



QXD-005 - Arquitetura de Computadores

Introdução à Disciplina

Prof. Pedro Botelho

Quem sou eu?

- **Formação**

- Engenharia de Computação (UFC - 2025)
- Técnico em Informática (EEEP - 2018)

- **Atuação**

- Projetista de Chips (von Braun Labs - 2025)
- Professor Particular (Profes/Superprof - 2023 a 2025)
- Monitor de Arquitetura (UFC - 2021 e 2024)
- Desenvolvedor C/C++ (LSBD - 2022 a 2023)

- **Interesses**

- Sistemas Embarcados e Sistemas Críticos
- Projeto de Sistemas Digitais e Circuitos Integrados
- Arquiteturas de Baixa Potência (AVR, ARM Cortex-M e RISC-V)
- Arquiteturas de Alta Performance (x86-64 e ARMv8)



Introdução ao Assunto

- Em “*Fundamentos de Programação*” → Alto Nível de Abstração (*Software*)
 - Interação com o computador via **linguagem de programação** e.g. C, C++, Go, etc...
 - Funcionamento abstraído pelo sistema operacional e pela linguagem de programação
- Em “*Arquitetura de Computadores*” → Baixo Nível de Abstração (*Hardware*)
 - Compreender o **funcionamento interno** do computador
 - “*Como o computador se comporta durante a execução do programa?*”
 - Fundamental para o desenvolvimento em baixo nível → **Sistemas Embarcados**¹
- Posteriormente em “*Sistemas Operacionais*” → Fornecer **abstração** para o usuário
 - Vários recursos de **abstração** e **gerenciamento** de *hardware*
 - Todos os programas interagem com o SO → SO interage com o *hardware*
- Teoria (Aulas) + Prática (Laboratórios)

¹ Computador mínimo que controla um sistema maior



Introdução à Disciplina

O Mundo de Arquitetura de Computadores

O que são computadores?

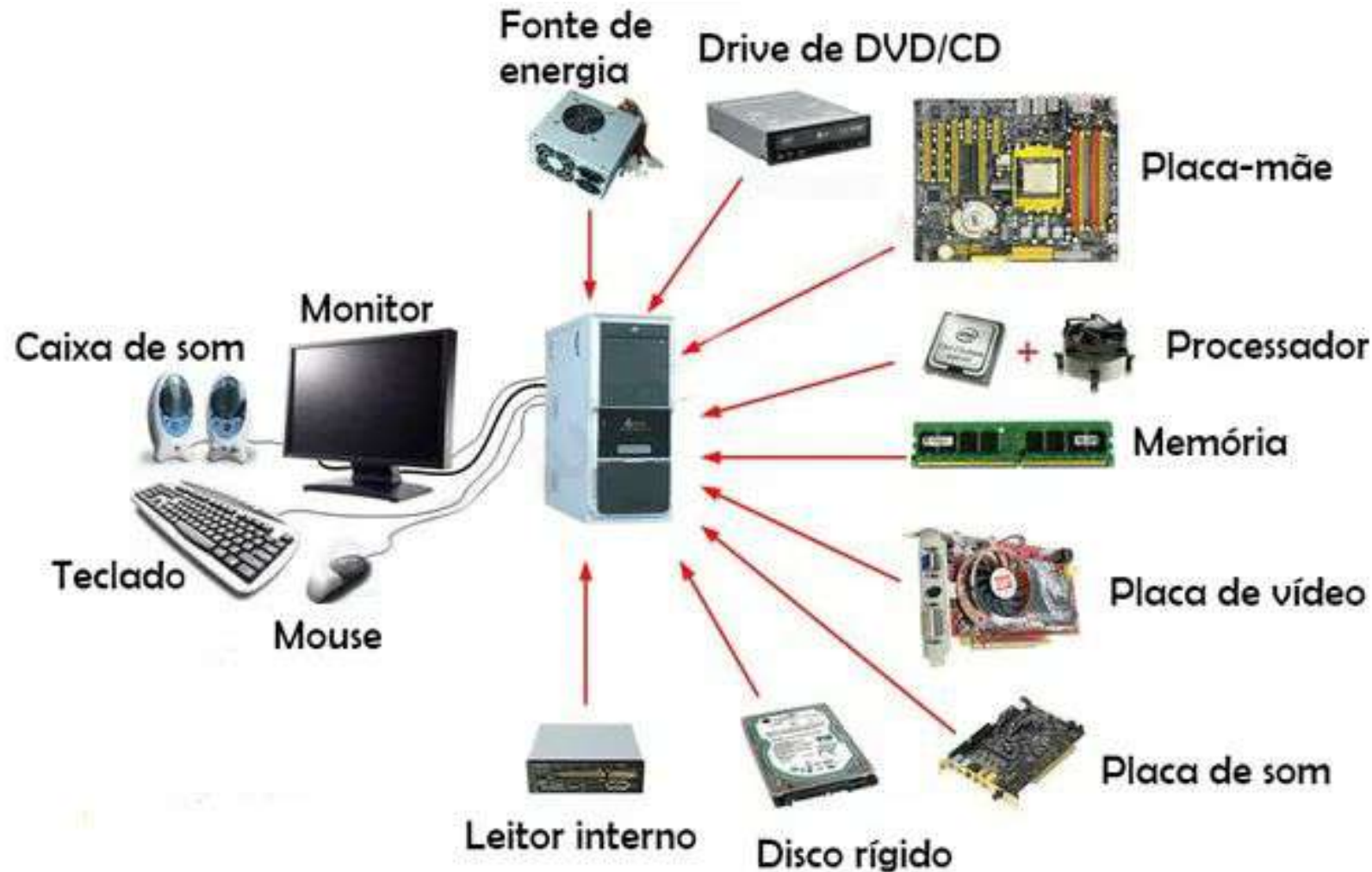
- Existem diversas definições → Em resumo, “**hardware** que executa **software**”
 - *Hardware* (equipamento físico) \rightleftharpoons *Software* (operações do hardware)
- “Máquina que pode resolver problemas, executando **instruções** que lhe são dadas”
 - Instruções → operações do hardware (**quais?**) → compõem o software
- “Sistema interconectado de **processadores, memórias e dispositivos de entrada/saída**”
- “Equipamento que realiza **processamento de dados** através de uma sequência de instruções”
 - **Como?** → Assunto dessa disciplina
 - Sequência de instruções descrevendo como realizar determinada tarefa → **Programa**

```
int main(void) {  
    printf("Hello World!\n");  
    return 0;  
}
```

Programa em C

Principais Componentes de um Computador

- De acordo com Von Neumann → Processador, memória e dispositivos de entrada e saída(E/S)



Exemplos de Computadores de Pequeno Porte



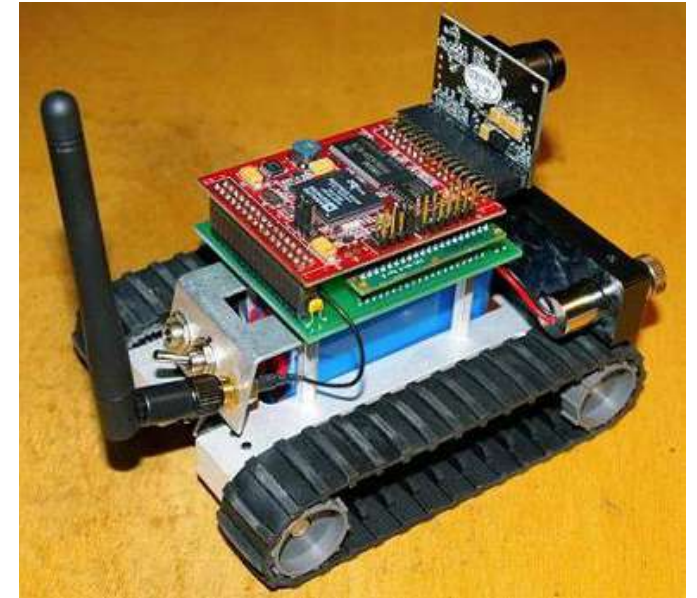
Desktop



Smartphone



Console de Jogos



Sistema Embarcado

Exemplos de Computadores de Grande Porte



Computador de Tubo
a Vácuo (ENIAC)



Mainframe (IBM S/360)



Supercomputador
(IBM Summit)

Estrutura e Função do Computador

- Computador → Sistema hierárquico (hierarquia de níveis ou camadas)
 - Abordagem *Top-down* → começar com uma visão de cima e decompor o sistema em suas subpartes
 - Abordagem *Bottom-up* → começar de baixo e subir até uma descrição completa
- Em cada nível o projetista se preocupa com...
 - **Estrutura** → o modo como os componentes são inter-relacionados.
 - **Função** → a operação individual de cada componente como parte da estrutura.

