógicas no sistema.**

DETALHAMENTO

1. Imersão na Experiência Concreta:
   - Os alunos investigam os problemas reais enfrentados pela empresa de e-commerce.
2. Observação Reflexiva e Obtendo Feedback:
   - Os alunos discutem as observações e compartilham feedback sobre as limitações atuais do sistema.
3. Conceituação Abstrata e Correção de Erros:
   - Apresentação dos conceitos de hardware, software, RISC, CISC, álgebra de Boole e sua aplicação ao problema.
4. Experimentação Ativa e Utilização de Metacognição:
   - Os alunos propõem soluções, implementam melhorias no sistema e refletem sobre os resultados.
5. Reprocessamento e Aprendizado Contínuo:
   - Os alunos revisitam as soluções, adaptando-as conforme necessário, e exploram a conexão com conceitos anteriores.
6. Colaboração e Envolvimento no Mundo Real:
   - Os alunos colaboram para implementar as soluções no sistema real da empresa, compartilham resultados e recebem feedback do desempenho aprimorado.

REFLEXÃO E FEEDBACK

Após a imersão no problema, os alunos devem refletir sobre os desafios encontrados durante a investigação.

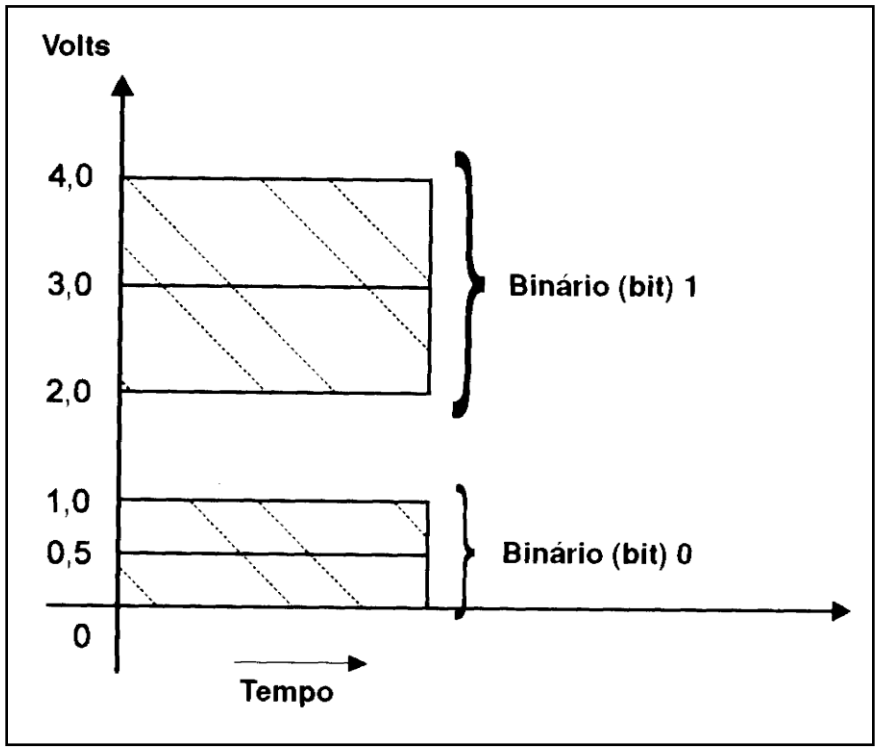
Encoraje discussões em grupo para que compartilhem suas experiências, erros e insights.

Ofereça feedback construtivo sobre as abordagens utilizadas pelos alunos, destacando pontos positivos e áreas de melhoria.

Conceitos Básicos de Eletrônica Digital

- **Computador digital** - máquina projetada para armazenar e manipular informações representadas apenas por algarismos (ou dígitos) e que só podem assumir dois valores distintos, 0 e 1.

- **Informação binária** (0 ou 1) - representada em um sistema digital por quantidades físicas (sinais elétricos).



Conceitos Básicos de Eletrônica Digital

- Operações de um computador digital - combinações de simples operações aritméticas e lógicas básicas: somar bits, complementar bits (para fazer subtrações), comparar bits, mover bits.

- As operações são fisicamente realizadas por circuitos eletrônicos, chamados **circuitos digitais**.

- Componentes básicos dos circuitos digitais - **"portas" (*gates*) lógicas**, por permitirem ou não a passagem dos sinais.

- **Circuitos lógicos** - circuitos que contêm as portas lógicas.

Conceitos Básicos de Eletrônica Digital

- **Computadores digitais** (binários) - construídos com circuitos eletrônicos digitais - as portas lógicas (circuitos lógicos).

- Um computador digital é construído, então, contendo circuitos lógicos (ou portas), convenientemente distribuídos e organizados, de modo que:
  - alguns servirão para armazenamento de valores,
  - outros permitirão e controlarão o fluxo de sinais entre componentes e
  - outros serão utilizados para realizar operações matemáticas.

Conceitos Básicos de Eletrônica Digital

- O projeto de circuitos digitais e a análise de seu comportamento podem ser realizados através do emprego de conceitos e regras estabelecidas pela **álgebra de chaveamentos**, um ramo da álgebra moderna ou **álgebra de Boole**, conceituada pelo matemático inglês George Boole (1815 - 1864).

Conceitos Básicos de Eletrônica Digital

É importante entender o significado dos seguintes conceitos: Lógica e Álgebra de Boole e como estes conceitos podem ser empregados para a implementação das portas lógicas e, conseqüentemente, dos circuitos lógicos (digitais) e computadores digitais.

Conceitos Básicos de Eletrônica Digital

- A **lógica** é a base da eletrônica digital e da informática. Esta surgiu na Grécia antiga com a contribuição de três filósofos: **Sócrates**, **Platão** e **Aristóteles**.

  - Sócrates não deixou seus ensinamentos por escrito.
  - Platão (seguidor de Sócrates) escreveu vários de seus diálogos e desenvolveu sua filosofia abrangendo a ética, a política e o conhecimento, tendo como princípio o método da investigação.
  - Aristóteles, baseado nos diálogos escritos por Platão, observou que a linguagem deve ter uma estrutura lógica, para que leve, necessariamente, a uma verdade.
  - Pelo método de investigação de Sócrates, se duas verdades são alcançadas individualmente, ao juntá-las tem-se uma única verdade.

Sócrates, considerado um dos homens mais sábios da humanidade, notabilizou-se por afirmar que era sábio justamente por "**saber que nada sabia**".

Conceitos Básicos de Eletrônica Digital

- No século XIX, a teoria de Aristóteles foi sintetizada em forma de álgebra, ganhando o nome de Álgebra Booleana.

- A Álgebra de Boole permite que **uma afirmação** (lógica) possa ser expressa matematicamente.

- Boole construiu sua lógica a partir de símbolos, representando as expressões por letras e ligando-as através de conectivos - **símbolos algébricos**.

- Boole, através de seu livro "*An investigation of the laws of thought*" (Uma investigação das leis do pensamento) apresentou a **lógica binária.**

Conceitos Básicos de Eletrônica Digital

- A **lógica** teve como objetivo modelar o raciocínio humano.

- Partindo de frases declarativas (*proposições*), que podem ser ***verdadeiras*** ou ***falsas***, estuda-se o processo de construção e a veracidade de outras proposições usando conectivos.

- Na lógica proposicional associa-se a cada proposição um valor lógico: ou verdade (1) ou falso (0).

Da **Lógica** nasceu a **Lógica Matemática** e, dentro desta, várias filosofias da lógica que interpretam os cálculos simbólicos e sua sistematização axiomática.

Álgebra de Boole

- **Operação lógica –** realizada sobre um ou mais valores lógicos para produzir um certo resultado (também um valor lógico).

- Assim como na álgebra comum, é necessário definir símbolos matemáticos e gráficos para representar as operações lógicas (e os operadores lógicos).

- Resultados possíveis de uma operação lógica:
  - 0 (FALSO, **F**= bit 0) - nível baixo
  - 1 (VERDADEIRO, **V** = bit 1) - nível alto (Lógica Positiva)

Álgebra de Boole

OPERADORES LÓGICOS BÁSICOS

- Os conectivos ou OPERADORES LÓGICOS ou FUNÇÕES LÓGICAS são:
  - **E (ou AND)** - uma sentença é verdadeira **SE - e somente se** - todos os termos forem verdadeiros.
  - **OU (ou OR)** - uma sentença resulta verdadeira se **QUALQUER UM** dos termos for verdadeiro.
  - **NÃO (ou NOT)** - este operador **INVERTE** um termo.

Álgebra de Boole

OPERADORES LÓGICOS BÁSICOS

- Os operadores lógicos são representados por:
  - **E** → • (um ponto, como se fosse uma multiplicação)
  - **OU** → **+** (o sinal de soma)
  - **NOT** → $\overline{}$ (ou ’) (uma barra horizontal sobre o termo a ser invertido ou negado).

Simbologia definida pela ANSI

Álgebra de Boole

FUNÇÕES LÓGICAS

- Operadores que possuem como entrada pelo menos uma variável lógica e uma saída.
- Dada uma variável lógica (**A**), é possível construir uma função desta variável, **f**(**A**).
- Operações da álgebra booleana aplicadas a uma ou mais variáveis lógicas.

- **Funções básicas**: E, OU e INVERSORA (AND, OR e NOT ou INVERTER)

- **Derivadas**: (NAND, NOR, XOR e XNOR).

Álgebra de Boole

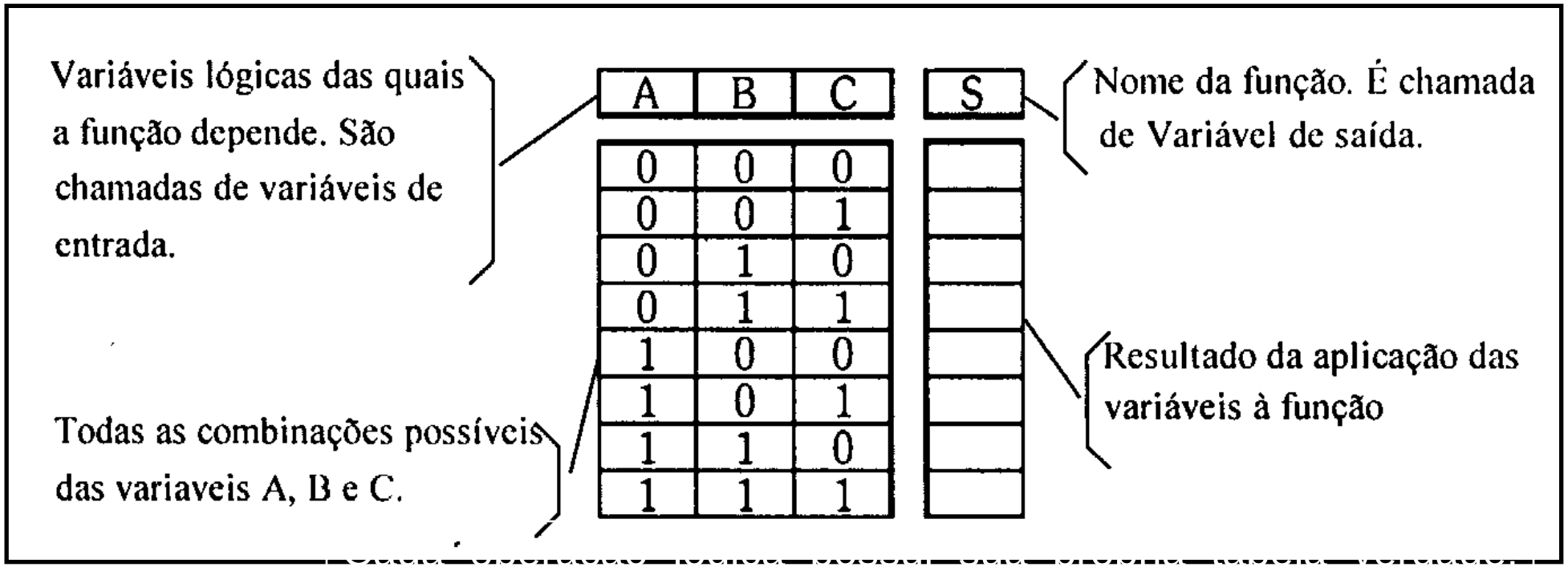
- A partir das combinações dos valores de entrada, determina-se todos os valores possíveis de resultado de uma dada operação lógica.

- Essas possibilidades podem ser representadas de forma tabular, e o conjunto se chama **TABELA VERDADE**.

- **TABELA VERDADE** - tabela que representa todas as possíveis combinações das variáveis de entrada de uma função, e os seus respectivos valores de saída.

Álgebra de Boole

Tabela-verdade



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

Álgebra de Boole

FUNÇÃO AND (E)

$$S = A \cdot B$$

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

FUNÇÃO OR (OU)

$$S = A + B$$

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

FUNÇÃO NOT (INVERTER OU NÃO)

$$S = \overline{A}$$

A	S
0	1
1	0

Álgebra de Boole

FUNÇÃO NAND (NÃO E)

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

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

FUNÇÃO NOR (NÃO OU)

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

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Álgebra de Boole

FUNÇÃO XOR (OU EXCLUSIVO)

$S = A \oplus B$

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

FUNÇÃO XNOR (OU COINCIDÊNCIA)

$S = A \otimes B$

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

XOR - a saída será verdade se **exclusivamente uma ou outra entrada** for verdade. (XNOR - inverso da XOR). Isto só se aplica se houver apenas 2 entradas.

Álgebra de Boole e os Computadores Digitais

- O projeto de elementos digitais está relacionado com a **conversão de idéias em *hardware* real**, e os elementos encontrados na **álgebra booleana** permitem que uma idéia, uma afirmação, possa ser expressa matematicamente.

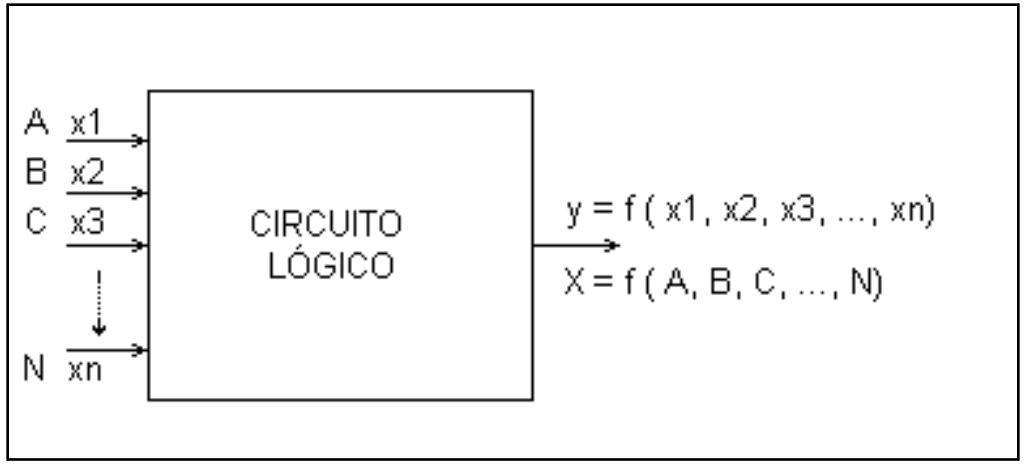
- A **álgebra booleana** permite também que a expressão resultante da formulação matemática da idéia possa ser simplificada e, finalmente, **convertida no mundo real do *hardware* de portas lógicas e outros elementos digitais**.

O que são exatamente?

Álgebra de Boole e os Computadores Digitais

- **Portas lógicas**: dispositivos dos circuitos digitais - implementam funções lógicas.

- São dispositivos ou circuitos lógicos que operam um ou mais sinais lógicos de entrada para produzir uma (e somente uma) saída, a qual é dependente da função implementada no circuito.



Álgebra de Boole e os Computadores Digitais

Como os conceitos da álgebra de chaveamentos (ramo da álgebra do Boole) são aplicados ao projeto dos computadores digitais?

- **Primeiros computadores fabricados** (Ex.: ENIAC) - trabalhavam em DECIMAL - grande complexidade ao projeto e construção dos computadores, tendo por conseqüência um custo muito elevado.

- **Aplicação da álgebra de Boole** – uso de apenas dois algarismos 0 (F) e 1(V) $\rightarrow$ simplificação do projeto e construção dos computadores.

Álgebra de Boole e os Computadores Digitais

- A chave de tudo é um circuito eletrônico chamado **CHAVE AUTOMÁTICA**.

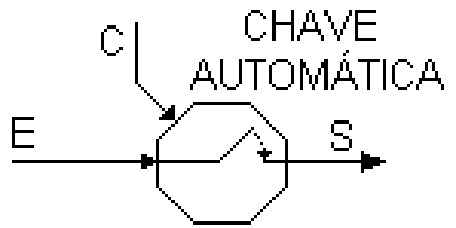
- **Como funciona uma chave automática?**

- Considerar um circuito chaveador com as seguintes entradas:

  – uma fonte de alimentação (fornece energia para o circuito)

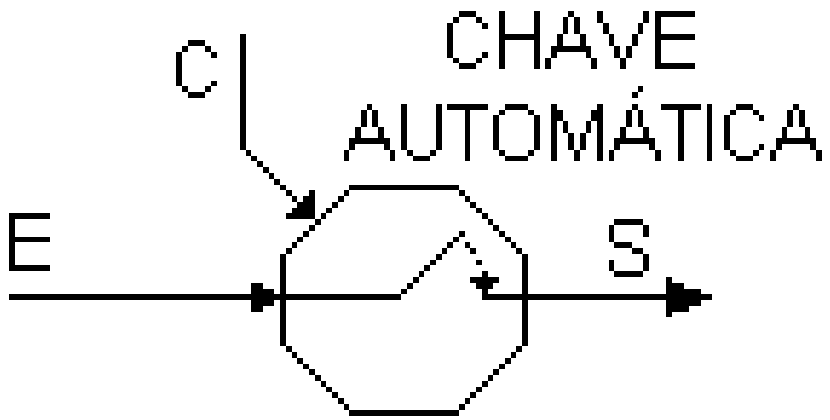
  – um fio de controle (comanda a operação do circuito)

  – um fio de saída (conduz o resultado)



Álgebra de Boole e os Computadores Digitais

- Sinal **C = 0** (ou F) $\Rightarrow$ **S = 0** (ou Falso). A chave permanece aberta.
- Sinal **C = 1** (ou V) $\Rightarrow$ **S = 1** (ou V). A chave muda de posição.
- A posição da chave se manterá enquanto não ocorrer um novo sinal na entrada.



Álgebra de Boole e os Computadores Digitais

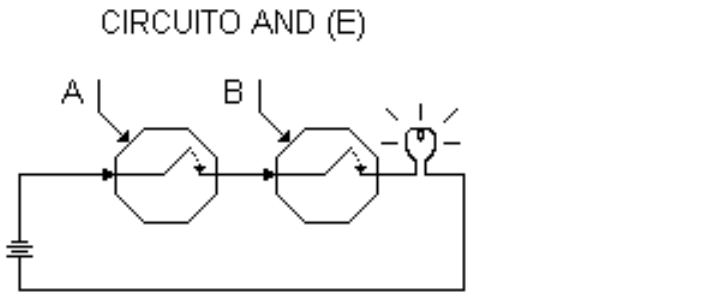
- A chave automática foi inicialmente implementada com relés eletromecânicos e depois com válvulas eletrônicas.

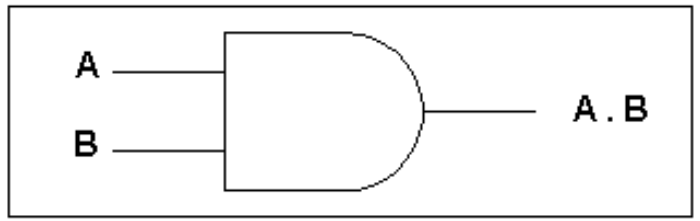
- A partir da metade da década de 50, passaram a ser utilizados dispositivos em estado sólido - os TRANSISTORES, inventados em Stanford em 1947.

- Modernos Circuitos Integrados e microprocessadores são implementados com milhões de transistores "impressos" em minúsculas pastilhas.

Álgebra de Boole e os Computadores Digitais

- **Ligação em SÉRIE** de duas chaves automáticas (com uma lâmpada ligada ao circuito).



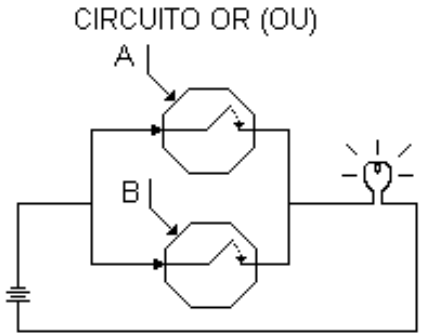


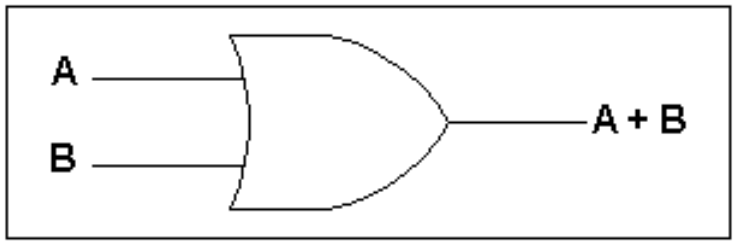
A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

PORTA E (AND GATE) - circuito que implementa a **função E**.

Álgebra de Boole e os Computadores Digitais

- **Ligação em PARALELO** de duas chaves automáticas (com uma lâmpada ligada ao circuito).



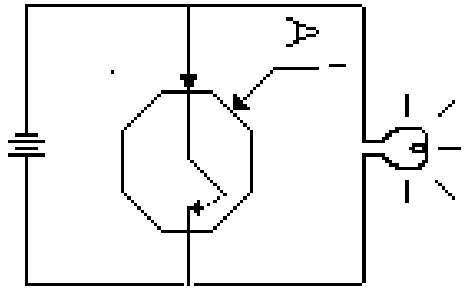


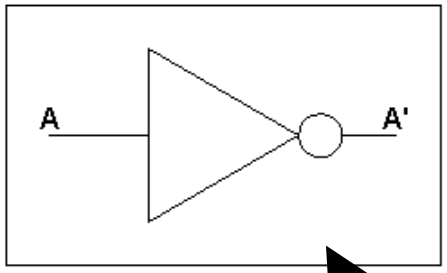
A	B	L
0	0	0
0	1	1
1	0	1
1	1	1

PORTA OU (OR GATE) - circuito que implementa a **função OR**.

Álgebra de Boole e os Computadores Digitais

- **Ligação** de uma chave automática (com uma lâmpada ligada ao circuito).





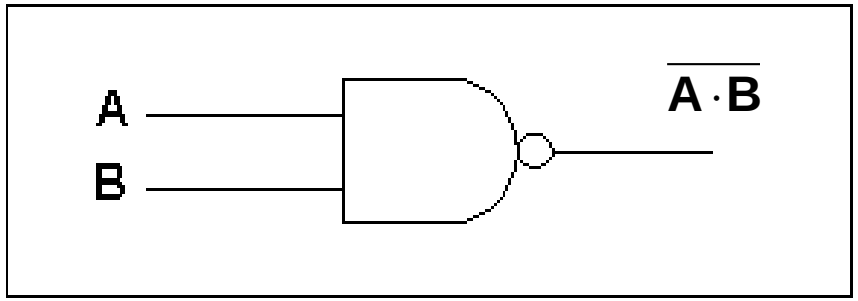
A	L
0	1
1	0

PORTA NÃO (NOT GATE ou INVERTER GATE) - circuito que implementa a **função NÃO**.

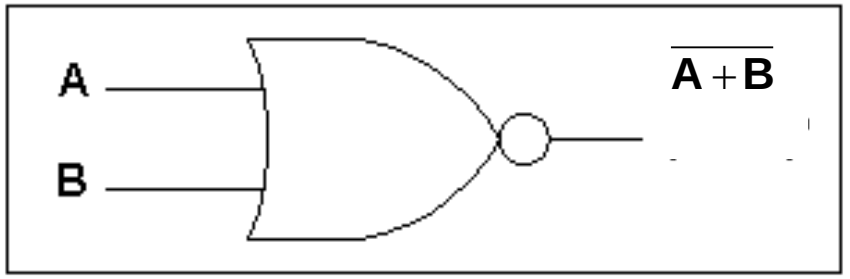
Álgebra de Boole e os Computadores Digitais

- **Demais portas lógicas:**

PORTA NAND (NAND GATE) - circuito que implementa a **função NAND**.



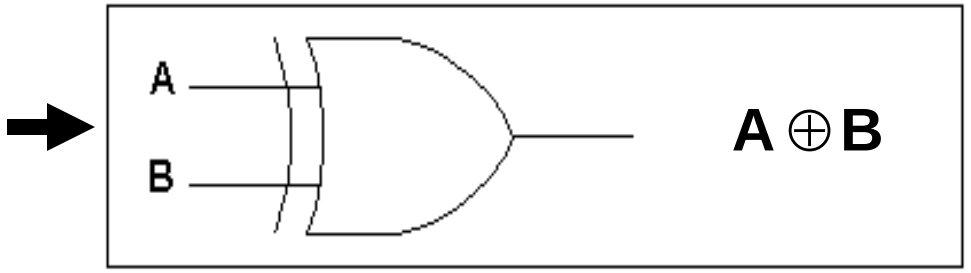
PORTA NOR (NOR GATE) - circuito que implementa a **função NOR**.



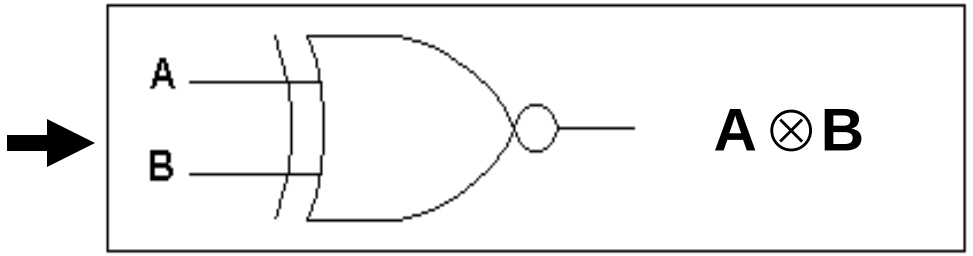
Álgebra de Boole e os Computadores Digitais

- **Demais portas lógicas:**

PORTA XOR (XOR GATE) - circuito que implementa a **função XOR**.



PORTA XNOR (XNOR GATE) - circuito que implementa a **função XNOR**.

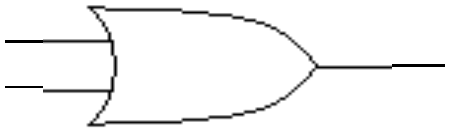
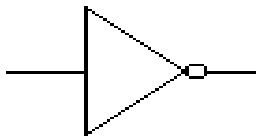


Número par de entradas - portas XOR e XNOR possuem **saídas complementares** entre si.
Número ímpar de entradas, as saídas das portas XOR e XNOR são **iguais entre si**.

Álgebra de Boole e os Computadores Digitais

- **Quadro Resumo**

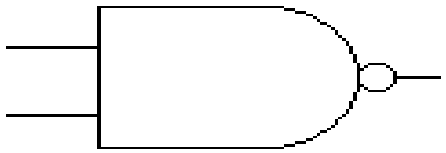
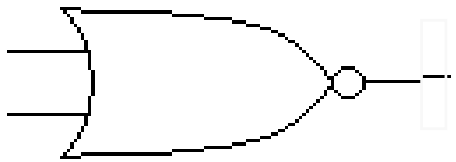
BLOCOS LÓGICOS BÁSICOS

Porta	Símbolo usual	Tabela Verdade	Função Lógica	Expressão
E AND		A B S 0 0 0 0 1 0 1 0 0 1 1 1	**Função E**: assume 1 quando todas as variáveis forem 1 e 0 nos outros casos.	$S = A \cdot B$
OU OR		A B S 0 0 0 0 1 1 1 0 1 1 1 1	**Função OU**: assume 0 quando todas as variáveis forem 0 e 1 nos outros casos.	$S = A + B$
NÃO NOT		A S 0 1 1 0	**Função NOT**: inverte a variável aplicada a sua entrada.	$S = \overline{A}$

Álgebra de Boole e os Computadores Digitais

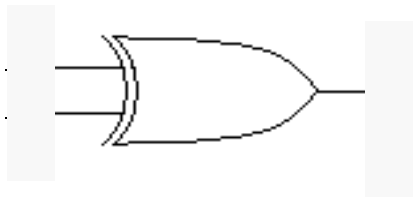
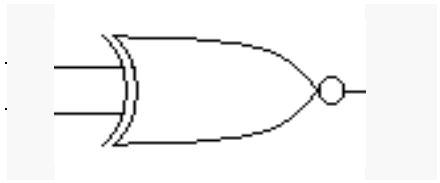
- **Quadro Resumo**

BLOCOS LÓGICOS BÁSICOS

Porta	Símbolo usual	Tabela Verdade	Função Lógica	Expressão
NÃO E NAND		A B S 0 0 1 0 1 1 1 0 1 1 1 0	**Função NÃO E**: Inverso da função E	$S = \overline{A \cdot B}$
NÃO OU NOR		A B S 0 0 1 0 1 0 1 0 0 1 1 0	**Função NÃO OU**: Inverso da função OU	$S = \overline{A + B}$

Álgebra de Boole e os Computadores Digitais

- **Quadro Resumo**

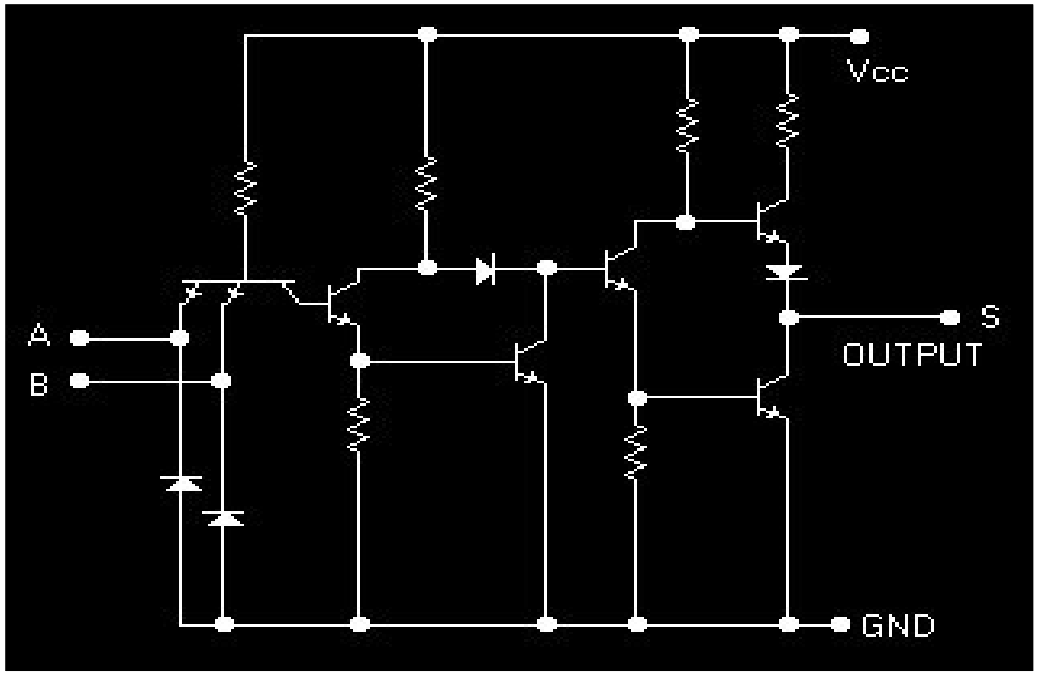
BLOCOS LÓGICOS BÁSICOS				
Porta	**Símbolo usual**	**Tabela Verdade**	**Função Lógica**	**Expressão**
OU EXCLUSIVO XOR		A B S 0 0 0 0 1 1 1 0 1 1 1 0	**Função XOR**: Assume 1 quando as duas variáveis assumirem valores diferentes entre si.	$S = A \oplus B$
COINCIDÊNCIA XNOR		A B S 0 0 1 0 1 0 1 0 0 1 1 1	**Função XNOR**: Assume 1 quando houver coincidência entre os valores das variáveis.	$S = A \otimes B$

As Portas lógicas **XOR e XNOR** são na verdade circuitos obtidos de portas lógicas básicas.

$$S = A \oplus B = \overline{A} \cdot B + A \cdot \overline{B} \qquad S = A \otimes B = A \cdot B + \overline{A} \cdot \overline{B}$$

Álgebra de Boole e os Computadores Digitais

- **Obs**.: O circuito elétrico da porta lógica que implementa a função **AND** é :

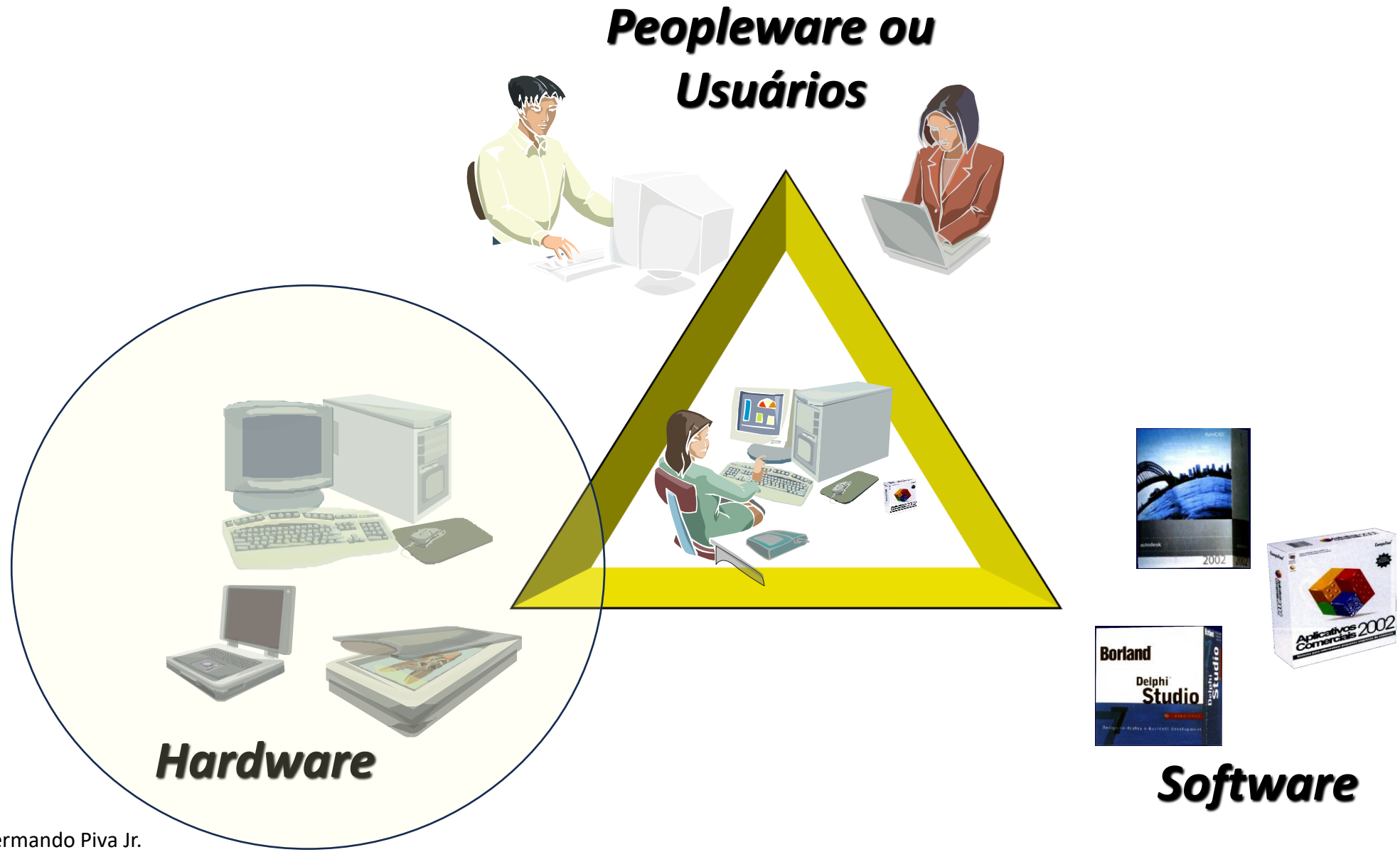


♦Torna-se difícil desenhar o esquema elétrico de um projeto composto por várias portas lógicas representadas desta forma.

