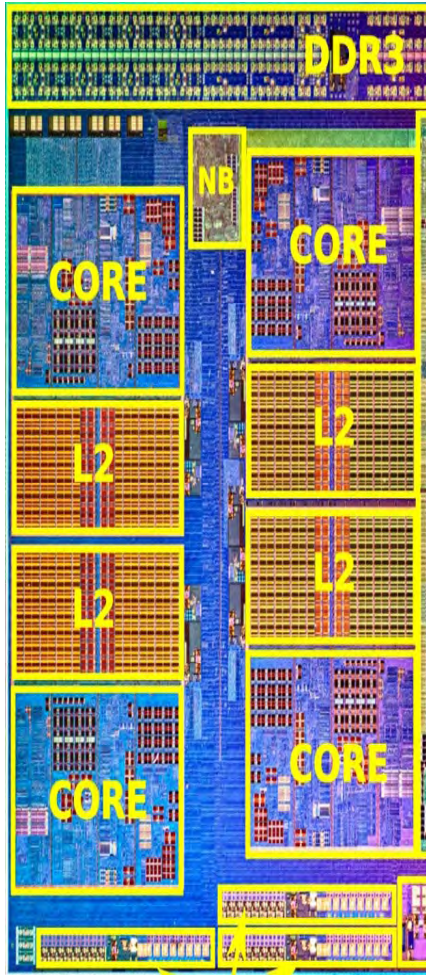


Arquitetura e Organização de Computadores



Fonte: <http://www.techspot.com/article/904-history-of-the-personal-computer-part-5/>

Fundamentos

Luciano de Oliveira Neris

luciano@dc.ufscar.br

Adaptado de slides do prof. Marcio Merino Fernandes

Departamento de Computação
Universidade Federal de São Carlos



Sistemas

- **Sistema:** conjunto de componentes que trabalham de maneira coordenada para realizar alguma atividade.
- Principais características:
 - Um sistema é composto por partes.
 - Todas as partes de um sistema devem se relacionar de forma direta ou indireta.
 - Um sistema pode abrigar outro sistema.

Abordagem "sistêmica": Facilita a compreensão do funcionamento como um todo e a participação de cada uma das partes envolvidas

Sistemas de Computação

- **Sistema de Computação:** realiza algum tipo de processamento de informações de entrada para gerar algum tipo de saída
 - O processamento (computação) é especificado através de um conjunto de **instruções** (programa) que definem o que, quando e como deve ser feito
 - *Processar informação significa, abstratamente, transformar elementos de entrada com o objetivo de produzir elementos na saída, de uma forma coerente, desejável e previsível.*
 - *Composto por subsistemas*

Sistemas de Computação

- **Sistema de Computação:** estrutura dividida em 3 componentes:
 - *Hardware*
 - *Software*
 - *Dados*

Sistemas de Computação

- Onipresentes: estão em todo lugar
 - Uso geral: servidores, desktops, laptops, PDAs, etc.
 - Uso específico: supercomputadores, vídeo games, etc.
 - Embarcados: carros, impressoras, DVDs, telefone celular, equipamentos industriais, equipamentos médicos, etc.

Sistemas de Computação

- Características Diferenciais
 - Velocidade
 - Custo
 - Facilidade de uso, interface, suporte
 - Escalabilidade
 - Consumo de Energia
 - Dispositivos portáteis
 - Dispositivos de grande porte (datacenters, supercomputadores)

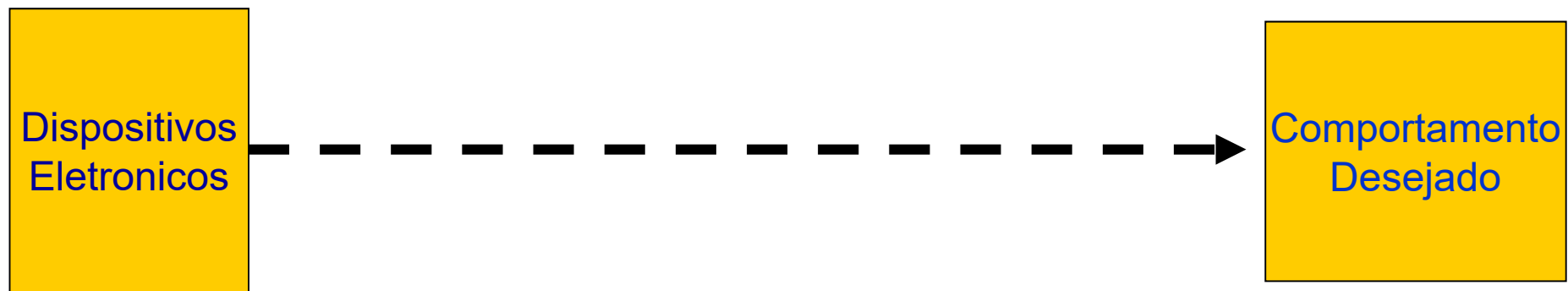
Sistemas de Computação

- Três questões básicas associadas com sistemas de computação:
 - Para que eles são utilizados?
 - Como são implementados?
 - O que eles podem fazer, e o que não podem?

Sistemas de Computação

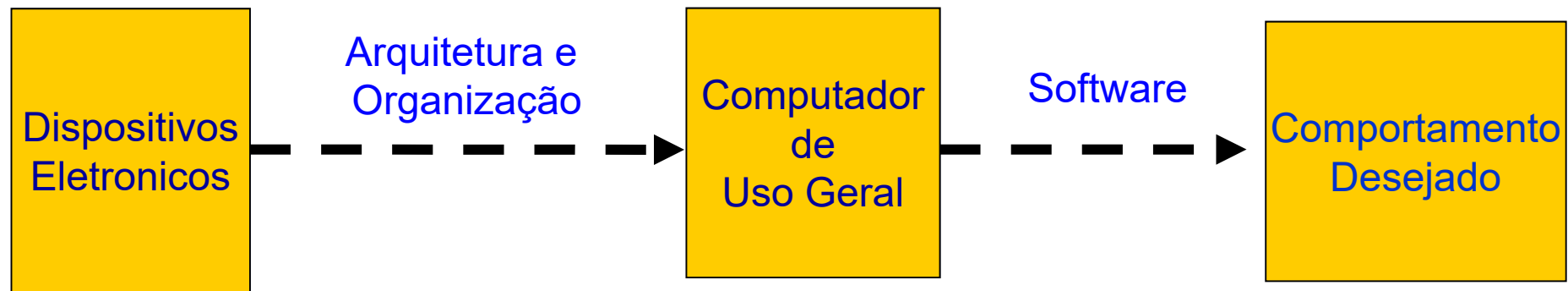
- Entrada:
 - Como consigo interagir com o sistema?
 - Qual a linguagem que ele entende?
 - Quanto de conhecimento preciso ter para "inserir coisas" no sistema?
 - Qual o formato ou modalidade devo usar?
 - ...
- Saída:
 - Qual a linguagem que o usuário entende?
 - Qual o resultado esperado pelo usuário?
 - Quanto de conhecimento preciso para entender os resultados produzidos?
 - Qual o formato, linguagem e modalidade a ser utilizada?
 - ...

Sistemas de Computação



Existe uma longa distancia entre um determinado comportamento desejado, e um conjunto de dispositivos eletrônicos (desorganizados).

Sistemas de Computação



Um computador de uso geral pode ser visto como o ponto central de uma ponte para se caminhar de um conjunto de dispositivos eletrônicos até a obtenção de um comportamento desejado (função).

Arquitetura x Organização

- **Arquitetura** refere-se aos **atributos** que são **visíveis** para o **programador**, ou seja, os atributos que tem impacto direto na execução do programa.
- Atributos:
 - Conjunto de instruções
 - Número de bits para representar diferentes categorias de dados (e.g., números e caracteres).
 - Mecanismos de E/S

Arquitetura x Organização

- **Organização** diz respeito às unidades operacionais e suas interconexões que implementam as especificações de sua arquitetura, ou seja, **como as características da arquitetura serão implementadas.**
- Atributos:
 - Sinais de controle
 - Tecnologia de memória, tecnologia de transistores etc.

Detalhes do hardware que são transparentes ao programador fazem parte da organização do sistema.

Arquitetura x Organização

- Especificar se um computador deve ou não ter uma instrução de multiplicação constitui uma decisão de projeto de **Arquitetura**
- Definir se essa instrução será implementada por uma unidade específica de multiplicação ou por um mecanismo que utiliza repetidamente sua unidade de soma é uma decisão de **Organização**

Sistemas de Computação

- O estudo de sistemas (incluindo os computacionais) são extremamente complexos se forem analisados detalhadamente.
- Utilizando diferentes níveis de abstração é possível reduzir a complexidade da análise de sistemas pois omite detalhes
- Abstração: distinção entre as propriedades externas de um componente e os detalhes internos de sua construção.