Sistema Computacional		
Tipos de Memórias	Dispositivos de Entrada e Saída	Processador
Tecnologias Utilizadas	Barramentos de E/S	Instruções e Assembly
Recursos Modernos	Técnicas de E/S	<i>Pipeline</i> e Controle
Eletrônica		

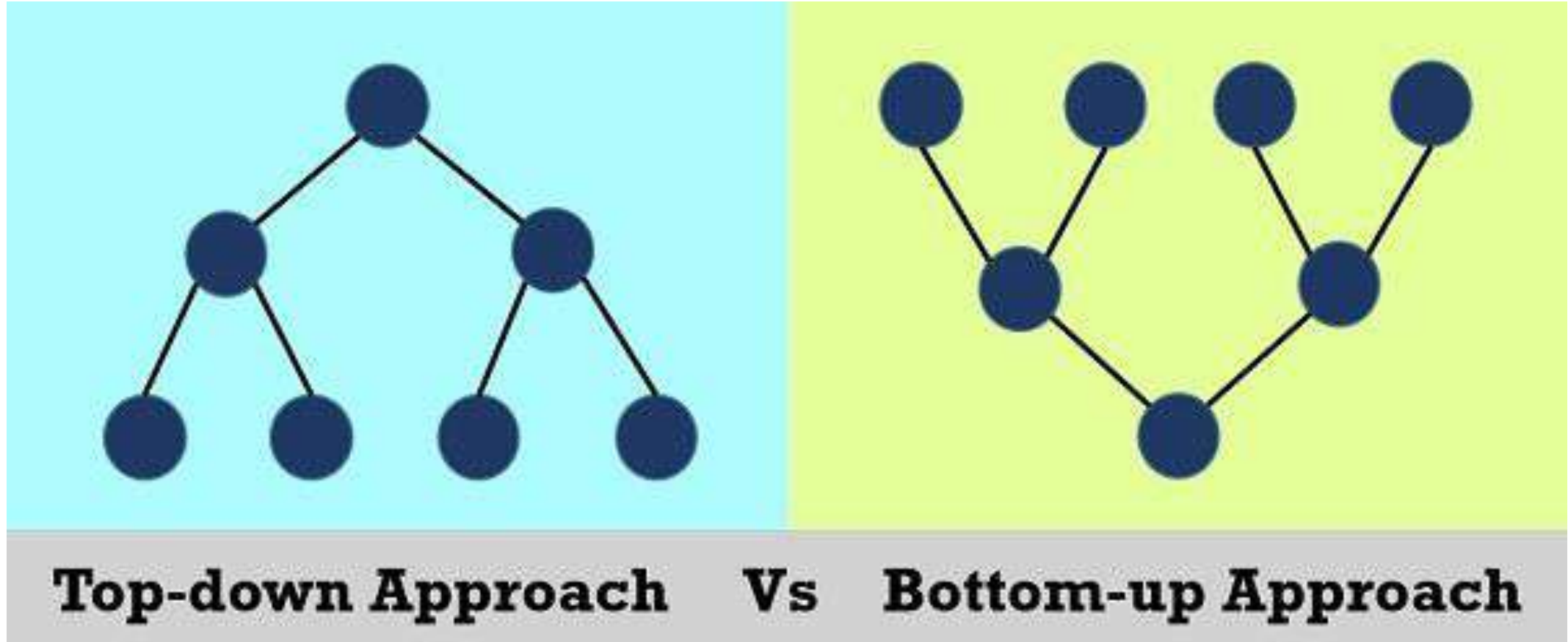
Camadas do Computador

Aplicações
Sistema Operacional
<i>Firmware</i>
<i>Hardware</i>

Camadas
do Sistema

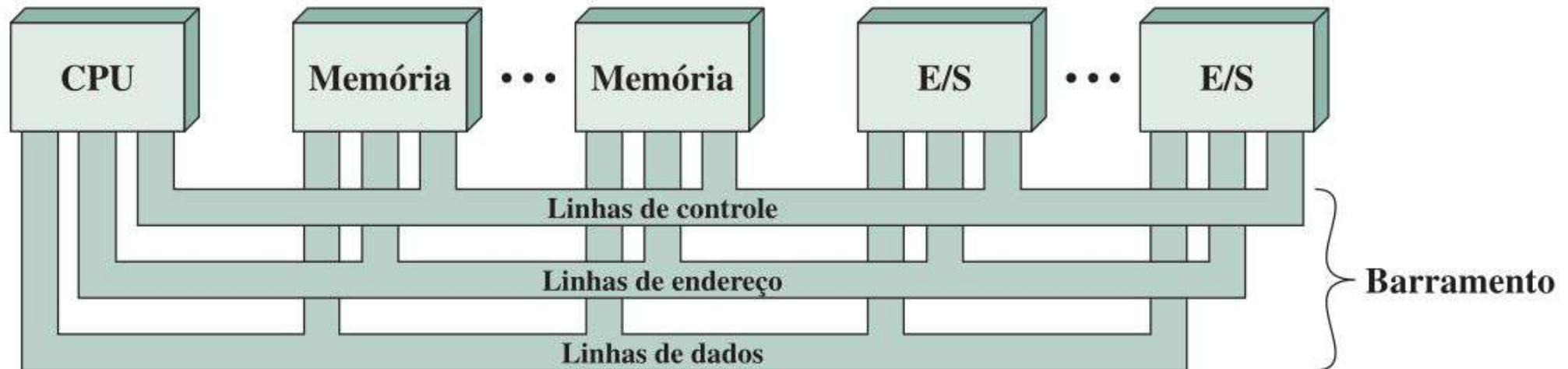
Top-down vs. Bottom-up

- Qual seria melhor para o aprendizado?

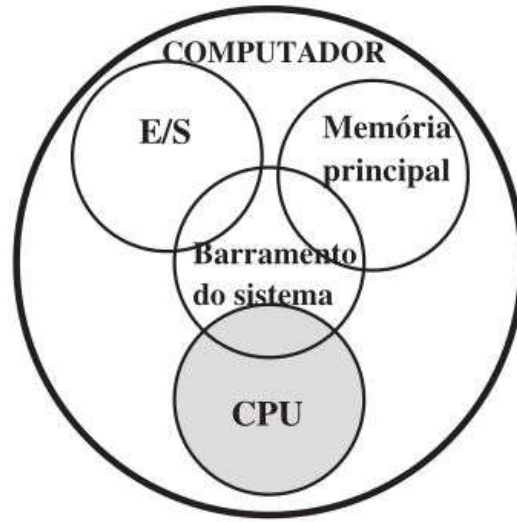


Estrutura do Computador

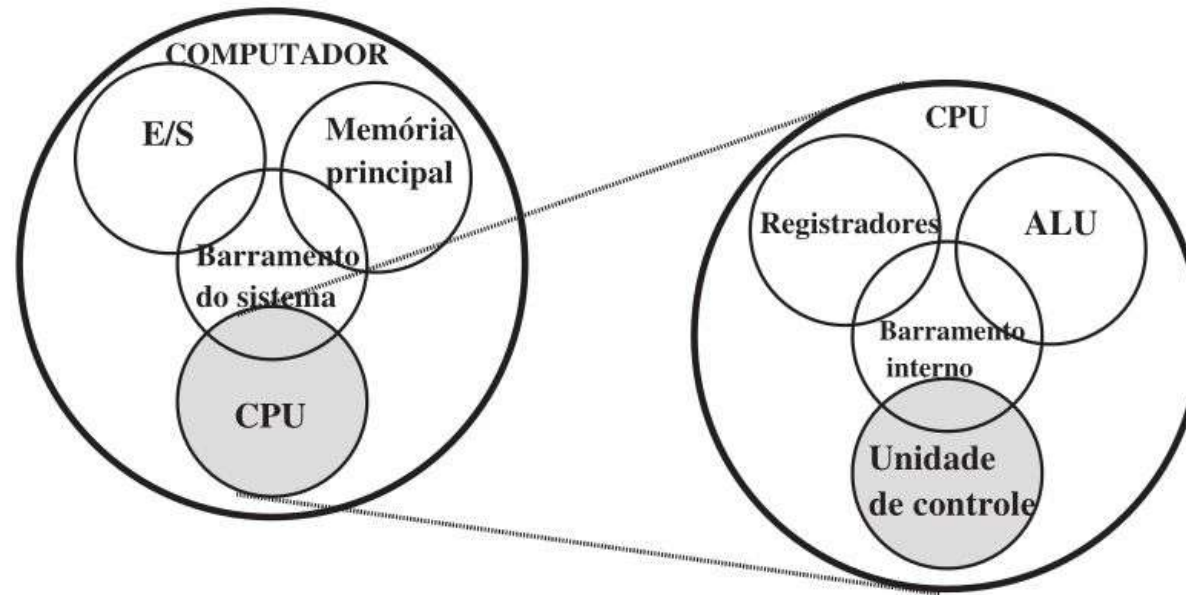
- O Computador possui quatro componentes principais
- **Unidade Central de Processamento (CPU)** → Controla a operação e realiza o processamento
- **Memória Principal** → Armazena dados
- **Entrada e Saída (E/S)** → Move dados entre o computador e seu ambiente externo
 - Interação entre usuário e computador
- **Sistema de Interconexão** → Interconecta todos os componentes
 - Ex: **Barramento do Sistema** → Conjunto de fios que interliga todos os componentes