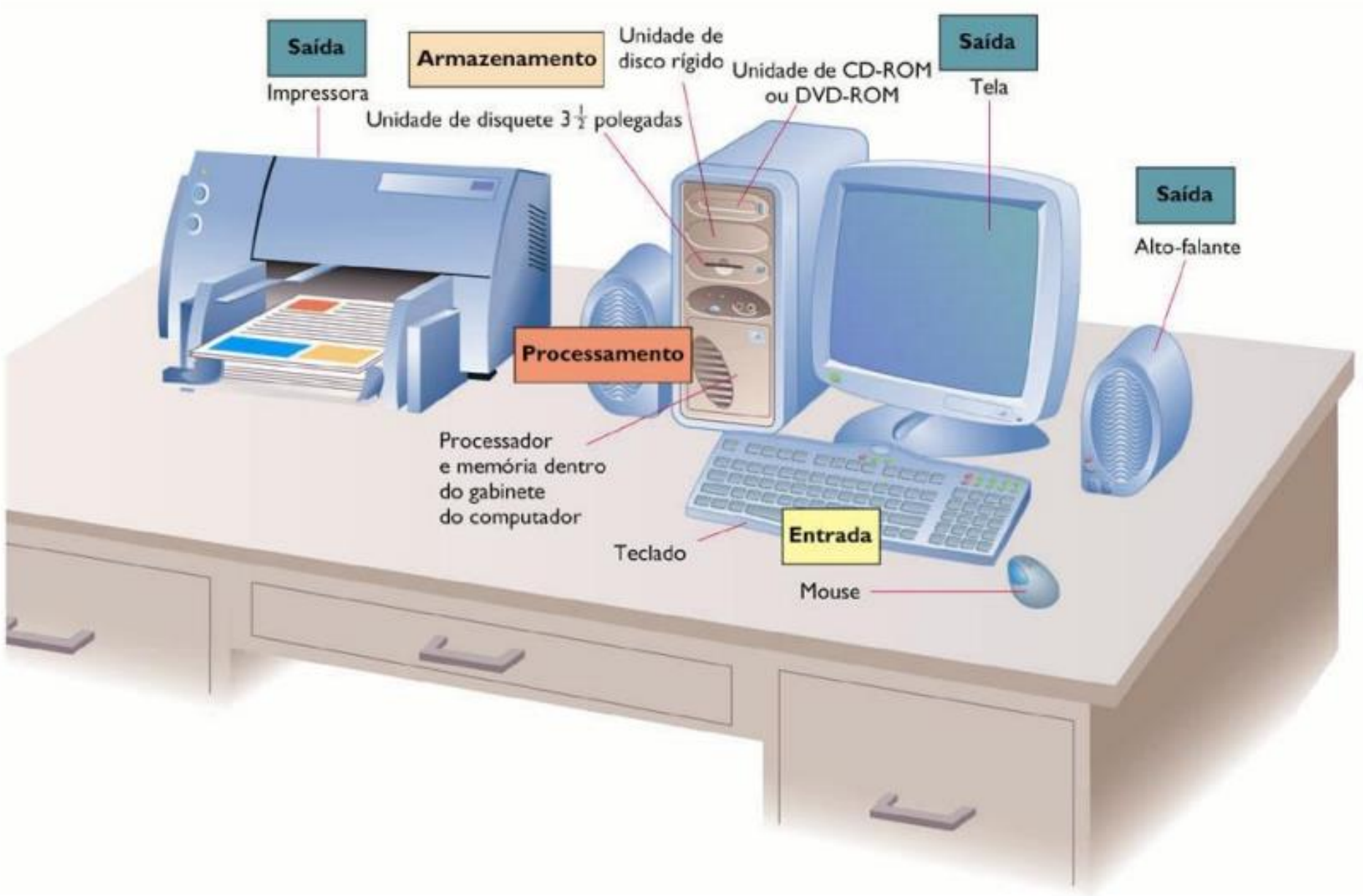
♦**Solução**: uso de uma **SIMBOLOGIA**.

SISTEMA COMPUTACIONAL

HARDWARE

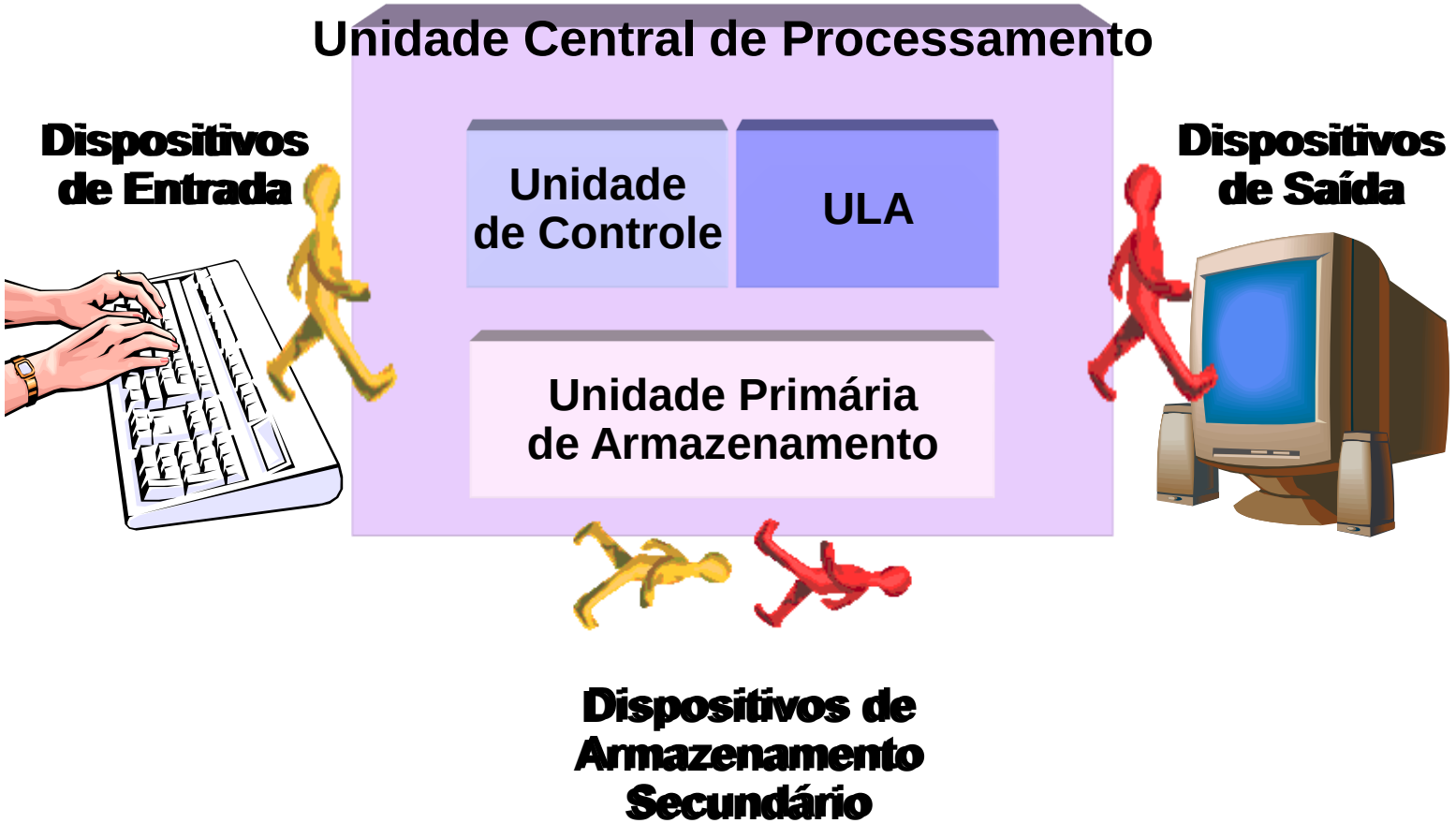


HARDWARE – Conceitos Básicos



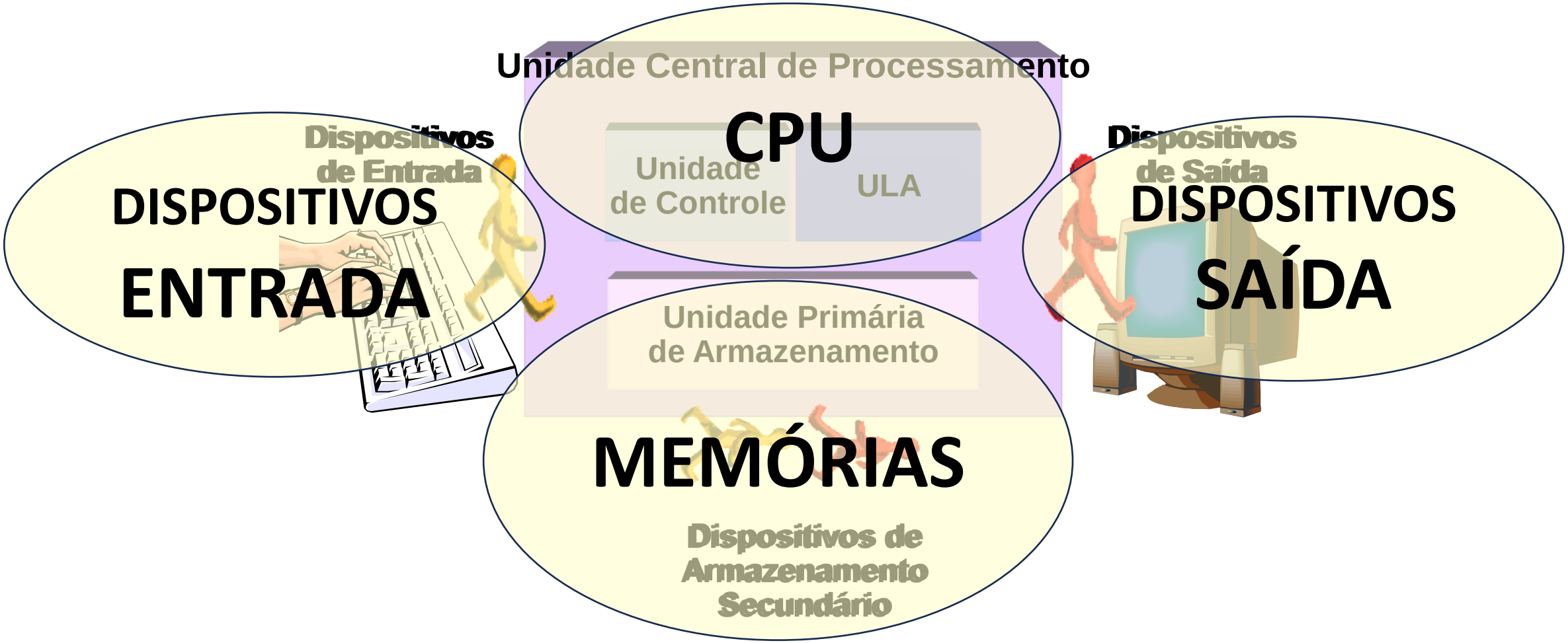
HARDWARE – Conceitos Básicos

Computadores atuais



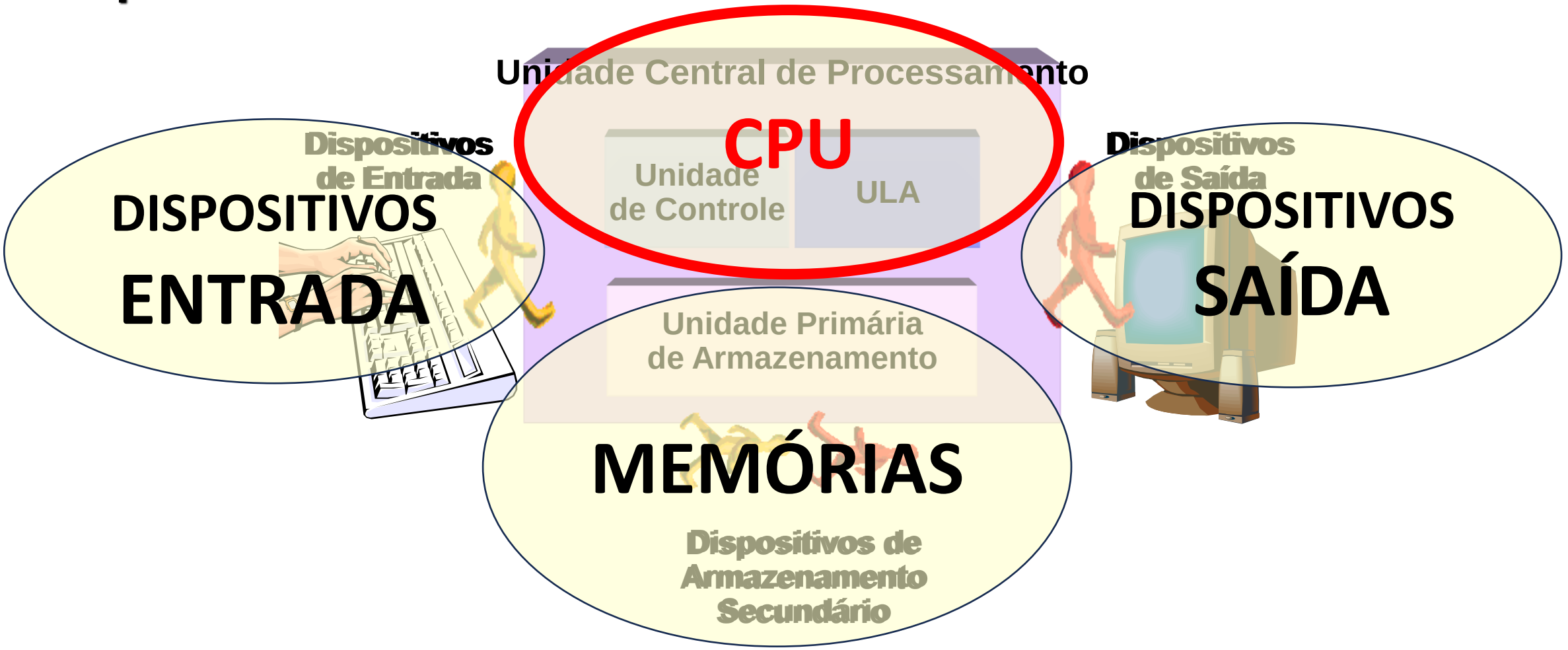
HARDWARE – Conceitos Básicos

Computadores atuais

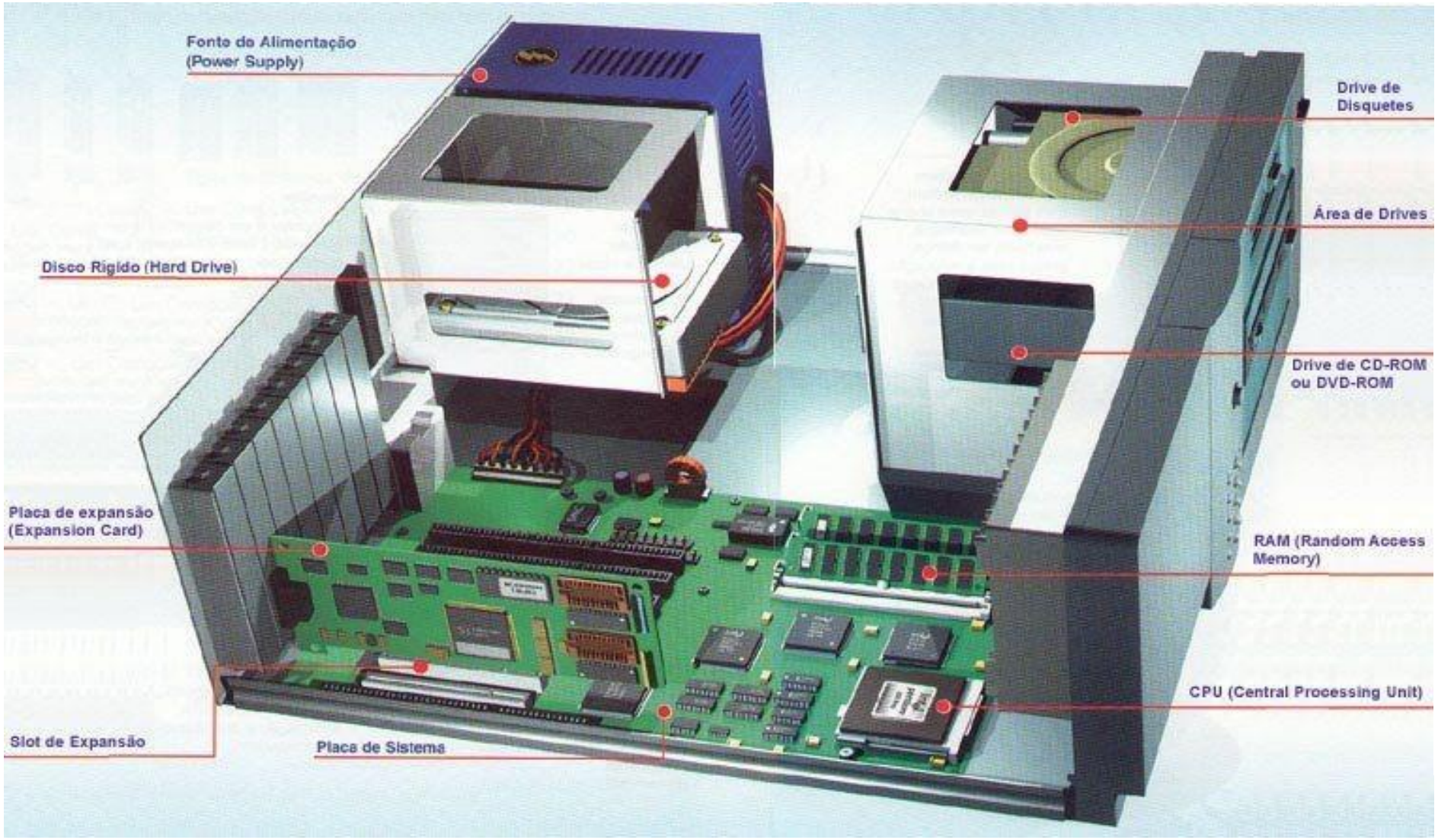


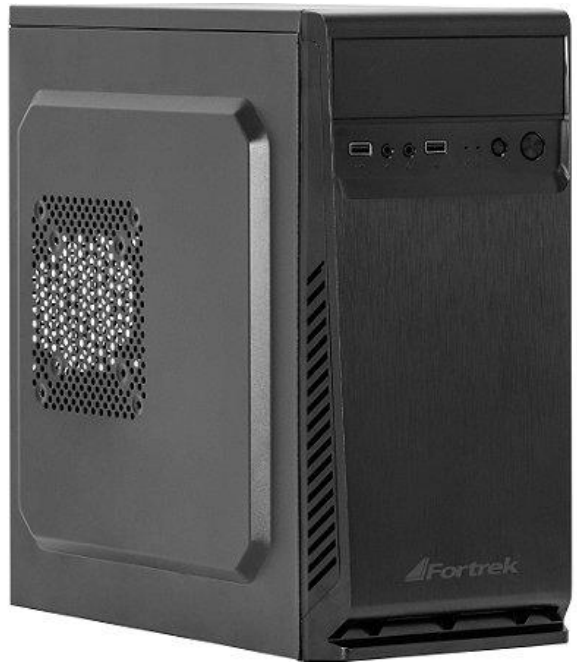
HARDWARE – UCP ou CPU

Computadores atuais



HARDWARE – UCP ou CPU



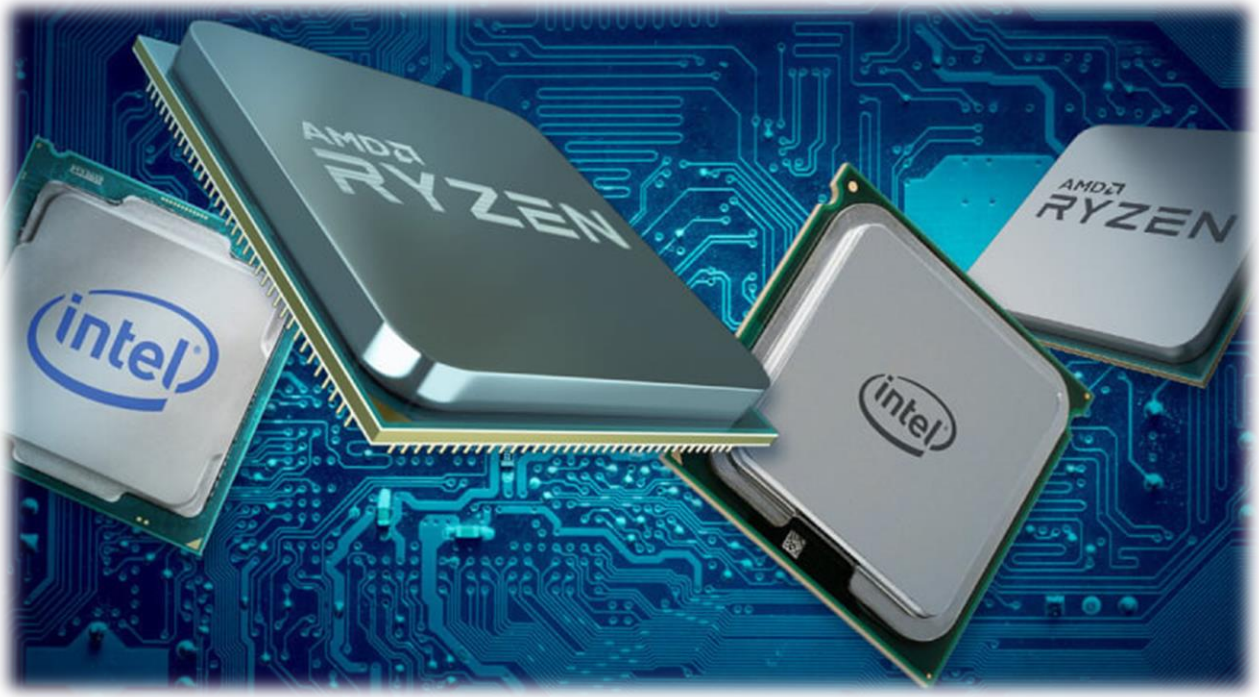


HARDWARE – UCP ou CPU

Unidade Central de Processamento – UCP (*Central Processing Unity* – CPU)

- Desempenha um papel crucial no funcionamento do sistema de computação.
- Responsável pela atividade-fim do sistema ⇒ computar, calcular e processar.
- Surgiu pela primeira vez em 1971, característica principal - em um único invólucro (pastilha-chip), são inseridos todos os elementos necessários à realização de suas funções, diminuindo o tamanho, o consumo de energia e o preço.
- Outra denominação - microprocessador (μ P).
- Primeiro processador - Intel 4004.

HARDWARE – UCP ou CPU

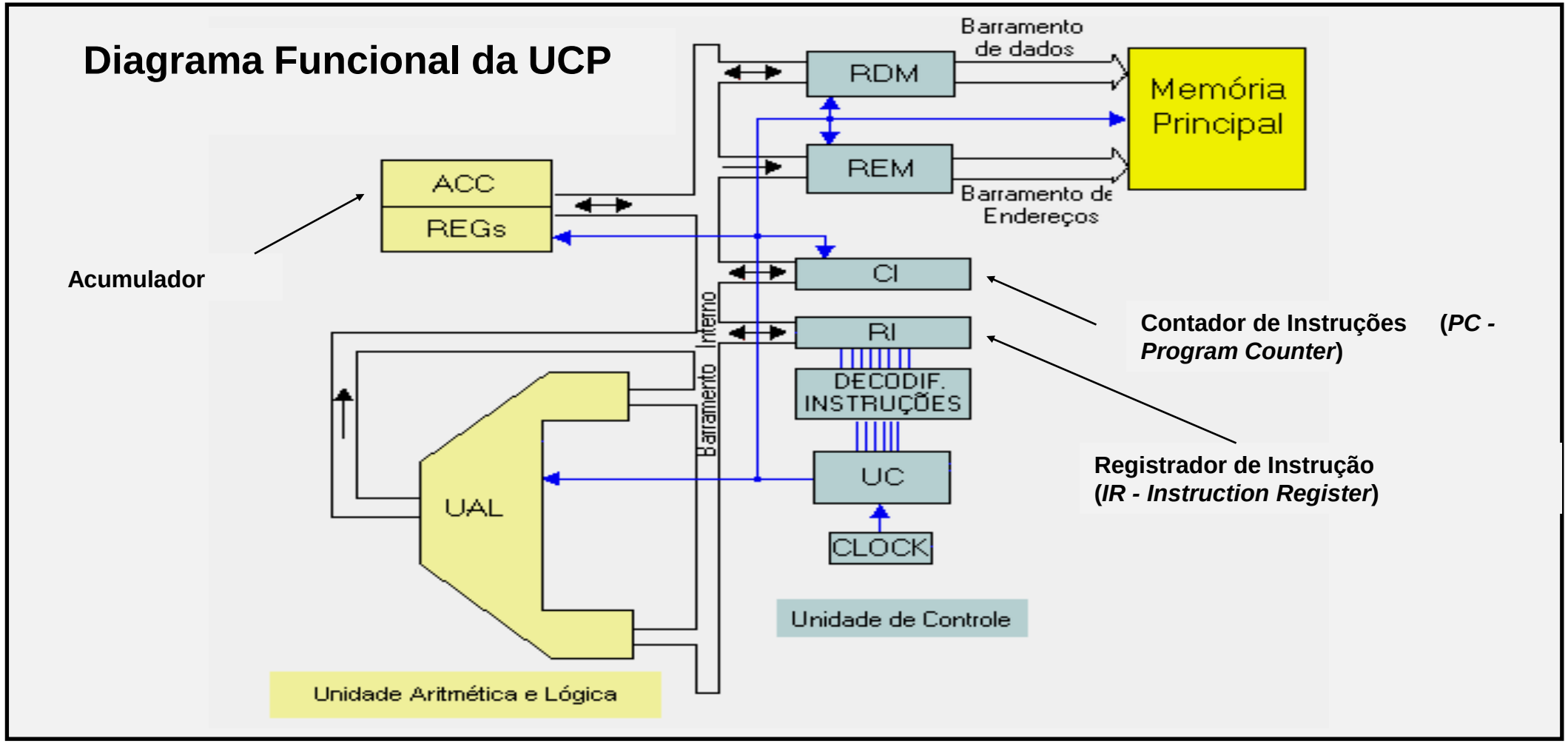


Funções realizadas pela UCP

- **Processamento**
  - operações aritméticas e lógicas
  - movimentação de dados
  - desvios
  - operações de entrada ou saída

- **Controle**
  - Busca, interpretação e controle da execução das instruções.
  - Controle da ação dos demais componentes do sistema de computação (memória, entrada/saída).

HARDWARE – UCP ou CPU



Obs.: Por simplicidade, será considerado o funcionamento serial em uma UCP.

HARDWARE – UCP ou CPU

UC - Unidade de Controle

- **Funções**: busca, interpretação e controle de execução das instruções, e o controle dos demais componentes do computador.
- Envia ordens de cálculo para a UAL, que indica os valores a processar, e os coloca nos registradores para esse efeito.
- A partir da UC a informação é transferida para as outras partes que constituem o computador, como a memória, os sistemas de E/S, etc..
- É o dispositivo mais complexo da UCP.
- Possui a lógica necessária para realizar a movimentação de dados e instruções de e para a UCP, através de sinais de controle que emite em instantes de tempo programados.
- Os sinais de controle, emitidos pela UC, ocorrem em vários instantes durante o período de realização de um ciclo de instrução e, de modo geral, todos possuem uma duração fixa e igual, originada em um gerador de sinais denominado relógio (***clock***).