Sistemas de Computação

- *O computador pode ser visto por várias perspectivas ou níveis, do mais alto nível, "do usuário", até o mais baixo nível, "de transistores". Cada um desses níveis representa uma abstração do computador;*
- *Uma das razões para o grande sucesso dos computadores digitais é o grau de separação desses níveis, ou seja, a independência entre os níveis.*

Sistemas de Computação

- *Sistemas em que o hardware é dedicado para uma aplicação particular não são flexíveis*
- *Sistemas de propósito geral podem executar diferentes tarefas através dos sinais de controle*
 - *Ao invés de se reprogramar o hardware, muda-se o conjunto de sinais de controle*

Arquitetura de Computadores

- Dispositivos Universais de Computação
 - Dados tempo e memória suficiente, todos os computadores são capazes de executar as mesmas tarefas;
 - Tese de Turing: toda computação pode ser executada por uma máquina de Turing (um dispositivo computacional teoricamente universal);
- Transformação de um Problema
 - O objetivo final é transformar um problema, descrito em linguagem natural, em elétrons circulando através de um circuito!
 - Isto é a essência da Ciência e Engenharia da Computação.

Aspectos teóricos e práticos :
SW: 80% , HW: 20%

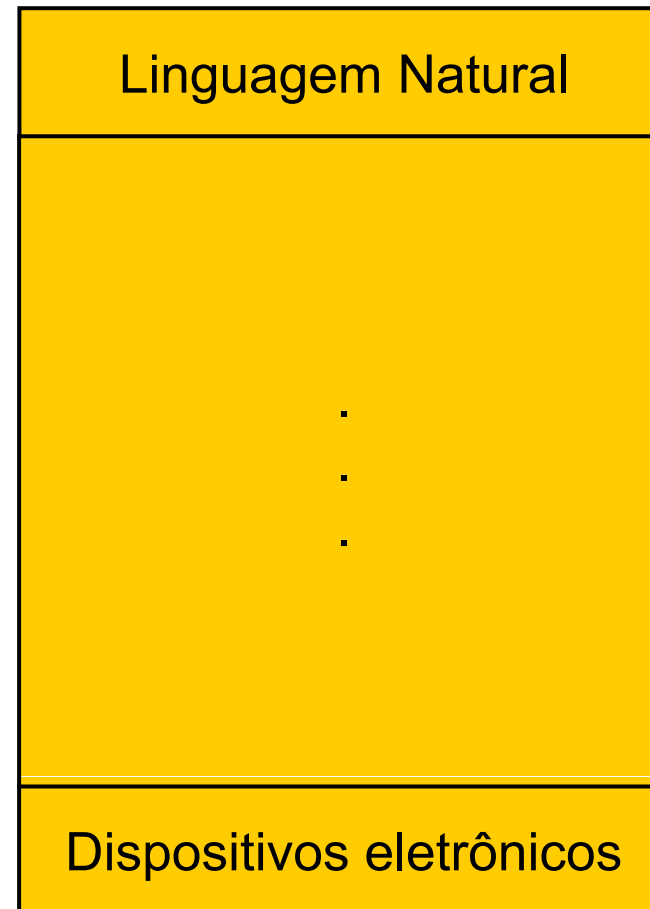
Aspectos práticos :
SW: 50% , HW: 50%

Transformação de um Problema

Comportamento Desejado:
Aplicação



Matéria Prima:
dispositivos
eletrônicos

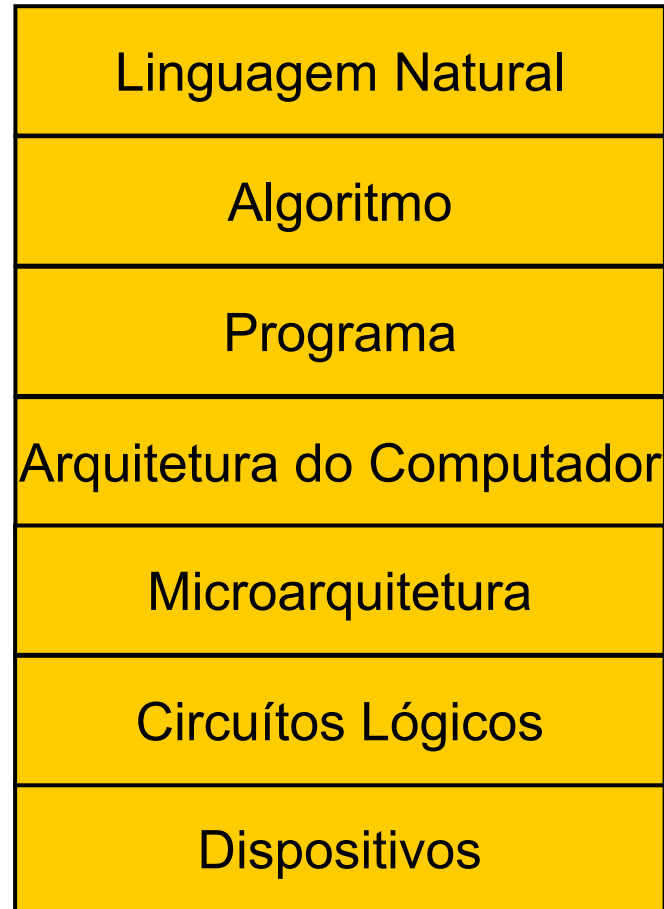


Transformação de um Problema

Comportamento Desejado:
Aplicação



Matéria Prima:
dispositivos
eletrônicos



Níveis de Descrição

- Esses níveis não correspondem necessariamente a componentes individuais, porém determinam uma *série de interfaces padronizadas*.
- Interfaces padronizadas permitem:
 - Portabilidade
 - Uso de Software/Hardware desenvolvido por terceiros
 - Uso mais amplo

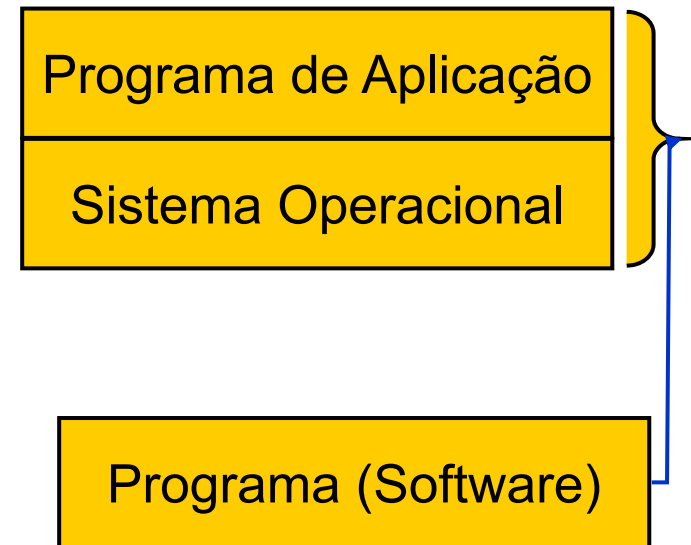
| |
|---------------------------|
| Linguagem Natural |
| Algoritmo |
| Programa |
| Arquitetura do Computador |
| Microarquitetura |
| Circuitos Logicos |
| Dispositivos |

Nível de Programa

- Programa
 - Sequência de passos
 - Para cada passo:
 - uma operação lógica ou aritmética é realizada
 - um conjunto diferente de sinais deve ser fornecido
 - Para cada operação, um código único é fornecido:
 - Exemplo: 10 (ADD), 11 (MOVE), etc
 - Função da Unidade de Controle:
 - Interpretar o código e gerar os sinais de controle que executarão a instrução requerida

Nível de Programa

- A maioria dos computadores executa um programa de gerenciamento, chamado sistema operacional (S.O.).
- Os programas de aplicação interagem com a arquitetura da máquina através do S.O.



Exemplo:

| |
|--------------|
| Estes Slides |
| PowerPoint |
| Windows 10 |

Dados

Programa de Aplicação

S.O.

Nível de Programa

- SO gerencia os recursos da máquina durante a execução dos programas
 - Operações de Entrada/Saída (E/S), "carga" do programa na memória, exceções, etc.
 - Gerente dos recursos, escondendo o acesso direto ao hardware dos usuários
 - Também: multiprocessamento, gerência de arquivos, processamento distribuído, ...