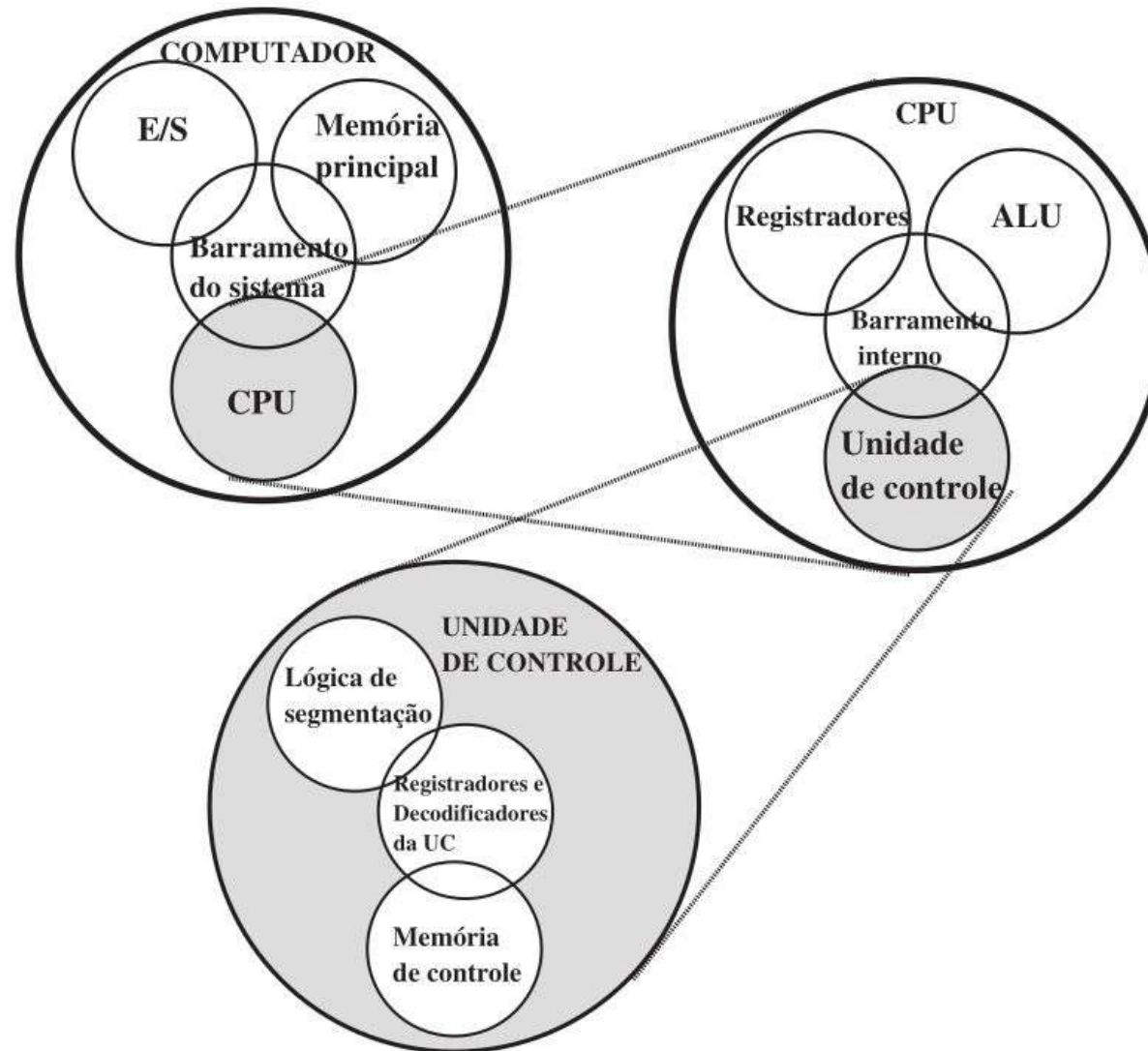
Estrutura de Alto Nível da CPU



Estrutura de Alto Nível da CPU

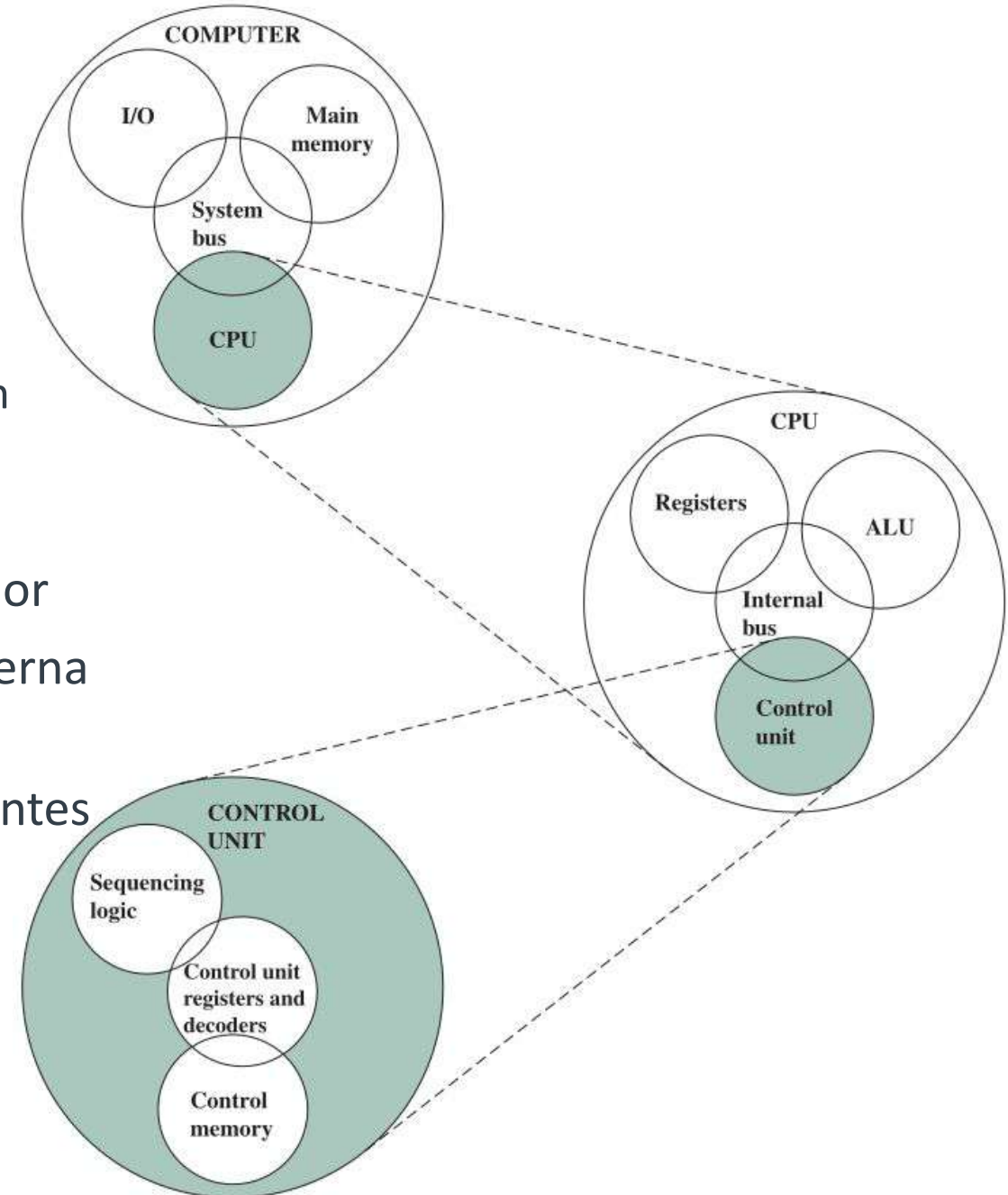


Estrutura de Alto Nível da CPU



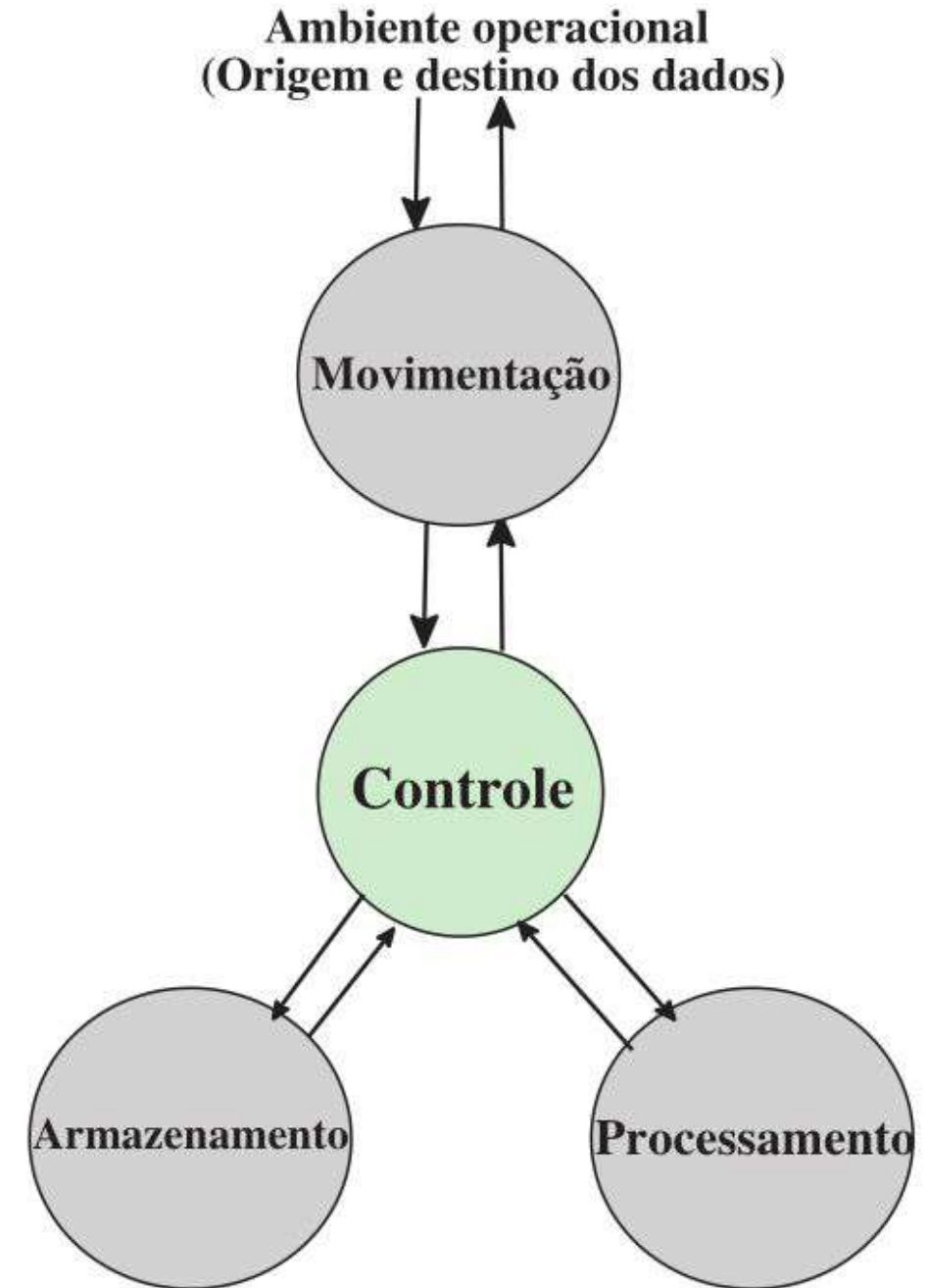
Estrutura de Alto Nível da CPU

- A CPU tem como principais componentes:
- **Unidade de Controle** → Controla a operação da CPU e, portanto, do computador
 - UC Microprogramada → Quebra cada instrução em várias microinstruções
- **Unidade Lógica e Aritmética (ULA)** → executa as funções de processamento de dados do computador
- **Registradores** → Proporciona armazenamento interna na CPU
- **Interconexão da CPU** → Interconecta os componentes da CPU
- Um processador pode ter vários núcleos (**cores**)
 - Maior poder de processamento (mais adiante)