HARDWARE – UCP ou CPU

UAL - Unidade Aritmética e Lógica

- **Função**: a efetiva execução das instruções.
- Aglomerado de circuitos lógicos e componentes eletrônicos simples que, integrados, realizam as operações aritméticas e lógicas (soma, subtração, multiplicação, divisão, AND, OR, XOR, complemento, deslocamento, incremento e decremento).
- Processadores modernos utilizam mais de uma UAL. Exemplo: Processadores Pentium contêm 3 UAL - 2 para processar números inteiros e a terceira, FPU (*Floating Point Unit*), para processar números fracionários.

HARDWARE – UCP ou CPU

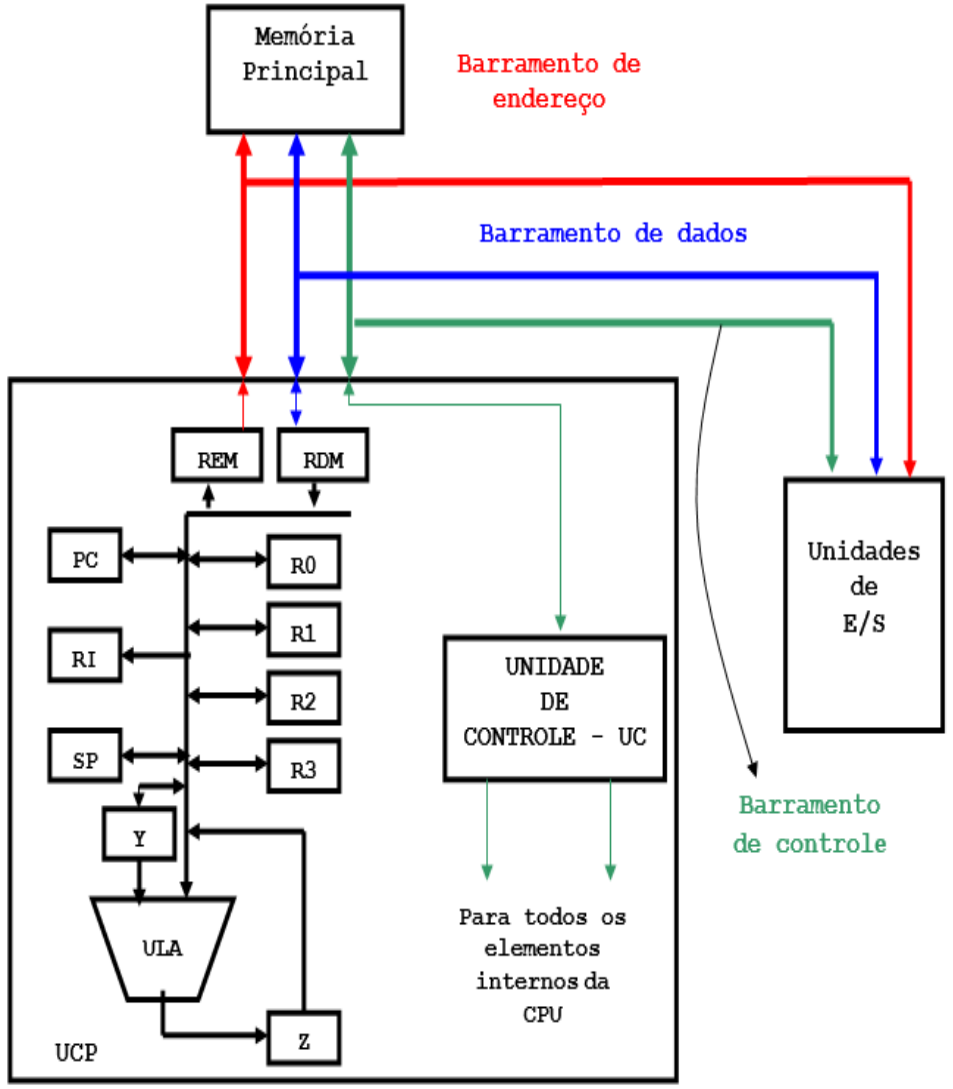
Registradores

- **Função**: armazenamento de dados e resultados que serão usados pela UAL.
- Servem de memória auxiliar básica para a UAL.
- Classificação (atual): registradores de uso geral e registradores de uso específico.
- Em geral, os registradores de dados da UCP têm uma largura (quantidade de bits que podem armazenar) igual ao tamanho estabelecido pelo fabricante para a palavra do referido processador.
- A quantidade e o emprego dos registradores variam bastante de modelo para modelo de UCP.

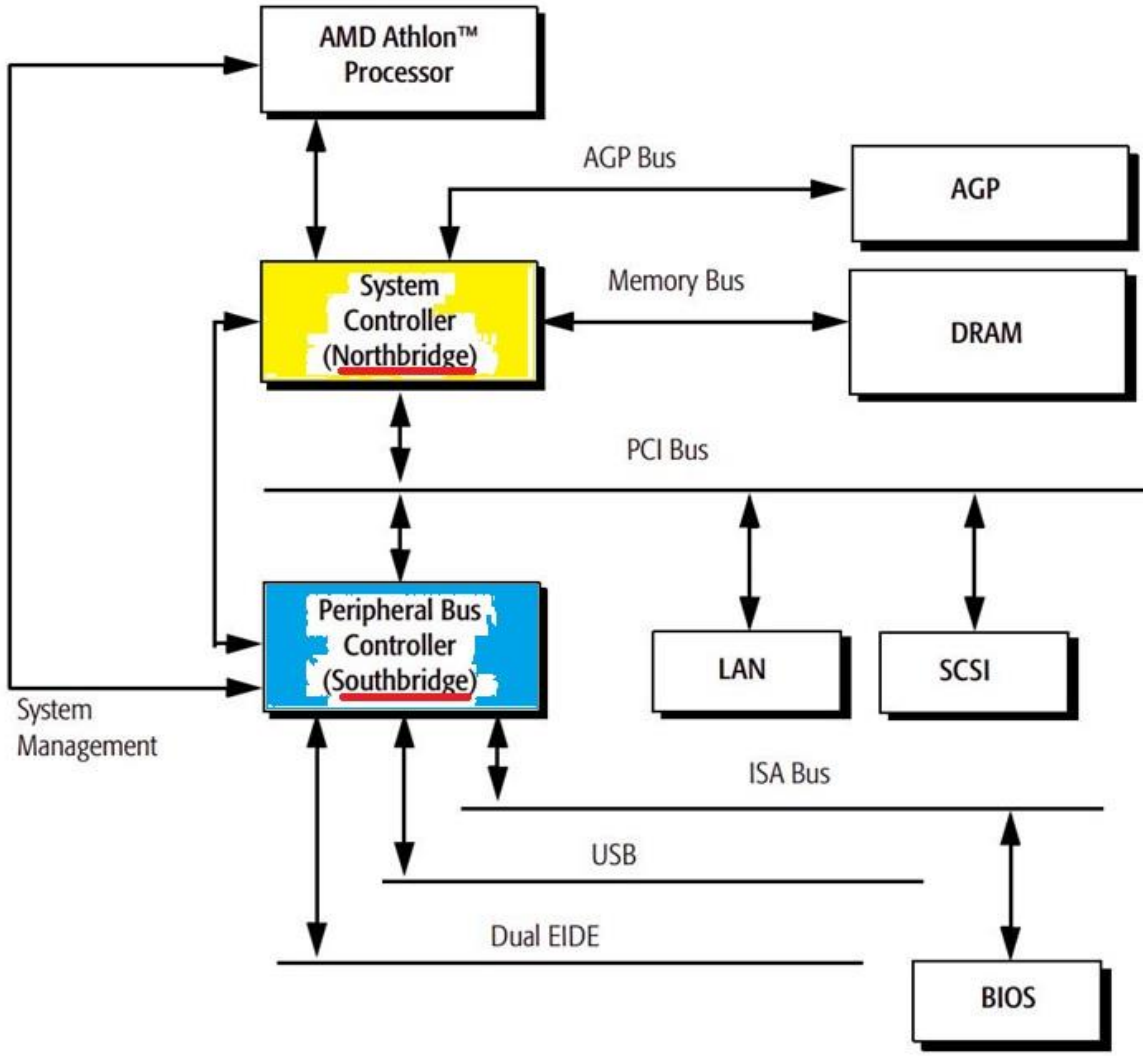
HARDWARE – UCP ou CPU

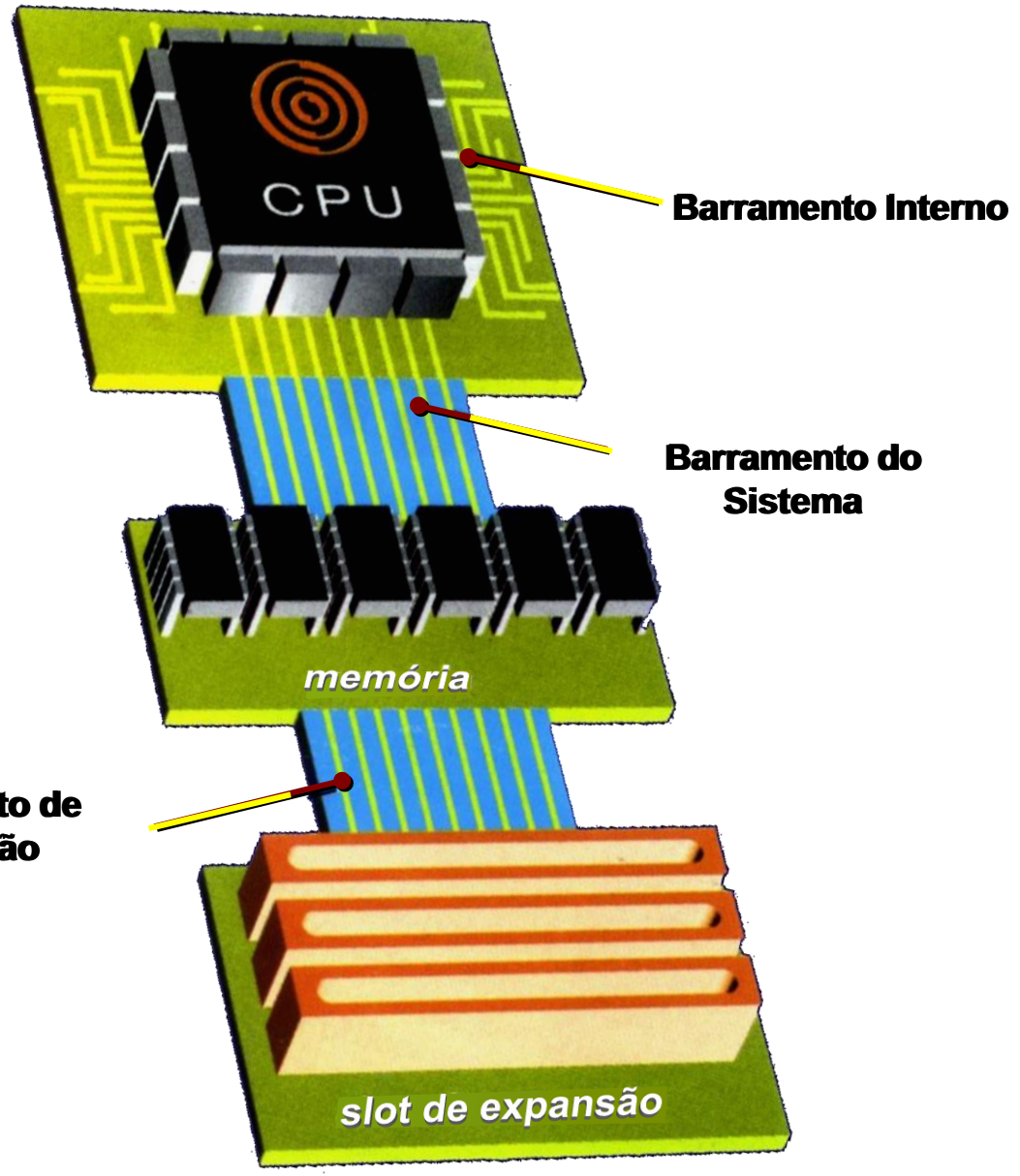
Barramentos

- Rede de linhas de comunicação que conecta os elementos internos do processador e que também conduz até os conectores externos que ligam o processador com os demais elementos do sistema de informática.
- Como um dado é composto por bits (geralmente um ou mais bytes) o barramento deverá ter tantas linhas condutoras quanto forem os bits a serem transportados de cada vez.
- Em alguns computadores (usando uma abordagem que visa a redução de custos), os dados podem ser transportados usando mais de um ciclo do barramento.



HARDWARE – Barramentos





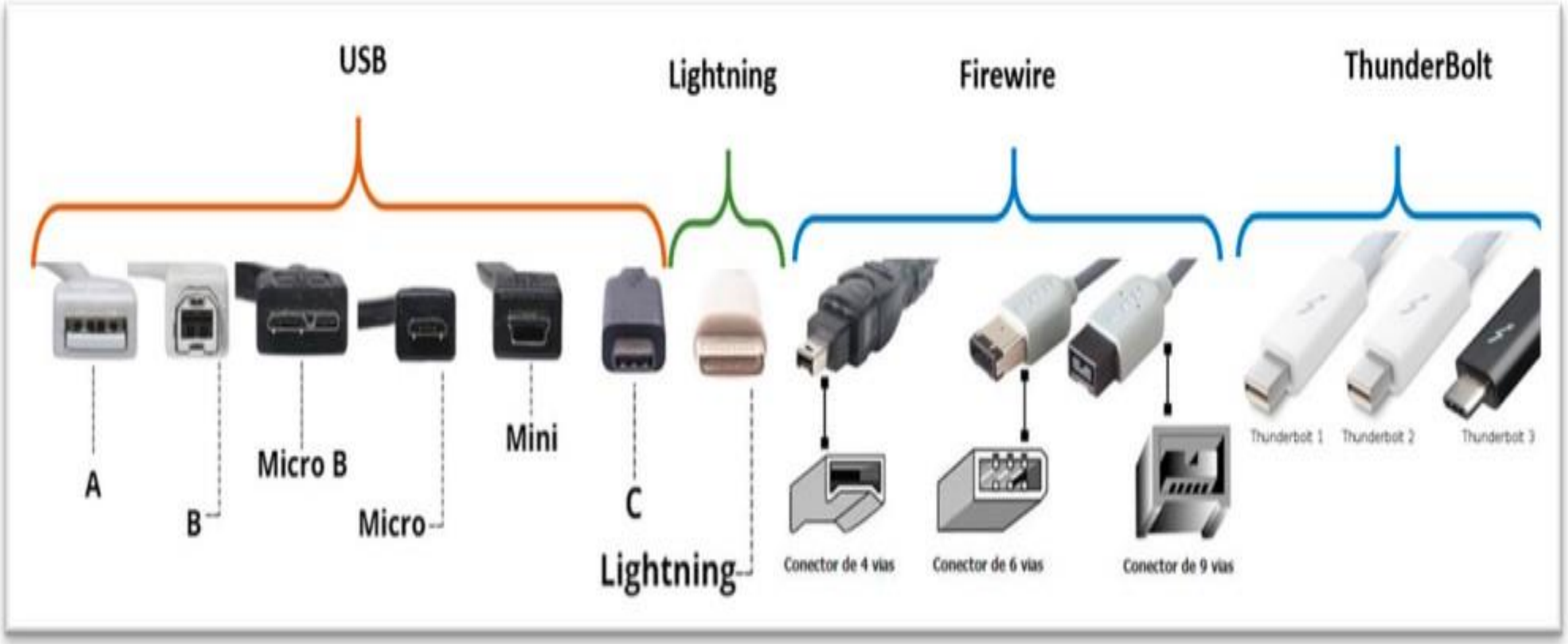
HARDWARE – Barramentos

Barramentos - Protocolos - Padronização

- ***UNIBUS*** **-** definido pela DEC, praticamente fora de uso.
- ***MCA*** (*Micro Channel Architecture*) - definido pela IBM, sistemas PS-2.
- ***ISA*** (*Industry Standard Adapter*) - definido pela IBM para o PC-AT e adotado por toda a indústria.
- ***EISA*** (*Extended ISA*) - praticamente abandonado.
- ***PCI*** (*Peripheral Component Interconnect*) - desenvolvido pela Intel, quase um padrão para o mercado, com barramento de E/S de alta velocidade.
- ***USB*** (*Universal Serial Bus*) - permite a conexão de muitos periféricos simultaneamente ao barramento e este, por uma única tomada, se conecta a placa mãe. Pretende ser norma os dispositivos que necessitem de baixo desempenho (Ex.: teclado, rato, *modem*, *scanner*, impressoras, etc).
- ***AGP*** (*Accelerated Graphics Port*) - visa acelerar as transferências de dados do vídeo para a memória, especialmente dados para 3D.

HARDWARE – Barramentos

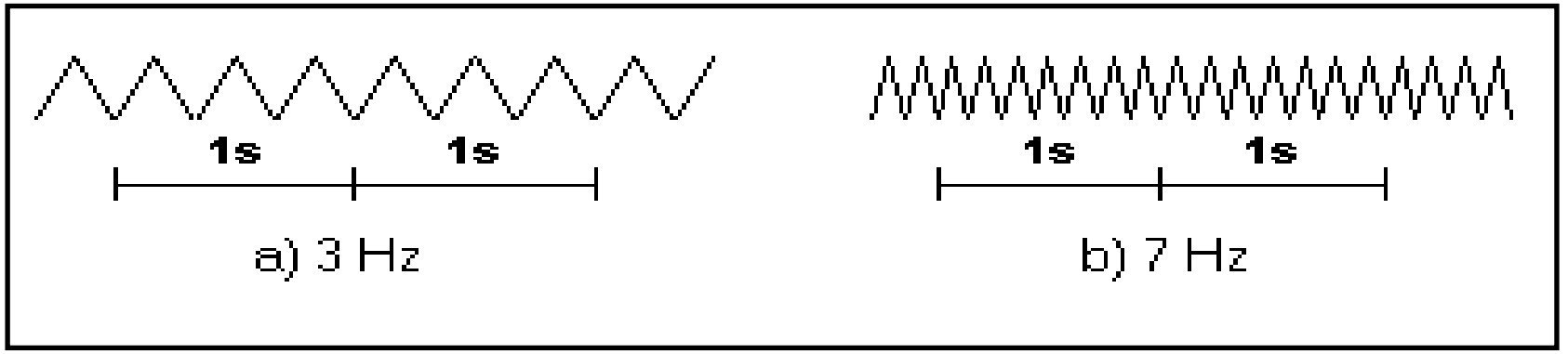
HARDWARE – Barramentos



HARDWARE – Demais componentes da CPU

- **Relógio (*clock*)**
  - dispositivo gerador de pulsos cuja duração é chamada de ciclo.
  - Freqüência - número de ciclos por segundo (Hz), usada também para definir a **velocidade do processador**.

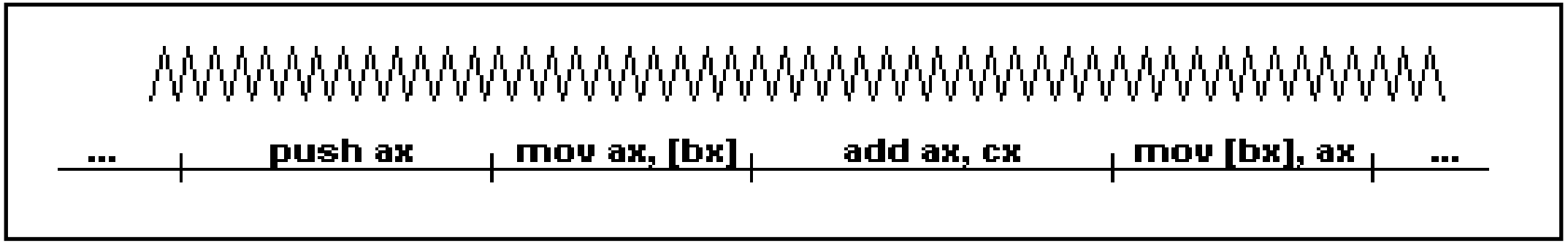
- O relógio nada mais é do que um oscilador externo ao microprocessador, que gera pulsos a intervalos regulares de tempo. A cada pulso, uma ou mais microoperações são realizadas.



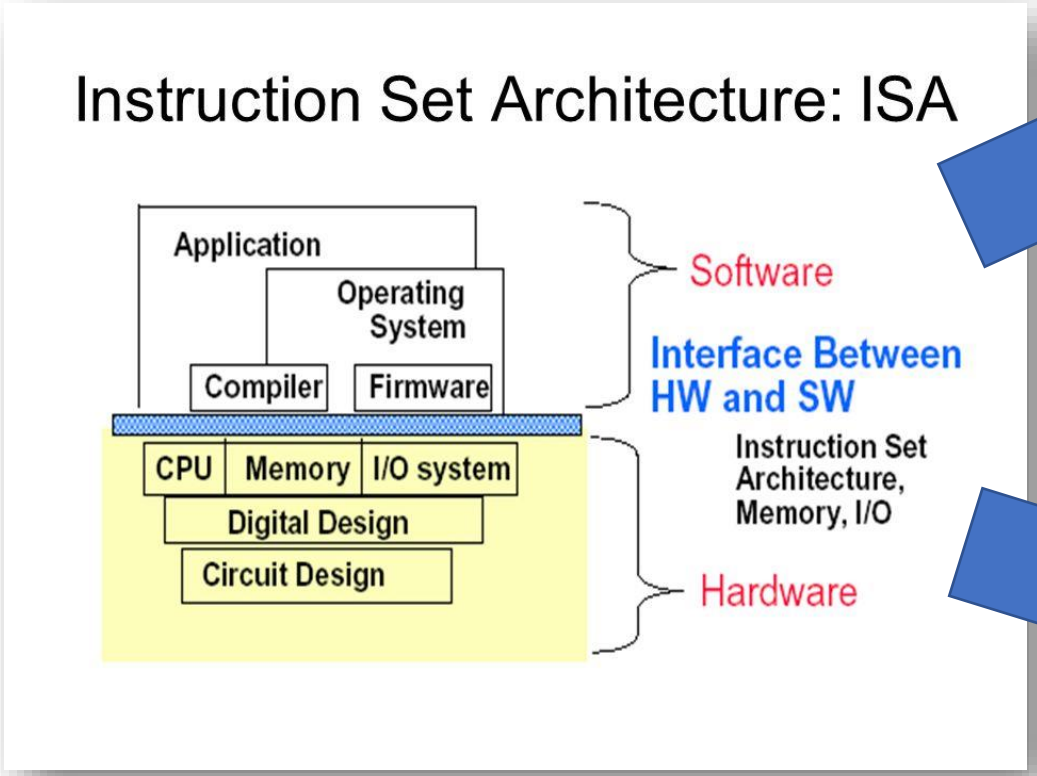
HARDWARE – Demais componentes da CPU

Número de ciclos por segundo

- Esta medida é confundida com o número de instruções que o processador realiza por segundo.
- Cada instrução é realizada em um número específico de ciclos.
- Existem instruções que são realizadas em um único ciclo de relógio (no limite é o que se deseja), enquanto outras demoram várias dezenas.
- Exemplos:



Microprocessador (Arquitetura)



JOGO DE INSTRUÇÕES:

- Jogo de Instruções (códigos)
- Quantidade de bits usados para representar as instruções
- Os operadores
- Tipos de operadores
- Técnicas de endereçamento dos comandos
- Interface de Entrada e Saída

Essencialmente distinguem-se cinco tipos de instruções na **linguagem da máquina**:

- transferência de operandos entre a memória e os registradores da CPU,
- operações lógico-aritméticas,
- operações de controle de fluxo de execução,
- operações do co-processador, caso exista, e
- operações de controle do sistema.

CISC x RISC x EPIC

Além dos tipos de intrução... Outro fator → complexidade conjunto instruções

- **CISC (Complex Instruction Set Computer) :** Dominou a dédada de 80. Várias centenas de instruções (simples e complexas) → vários ciclos de clock // Baseadas em microinstruções.
  - Grande gama de instruções→ tamanho de instruções variados

- **RISC (Reduced Instruction Set Computer):** Conjunto pequeno de instruções com poucos modos de endereçamento. Instruções lógico-aritméticas são as mesmas (tamanho). Tempo: 1 ciclo de clock
  - Operandos das instruções são previamente carregados para os registradores. Os acessos de leitura e escrita → mesmos para todas as transferências de dados.
  - Arquitetura ARM (Advanced Risc Machine) → 32 bits -> arq. RISC mais popular
  - Núcleos Krait e Kryo da Qualcomm (Microcontroladores Snapdragon) → RISC da família ARM
  - Não existem microinstruções → As instruções reduzidas já são as microinstruções.

- **EPIC (Explicitly Parallel Instruction Computing):** Resposta ao limite de desempenho da arq RISC. A ideia é codificar em uma palavra de instrução muito longa várias operações (paralelismo a nível de instrução)

CISC x RISC (Video 01)

CISC x RISC (EXEMPLO)

CISC

```
mov ax, 10
mov bx, 5
mul bx, ax
```

RISC

```
        mov ax, 0
        mov bx, 10
        mov cx, 5
Inicio: add ax, bx
        loop Inicio
```

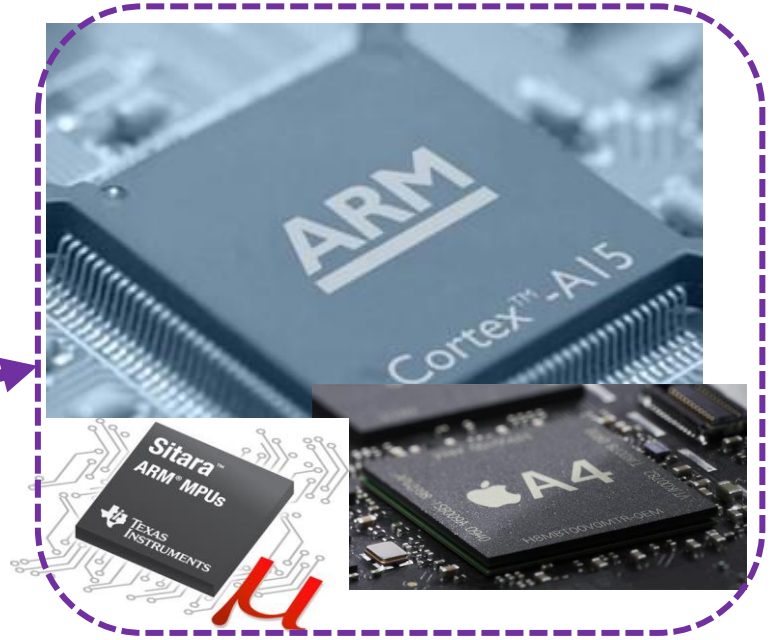
$$\text{Tempo de CPU} = \frac{\text{segundos}}{\text{programa}} = \frac{\text{instruções}}{\text{programa}} \times \frac{\text{ciclos}}{\text{instrução}} \times \frac{\text{segundos}}{\text{ciclo}}$$

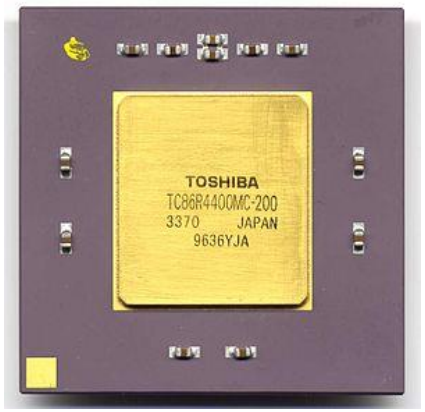
CISC: (2 movs x 1 ciclo) + (1 mul x 30 ciclos) = **32 ciclos**

RISC: (3 movs x 1 ciclo) + (5 adds x 1 ciclo) + (5 loops x 1 ciclo) = **13 ciclos**

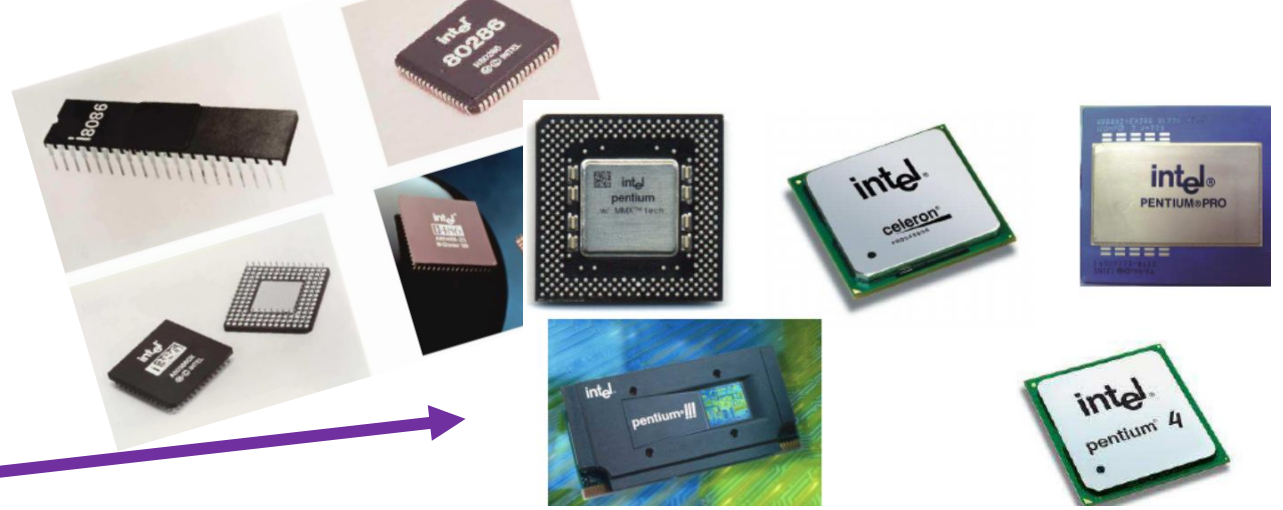
Exemplos CISC e RISC

- ARM – domina o mercado de baixa potência – 90% do mercado de dispositivos móveis
- PowerPC (roteadores e switches) e equipamentos de armazenamento
- Linha MIPS (playStation e Nintendo64)
- SPARC (Oracle)
- Etc.