Nível de Máquina

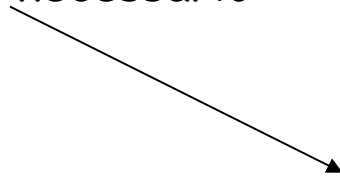
- **Arquitetura do Computador**
 - Especificação formal de todas as funções que uma determinada máquina pode executar. Essas funções são conhecidas como ISA (Instruction Set Architecture).
- **Microarquitetura**
 - Implementação da ISA em um microprocessador, ou seja, a forma como as especificações da ISA ocorrerão (registradores, ULA).
- **Circuitos Lógicos**
 - Cada elemento da microarquitetura é composto por circuitos lógicos simples (portas)
- **Dispositivos Eletrônicos (devices)**
 - Cada circuito lógico é construído com dispositivos eletrônicos, como transistores CMOS (*complementary metal-oxide-semiconductor*)

Nível de Máquina

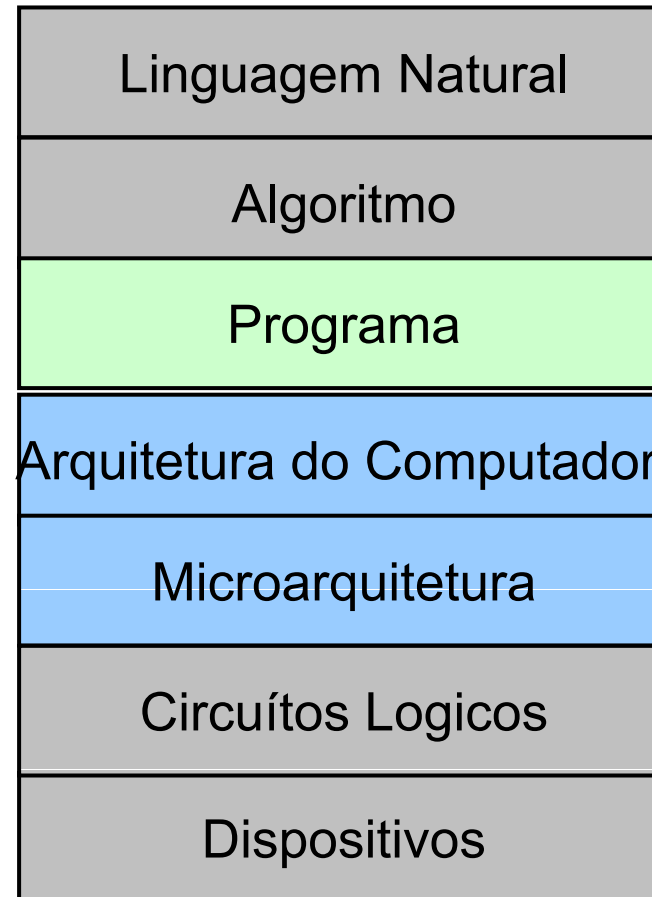
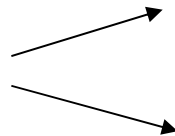
- Os números binários são base da teoria computacional
 - 100011000001 (bits): "Linguagem" do computador
 - 1. Primórdios: uso da linguagem nativa em binário.
 - 2. Linguagem de Montagem (Assembly)
 - Montador (Assembler): traduz uma versão simbólica das instruções para sua representação binária na arquitetura
 - `add A, B` -> montador -> 100011000001
 - 3. Linguagem de Programação de alto-nível
 - Compilador: traduz instruções de alto-nível para instruções binárias diretamente ou via um montador
 - `A + B` -> compilador -> `add A, B` -> montador -> 100011000001