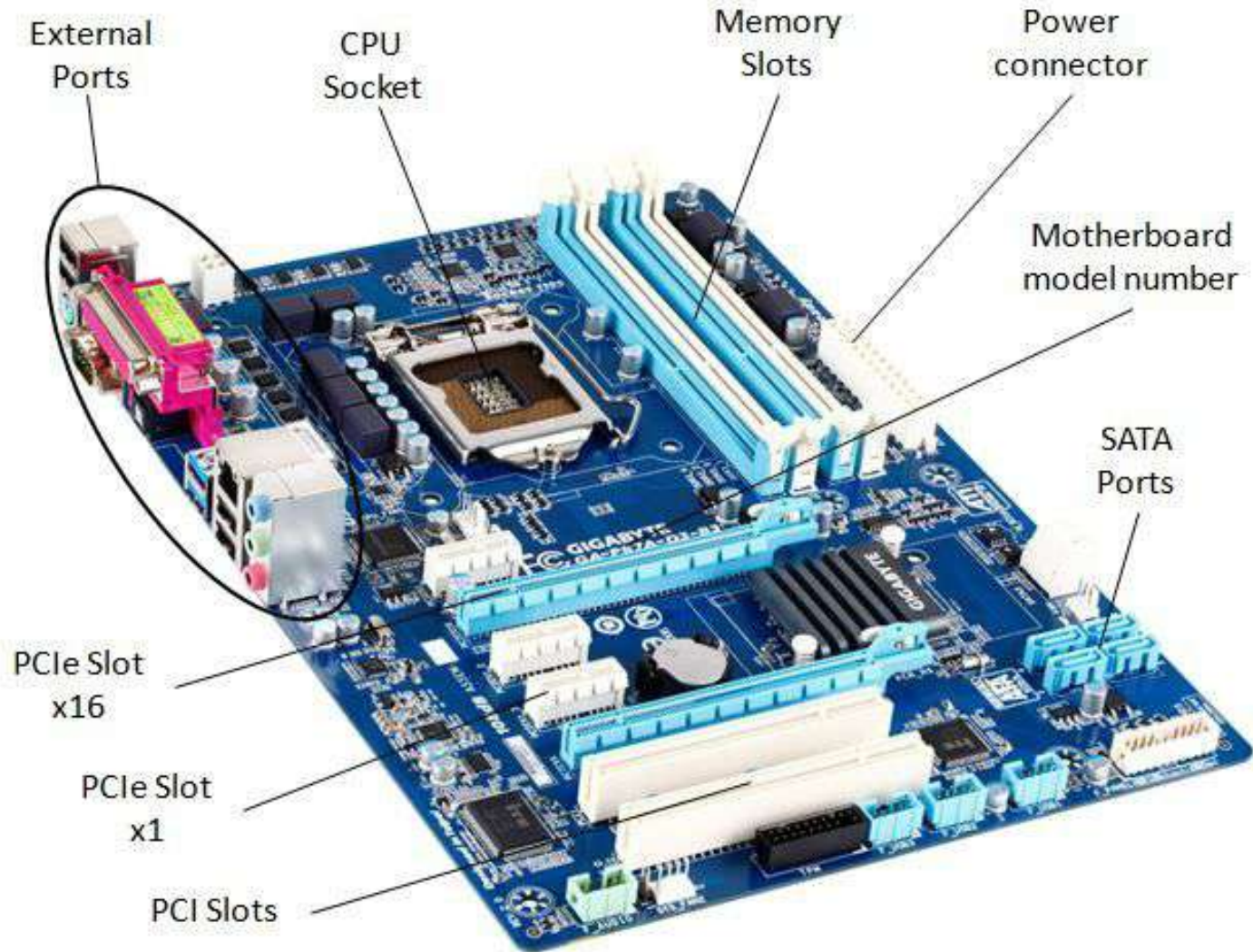
Funções Básicas do Computador

- **Processamento de dados:** Como obter a média aritmética de um *array*
- **Armazenamento de dados:** Guardar dados que está processando ou resultados
 - Armazenamento a curto prazo (memória RAM) \rightleftharpoons longo prazo (disco ou SSD)
 - Exemplo: Salvar (ou recuperar) dados na memória
- **Movimentação de dados:** Entre fonte e destino e.g. E/S
 - Exemplo: Obter dados inseridos no teclado
 - Movimentação a grandes distâncias \rightarrow Comunicação de dados
- **Controle:** Unidade de controle gerencia os recursos e orchestra o funcionamento
 - Exemplo: Sinais de controle gerado para a execução de uma instrução



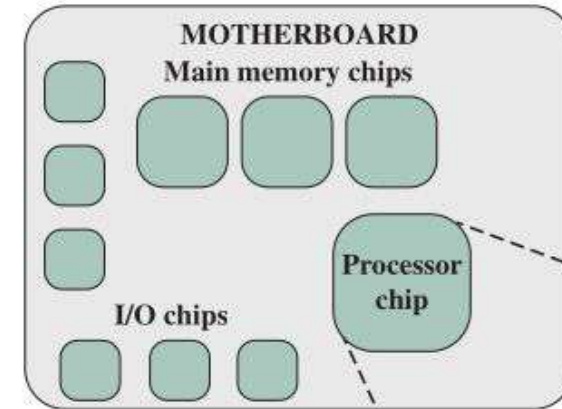
Placa-mãe do Computador

- **Placa de circuito impresso (PCB)** que interconecta todos os componentes do computador



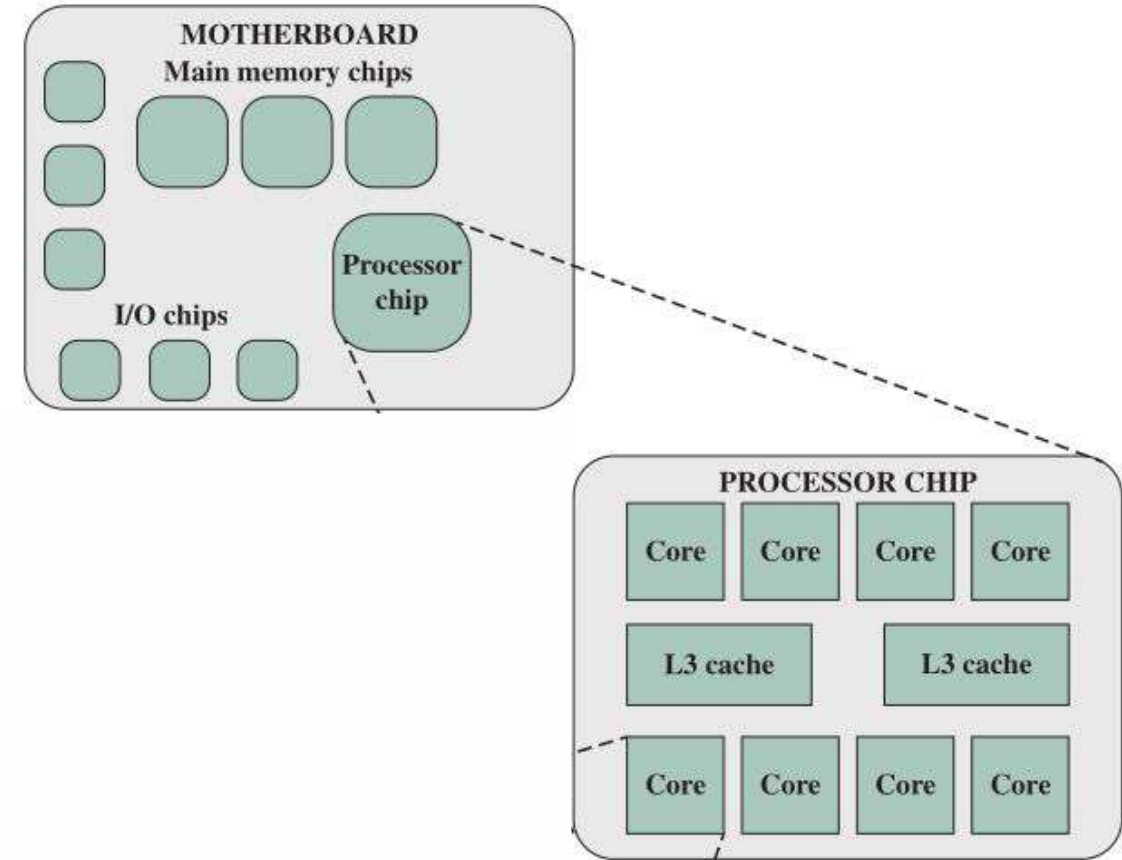
Estrutura de um Computador *Multicore*

- Computadores contemporâneos em geral têm **múltiplos processadores**
 - Quando no mesmo chip → Computador **Multicore**
 - Cada unidade de processamento (CPU) → **Core**
 - Um **processador** (chip de silícios) pode ter múltiplos núcleos (*cores*)
- Utiliza **memória cache** → Memória rápida no chip
 - Vários níveis → Dedicada ou compartilhada
 - **L1-I**: Cache ultra-rápida de instruções (dedicada)
 - **L1-D**: Cache ultra-rápida de dados (dedicada)
 - **L2**: Cache muito rápida (dedicada)
 - **L3**: Cache rápida (compartilhada)
- Exemplo: CPU com 8 núcleos e 3 níveis de cache