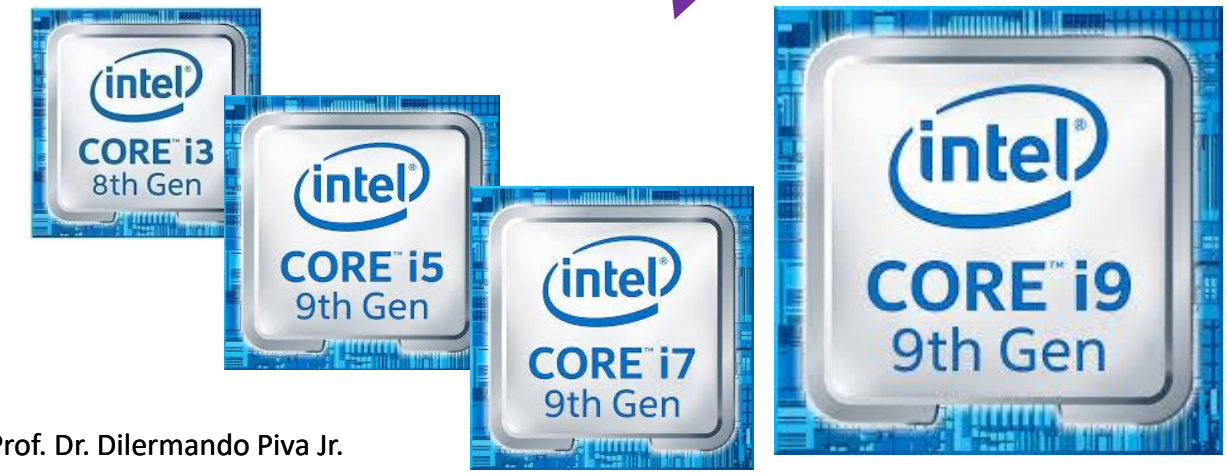


Exemplos CISC e RISC



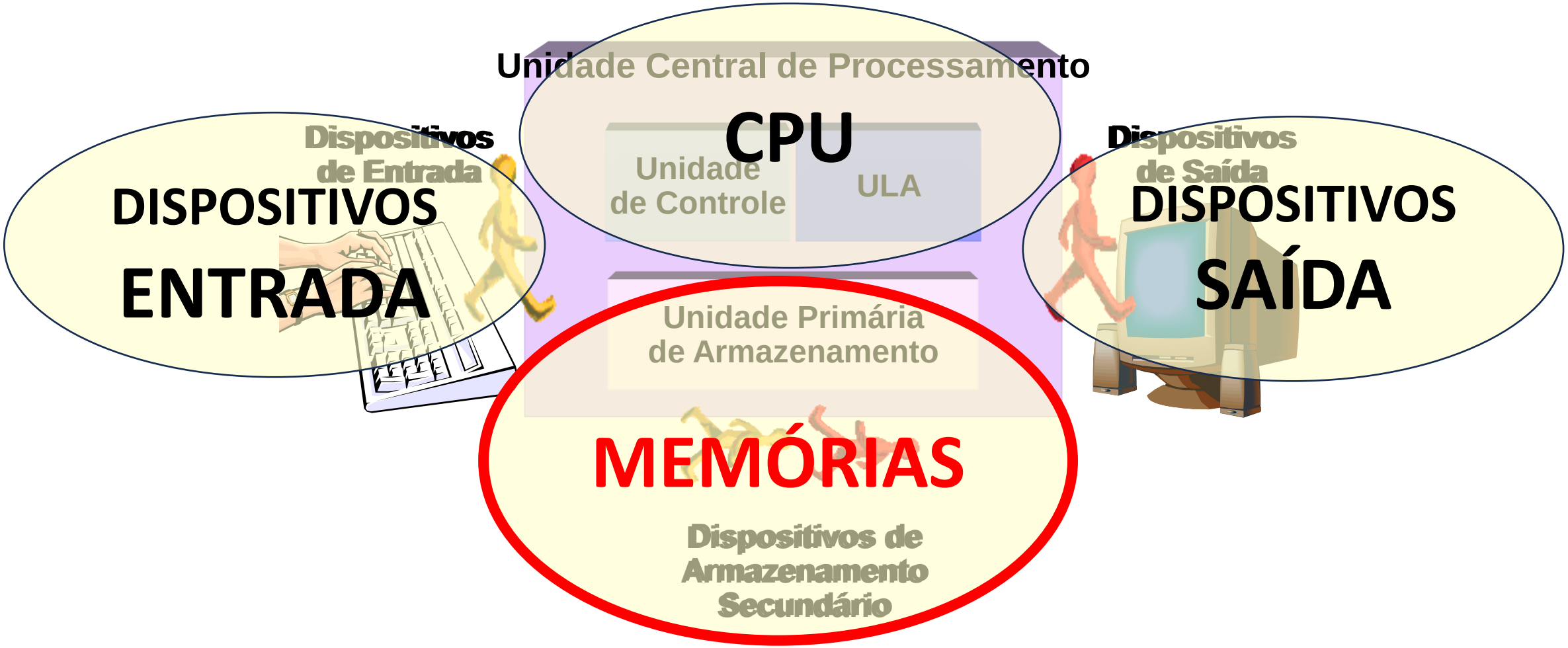
- Intel - 8088, 80386... Pentium...
- Motorola 68030/68040...

Processadores Mais ATUAIS... Seguem um mix entre CISC e RISC
Exemplo: x86 (i3, 5, 7, 9...)

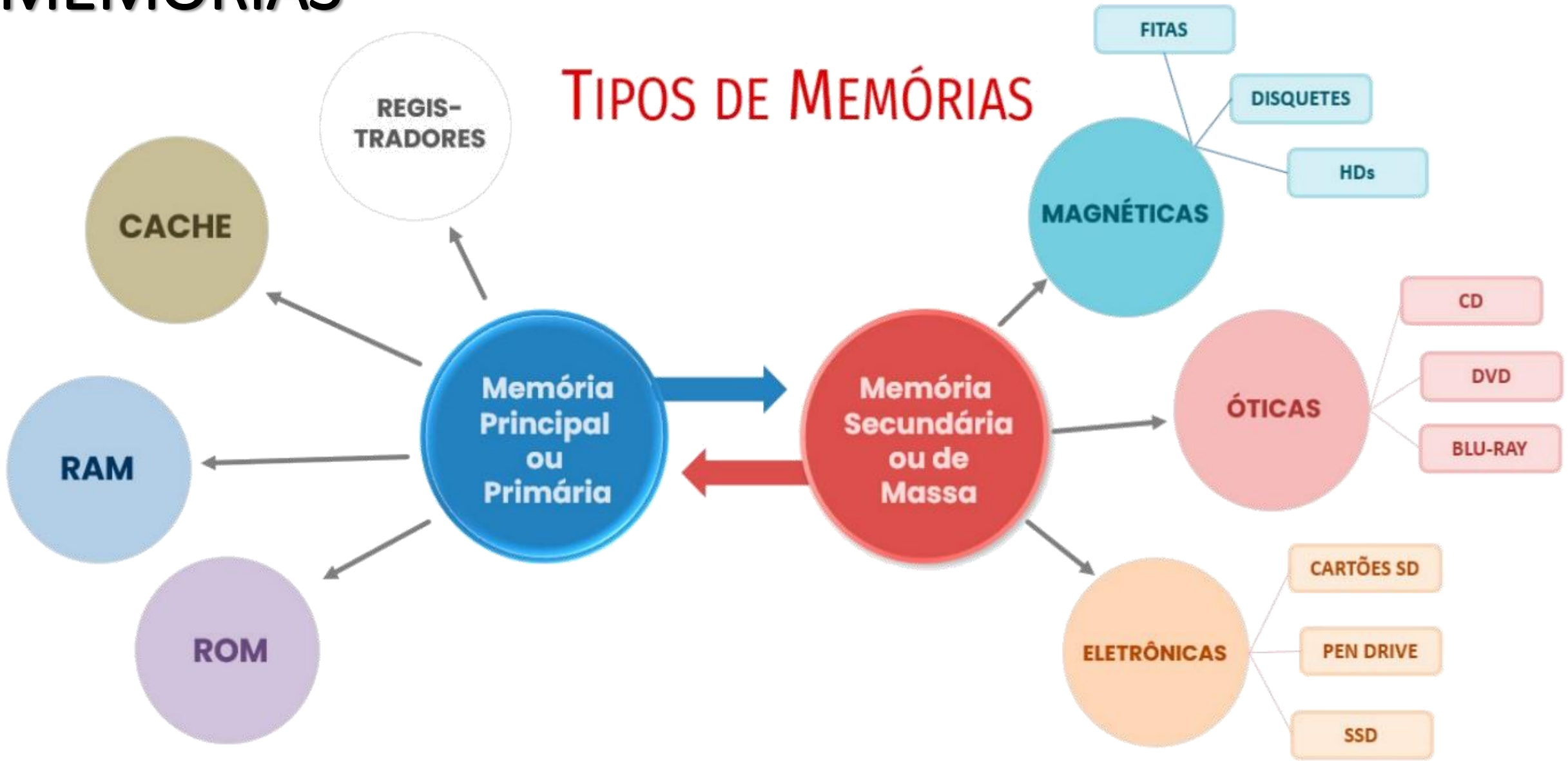


HARDWARE – MEMÓRIAS

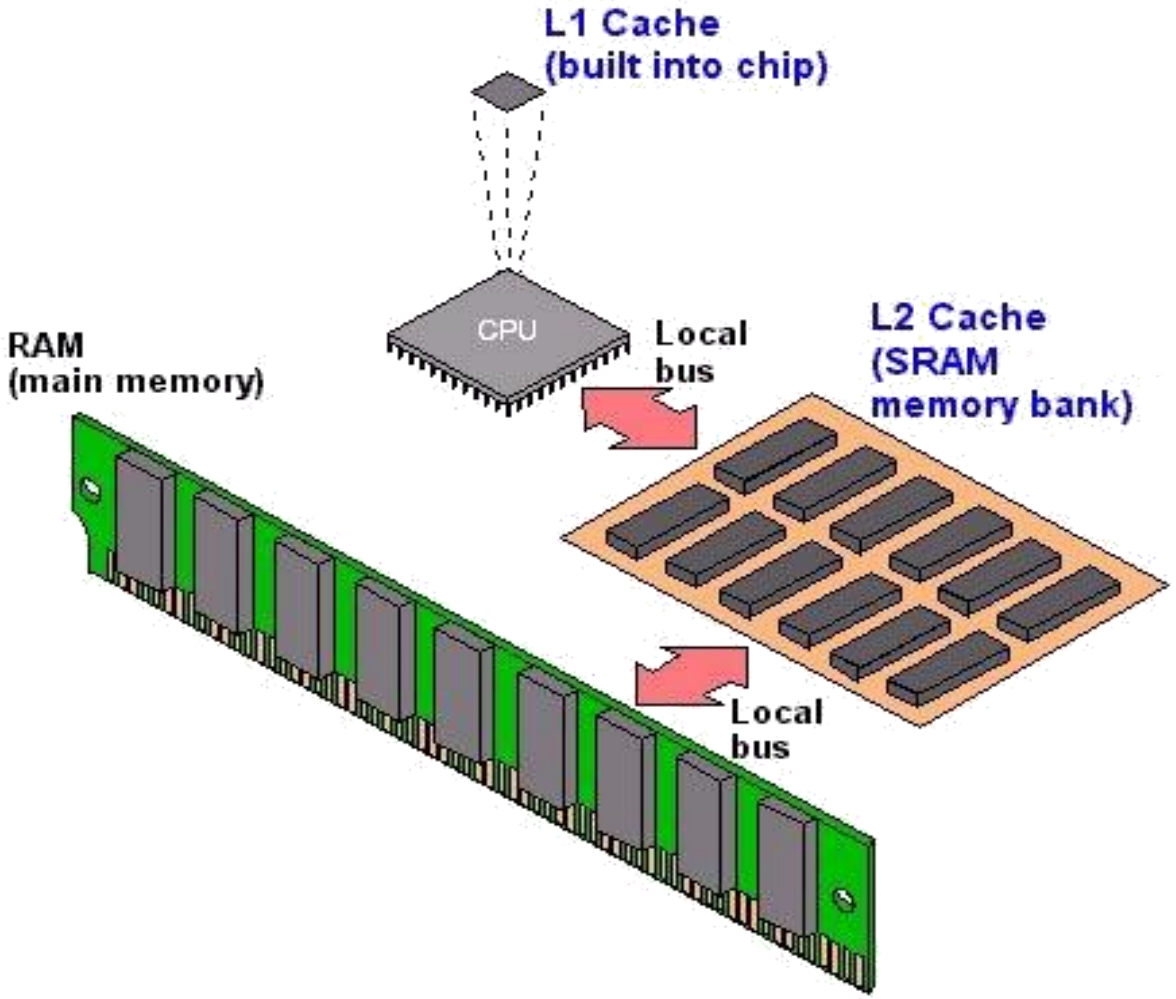
Computadores atuais

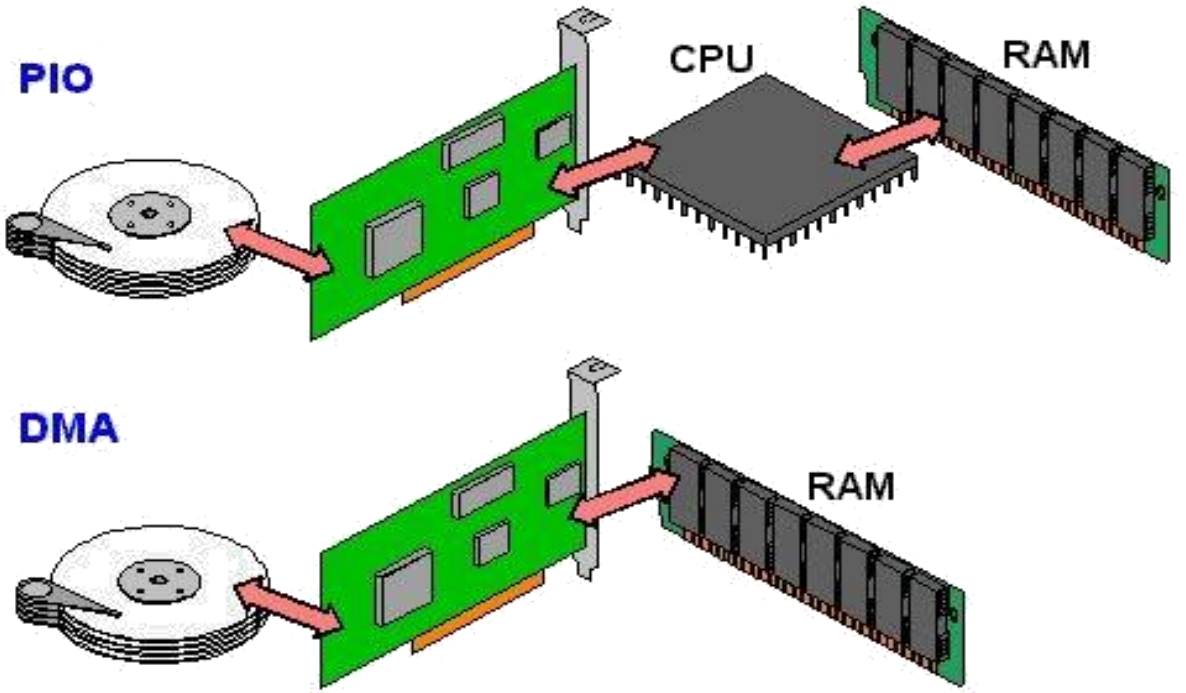


MEMÓRIAS

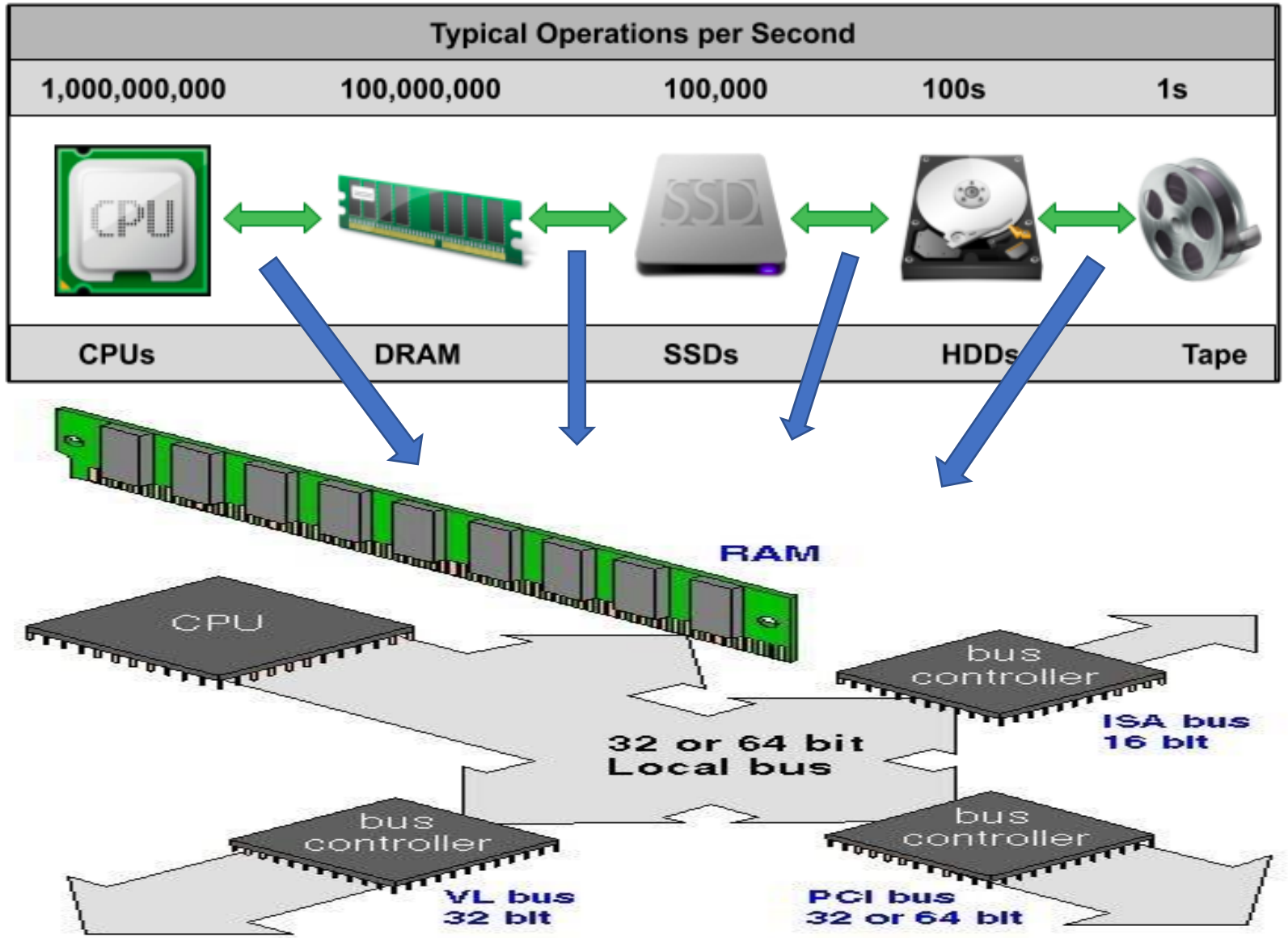


MEMÓRIAS



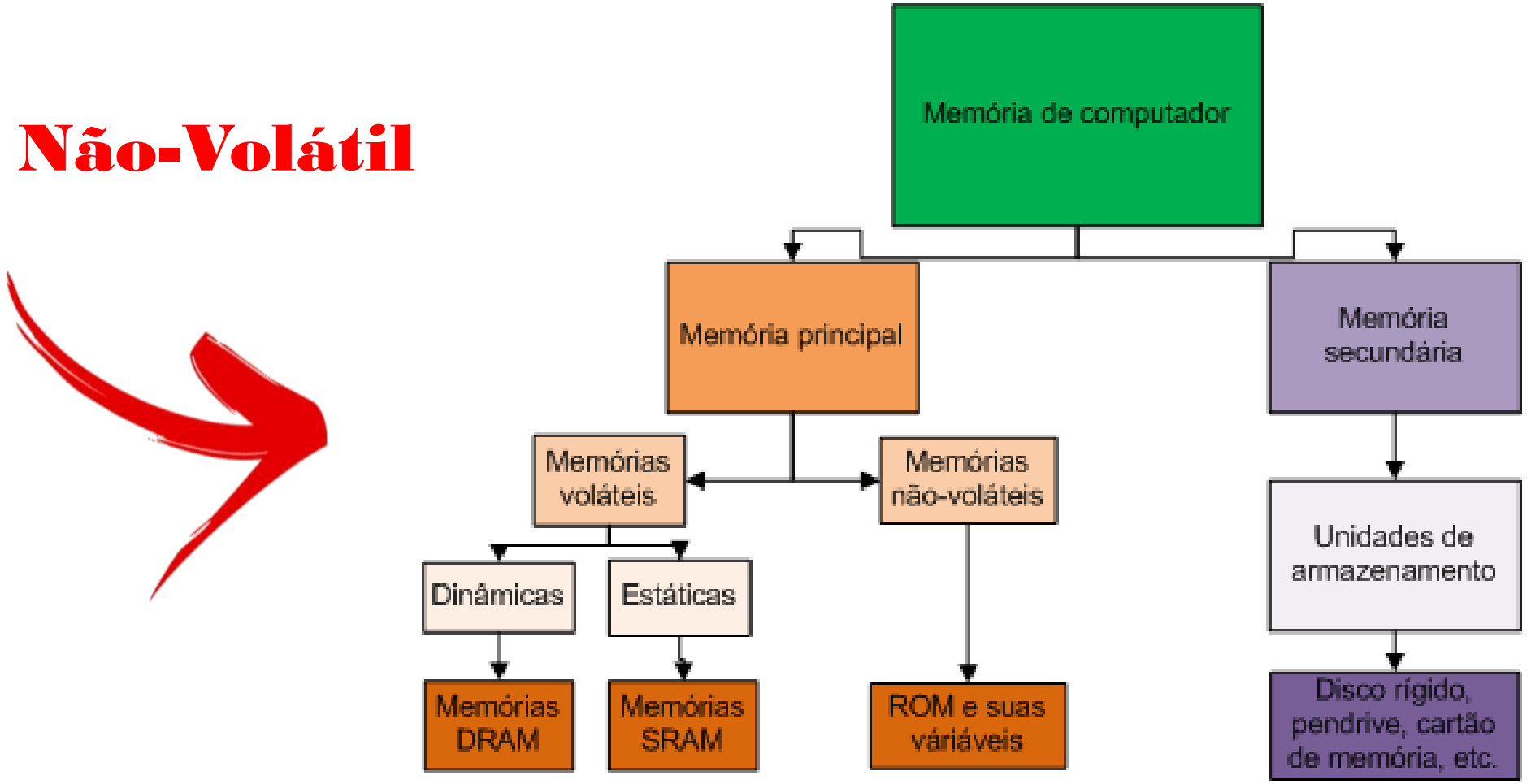


MEMÓRIAS x OPERAÇÕES p/ segundo



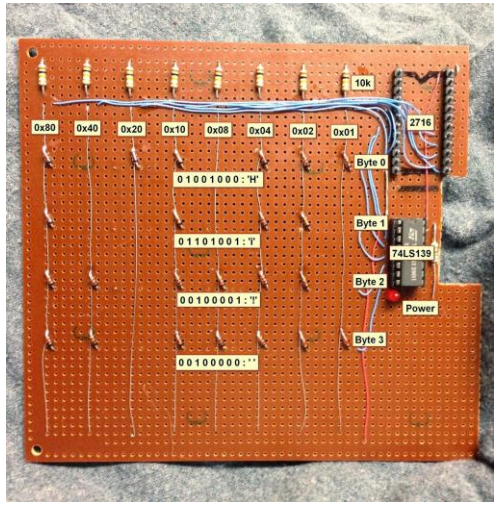
MEMÓRIAS (Tecnologias)

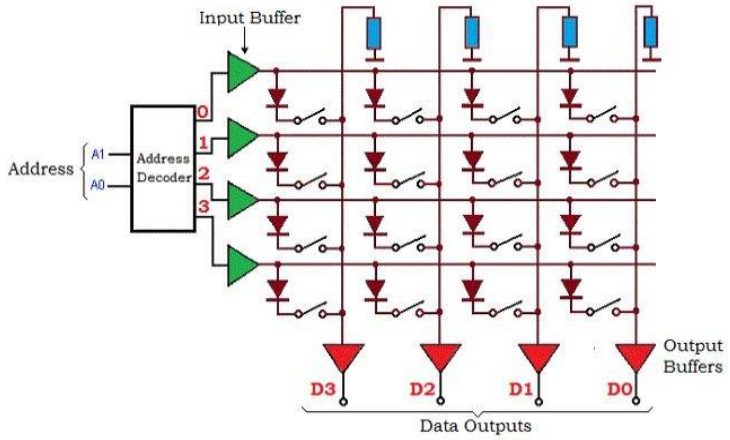
Volátil x Não-Volátil



MEMÓRIAS (Tecnologias) → Não-Volátil

ROM – Read Only Memory





Memória ROM como uma matriz de diodos.

Com o surgimento dos circuitos integrados...

Memórias somente de leitura de máscara (MROM)

Baseadas em conexões na lógica OU → or-wired

As MROMs são mais compactas, baratas e rápidas. Hoje em dia, elas são usadas para armazenar sistemas operacionais, fontes das impressoras, dados de áudio de instrumentos musicais, enfim dados para os quais a probabilidade de modificação seja ínfima

MEMÓRIAS (Tecnologias) → Não-Volátil

1956

PROM – Programmable
Read Only Memory

1k x 8 bits

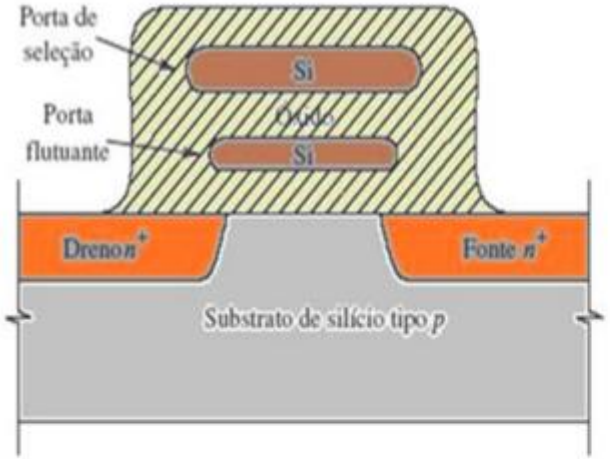
PROM e um Gravador de PROM

As memórias vinham "zeradas" de fábrica... **E com o gravador, era possível gravar uma informação**

MEMÓRIAS (Tecnologias) → Não-Volátil

1971

EPROM – Eresable Programmable Read Only Memory



(a) Célula EPROM

(b) EPROM com janela de quartzo

(c) gravador

(d) apagador

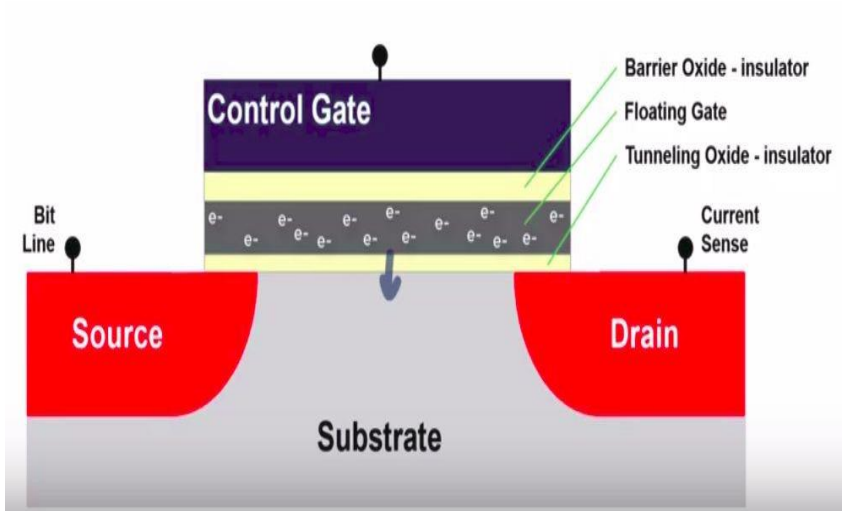
Existiam os EPROM, conhecidas por "programável por uma vez" - One time Programmable (OTP)

Não tinha Janelinha!

MEMÓRIAS (Tecnologias) → Não-Volátil

1972

EEPROM – Electrically Eresable and Programmable Read Only Memory



Limpeza por tunelamento de Fowler-Nordheim

EEPROM Encapsulada

Hoje em dia, EEPROM são amplamente aplicadas no armazenamento de instruções de tarefas dedicadas e reprogramáveis, como a BIOS e os dados de calibração de equipamentos de teste.

A grande limitação da memória EEPROM é o tempo e a quantidade de *bytes* (alguns *bytes*) numa limpeza ou numa reprogramação.

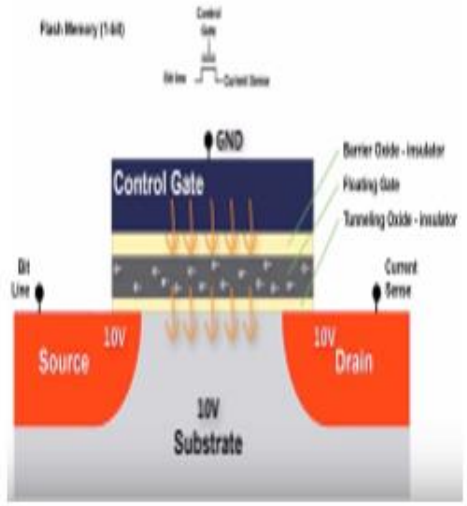
A sua vida útil é tipicamente um milhão de reprogramações.

MEMÓRIAS (Tecnologias) → Não-Volátil

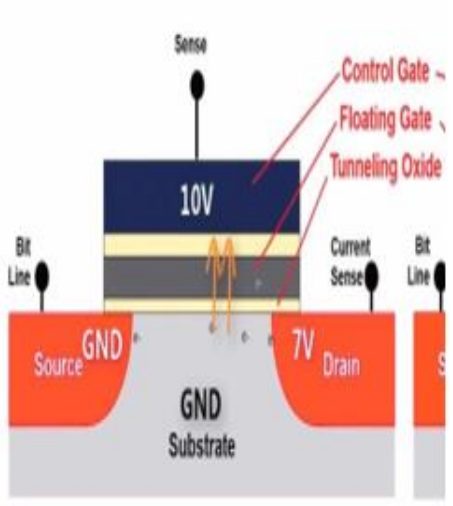
1980

EEPROM – Electrically Eresable and Programmable Read Only Memory

Memórias Flash



(a) Limpeza por bloco (Fonte: [38])



(b) Injeção de elétrons quentes (Fonte: [39])

(c) FLASH encapsulada

Duas principais organizações das suas células são as que se assemelham com a organização da porta lógica NOR (FLASH NOR) e as com a organização da porta lógica NAND (FLASH NAND)

Nas memórias *FLASH NOR* a linha de carga, em inglês *source line*, é conectada em cada célula, enquanto nas memórias NAND, esta linha de carga é conectada com até 8 células em série.

Portanto, enquanto as memórias FLASH NOR permitem acessos por *bytes* ou palavras, as memórias FLASH NAND fazem acessos por páginas.