Este Curso

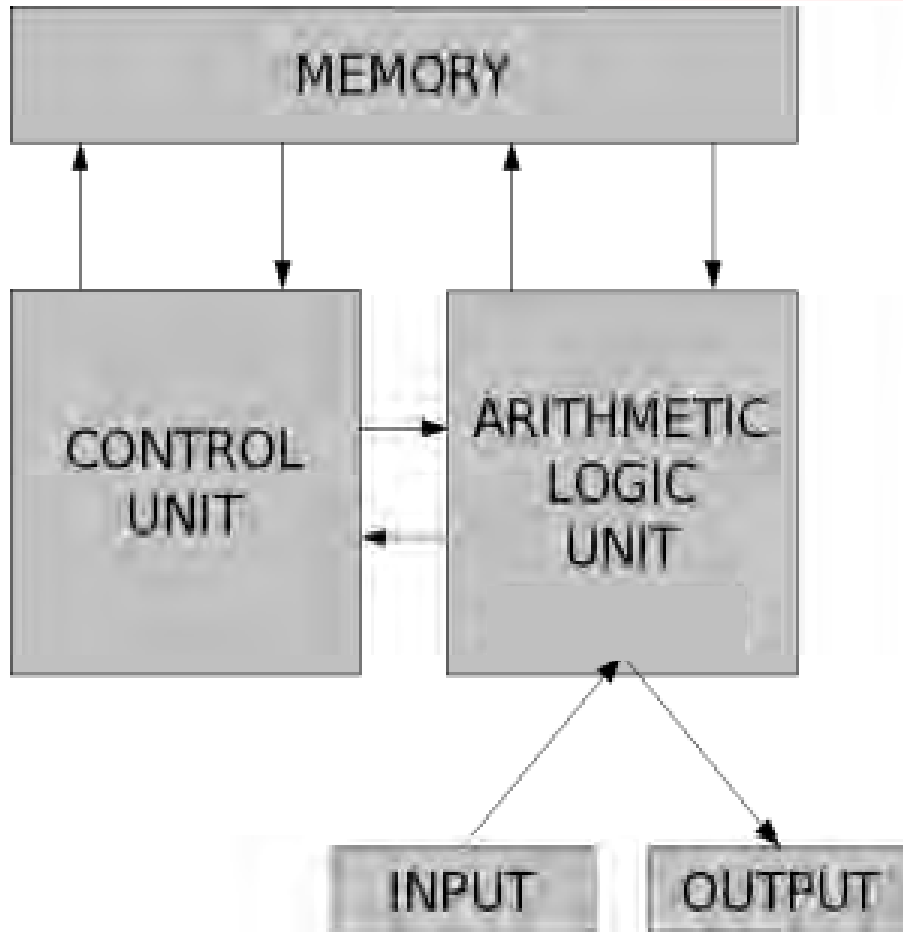
Foco secundário, mas necessário



Foco principal



Computador de von Neumann



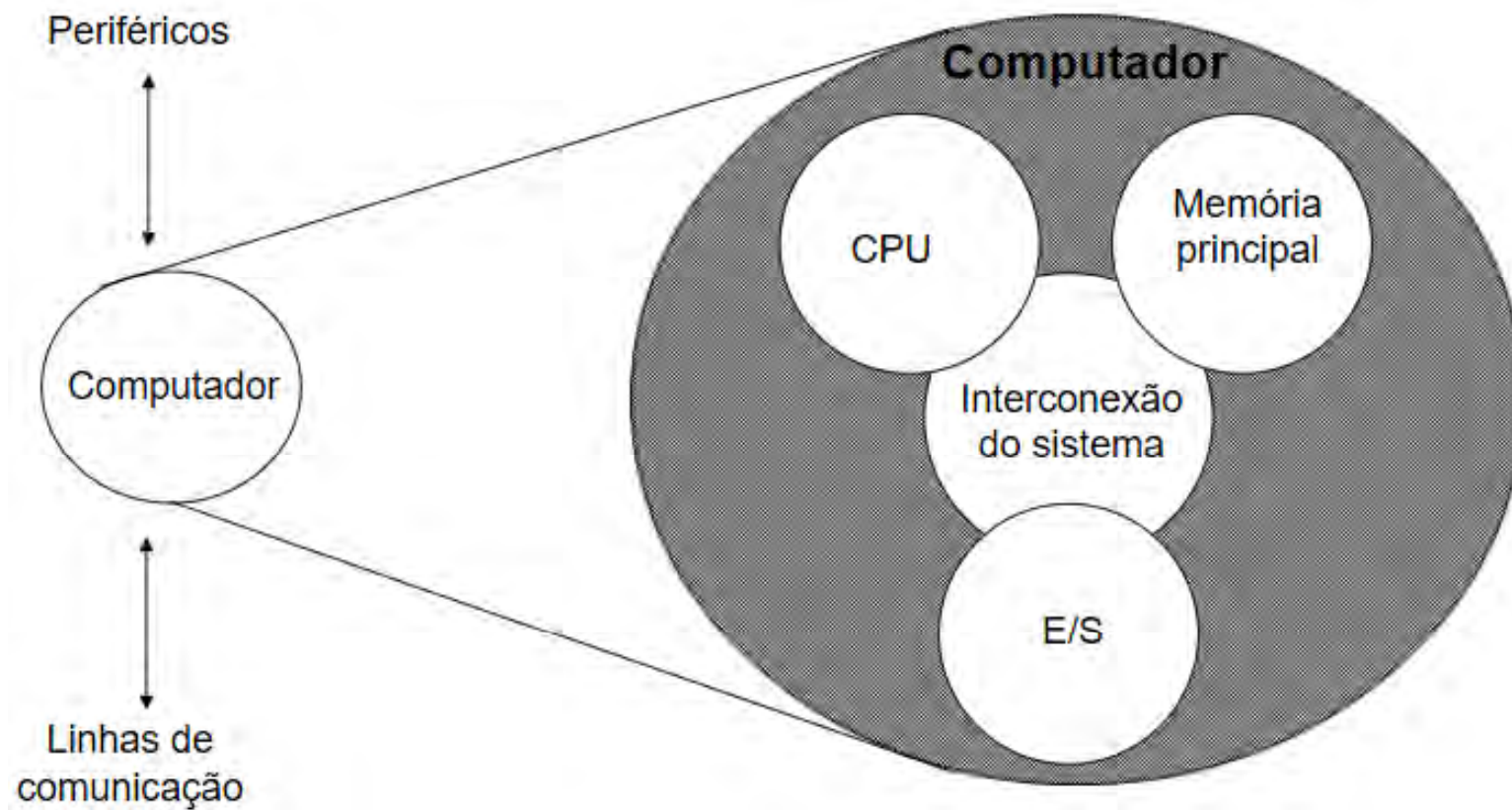
Computador de Von Neumann

- Princípio do programa armazenado
 - Dados e Instruções no mesmo espaço de memória
- Execução Sequencial de instruções.
- Implementação da máquina universal de Turing;
- Base para 99% das máquinas até os dias atuais;

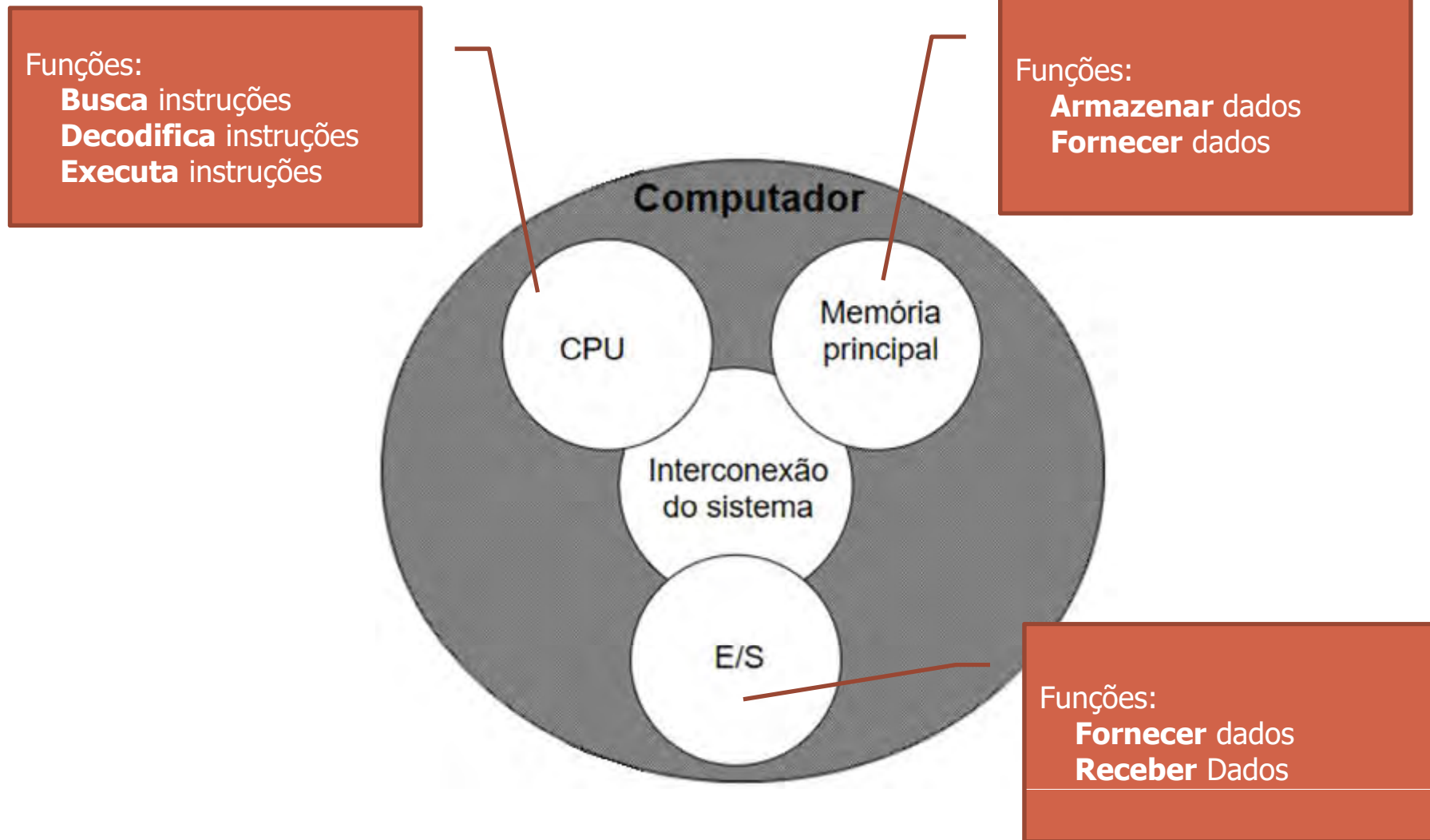
Computador de von Neumann

- Palavras (= conjunto de bits)
 - Podem ter diferentes significados agrupados em: dados e instruções
- Instruções
 - Contêm as informações que o computador necessita para executar as várias operações
- Cada máquina possui um conjunto de instruções
 - Coleção completa de instruções que será entendida pela CPU)