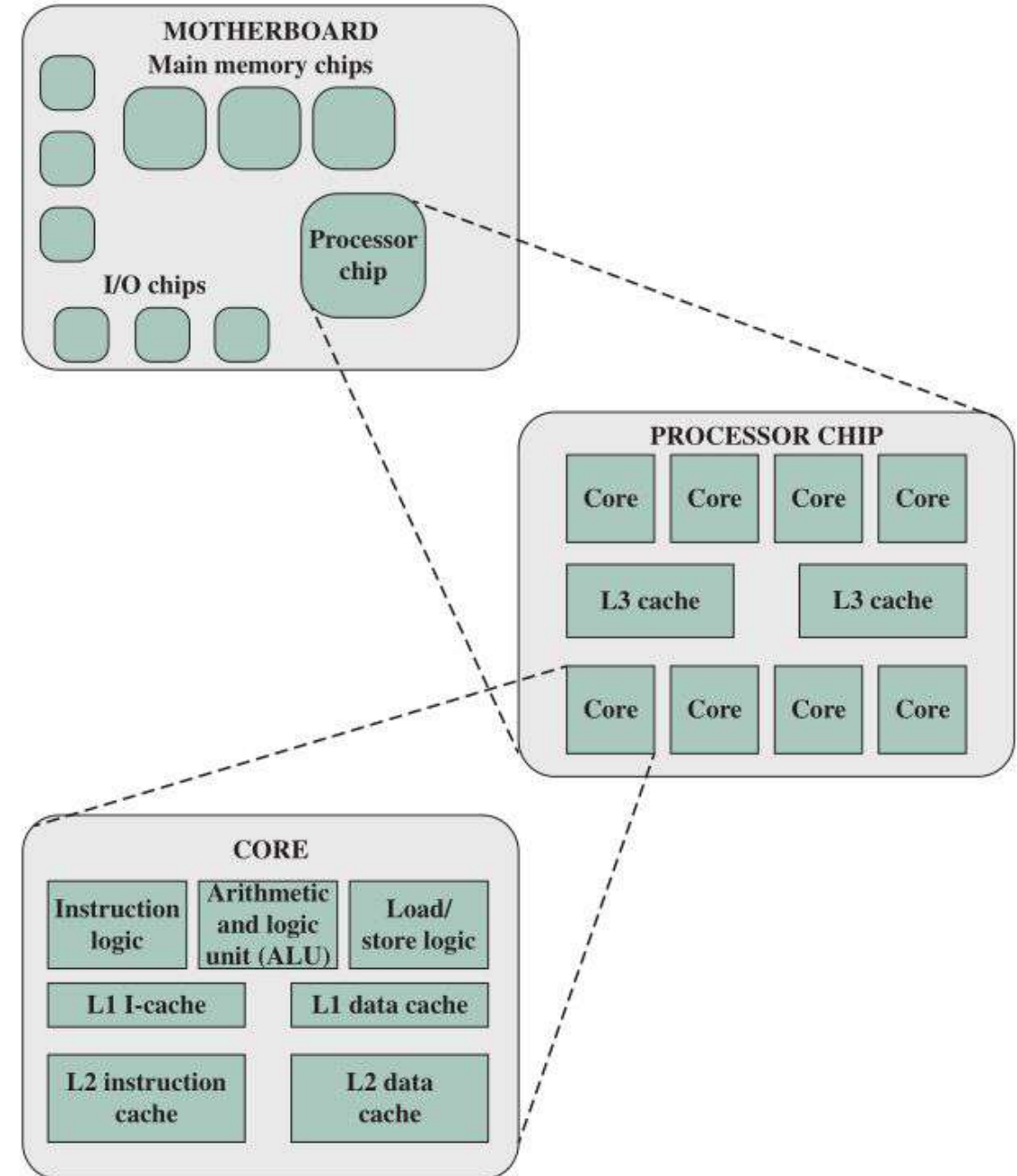
Estrutura de um Computador *Multicore*

- Computadores contemporâneos em geral têm **múltiplos processadores**
 - Quando no mesmo chip → Computador **Multicore**
 - Cada unidade de processamento (CPU) → **Core**
 - Um **processador** (chip de silícios) pode ter múltiplos núcleos (*cores*)
- Utiliza **memória cache** → Memória rápida no chip
 - Vários níveis → Dedicada ou compartilhada
 - **L1-I**: Cache ultra-rápida de instruções (dedicada)
 - **L1-D**: Cache ultra-rápida de dados (dedicada)
 - **L2**: Cache muito rápida (dedicada)
 - **L3**: Cache rápida (compartilhada)
- Exemplo: CPU com 8 núcleos e 3 níveis de cache



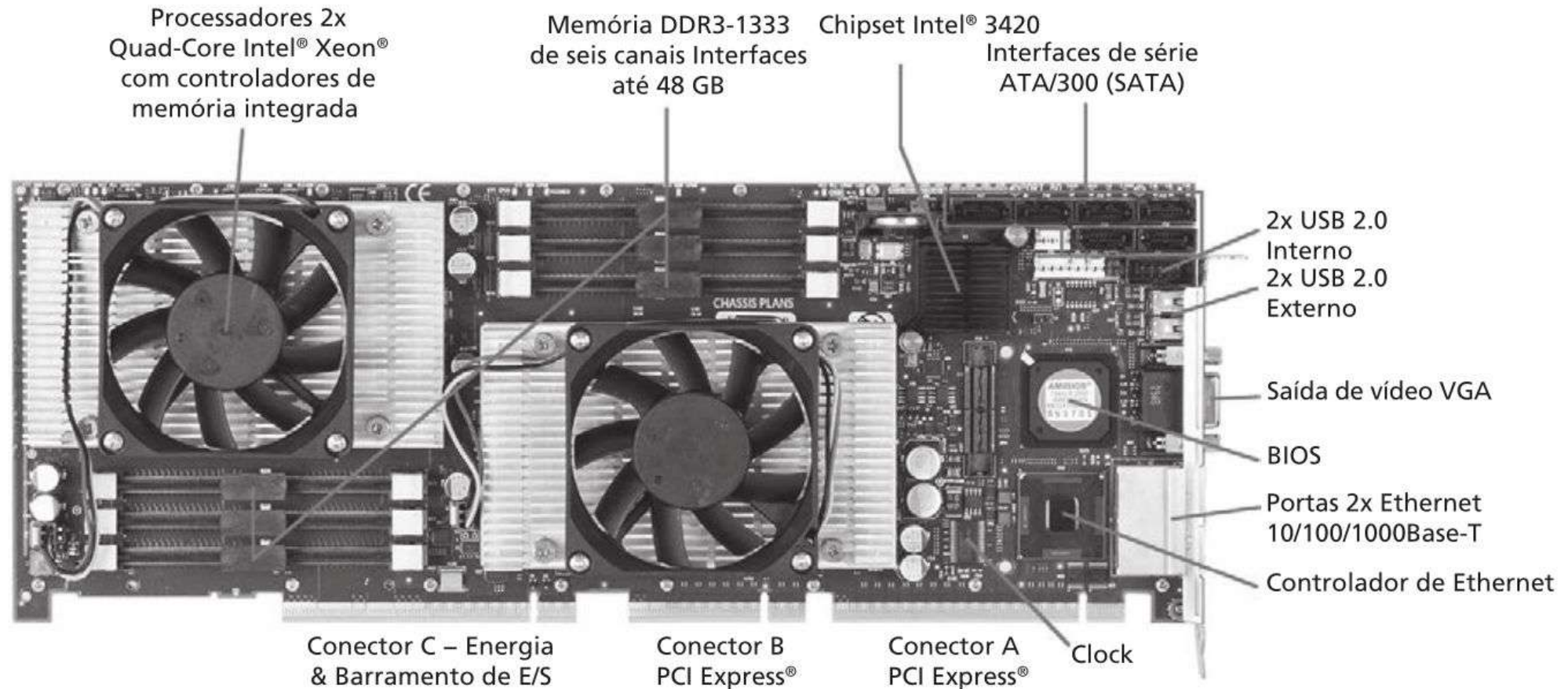
Estrutura de um Computador *Multicore*

- Computadores contemporâneos em geral têm **múltiplos processadores**
 - Quando no mesmo chip → Computador **Multicore**
 - Cada unidade de processamento (CPU) → **Core**
 - Um **processador** (chip de silícios) pode ter múltiplos núcleos (*cores*)
- Utiliza **memória cache** → Memória rápida no chip
 - Vários níveis → Dedicada ou compartilhada
 - **L1-I**: Cache ultra-rápida de instruções (dedicada)
 - **L1-D**: Cache ultra-rápida de dados (dedicada)
 - **L2**: Cache muito rápida (dedicada)
 - **L3**: Cache rápida (compartilhada)
- Exemplo: CPU com 8 núcleos e 3 níveis de cache



Exemplo de Computador Moderno

- Placa-mãe com dois processadores Intel Quad-Core Xeon



Arquitetura vs. Organização

- Dois termos usados no estudo de computadores
- **Arquitetura** de Computadores → Atributos do sistema visíveis ao programador¹
 - Possuem impacto direto sobre a execução lógica de um programa
 - Define o **conjunto de instruções**, **registradores**, modelo de **memória**, número de **bits**, etc...
 - Exemplo: *“Existe instrução de multiplicação?”*
- **Organização** de Computadores → Unidades operacionais (componentes) e suas interconexões
 - Atende às especificações da arquitetura ⇔ Transparente ao programador
 - Tecnologia da memória (DDR4 ou DDR5), interface de armazenamento (SATA ou NVMe), etc...
 - Define como é implementada a arquitetura ⇔ Microarquitetura
 - Exemplo: *“Existe uma unidade de multiplicação no hardware ou ela é feita pela adição repetitiva?”*
- Estudaremos ambos na disciplina → Começaremos pela organização