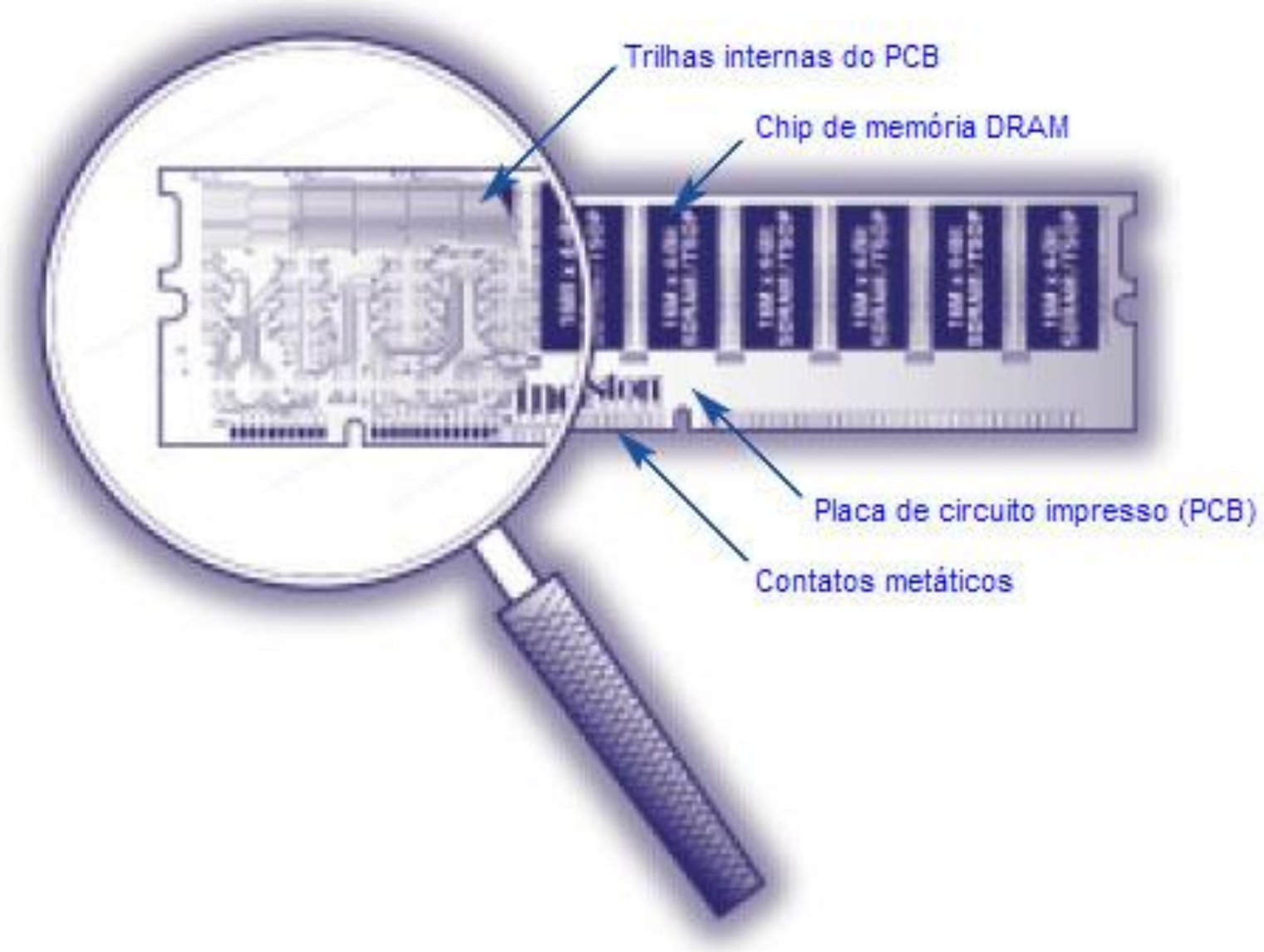
FLASH NOR → MAIOR CUSTO ; + GARANTIA ; + LENTA

FLASH NAND → MAIS BARATO, MAIOR % BLOCOS DEFEITUOSOS, + RÁPIDA

MEMÓRIAS (Tecnologias) → Voláteis

RAM – Random Access Memory

SRAM **DRAM**



MEMÓRIAS (Tecnologias) → Voláteis

RAM – Random Access Memory

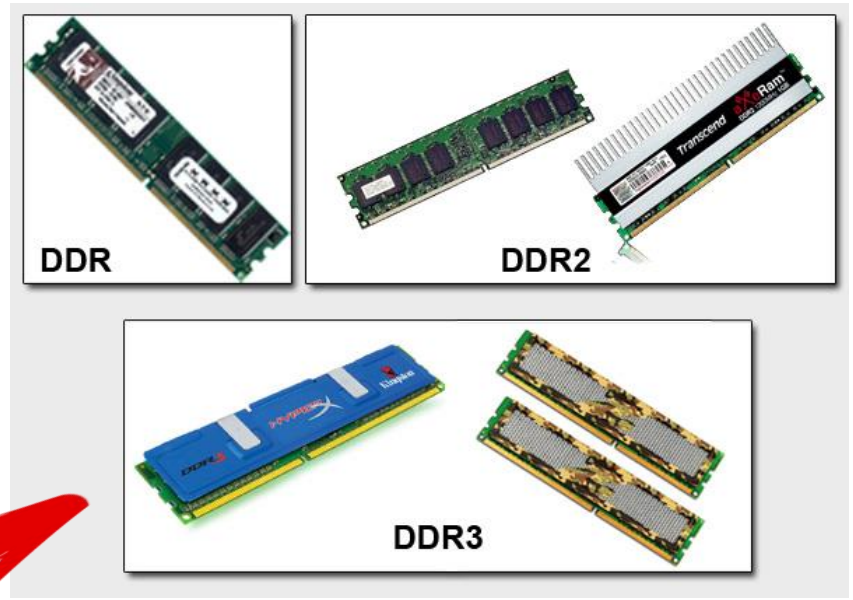
SRAM – Static RAM

Cada célula formada por 6 transistores integrados
Não precisa de “reflesh”
Mais cara e ocupa um maior espaço (físico)
Mais rápida!!

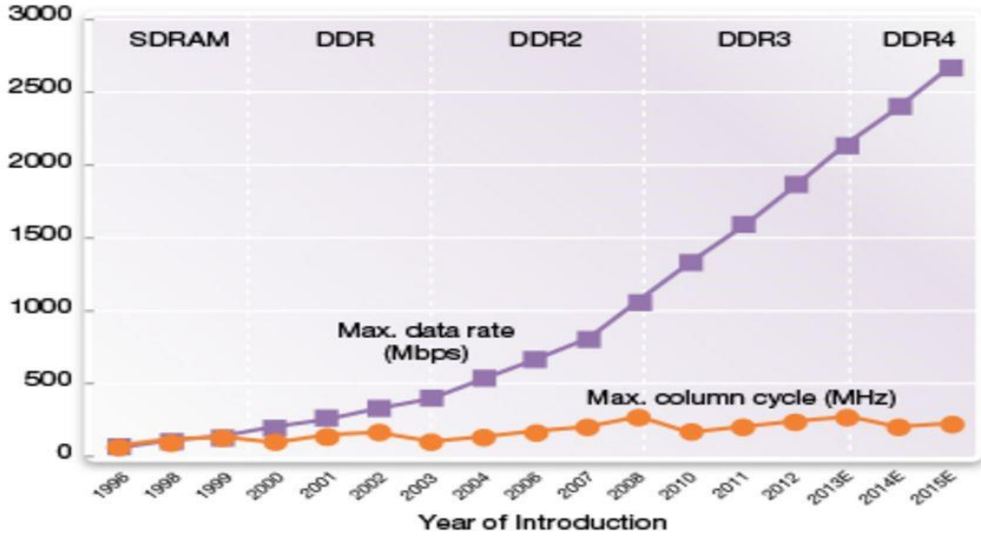
MEMÓRIAS (Tecnologias) → Voláteis

RAM – Random Access Memory

DRAM – Dinamic RAM



Cada célula formada por 1 capacitor e 1 transistor
Precisa de "reflesh" (2ms a 128 ms)
Mais barata e ocupa um menor espaço (físico)
Mais lenta (quando comparada a uma SRAM

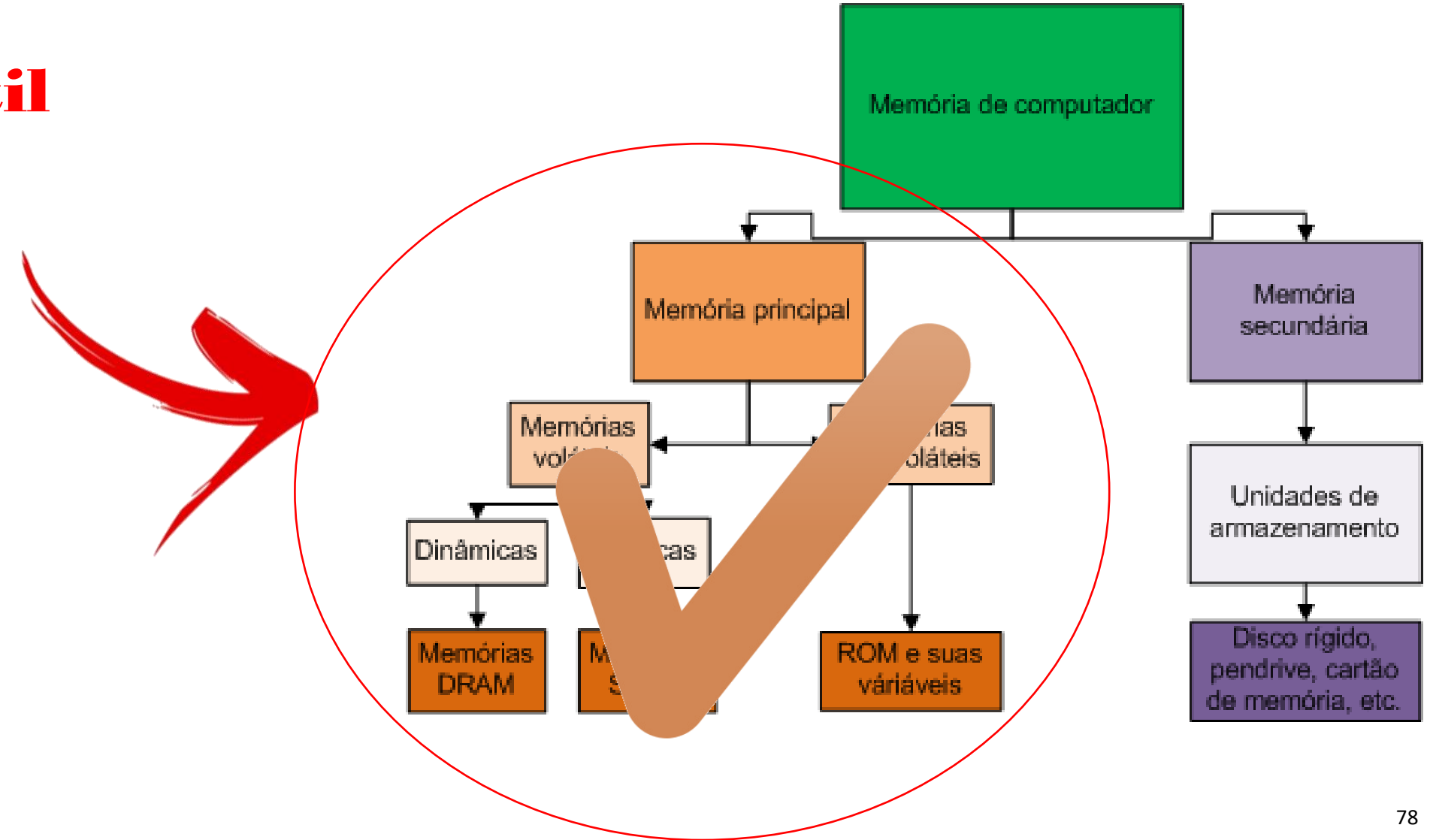


SIMM (*Single In-Line Memory Module*) ou
DIMM (*Dual In-Line Memory Module*).

A memória RAM

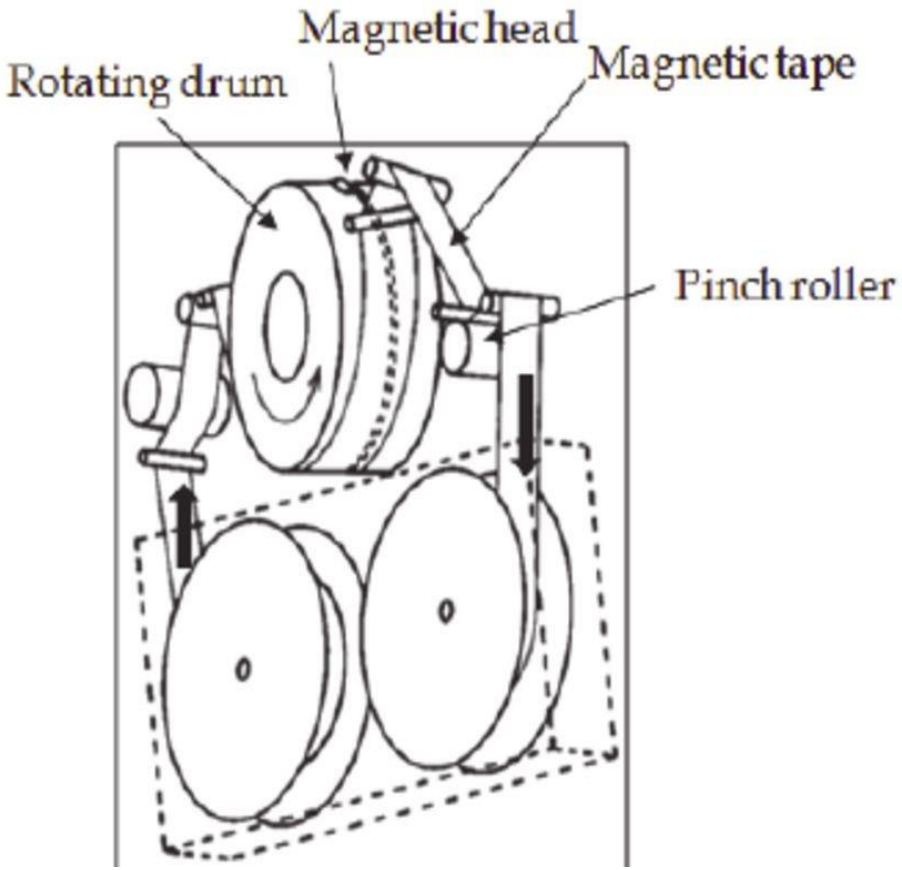
MEMÓRIAS SECUNDÁRIAS

Não-Volátil



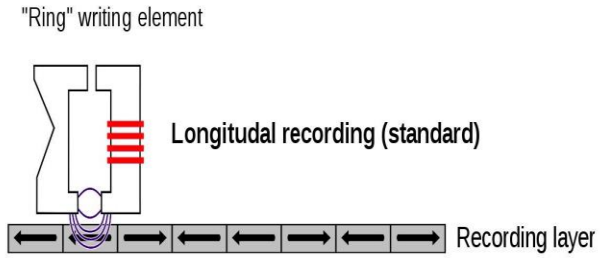
MEMÓRIAS SECUNDÁRIAS

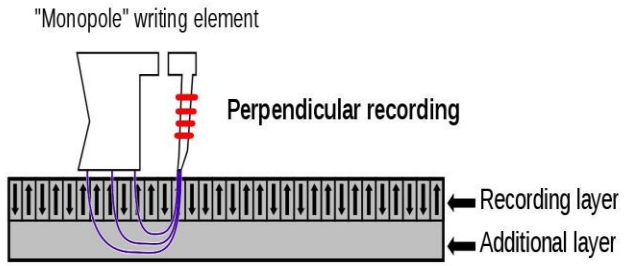
Memórias de Armazenamento de Massa – não-volátil

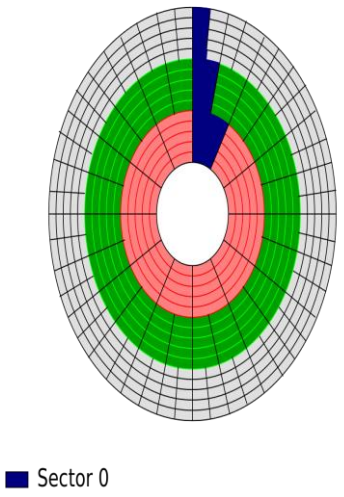


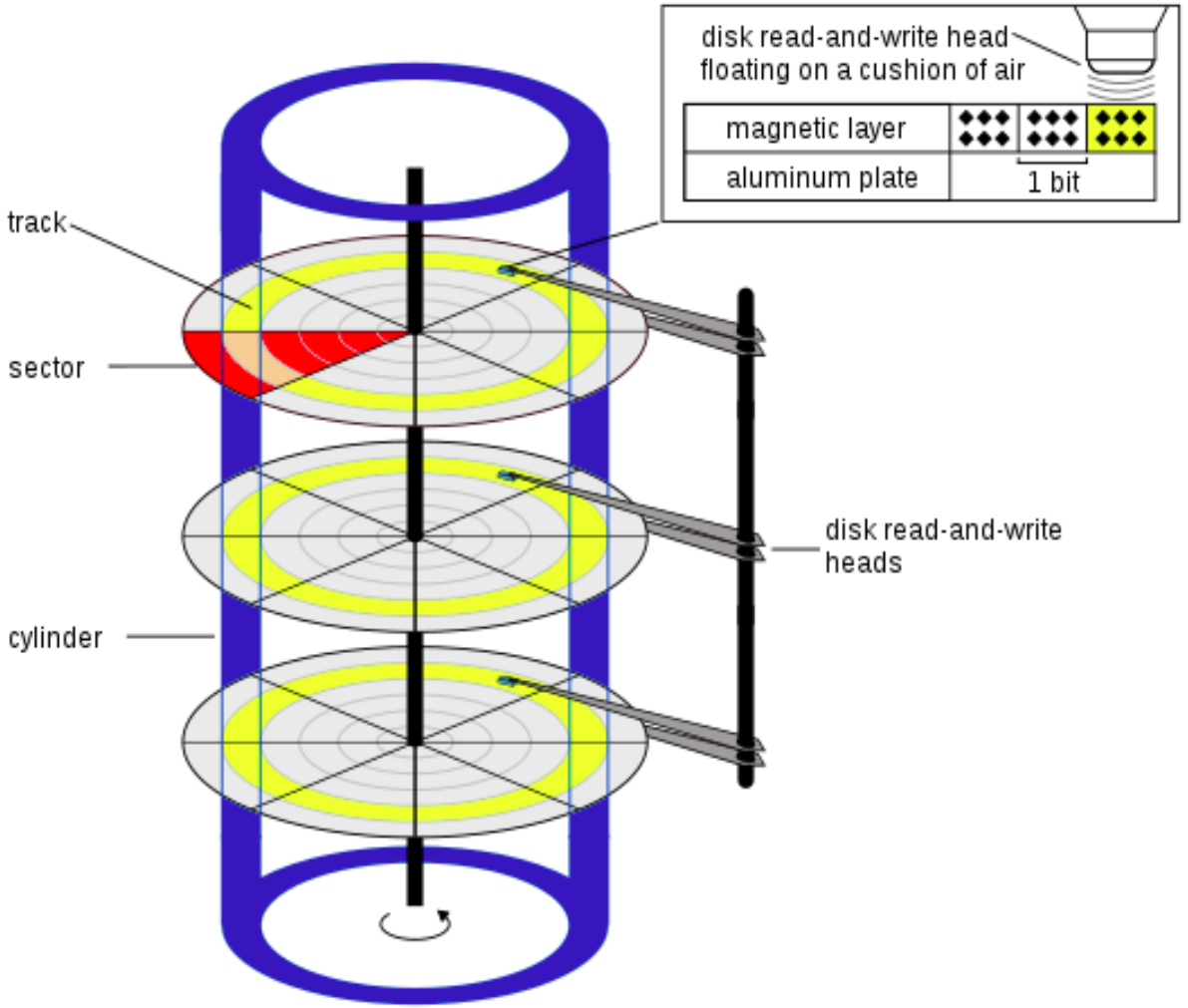
MEMÓRIAS SECUNDÁRIAS – Disco Rígido (HD)

Memórias de Armazenamento de Massa – não-volátil



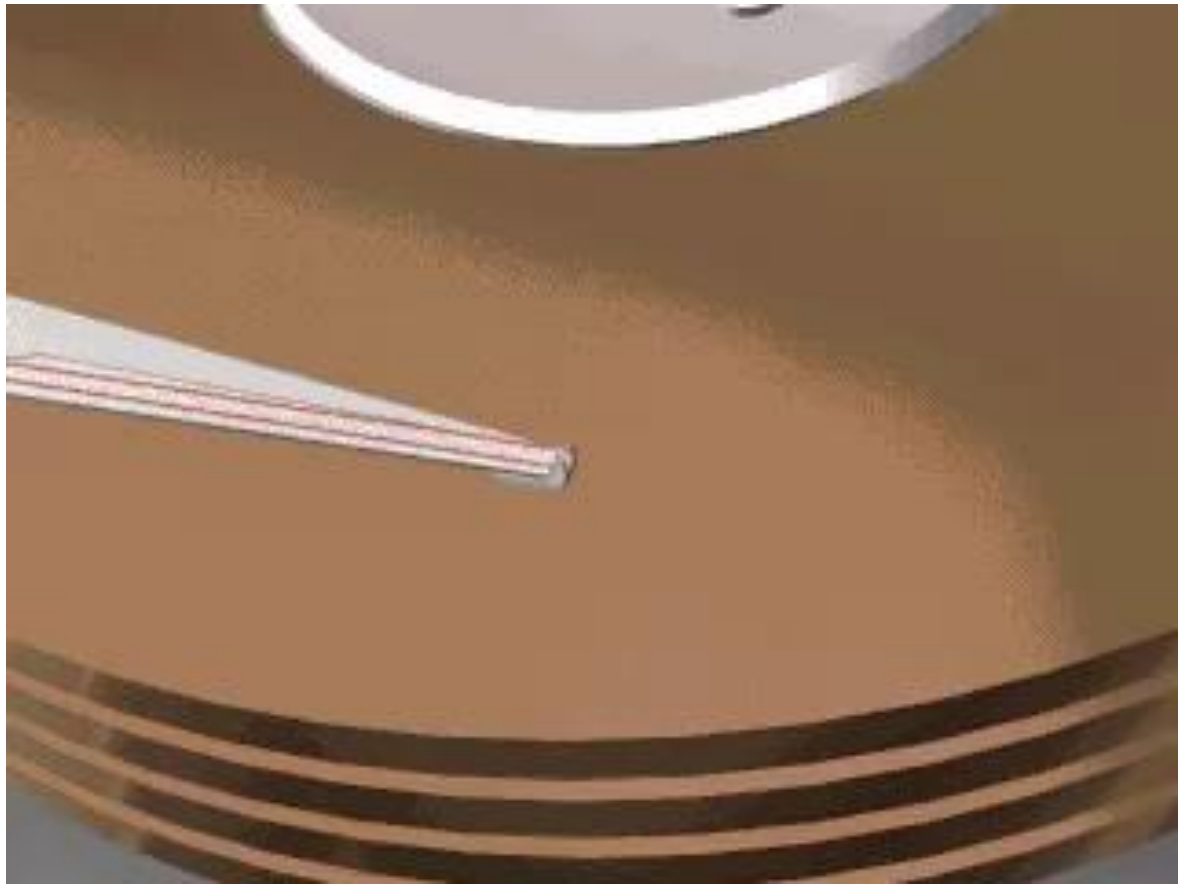






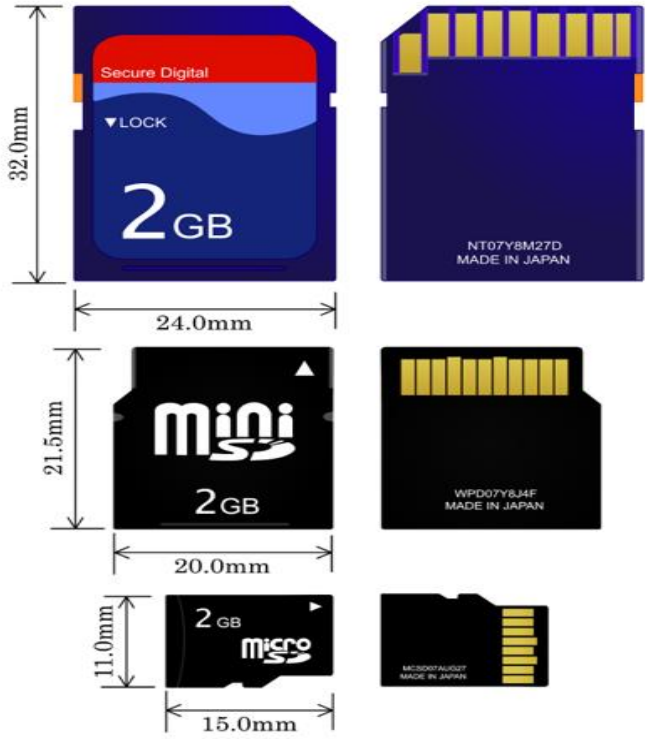
MEMÓRIAS SECUNDÁRIAS – Disco Rígido (HD)

Memórias de Armazenamento de Massa – não-volátil

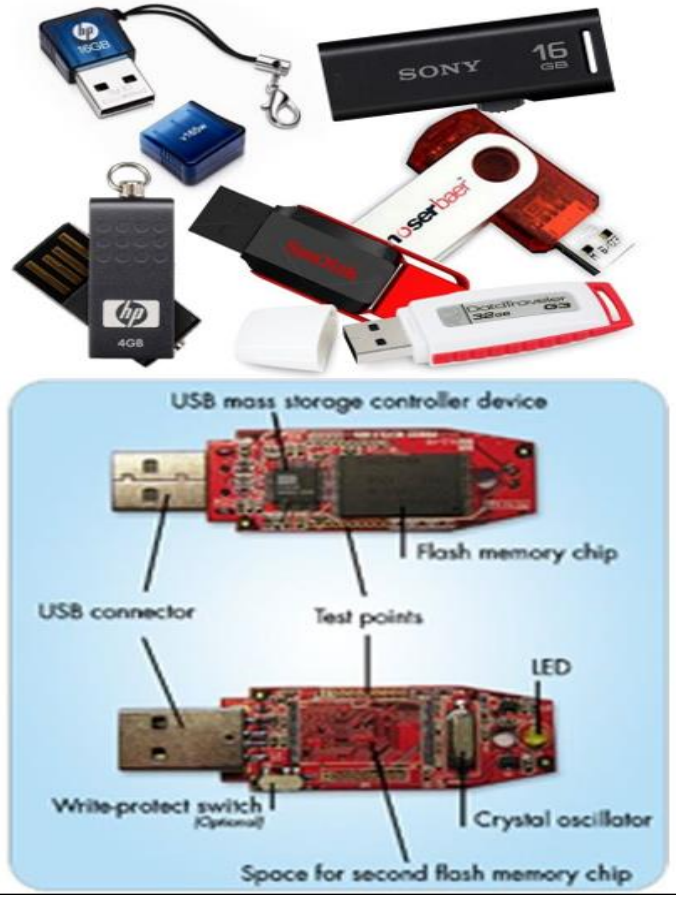


MEMÓRIAS SECUNDÁRIAS – FLASH NAND

Memórias de Armazenamento de Massa – não-volátil
Pendrives, flashdrives e SSD (state storage drive)



Cartões de Memória

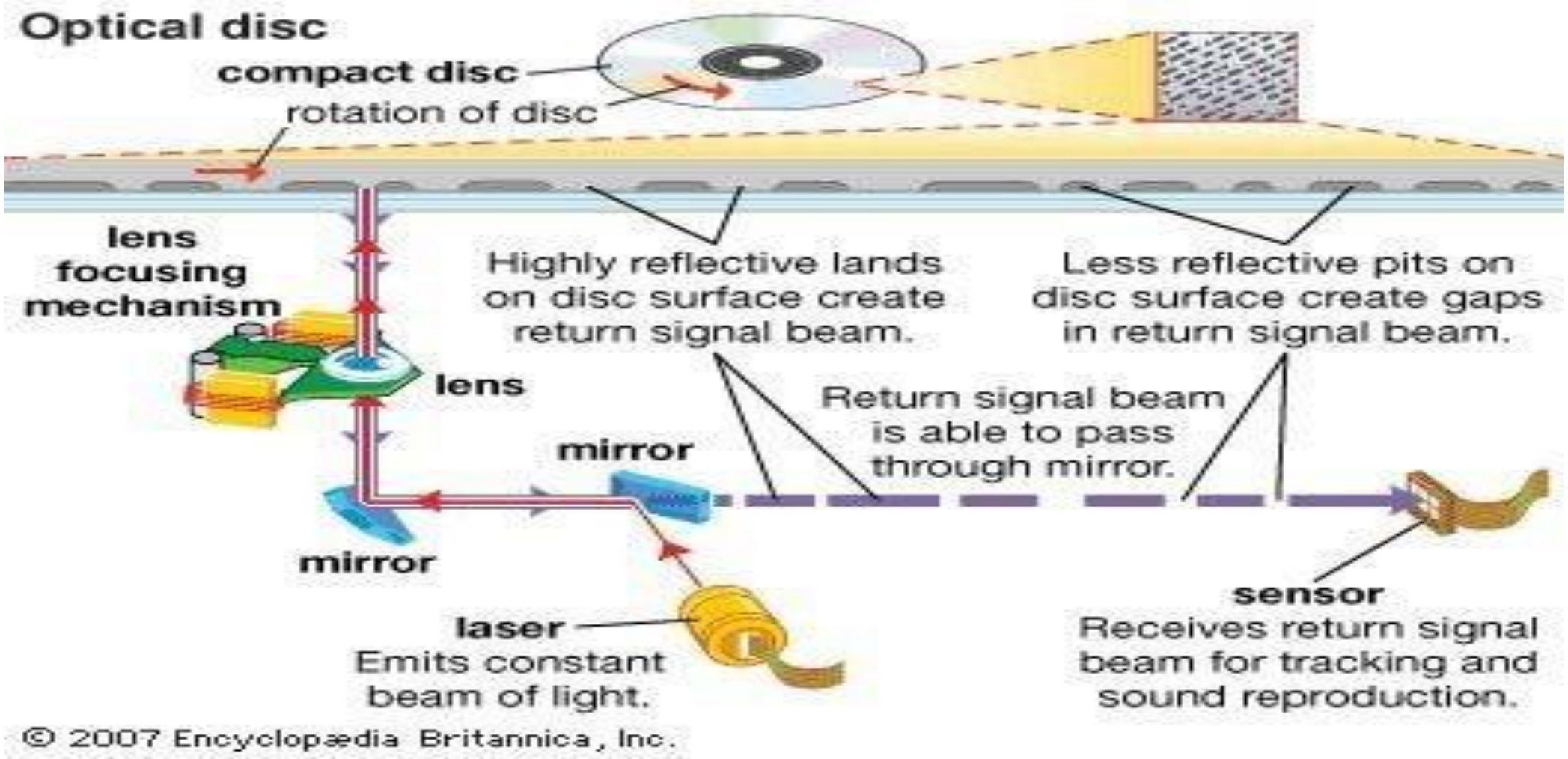


Pendrives

SSDs

MEMÓRIAS SECUNDÁRIAS – Discos Ópticos

Memórias de Armazenamento de Massa – não-volátil



Resumo do funcionamento da memória...