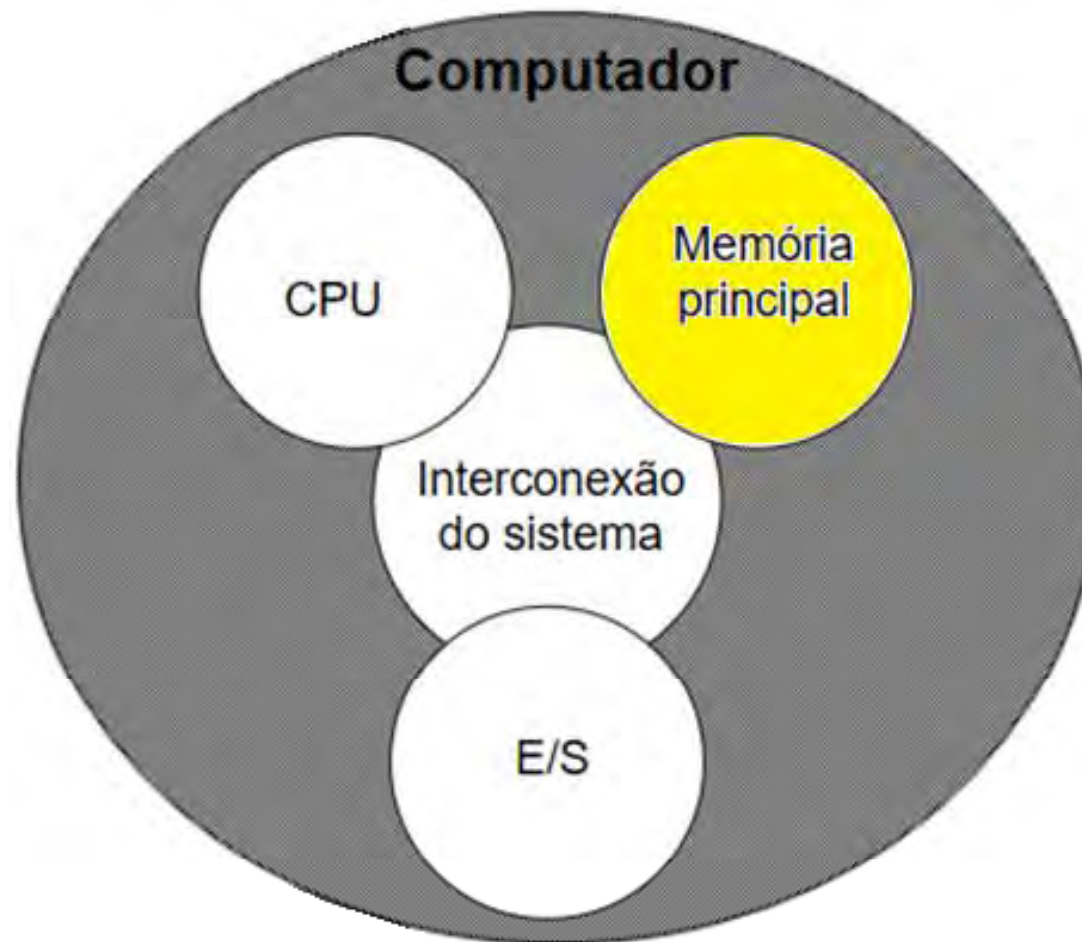
Computador: Componentes Básicos



Computador: Componentes Básicos



Computador: Componentes Básicos

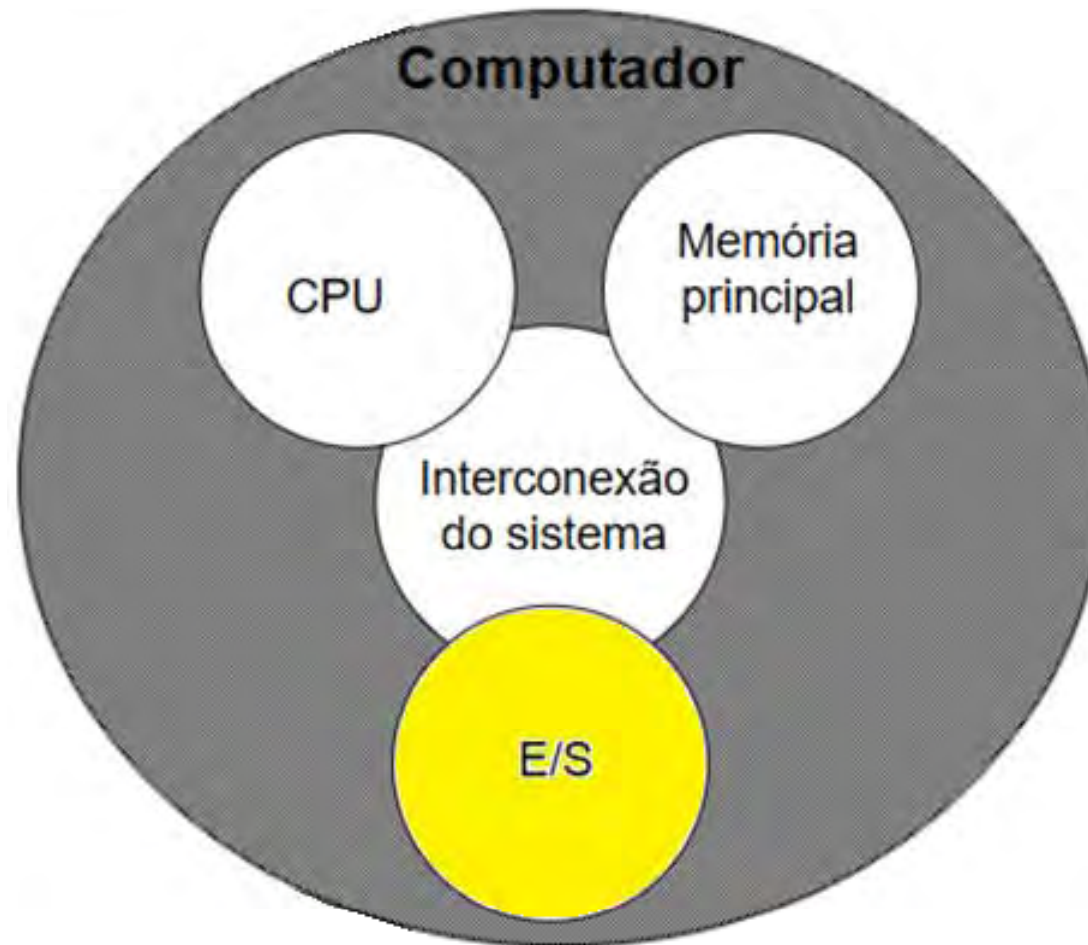


Computador: Componentes Básicos

- **Memória Principal**

- Consiste de um arranjo linear de células de armazenamento endereçáveis similares aos registradores, porém em quantidade muito maior;
- O endereçamento pode ser byte a byte, ou palavra a palavra, a qual geralmente é constituída por 1 ou mais bytes (ex: palavra de 32 bits, ou 4 bytes);
- Cada palavra possui um único endereço e pode ser lida ou escrita na memória. A natureza da operação é indicada por meio de sinais de controle. A posição de memória em que deve ser efetuada a operação é especificada por um endereço.

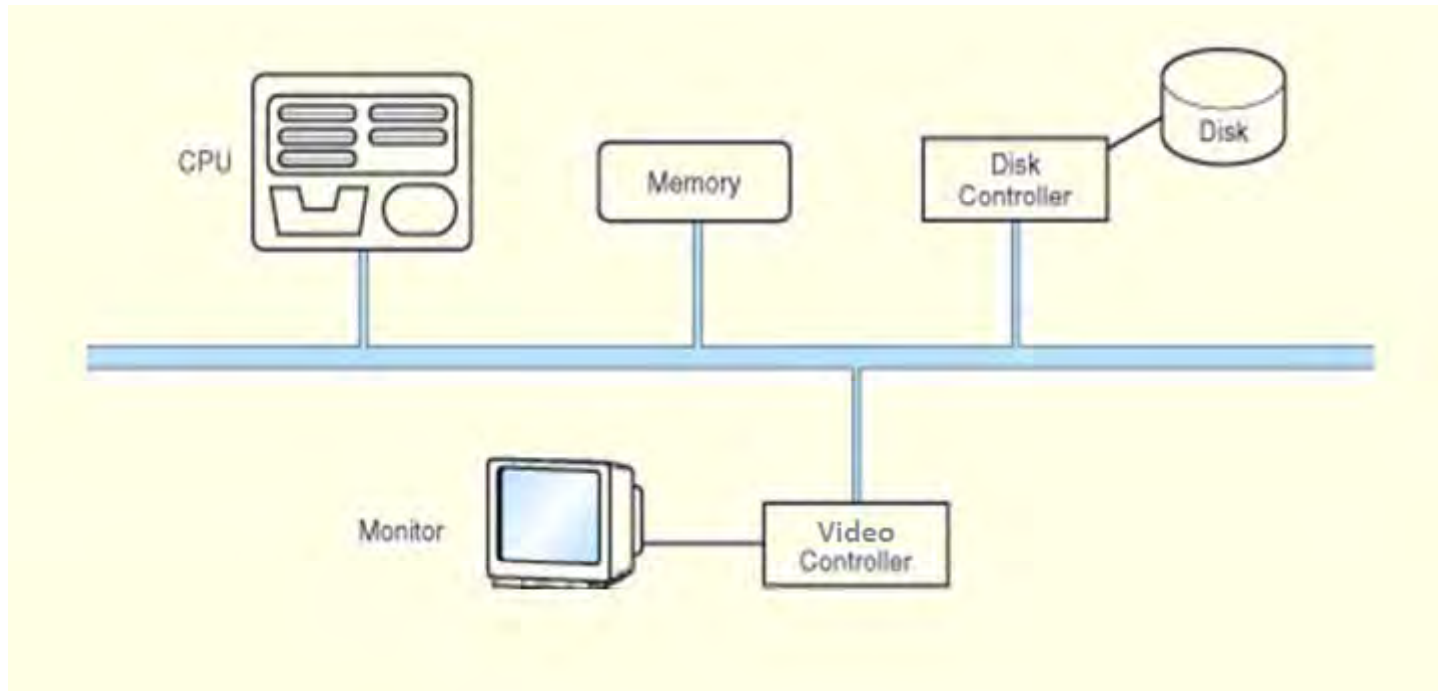
Computador: Componentes Básicos



Computador: Componentes Básicos

- E/S (I/O)
 - O computador se comunica com o "mundo exterior" através do Sistema de I/O (Input/Output);
 - Dispositivos de I/O: Monitor, teclado, disco (HD), placa de rede, microfone, alto-falante, memória flash, etc..;
 - Os dispositivos de I/O não se conectam diretamente à CPU, mas através de interfaces (ex: controladora de disco), estas sim conectadas ao barramento;