¹ Programador de baixo nível (e.g. assembly)

Arquitetura do Conjunto de Instruções (ISA)

- Define as características de uma **família de computadores** → Características semelhantes
 - Implementação física (organização) muda → Mas a arquitetura não!
- Exemplo: Arquitetura x86 → Processadores diferentes de uma mesma família
 - Retrocompatibilidade em *Software* ⇔ Mesmo programa funciona em todos!*



Exemplo: Mesma Arquitetura e Organização Diferente

- Comparação entre CPUs x86
 - Intel Core i3-10100
 - Intel Core i5-10400
- Mesmo programa → Performance Diferente
 - Mesma arquitetura → Mesmas instruções
 - Microarquitetura diferente → Implementação!
- Vantagem do i5 → **Maior performance**
- Vantagem do i3 → **Menor custo**
- *Trade-off* sempre será um fator na escolha

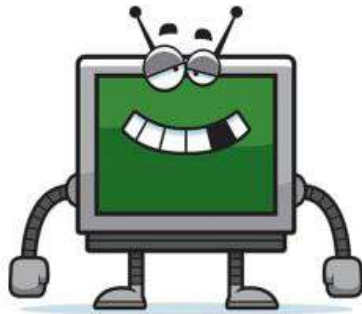
Por que Intel Core i5-10400 é melhor que Intel Core i3-10100?

- ✓ CPU 20.83% mais rápida ?
6 x 2.9 GHz vs 4 x 3.6 GHz
- ✓ 4 threads threads de CPU a mais ?
12 threads vs 8 threads
- ✓ 40% pontos a mais no PassMark ?
12113 vs 8652
- ✓ Cache L2 0.5 MB maior ?
1.5 MB vs 1 MB
- ✓ Possui um multiplicador desbloqueado ?
- ✓ Cache L3 6 MB maior ?
12 MB vs 6 MB
- ✓ Cache L1 128 KB maior ?
384 KB vs 256 KB
- ✓ Melhor para jogos
Segundo avaliações de usuários - 9.9 vs 9

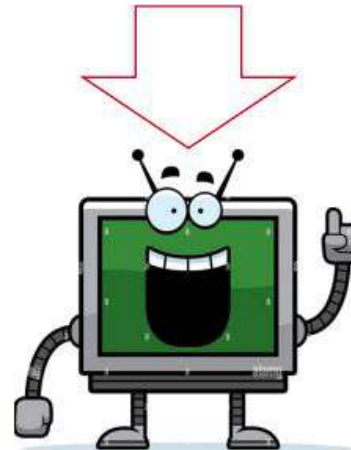
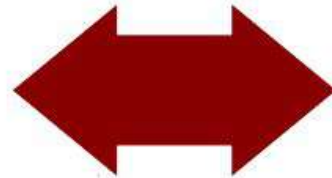
Execução de Programas

- Programas em linguagens de alto nível não entendidos pelo computador, mas pelo ser humano
 - Devem ser convertidos para instruções que o processador entende → **Código de Máquina**
 - Geralmente primeiro para linguagens intermediárias → **Assembly**

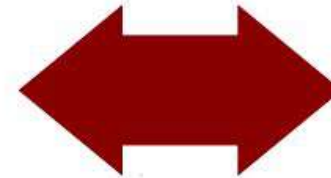
Computador hipotético com uma
linguagem de máquina intermediária



Linguagem de
máquina é **muito
simples**

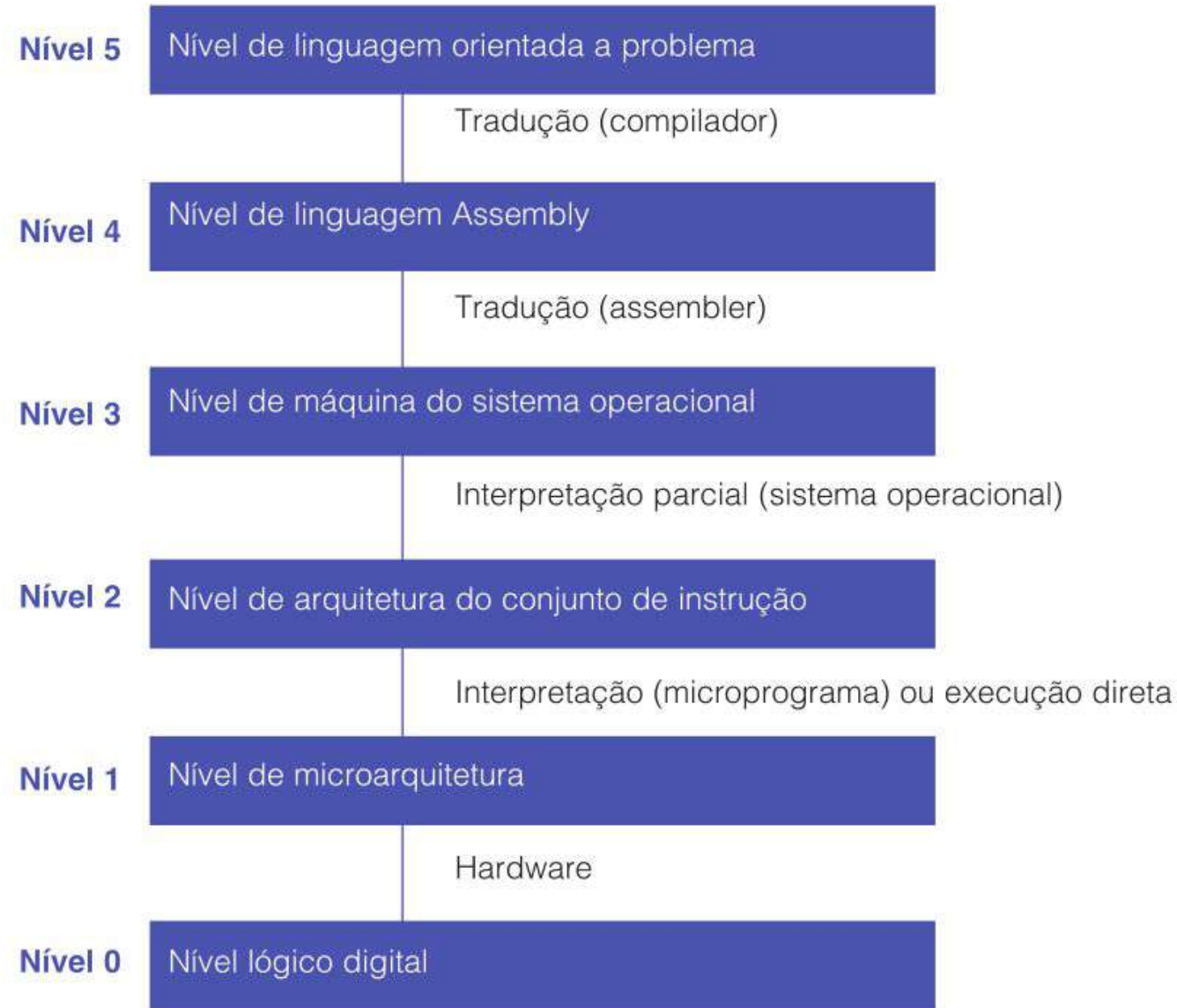


Solução
Criar uma linguagem
intermediária



Linguagem humana é
muito sofisticada

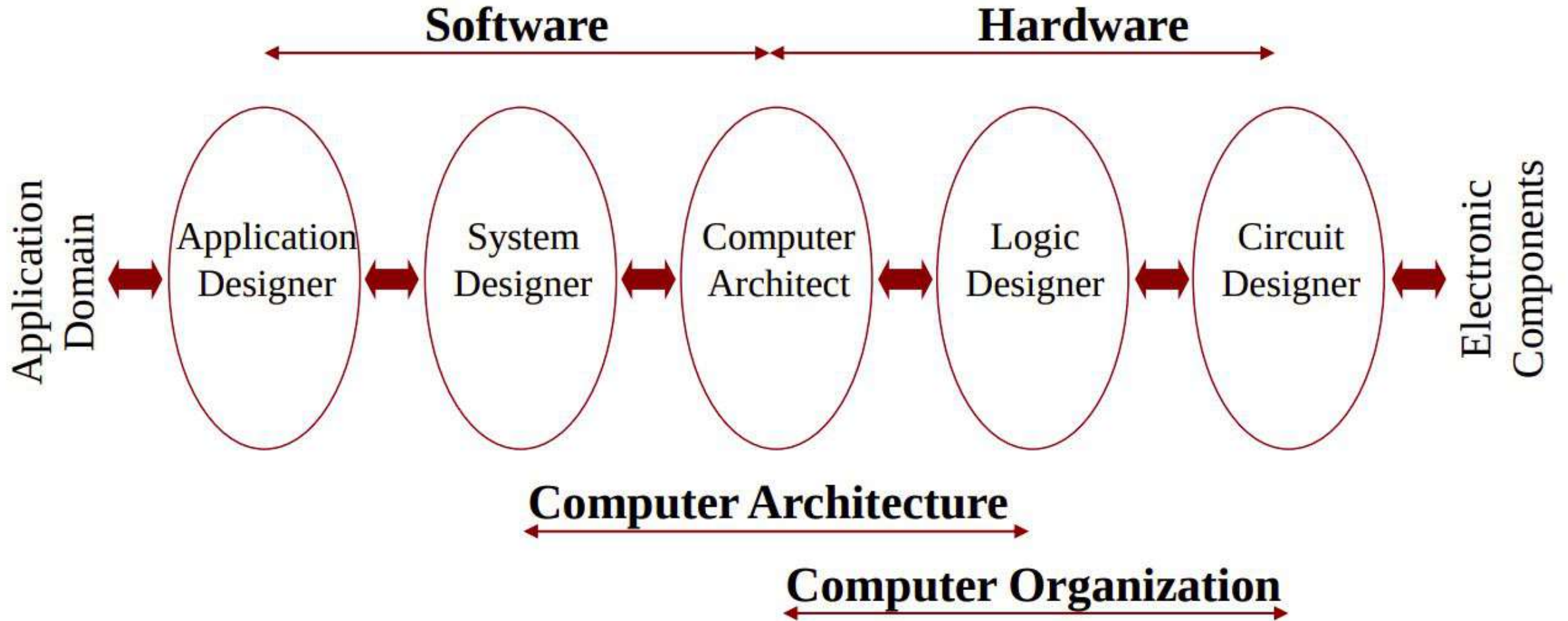
Máquina Multinível



Por que estudar arquitetura?