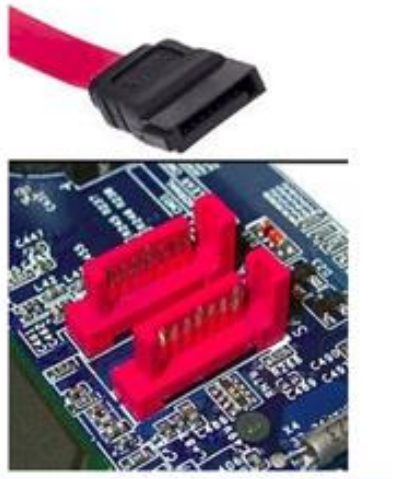
INTERFACES DAS MEMÓRIAS SECUNDÁRIAS

IDE (Integrated Drive Eletronics) / **ATA** (Advanced Technology Attachment

- **Foi o primeiro padrão de interface dos HDs, em que o controlador e o disco são integrados.**
- **Os primeiros HDs com interface IDE foram lançados por volta de 1986.**
- **Permite conectar a uma mesma interface dois discos no esquema mestre/escravo.**
- **Os cabos têm 40 fios, 16 para enviar e receber dados em paralelo, 7 terra, e 1 para seleção do modo mestre/escravo.**

INTERFACES DAS MEMÓRIAS SECUNDÁRIAS

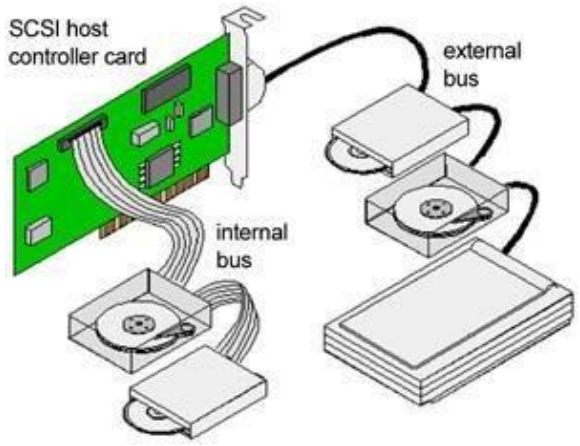
SATA (Serial AT Attachment) – AT - Advanced Technology



- **É o sucessor da tecnologia ATA, substituindo a interface paralela pela serial.**
- **Os cabos são formados por dois pares de fios (um para transmissão e outro para recepção) e mais três fios de terra, totalizando 7 fios.**
- **Usando transmissão diferencial, reduziu interferências eletromagnéticas entre os fios. Isso permitiu aumentar a velocidade de transferência, reduzir o diâmetro dos fios e melhorar a ventilação do gabinete.**
- **Reduzindo os fios, viabilizou a troca do dispositivo enquanto ligado, em inglês *hot-swap*.**
- **Há diferentes gerações da interface SATA. Elas diferem essencialmente em velocidade de transferência.**

INTERFACES DAS MEMÓRIAS SECUNDÁRIAS

SCSI (Small Computer System Interface)



- **Foi padronizada em 1986.**
- **Embora seja mais comumente encontrada como interface de discos rígidos e unidades de fita, foi originalmente concebida para conectar paralelamente uma larga gama de periféricos.**
- **Ele permite conexões de até 15 dispositivos, identificados por um código binário, chamado ID SCSI.**
- **As transmissões ocorrem, porém, sempre entre dois dispositivos de cada vez.**
- **Diferentes versões de SCSI diferem na quantidade de fios de dados e na velocidade de transmissão.**

INTERFACES DAS MEMÓRIAS SECUNDÁRIAS

SAS (Serial Attached SCSI)

- **É a versão serial de SCSI, com vantagens de maior velocidade na transferência de dados, *hot-swapping* e melhor controle dos erros na transmissão.**

INTERFACES DAS MEMÓRIAS SECUNDÁRIAS

FC (Fiber Channel)

- **É o sucessor serial de SCSI em mercados corporativos. Foi aprovado pela ANSI em 1994.**
- **Adotando fibras ópticas como cabos de conexão, é um protocolo sem perda e de altíssima taxa de transferência (na ordem de *gigabits* por segundo).**
- **Portanto, é tipicamente usado para conexão dos discos de *storage* com os servidores nas redes de áreas de storage, em inglês *storage area networks* (SAN) em centros de dados comerciais.**

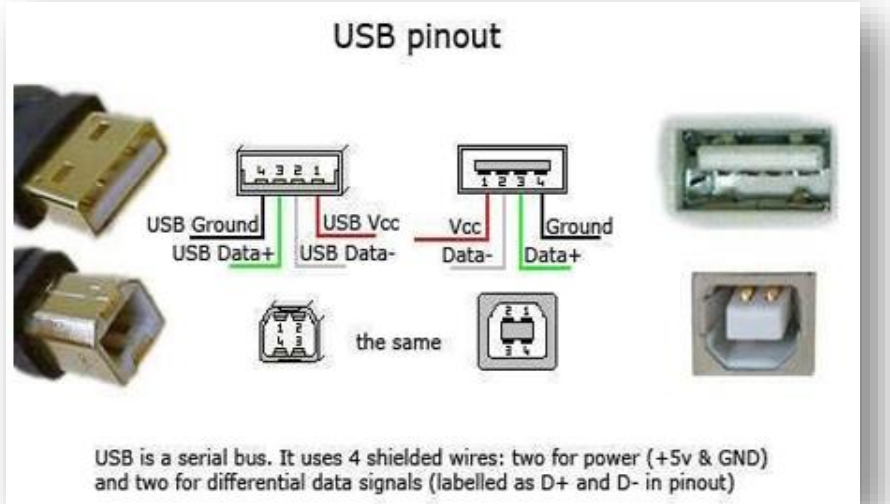
INTERFACES DAS MEMÓRIAS SECUNDÁRIAS

USB (Universal Serial Board)

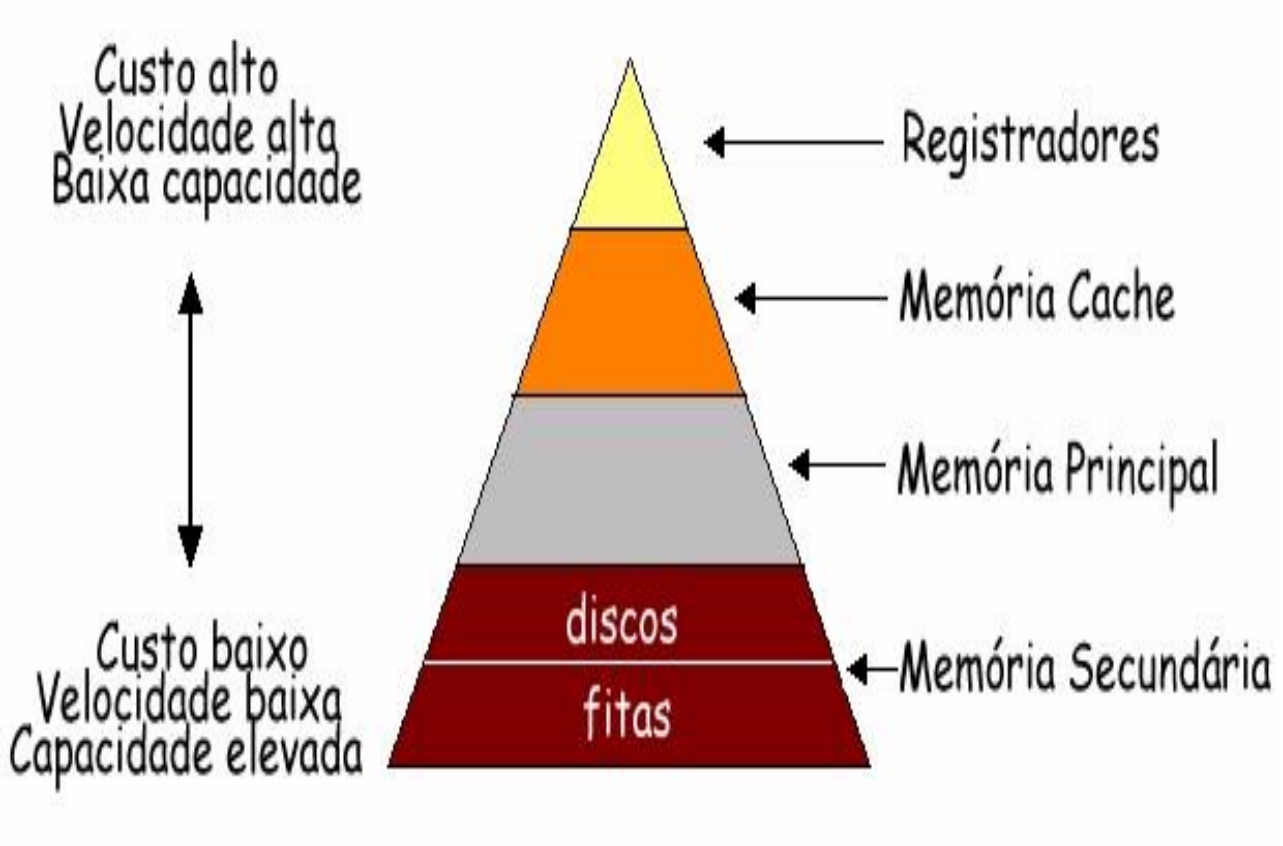
(a) 1.0, tipo A e tipo B

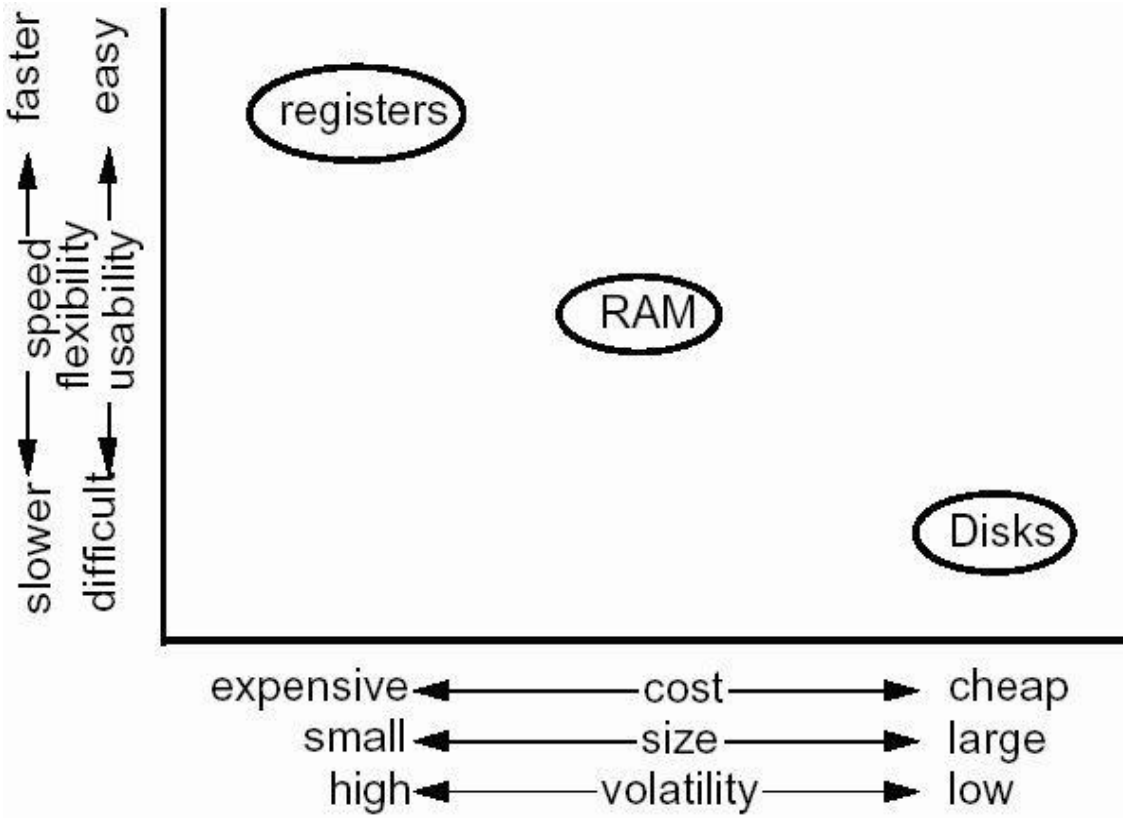
(b) 2.0, tipo A

- **Uma conexão USB é formada por 4 fios: 2 de alimentação e 1 par diferencial para dados.**
- **A alimentação típica da USB é 5V em corrente contínua, e a corrente acionável pelos pinos varia de 0,5A (USB 1.0) até 0.9A (USB 3.0).**

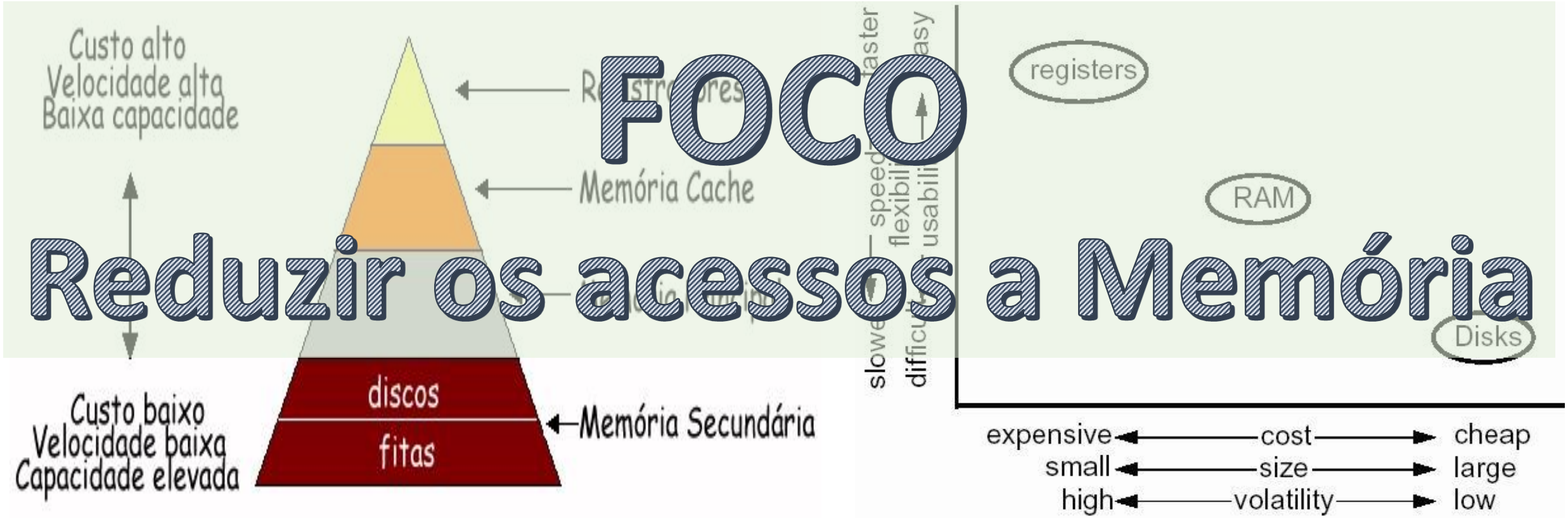


MEMÓRIAS - Hierarquia

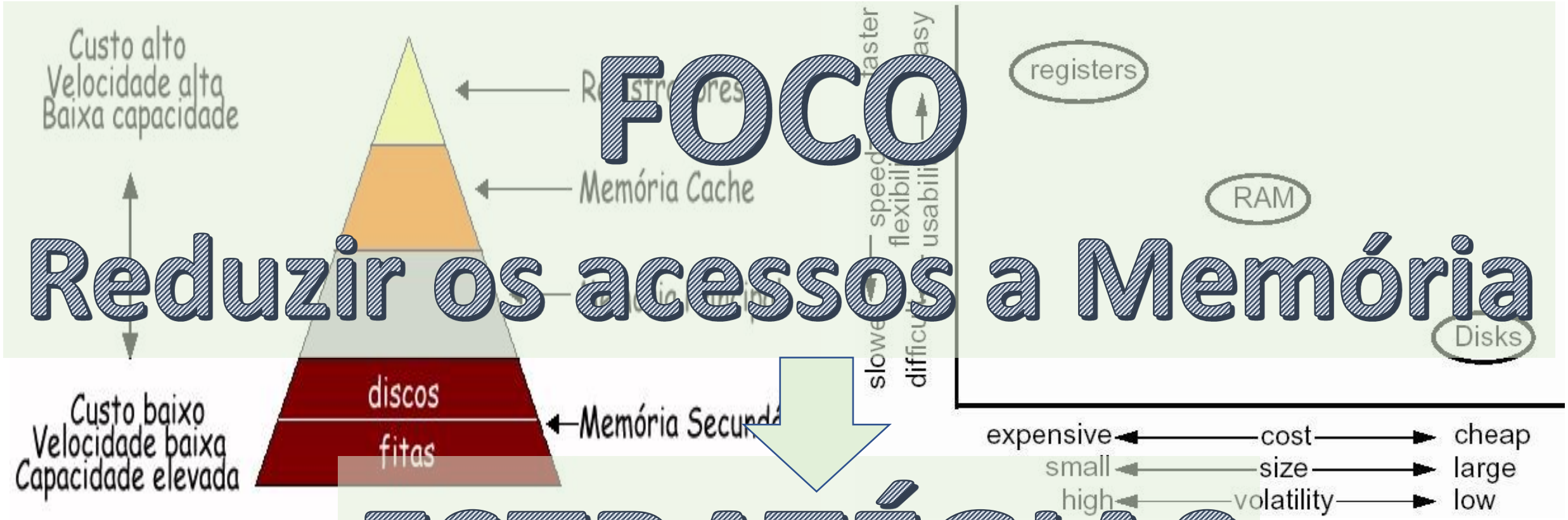




MEMÓRIAS - Hierarquia



MEMÓRIAS - Hierarquia

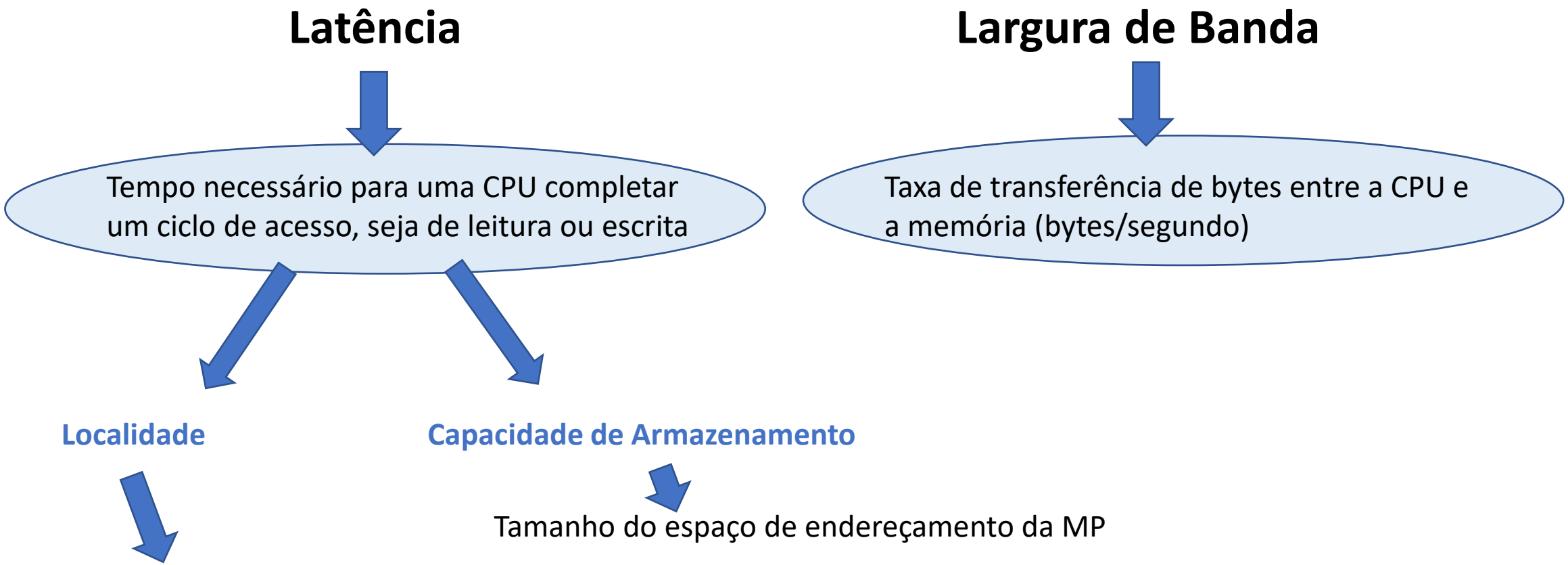


FOCO

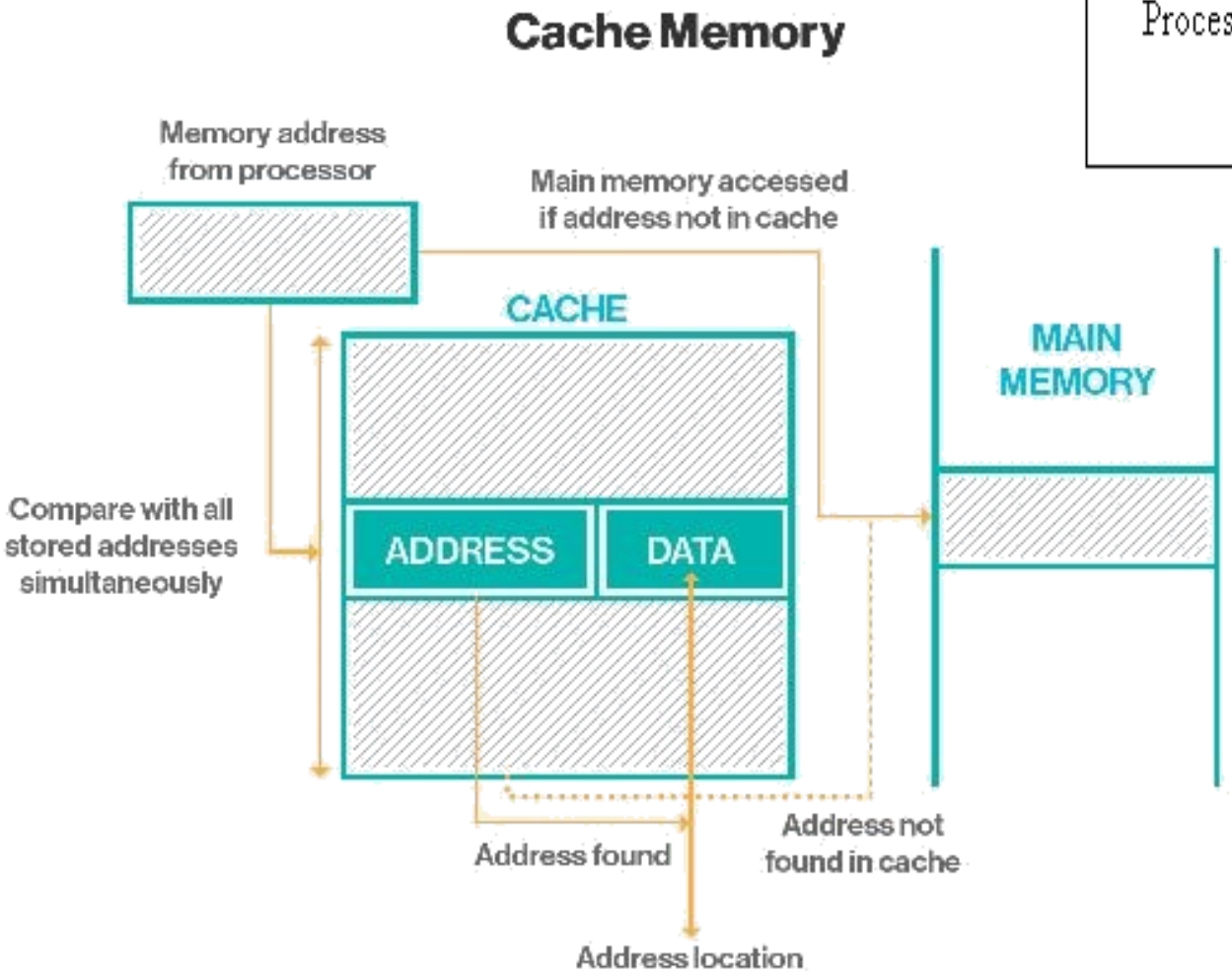
Reduzir os acessos a Memória

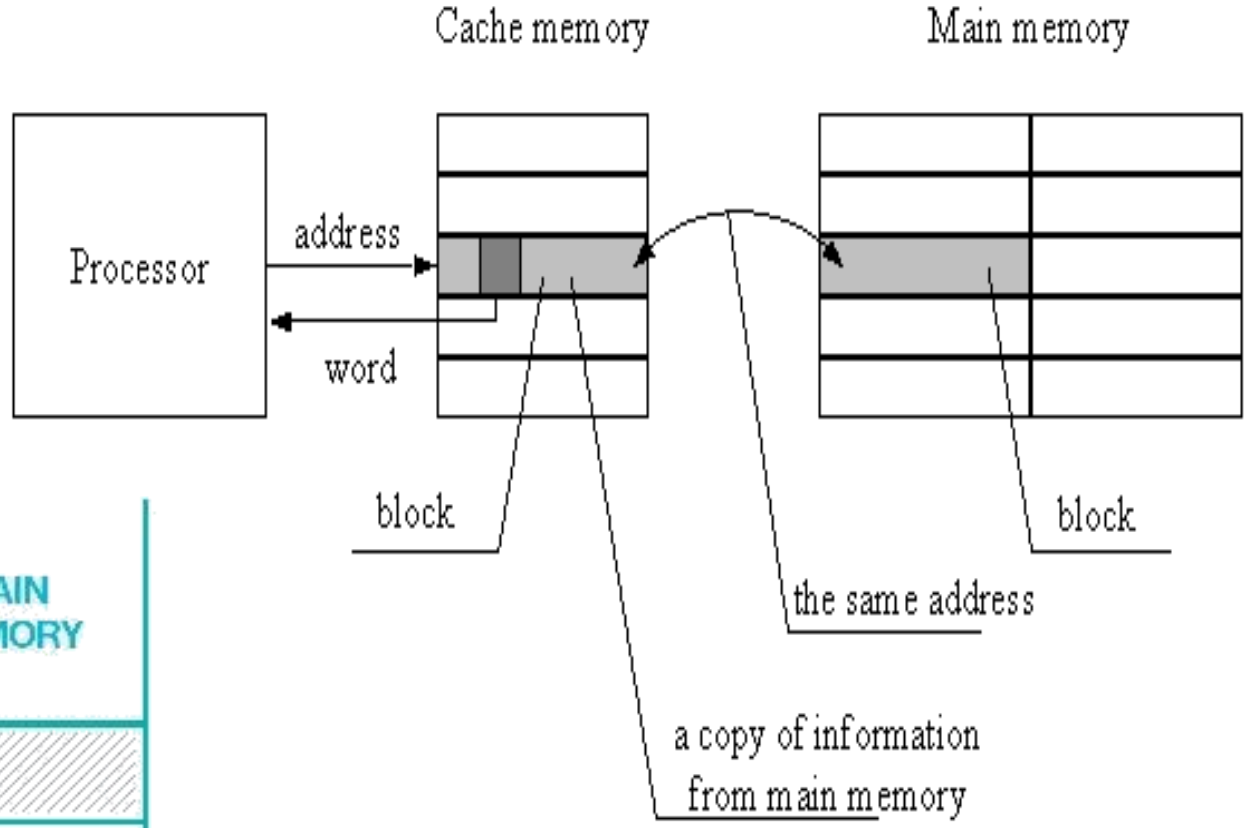
ESTRATÉGIAS

MEMÓRIA PRINCIPAL



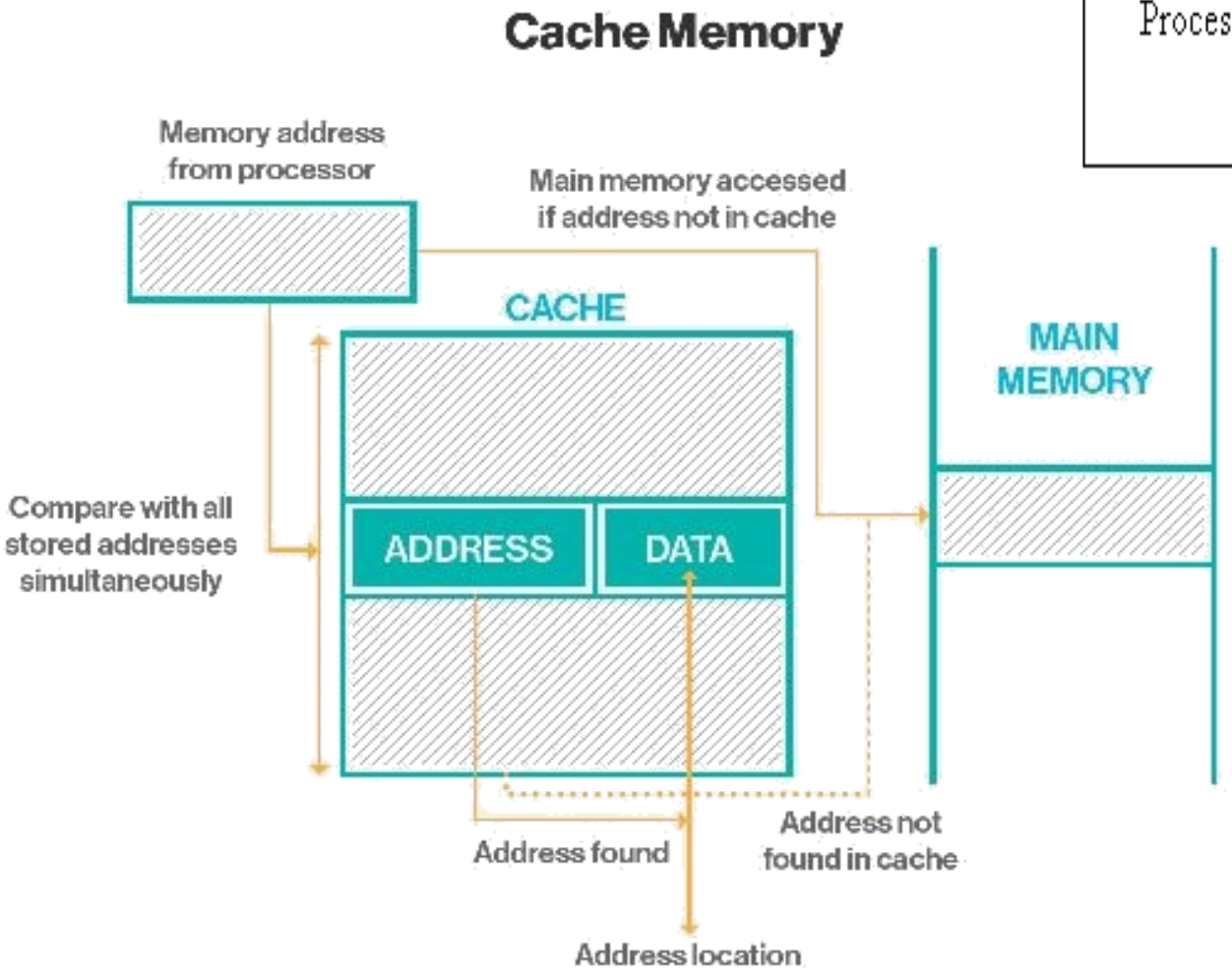
MEMÓRIA CACHE

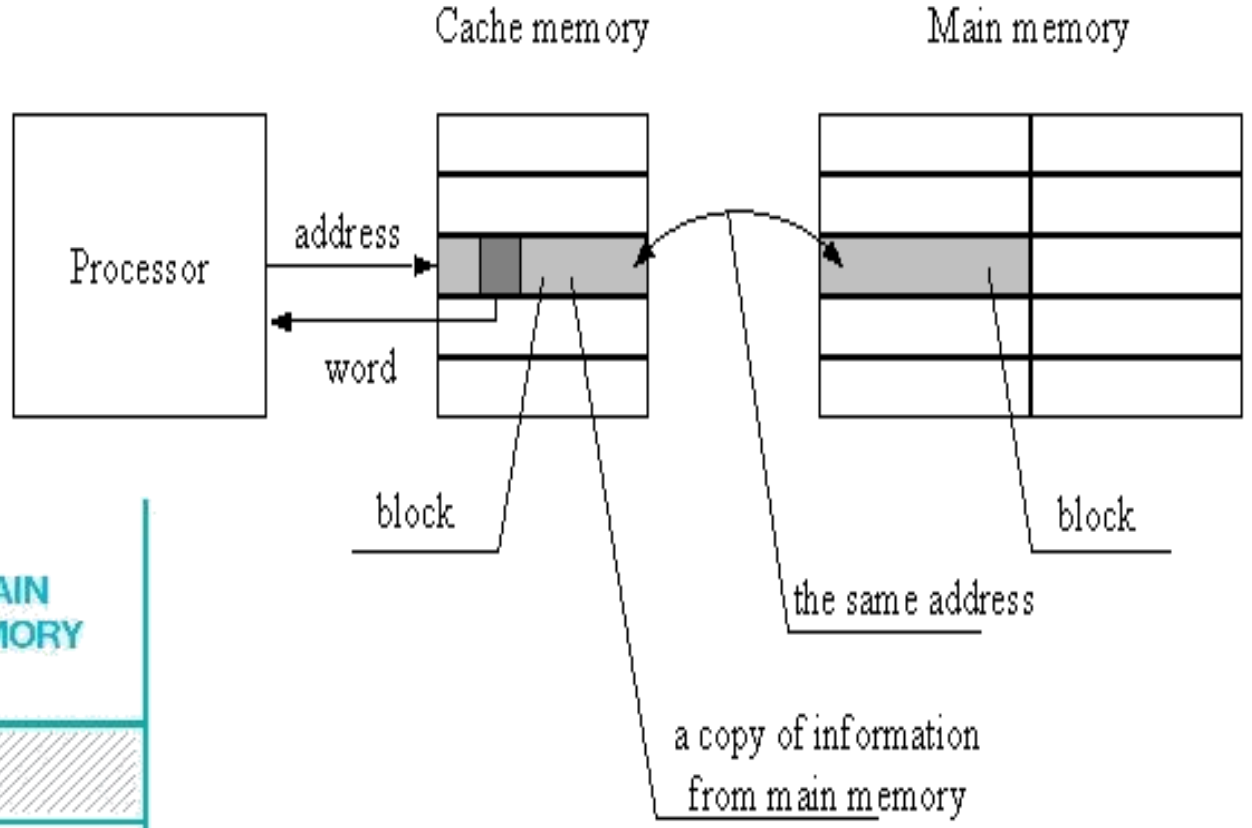




Memórias Cache (L1) integradas à CPU tem uma frequência 100 vezes maior que a MP. Lat. 1ns
Baixa qtd de armazenamento: +/- 256 Kbytes