Computador: Componentes Básicos



Computador: Componentes Básicos

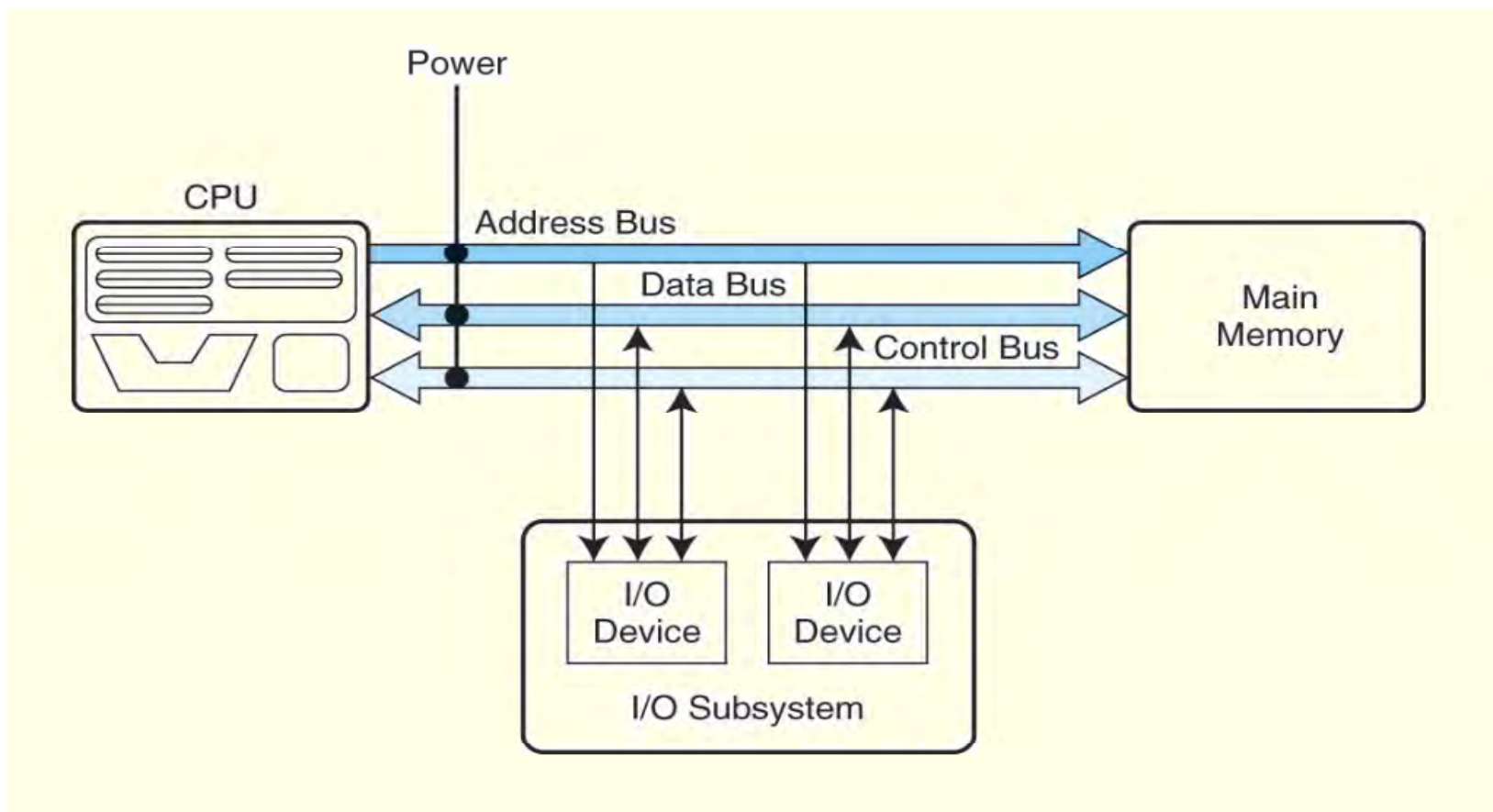
- E/S (I/O)
 - A CPU se comunica c/ esses dispositivos externos através de **registradores especiais**, ou registradores de I/O;
 - Essa troca de dados pode ser feita de 2 maneiras:
 - I/O **mapeado** na **memória**: os registradores de I/O aparecem como endereços de memória;
 - I/O via **instruções** especializadas, que utilizam os registradores de I/O;
 - **Interrupções** são utilizadas p/ notificar a CPU sobre eventos de I/O;

Computador: Componentes Básicos

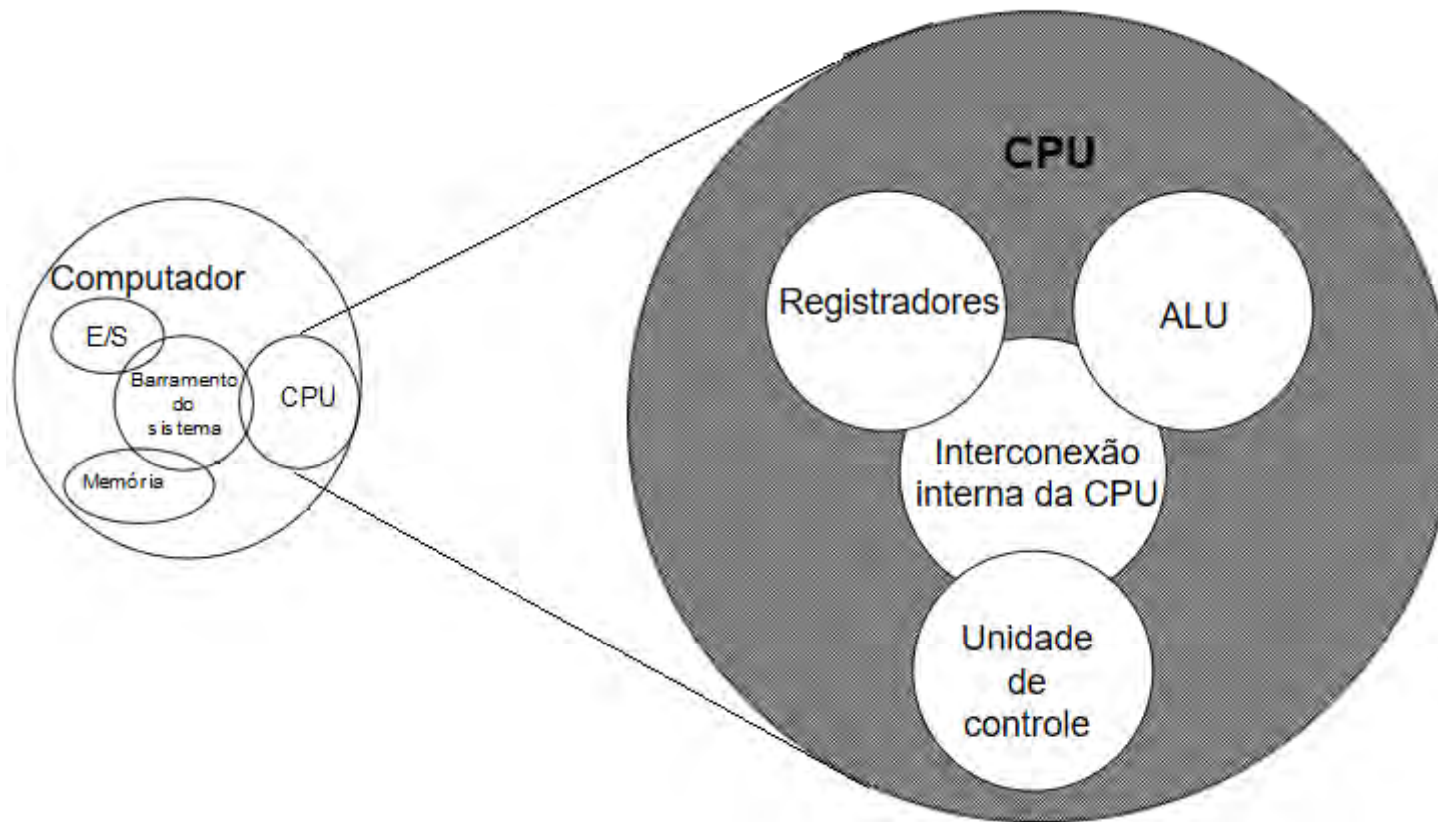
- Interconexão (barramento)
 - **Barramento**: um caminho de comunicação entre dois ou mais dispositivos. Conjunto de fios (linhas), os quais transmitem simultaneamente um bit (0 ou 1);
 - Barramentos podem ser de 3 tipos:
 - **Dados**: transmitem dados de um componente a outro
 - **Endereço**: determinam o local de um dado sendo acessado
 - **Controle**: determinam a direção de um determinado fluxo de dados, ou quando um componente pode acessar o barramento

Computador: Componentes Básicos

- Interconexão (barramento)