- Conhecer o *hardware* do sistema computacional é **fundamental** para o desenvolvedor
 - “Conhecer o campo de batalha é antecipar o movimento do inimigo. Conhecer o inimigo é antecipar a vitória”, Sun Tzu, Arte da Guerra
- **Disciplina fundamental** de todo curso de TI → Fundamentos do *hardware* do computador
 - Todo o sistema computacional depende do *hardware* → Essencial conhecê-lo
- Desenvolvedor de Sistemas Embarcados → **Obrigatório** conhecer a fundo o *hardware*
 - Qualquer programador que acesse o *hardware* diretamente irá usar isso (SO, drivers, etc...)
- Muitas outros setores de estudo usam e abusam destes conhecimentos:
 - Sistemas Operacionais, Sistemas Distribuídos, Sistemas Embarcados, Compiladores, e muitos outros
 - **Projetistas** de processadores precisam ter domínio total sobre esses assuntos
- Permite uma melhor escolha do *hardware* para cada necessidade de **desempenho**
 - É a base da avaliação de desempenho e *benchmark*
- Essencial para **otimização**, programação de **alta performance** e processamento **paralelo**

Campos de Atuação da Engenharia de Computação



Como escolher um Computador?

- Para cada cenário → Uma configuração específica
- O que levar em consideração?
 - *Trade-off Custo-Benefício* → “Se tirar de um lado, sobra do outro”
- **Exemplo 1: Desktop para trabalho** → Filosofia “*pode precisar depois*”
 - “Precisa de muitas aplicações abertas ao mesmo tempo?” → Memória RAM
 - “Realiza diversas tarefas ao mesmo tempo?” → Núcleos da CPU
 - “Precisa renderizar vídeos ou jogos?” → Memória de Vídeo
- **Exemplo 2: Sistema Embarcado** → Filosofia “*se não usa, não precisa*”
 - Otimização no **consumo** de energia e no **preço**
 - Só possui a memória que precisa e.g. programa ocupa 16 kB, então possui 16 kB de memória flash

Exemplos de Hardwares para Várias Aplicações

- Não devemos olhar só para processador e memória → Cada propósito tem uma necessidade



Desktop

CPU: 12x i7 12^a Gen./
16MB Cache L3 / 2.2GHz
GPU GDDR5: 4GB
RAM DDR4: 16 GB
SSD: 1 TB



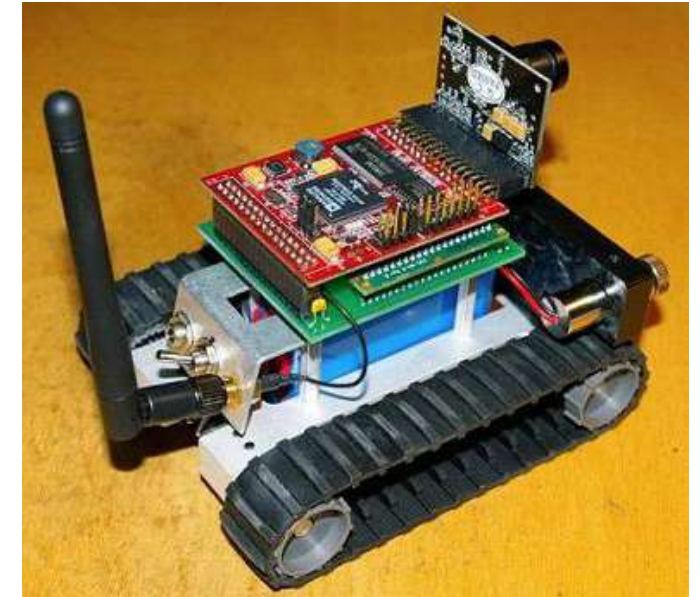
Smartphone

CPU: 2x 4.47 GHz
+ 6x 3.53 GHz
RAM LPDDR4: 12 GB
Flash: 512 GB



PlayStation 5

CPU: AMD 8x 3.5 GHz
GDDR6 SRAM: 16 GB
SSD: 825 GB



Carrinho Robô

CPU: 1x ARM Cortex-M
/ 48 MHz
SRAM: 64 kB
Flash: 128 kB



Introdução à Disciplina

Metodologia da Disciplina

Metodologia da Disciplina

- **Aulas:** Conteúdos mais importantes serão passados em sala (conteúdo muito extenso)
 - Visão geral sobre Arquitetura → Detalhes podem ser encontrados nos livros
- Disciplina dividida em **três partes** → **Avaliações** ao final de cada
 - **Parte 01** → Introdução ao Sistema Computacional (*prova escrita*)
 - **Parte 02** → Sistemas de Memória e E/S (*prova escrita*)
 - **Parte 03** → Unidade Central de Processamento (*trabalho final*)
- **Laboratórios:** Atividades práticas relacionadas aos conteúdos realizadas em grupos (4 pessoas)
 - Conteúdos geralmente muito teóricos → Aprender fazendo e com supervisão
 - Mini-laboratórios (**MLab**) ao final de algumas aulas → Em sala!

$$MF = \frac{AP_1 + AP_2 + TF + Labs}{4}$$

Assuntos da Disciplina

- **Parte 01** → Introdução ao Sistema Computacional
 - Sistemas Numéricos
 - Aritmética Binária
 - Visão de Alto Nível do Computador
- **Parte 02** → Sistemas de Memória e E/S
 - Memória Cache
 - Memória Interna
 - Memória Externa
 - Entrada e Saída
- **Parte 03** → Unidade Central de Processamento
 - Conjunto de Instruções e Linguagem de Montagem
 - Estrutura Interna e *Pipeline* de Instruções
 - Processadores Superescalares e RISC
 - Unidade de Controle

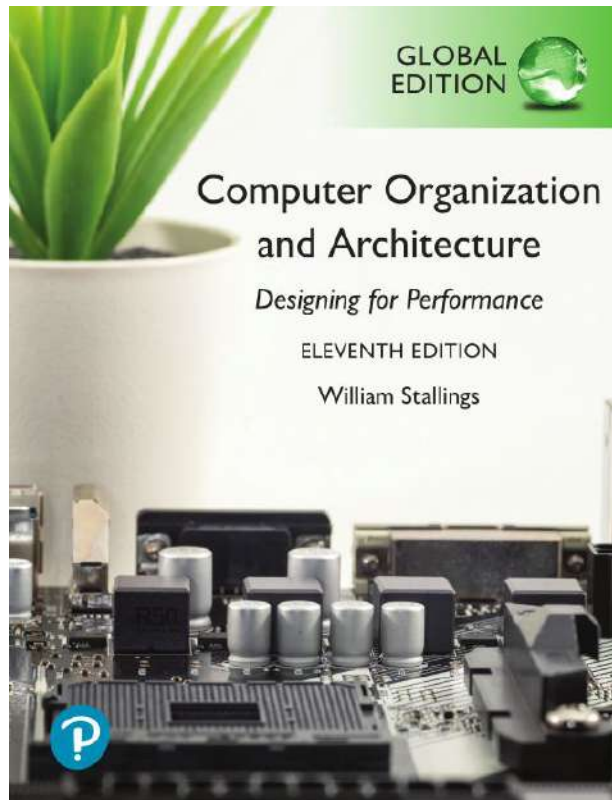
Nossos Objetivos

- “Fornecer aos alunos, através do estudo dos **componentes de hardware**, os conhecimentos básicos necessários para **entender o funcionamento e operação dos computadores**, capacitando-os no acompanhamento da evolução tecnológica na área de informática”
- Fornecer ao aluno informações sobre o funcionamento e a organização interna dos principais sistemas de computação
- Apresentar os conceitos básicos do funcionamento de **memória, unidade central de processamento, barramentos e dispositivos de entrada/saída**
- Apresentar os conceitos básicos sobre **conjunto de instruções** e linguagem **Assembly**
- Habilitar o aluno a compreender o *hardware* como **elemento facilitador/limitador do desempenho da aplicação**
- Permitir uma **melhor escolha do hardware** para cada necessidade de desempenho