MEMÓRIA CACHE

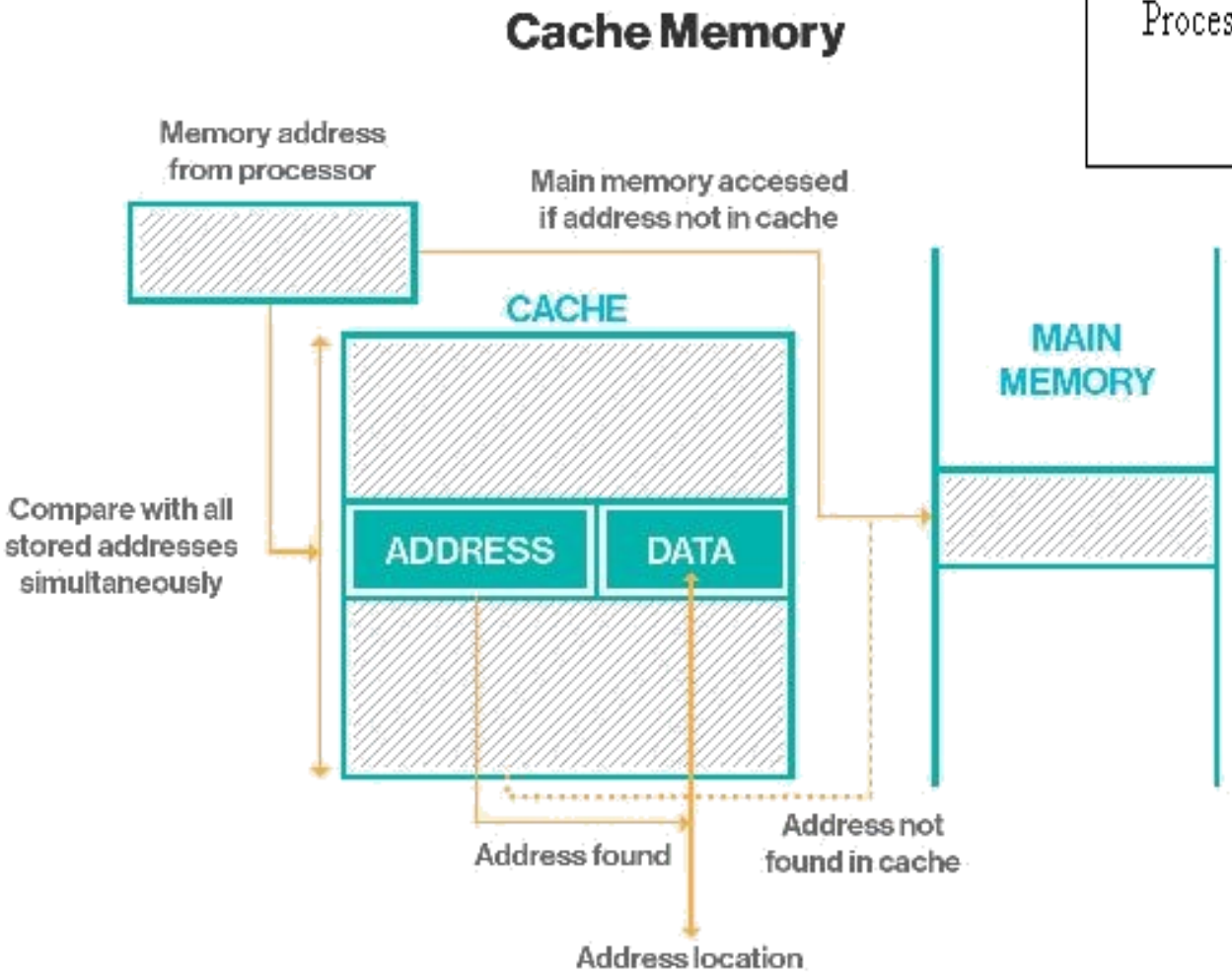


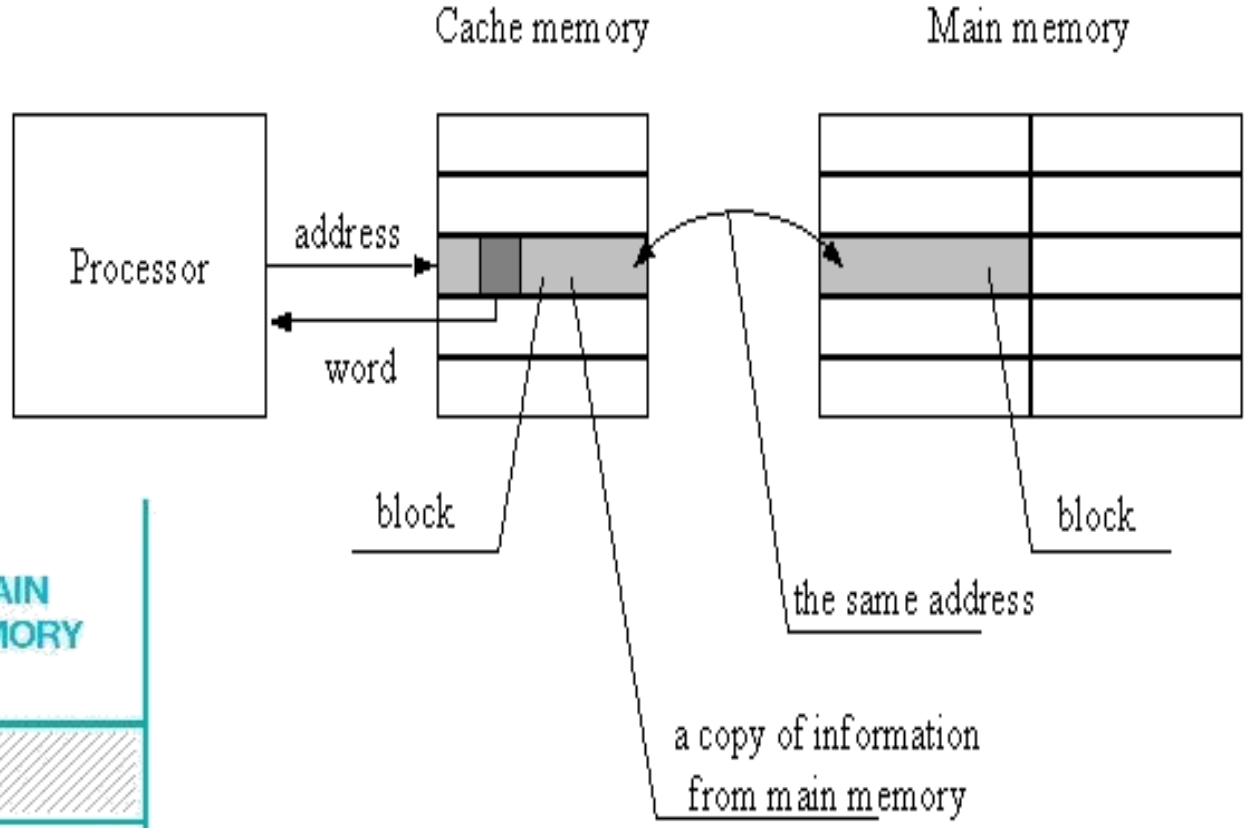


Memórias Cache (L1) integradas à CPU tem uma frequência 100 vezes maior que a MP. Latência 1ns.
Baixa qtd de armazenamento: +/- 256 Kbytes

Memórias Cache (L2) – Cache secundária. Localizada fora da CPU, mas integrada a pastilha do processador. Possível aumentar o tamanho de 256 Kb até 8 MB. Latência +/- 7ns

MEMÓRIA CACHE



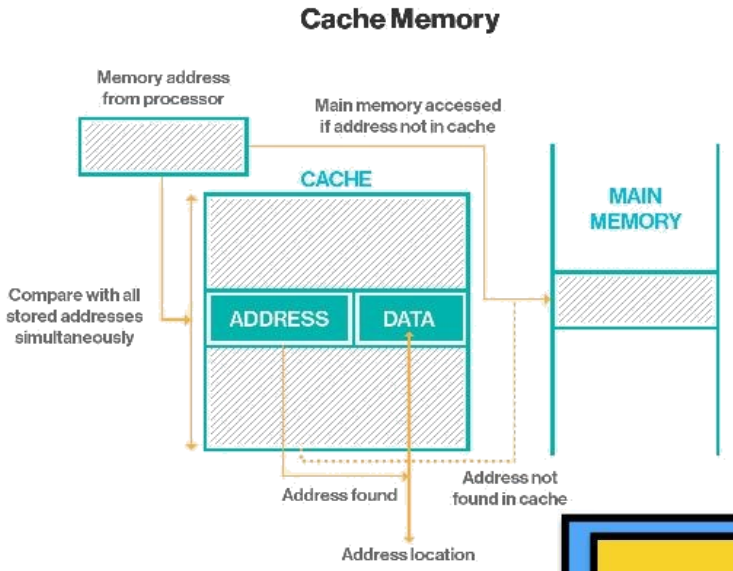


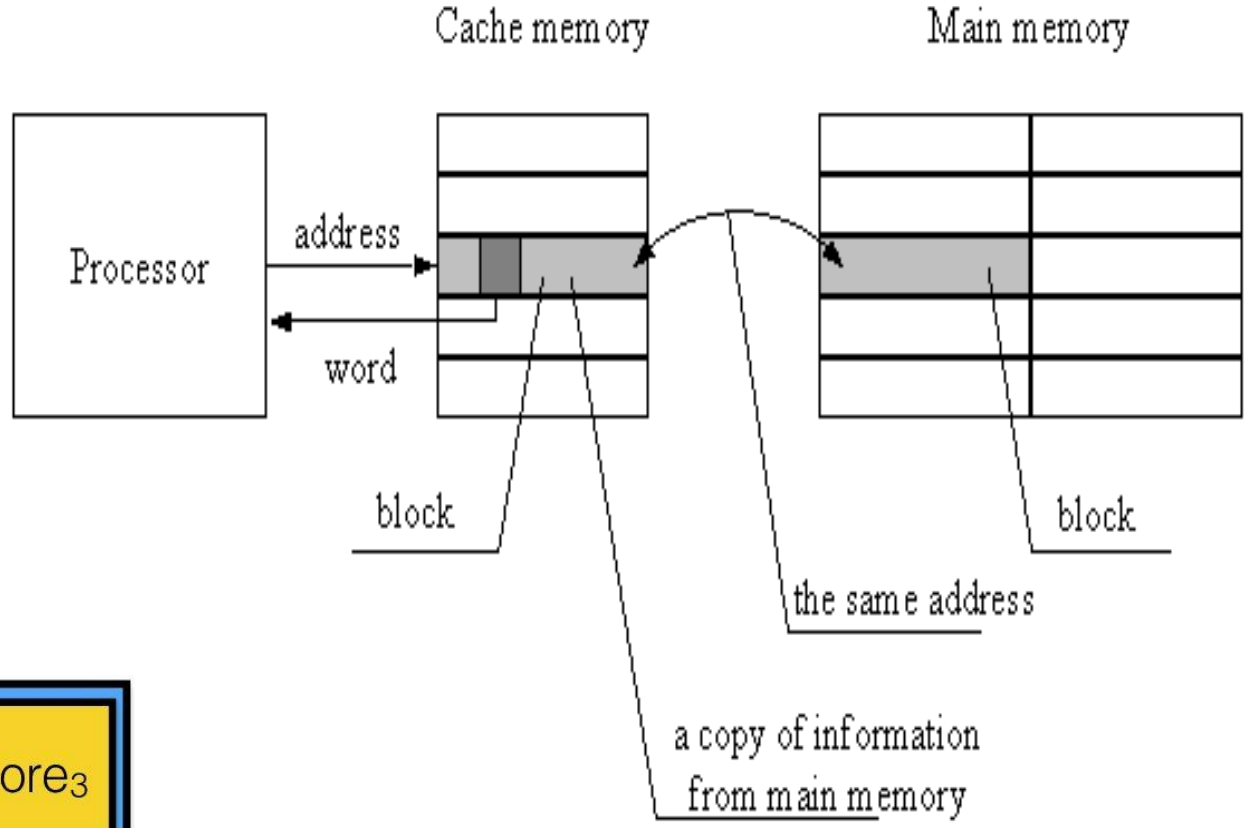
Memórias Cache (L1) integradas à CPU tem uma frequência 100 vezes maior que a MP. Latência 1ns.
Baixa qtd de armazenamento: +/- 256 Kbytes

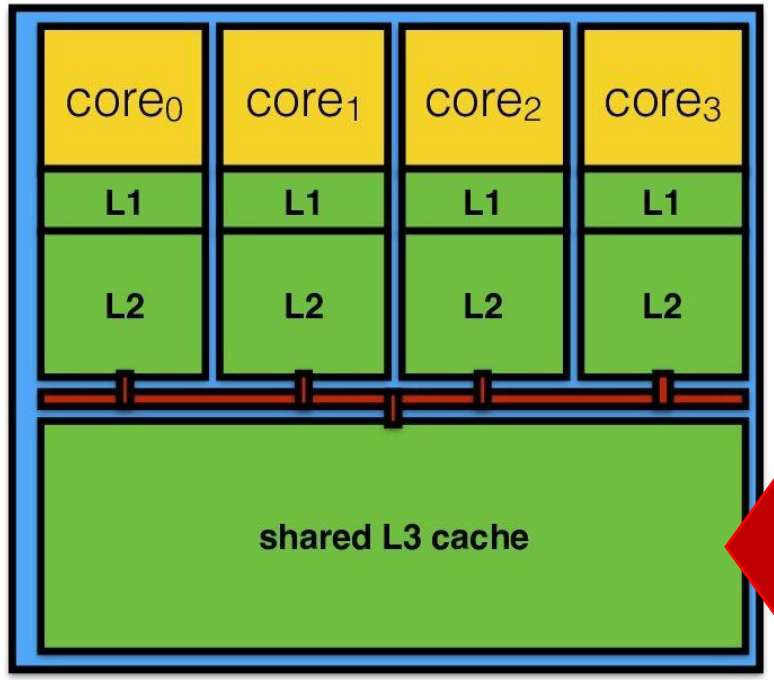
Memórias Cache (L2) – Cache secundária. Localizada fora da CPU, mas integrada a pastilha do processador. Possível aumentar o tamanho de 256 Kb até 8 MB. Latência +/- 7ns

Memórias Cache (L3) – compartilha com os multi-nucleos. Localizada na placa mãe e conectada a CPU por barramentos de altíssima velocidade.

MEMÓRIA CACHE







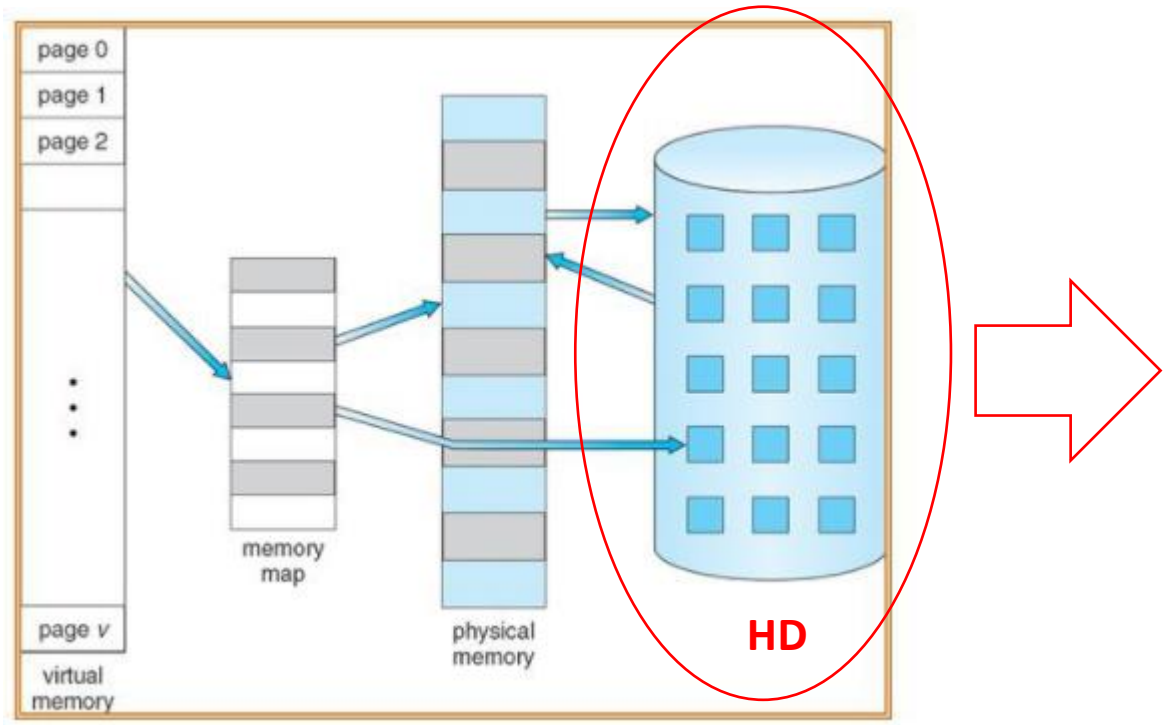
Memórias Cache (L1) integradas à CPU tem uma frequência 100 vezes maior que a MP. Latência 1ns.
Baixa qtd de armazenamento: +/- 256 Kbytes

Memórias Cache (L2) – Cache secundária. Localizada fora da CPU, mas integrada a pastilha do processador. Possível aumentar o tamanho de 256 Kb até 8 MB. Latência +/- 7ns

Memórias Cache (L3) – compartilha com os multi-nucleos. Localizada na placa mãe e conectada a CPU por barramentos de altíssima velocidade.

SISTEMA DE ARQUIVOS

Quando uma determinada informação não está na memória, a execução é "interrompida" e o SO busca a página que contém o dado demandado na memória secundária (HD), e a transfere para a MP.



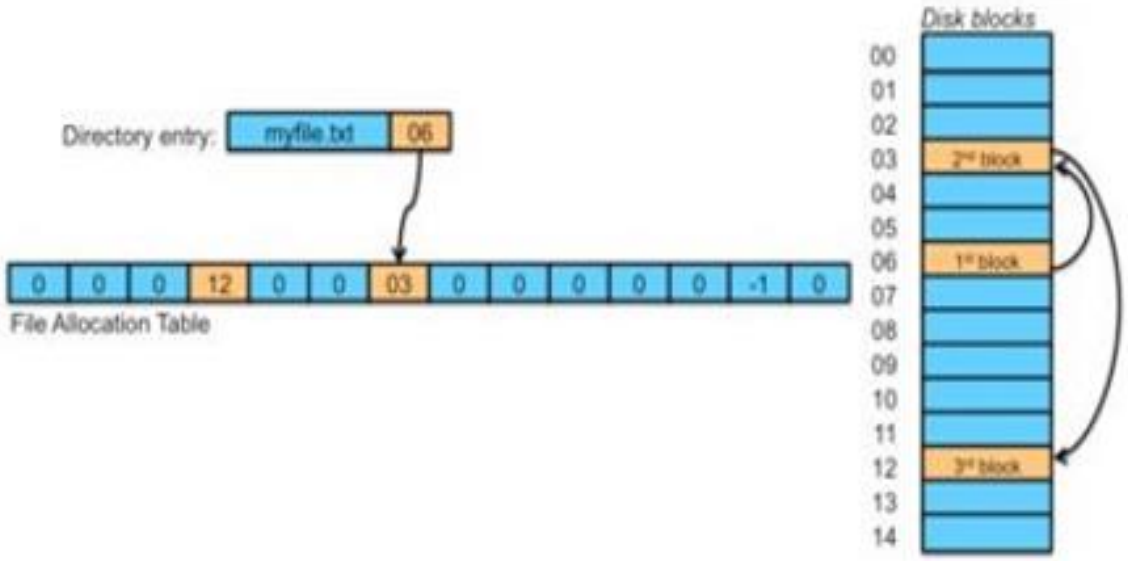
Sistema de Arquivo

Vários:
- FAT
- NTFS
- HFS (MacOS)
- UFS (Unix)

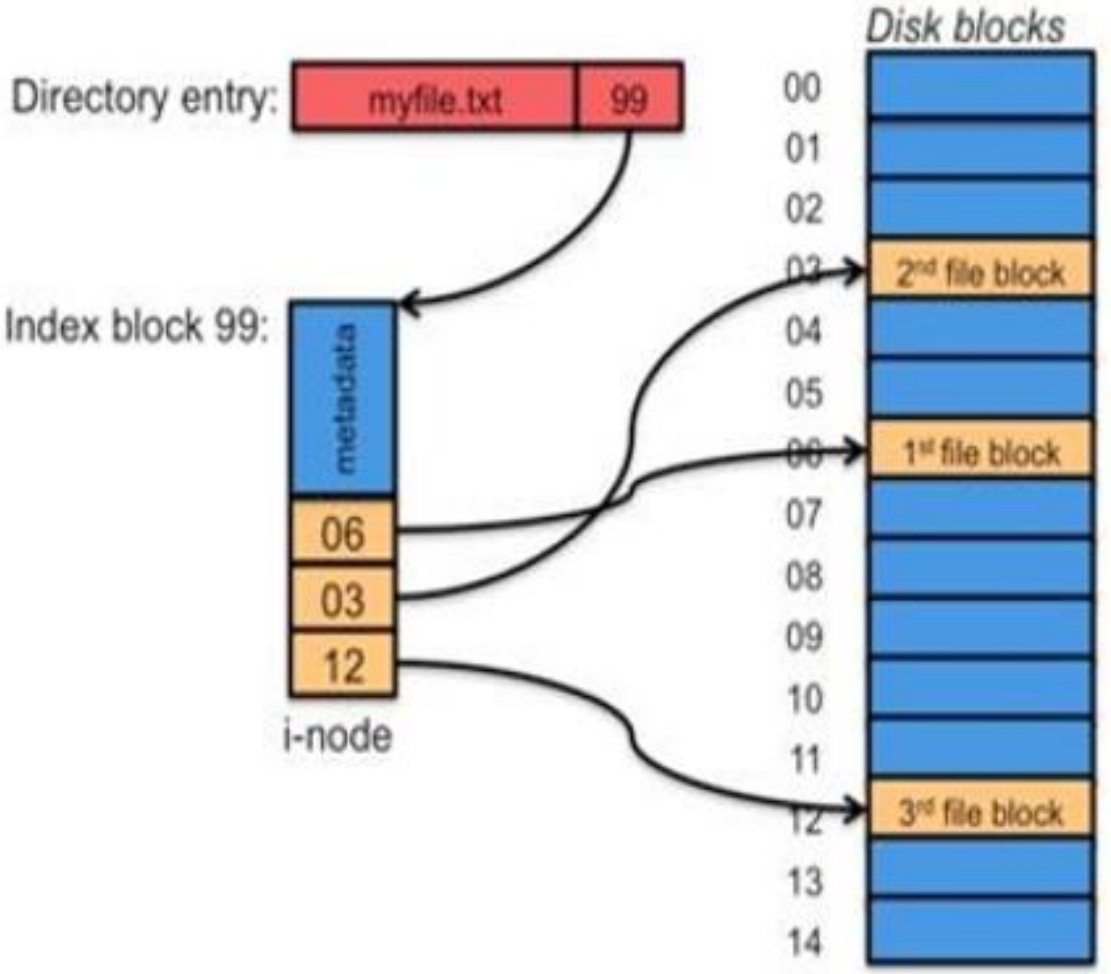
Mais difundido...
FAT8 – DOS 1971
FAT12 (12 bits)
FAT16 (16 bits)
FAT32 (32 bits)
exFAT (64 bits)

SISTEMA DE ARQUIVOS

SISTEMAS DE ARQUIVOS



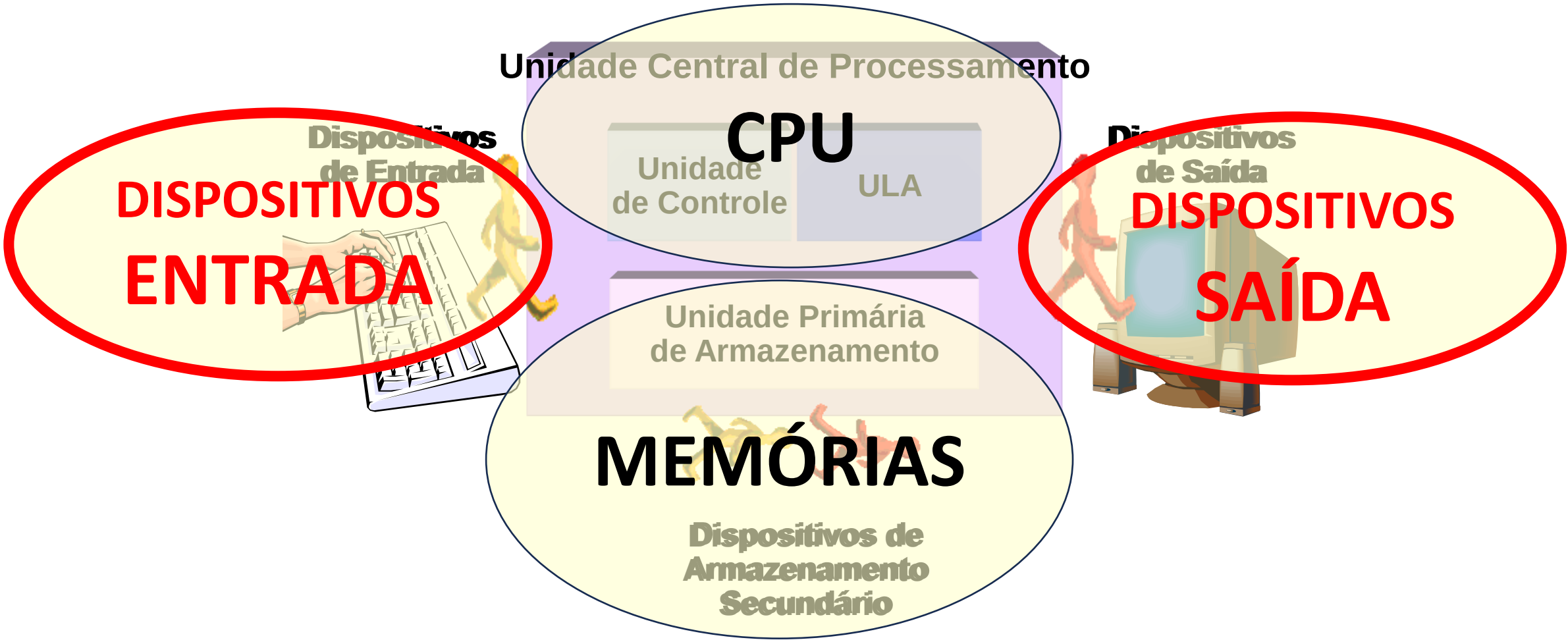
(a) FAT



(b) inode UFS (Unix)