Computador: Componentes Básicos

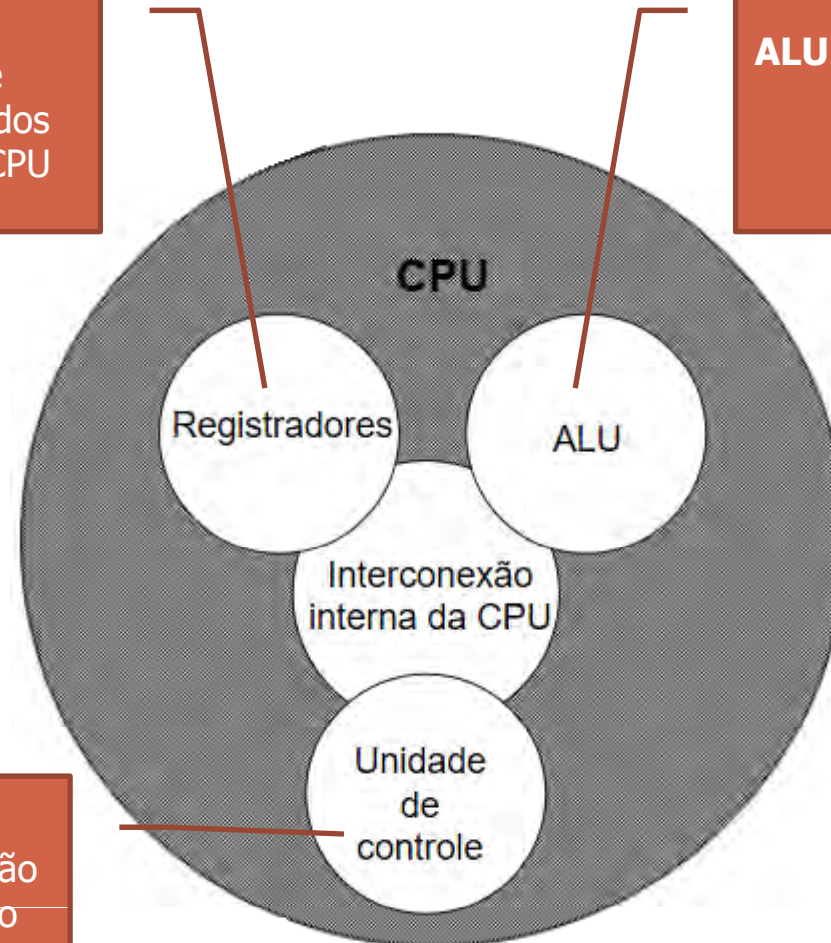


Computador: Componentes Básicos

Registradores:

Armazenam dados que podem ser acessados rapidamente pela CPU

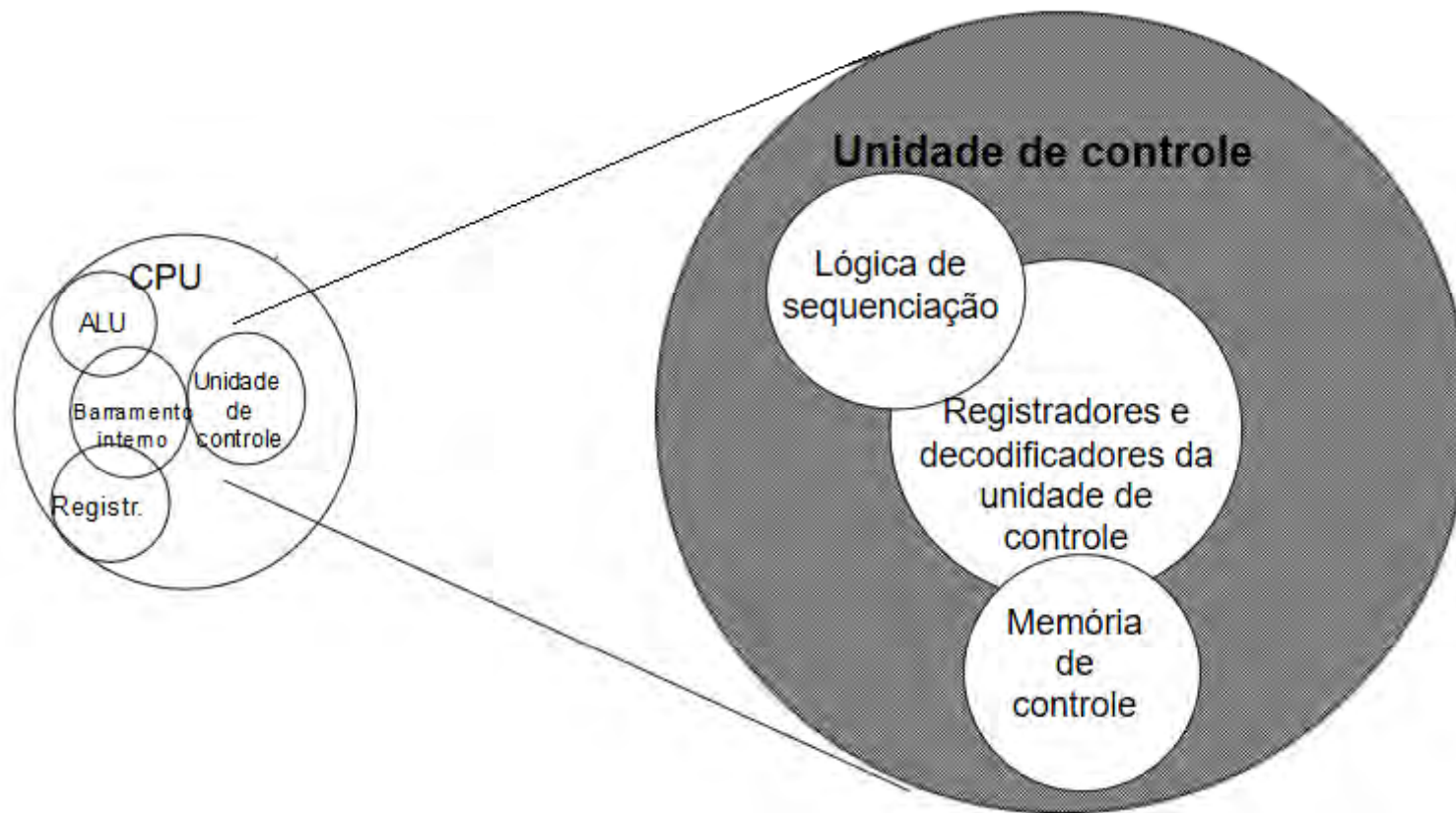
ALU (Aritmetic Logic Unit):
Executa operações lógicas e aritméticas;



UC:

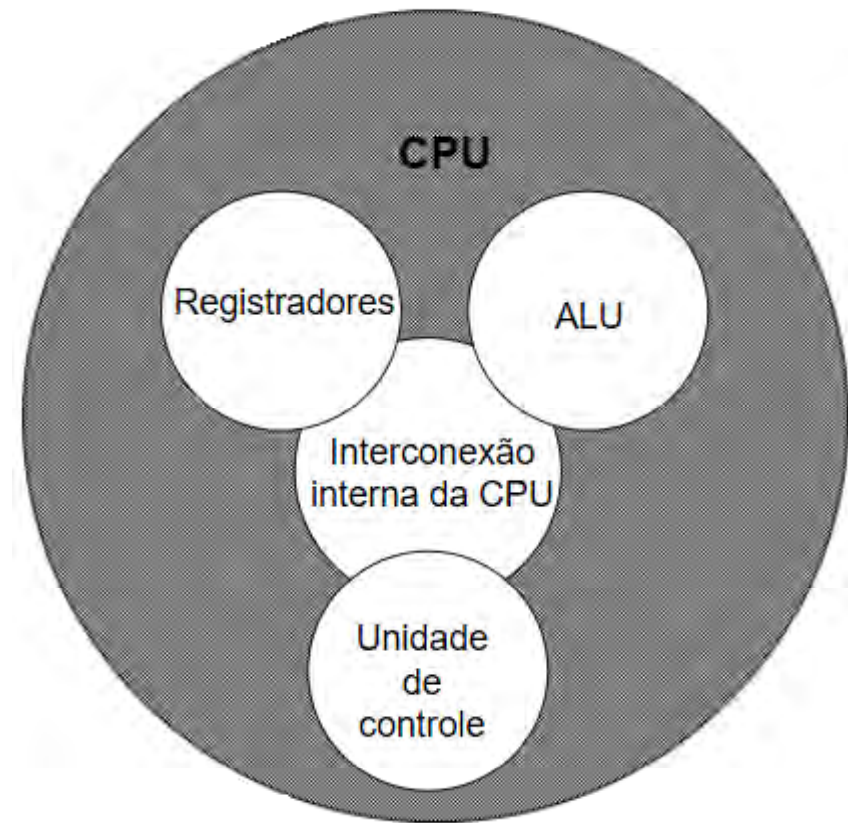
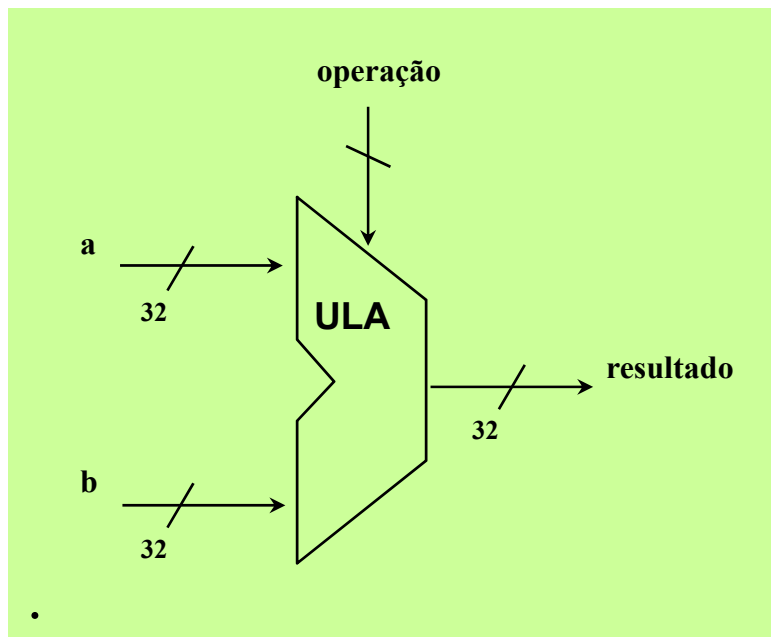
Determina quais ações são tomadas com base no valor de certos registradores especiais