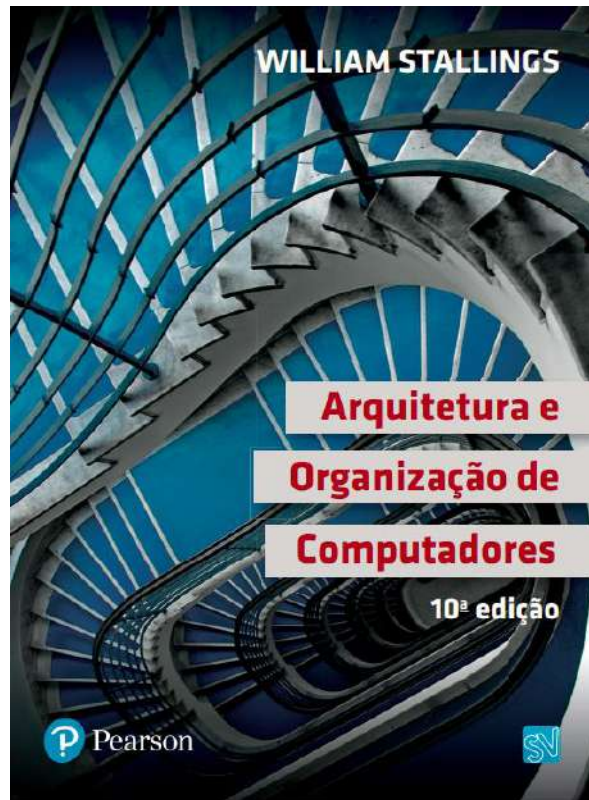
Trabalho Final

- Programação de um **Simulador de Processador** → CPU simplificada de 16 bits
 - Equipe pode optar em projetar um processador com **circuitos lógicos**
- Grupos de **4 pessoas** → Cada grupo deverá apresentar ao professor até o dia 16/01
 - Podem apresentar antes sem problemas → Alinhar com o professor
 - **Aviso:** Não deixem pra apresentar no último dia → Se não der tempo, a equipe ficará **sem a nota**
 - Serão aceitos grupos com menos/mais pessoas **apenas em casos excepcionais**
- Nota de acordo com os casos de teste → 5 casos de teste automáticos
 - **Caso 1** → Instruções simples
 - **Caso 2** → Instruções de salto
 - **Caso 3** → Operações com pilha
 - **Caso 4** → Chamada de função
 - **Caso 5** → Sistema de entrada e saída

Livros da Disciplina



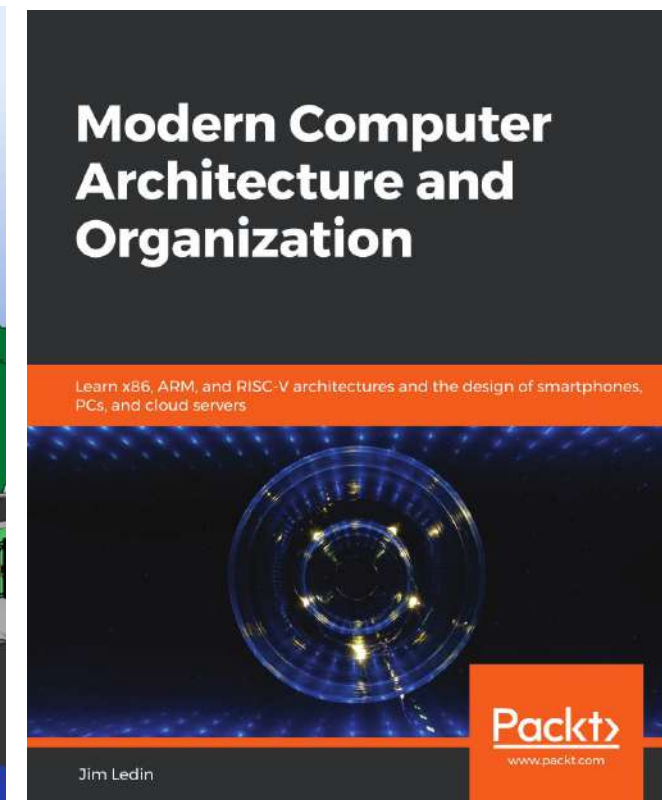
Melhor versão do Stallings
(Inglês)



Stallings mais atualizado
da biblioteca (Português)

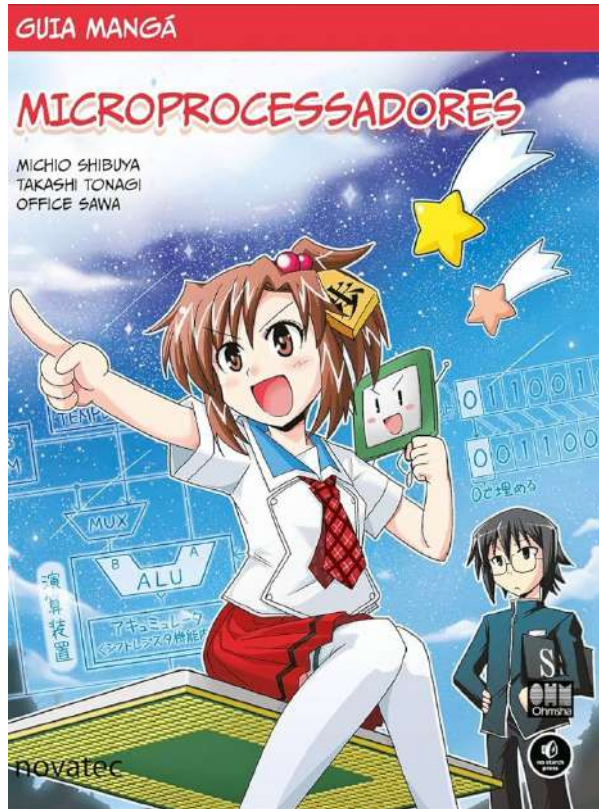


Ótimo complemento, para
mais detalhes (Português)



Abordagem moderna com
mais aplicações (Inglês)

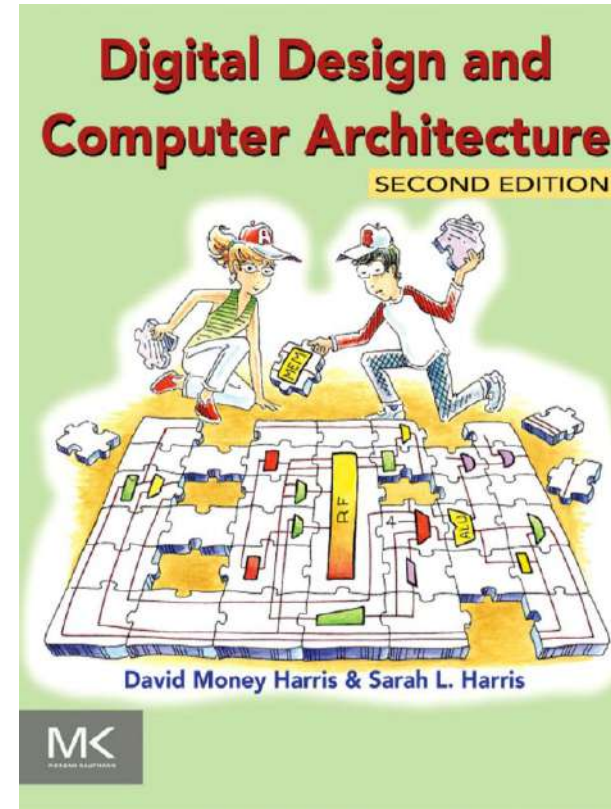
Outros Livros Interessantes



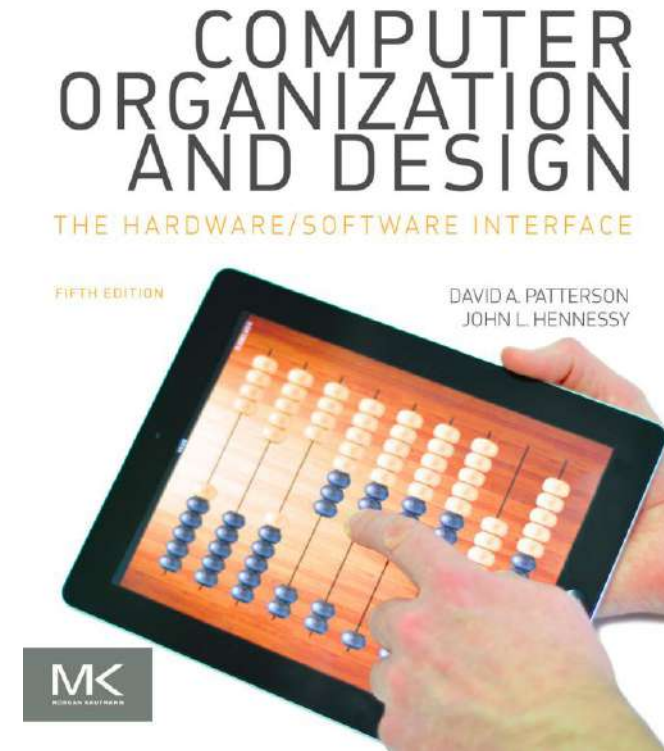
Abordagem descontraída
(Português)



Metodologia simples e
aplicada (Português)



Foca no projeto de
sistemas (Inglês)



Trata da análise e projeto de
computadores (Inglês)



Introdução à Disciplina

Conclusão

Resumo da Aula

- **Computador** → *Hardware* que executa *software*
 - Composto por **CPU, memória, sistema de E/S e interconexões**
 - CPU é composta por **unidade de controle, ULA, registradores e interconexão interna**
 - **Instruções** → Operações unitárias que a CPU consegue realizar
- **Computador Multicore** → Múltiplos processadores (núcleos, ou *core*)
- **Arquitetura** → Como o computador deve se comportar
- **Organização** → Como o computador é construído
 - Por vezes chamada de **microarquitetura** (implementação física da arquitetura)
- Computador tem **muitas camadas** → *Hardware* é a base
 - Essencial entender a arquitetura e a organização dos computadores

Conclusão

- Nessa Aula:
 - Introdução à Disciplina
- Bibliografia Principal:
 - Arquitetura e Organização de Computadores; Stallings, W.; 10ª Edição (Capítulo 1)
- Próxima Aula:
 - Conceitos Básicos e Evolução dos Computadores