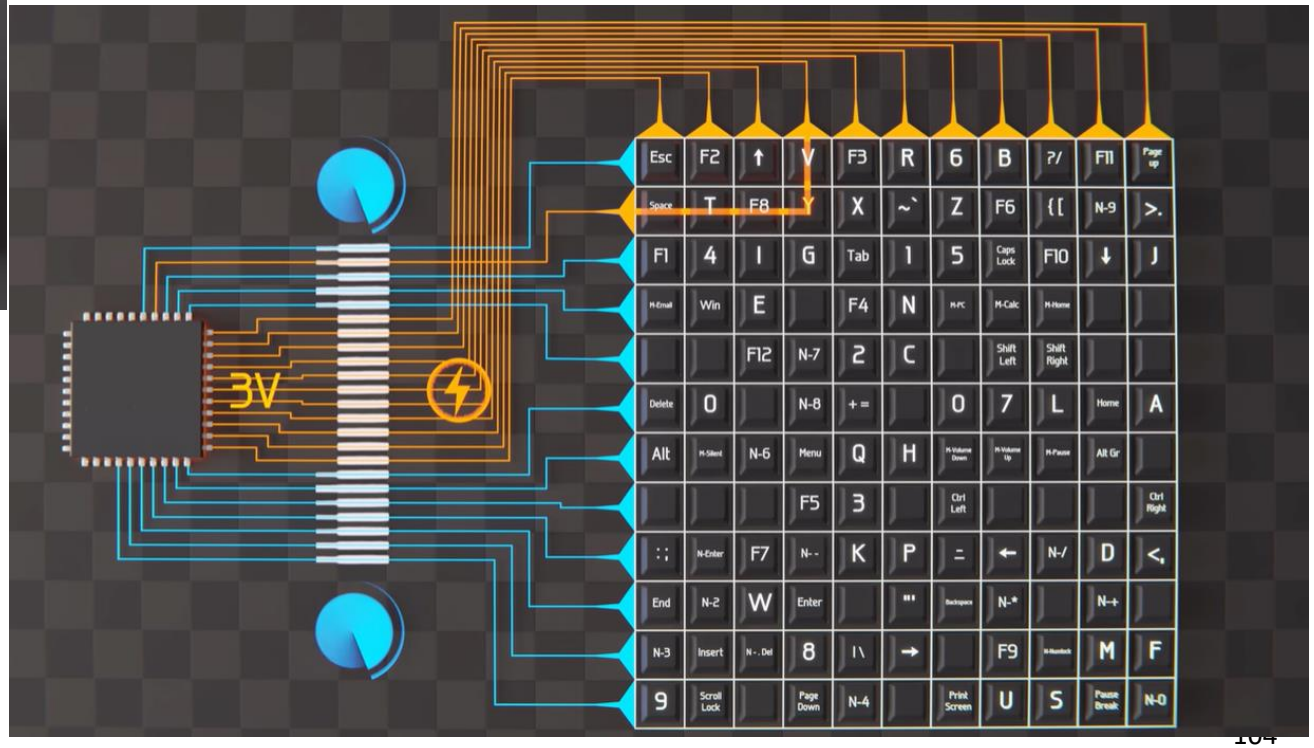
HARDWARE – DISPOSITIVOS I/O

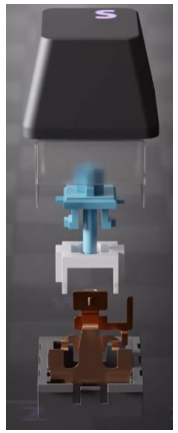
Computadores atuais

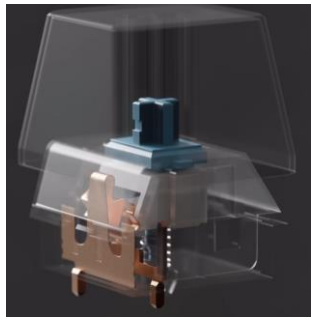


HARDWARE – DISPOSITIVOS I/O

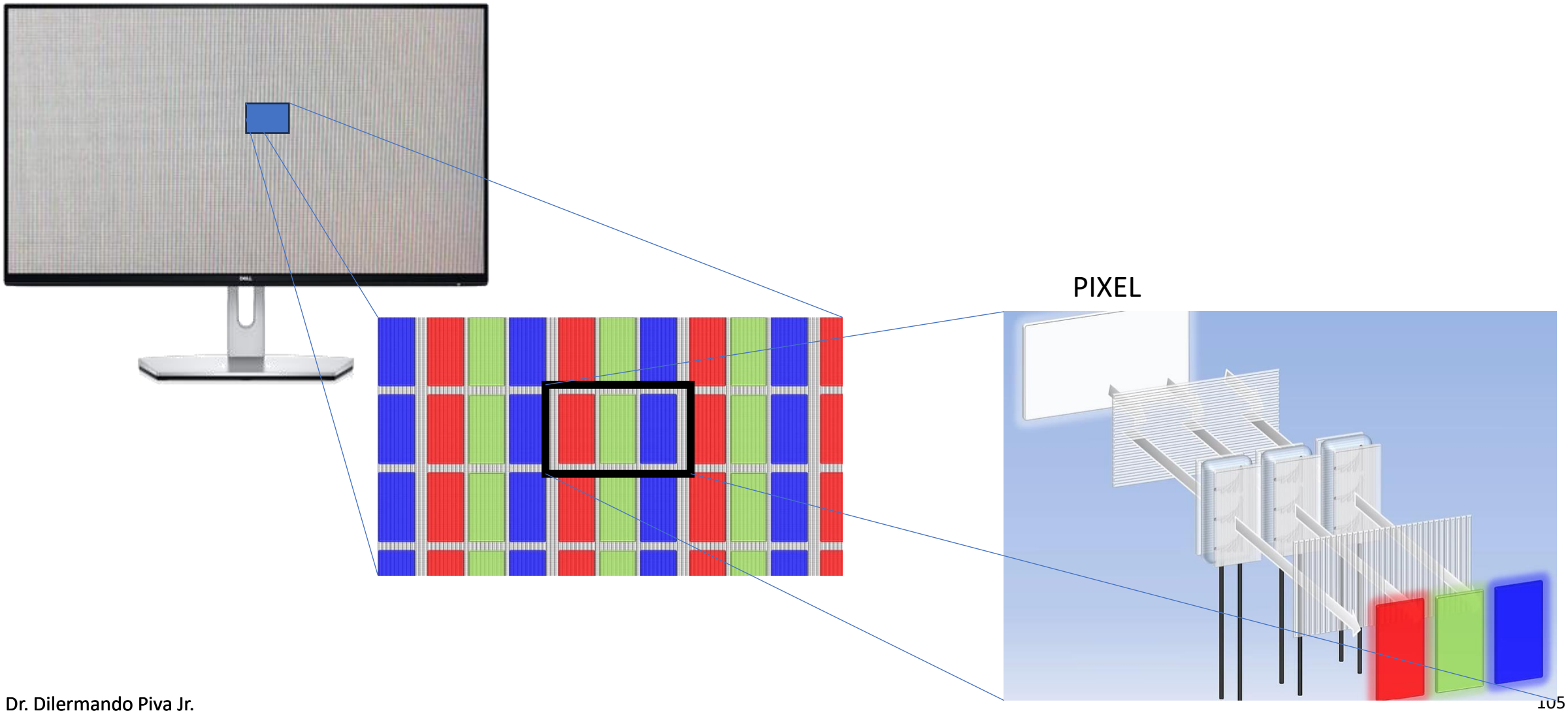
TECLADO



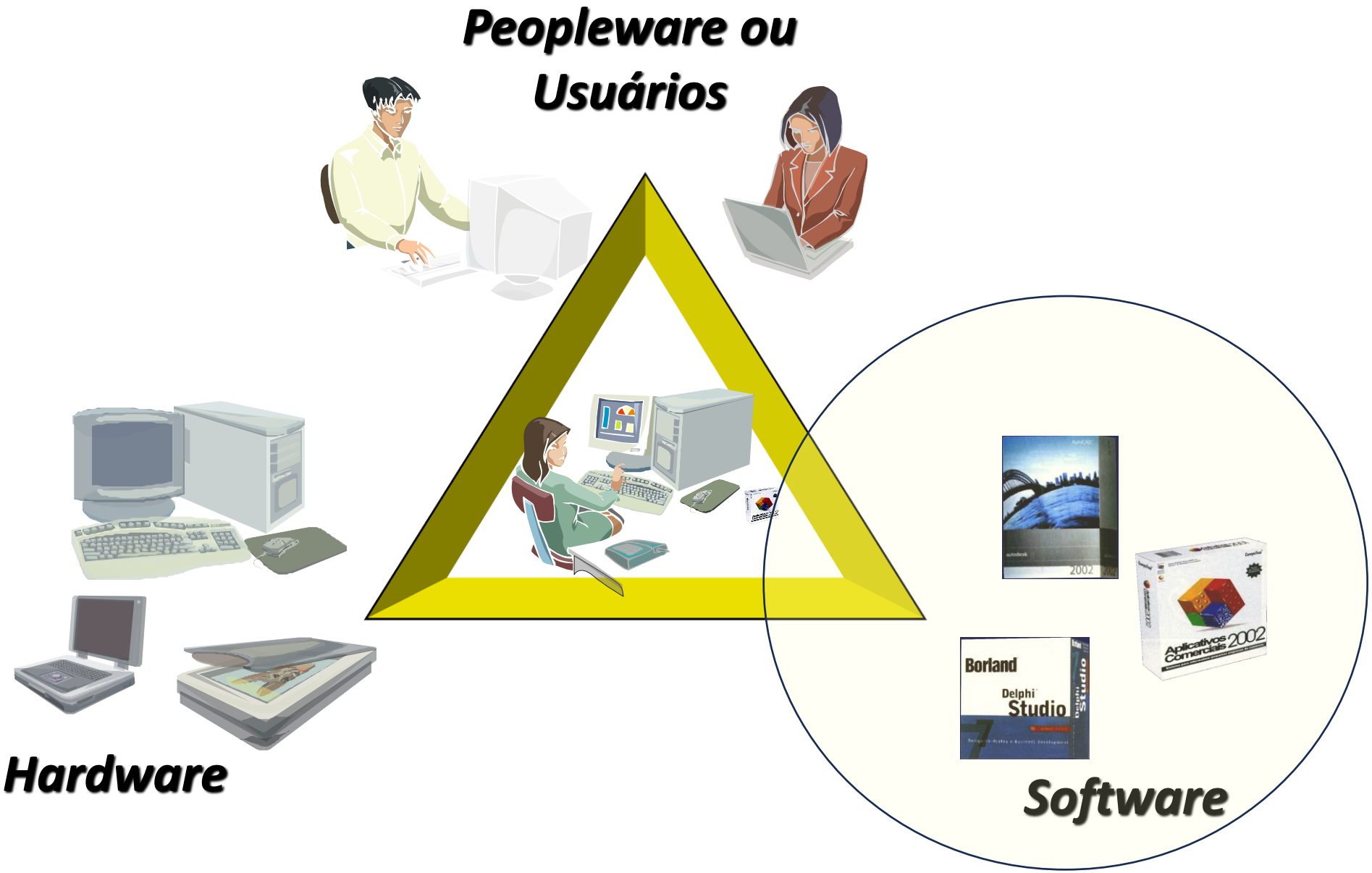




MONITOR (LCD / OLED)



SOFTWARE


Peopleware ou
Usuários
Hardware
Software
Borland
Delphi
Studio
Aplicativos Comerciais 2002

SOFTWARE

- **O que é Software?**

- Conjunto alterável de instruções, ordenadas e lógicas, fornecidas ao hardware para a execução de procedimentos necessários à solução dos problemas e tarefas do processamento de dados.

- É o que torna possível os computadores terem uma variedade ilimitada de utilizações.

SOFTWARE

Categorias de Software

- **Software Básico**

  Sistema Operacional

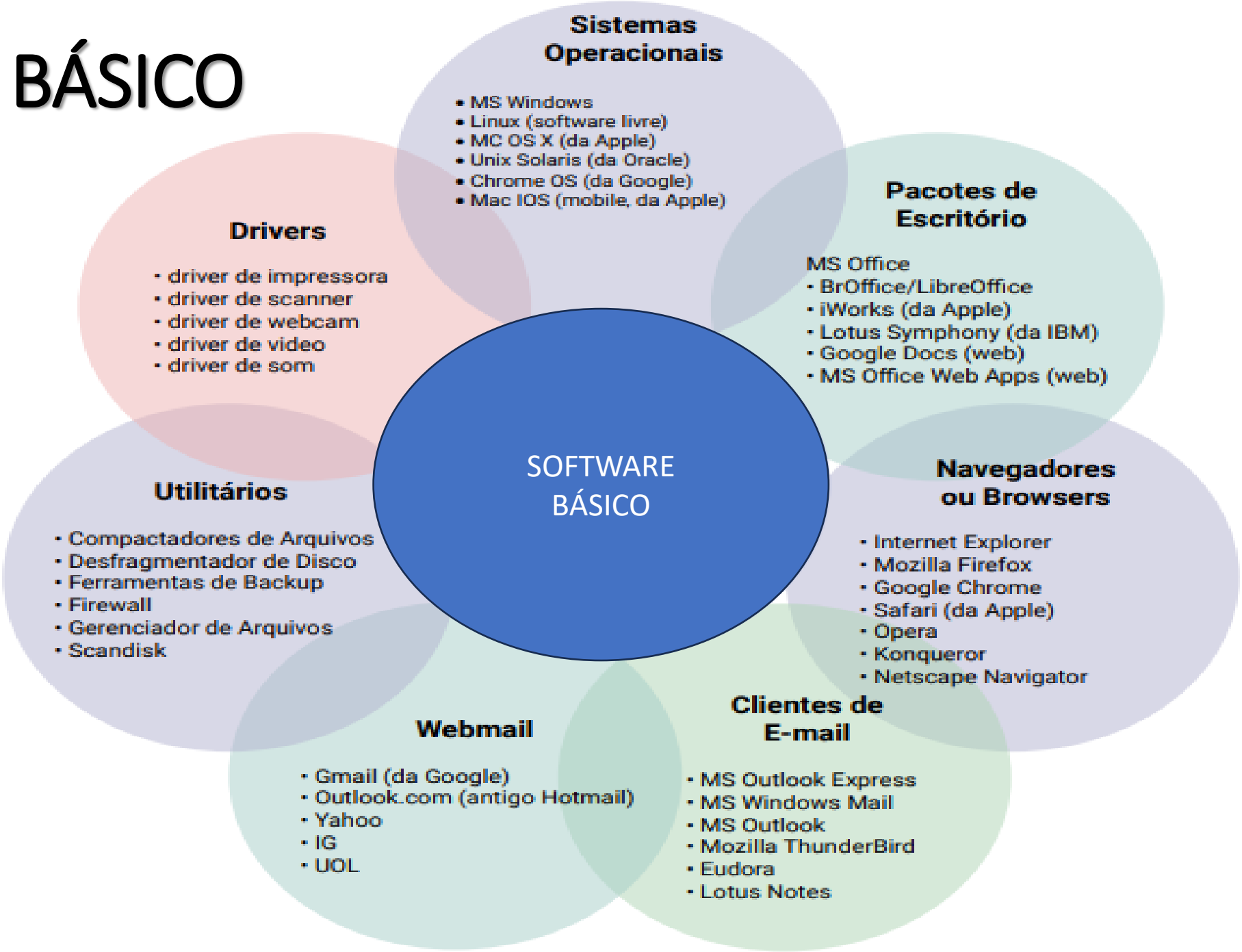
  Software Utilitário

- **Software Aplicativo ou Sistema Aplicativo**

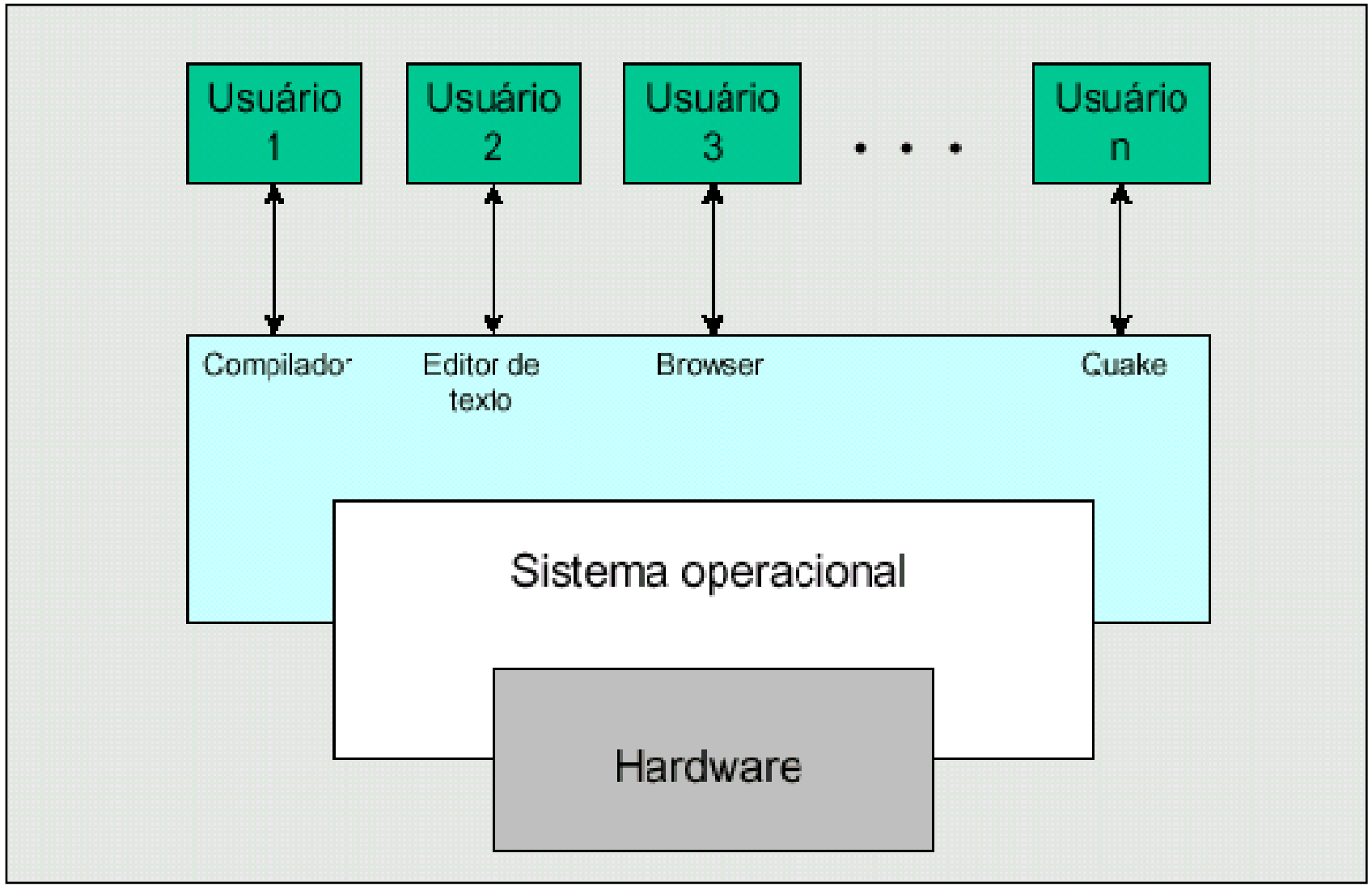
Outros autores ainda definem outra categoria:

Software de Linguagem ou Linguagem de Programação

SOFTWARE BÁSICO



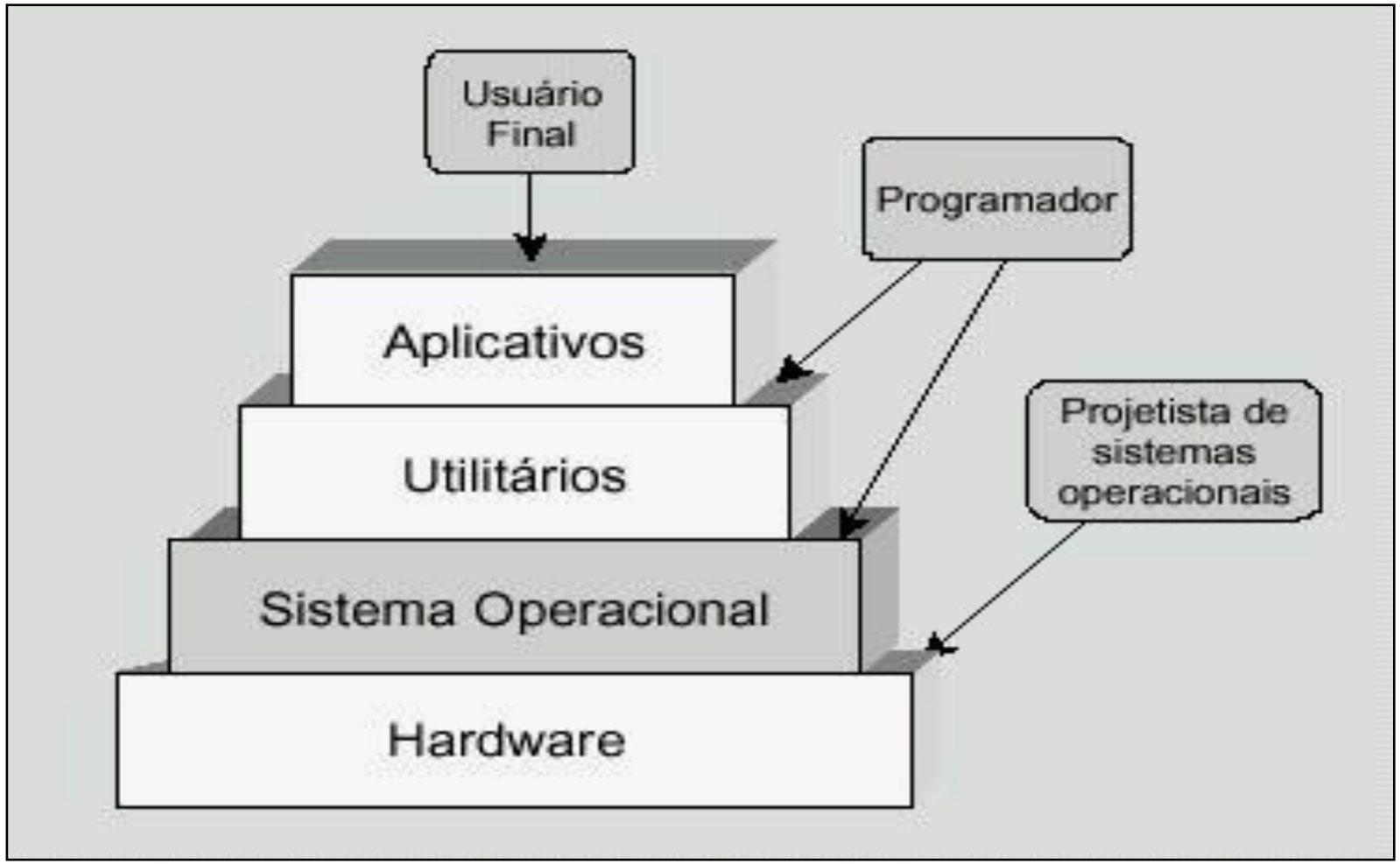
SOFTWARE



Componentes genéricos - sistema computacional

SOFTWARE

Diferentes visões- Sistema computacional



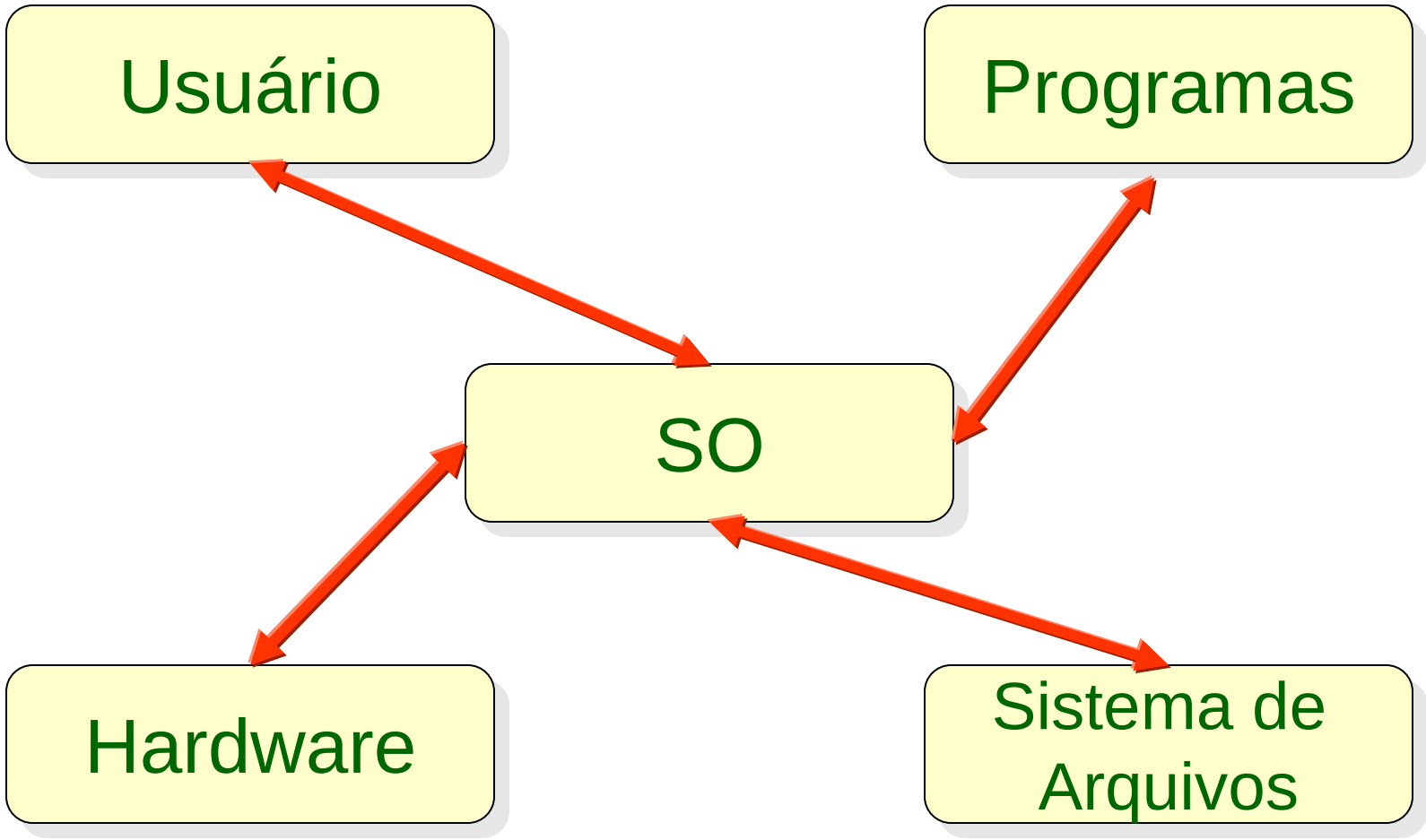
Visão Usuário Final X Programador X Projetista de Sistema Operacional

SOFTWARE - SISTEMA OPERACIONAL

Sistema Operacional

- Composto por um conjunto de **programas** e **rotinas**
- Controla a execução de qualquer *software* utilizado em um computador
- Gerencia os recursos do computador (*hardware* e *software*) de modo a:
  - Possibilitar sua utilização
  - Aumentar sua eficiência
  - Permitir a comunicação com outros equipamentos.

SOFTWARE - SISTEMA OPERACIONAL



SOFTWARE - SISTEMA OPERACIONAL

Kernel

- **Núcleo de um SO**
  - **Gestão de memória e dispositivos**
  - **Manutenção dos relógios do computador**
  - **Inicialização de aplicativos**
  - **Compartilhamento de recursos computacionais (programas, dispositivos, dados, informação)**
- **A cada inicialização do computador, o kernel e outras instruções de uso freqüente do SO são carregadas**

Residente na memória

- **Permanece na memória enquanto o computador estiver executando**
- **O *kernel* é residente na memória**

Não Residente na memória

- **Instruções permanecem no disco rígido até que sejam necessárias**
- **Outras partes do SO são não residentes**

SOFTWARE - SISTEMA OPERACIONAL

MICROSOFT WINDOWS

- Código **fechado**
- Sistema operacional **pago**
- Utilizado em **88%** dos computadores
- Versão mais recente: Windows 11

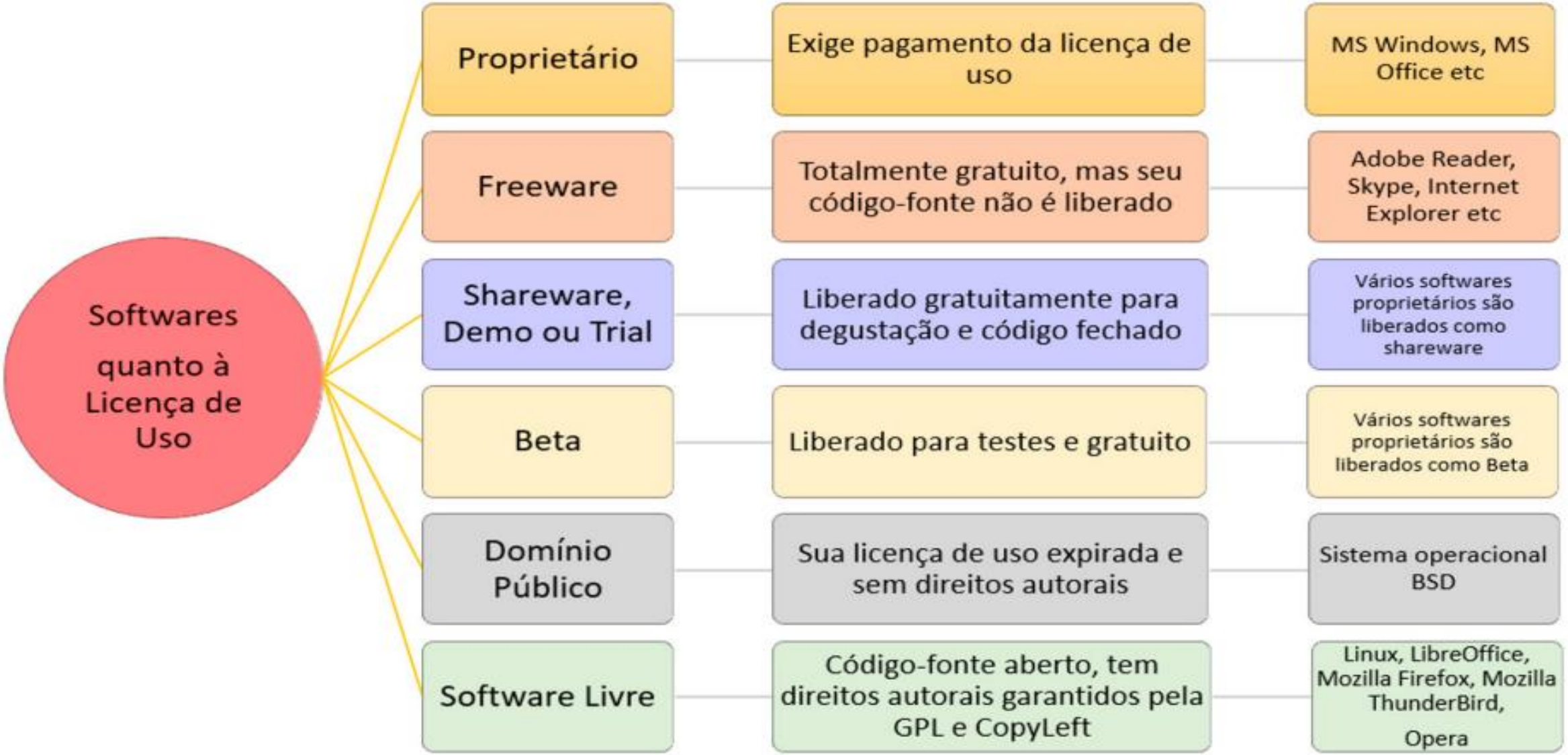
LINUX

- Código **aberto**
- Sistema operacional **gratuito**
- Utilizado em **1,14%** dos computadores
- Principais distribuições: Ubuntu, Mint, Debian, Pop!_OS, Arch, dentre outras

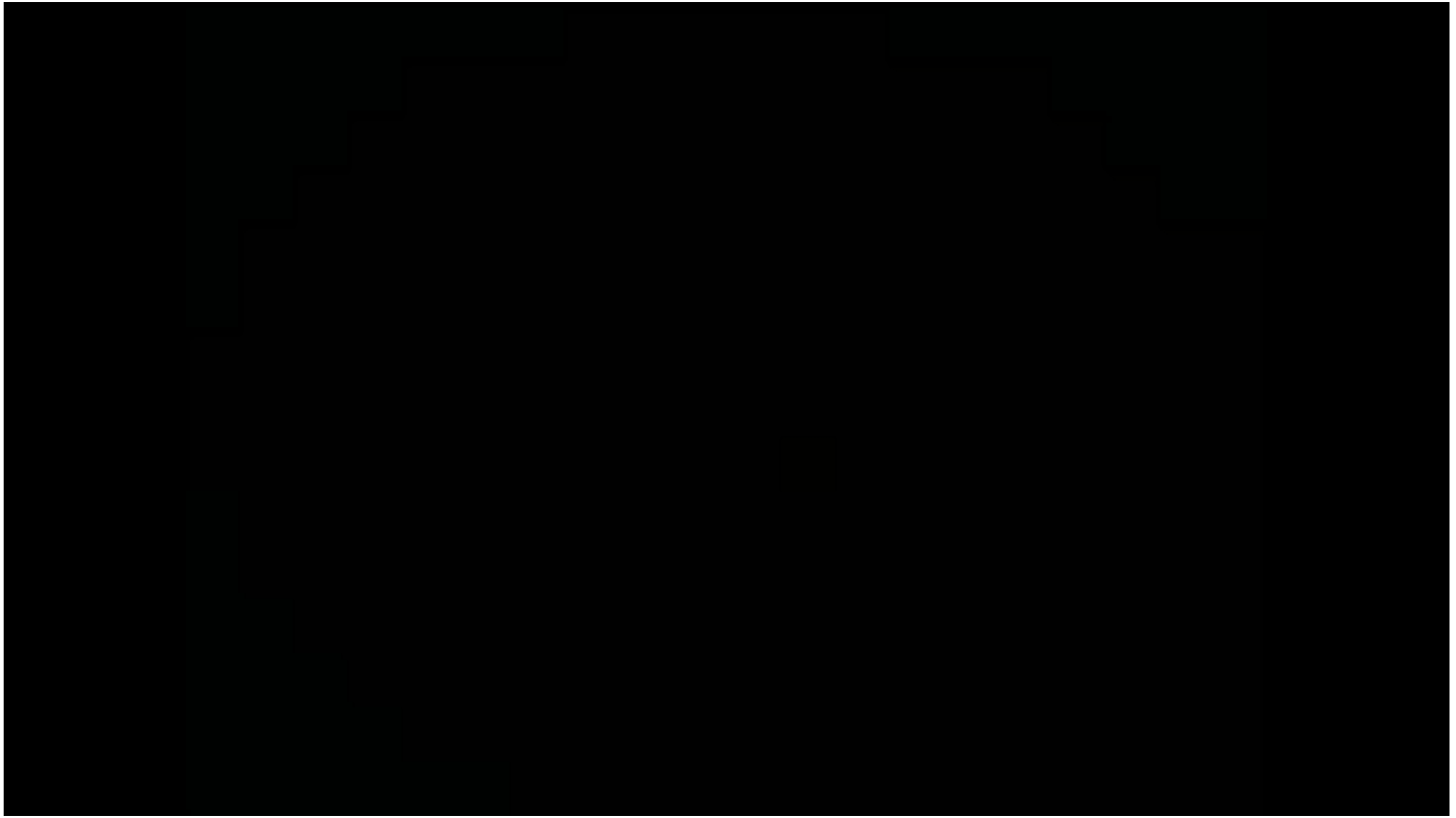
MacOS

- Código **fechado**
- Sistema operacional **pago**
- Utilizado em **3,7%** dos computadores
- Versão mais recente: macOS Big Sur

SOFTWARE - Modalidades



Funcionamento do Computador (PARÓDIA)



Experimentação ativa

Problema: **Otimização de Desempenho em Algoritmos de Processamento de Grandes Volumes de Dados em Tempo Real**

Contexto:

Uma empresa de análise de dados em tempo real está enfrentando desafios no processamento eficiente de grandes volumes de dados provenientes de fontes diversas, como redes sociais, sensores IoT e transações online.

O sistema atual de processamento em tempo real está enfrentando atrasos significativos, prejudicando a capacidade da empresa de fornecer insights em tempo hábil para seus clientes.

Experimentação ativa

Problema: ***Otimização de Desempenho em Algoritmos de Processamento de Grandes Volumes de Dados em Tempo Real***

Tarefas:

1. Análise de Hardware e Arquitetura de Software:
   - Avaliar o hardware utilizado no sistema de processamento em tempo real.
   - Identificar possíveis melhorias na arquitetura de software para otimizar o processamento de dados em tempo real.
2. Arquitetura de Sistemas Computacionais:
   - Aplicar conceitos de arquitetura de sistemas computacionais, incluindo a análise de trade-offs entre arquiteturas RISC e CISC.
   - Utilizar álgebra de Boole para otimizar operações lógicas no processamento de dados.
3. Memória e Hierarquias:
   - Analisar o uso da memória no sistema, considerando diferentes tipos e hierarquias de memória.
   - Propor estratégias para melhorar o acesso eficiente à memória durante o processamento de grandes volumes de dados.
4. Dispositivos de Entrada e Saída:
   - Avaliar os dispositivos de entrada e saída utilizados no sistema de processamento em tempo real.
   - Propor melhorias para otimizar a comunicação entre o sistema e as fontes de dados externas.

Colaboração e envolvimento no Mundo Real

Divida em Grupos os estudantes para colaborarem com suas soluções.

O grupo deve sair com uma solução única e unificada.

Cada integrante do grupo, deve fazer a publicação em seu portfólio da solução a que chegou.