Computador: Componentes Básicos



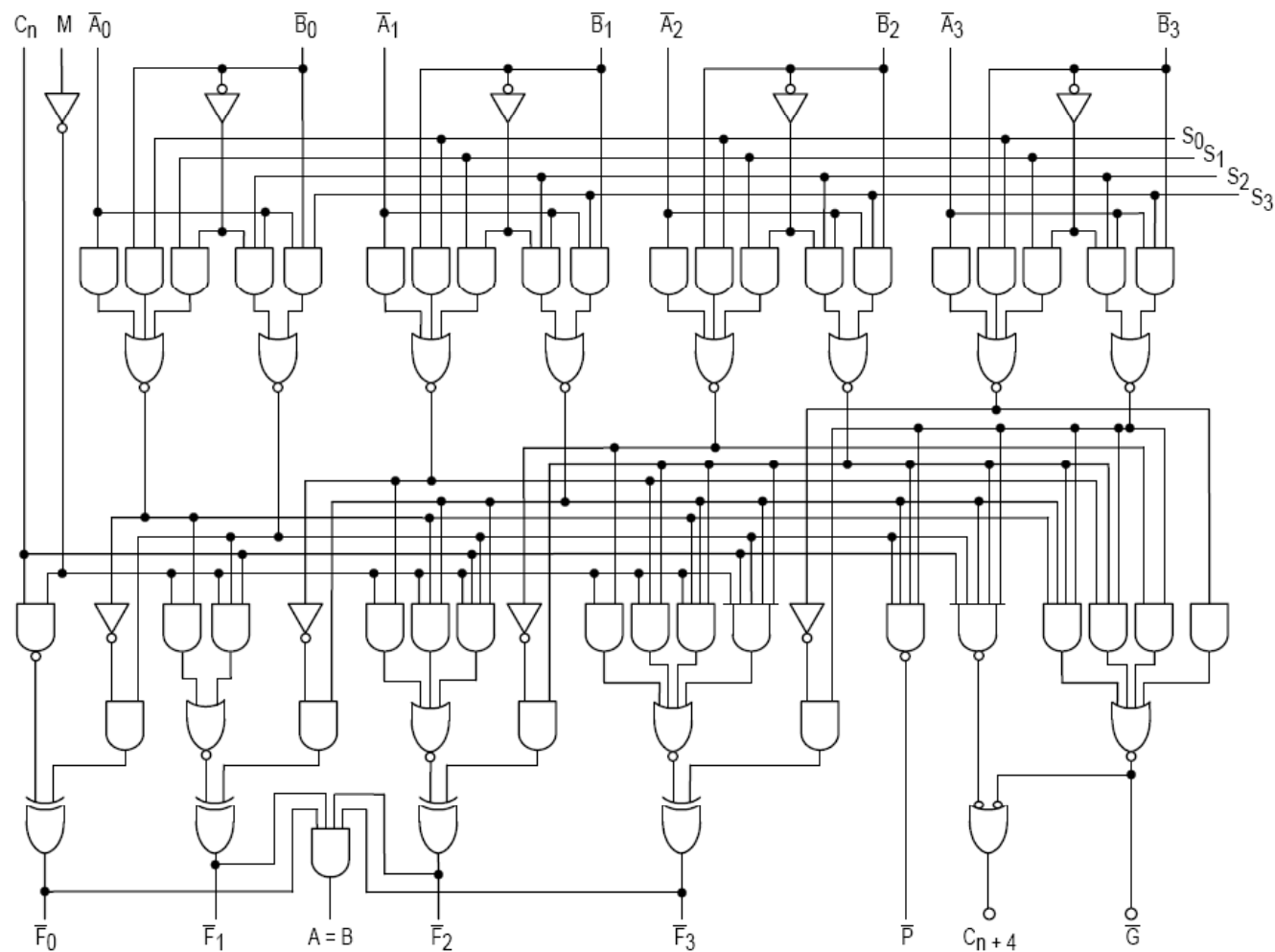
Computador: Componentes Básicos

- **ULA: "Motor" do computador** -> dispositivo que executa operações aritméticas (add, sub, etc) e lógicas (AND, OR, etc).



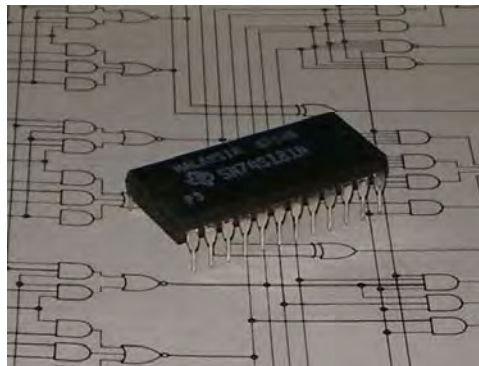
Computador: Componentes Básicos

- Como projetar e implementar uma ULA ?



Computador: Componentes Básicos

- Como projetar e implementar uma ULA ?



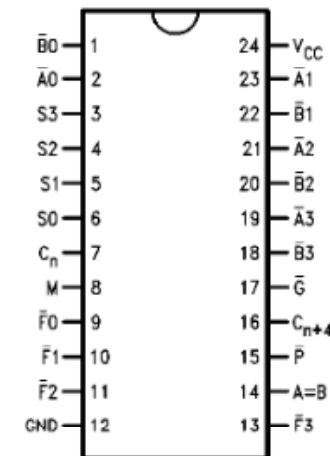
Function Table

| Mode Select Inputs | | | | Active LOW Operands & F _n Outputs | | Active HIGH Operands & F _n Outputs | |
|--------------------|----|----|----|--|--|---|--|
| S3 | S2 | S1 | S0 | Logic (M = H) | Arithmetic (Note 2) (M = L) (C _n = L) | Logic (M = H) | Arithmetic (Note 2) (M = L) (C _n = H) |
| L | L | L | L | \bar{A} | A minus 1 | \bar{A} | A |
| L | L | L | H | $\bar{A}\bar{B}$ | AB minus 1 | $\bar{A} + \bar{B}$ | A + B |
| L | L | H | L | $\bar{A} + \bar{B}$ | $\bar{A}\bar{B}$ minus 1 | $\bar{A}B$ | A + \bar{B} |
| L | L | H | H | Logic 1 | minus 1 | Logic 0 | minus 1 |
| L | H | L | L | $\bar{A} + \bar{B}$ | A plus (A + \bar{B}) | $\bar{A}\bar{B}$ | A plus $\bar{A}\bar{B}$ |
| L | H | L | H | \bar{B} | AB plus (A + \bar{B}) | \bar{B} | (A + \bar{B}) plus $\bar{A}\bar{B}$ |
| L | H | H | L | $\bar{A} \oplus \bar{B}$ | A minus B minus 1 | $A \oplus B$ | A minus B minus 1 |
| L | H | H | H | $A + \bar{B}$ | A + \bar{B} | $\bar{A}\bar{B}$ | AB minus 1 |
| H | L | L | L | $\bar{A}B$ | A plus (A + B) | $\bar{A} + \bar{B}$ | A plus AB |
| H | L | L | H | $A \oplus B$ | A plus B | $\bar{A} \oplus \bar{B}$ | A plus B |
| H | L | H | L | B | $\bar{A}\bar{B}$ plus (A + B) | B | (A + \bar{B}) plus AB |
| H | L | H | H | A + B | A + B | AB | AB minus 1 |
| H | H | L | L | Logic 0 | A plus A (Note 1) | Logic 1 | A plus A (Note 1) |
| H | H | L | H | $\bar{A}\bar{B}$ | AB plus A | $A + \bar{B}$ | (A + B) plus A |
| H | H | H | L | AB | $\bar{A}\bar{B}$ minus A | A + B | (A + \bar{B}) plus A |
| H | H | H | H | A | A | A | A minus 1 |

Note 1: Each bit is shifted to the next most significant position.

Note 2: Arithmetic operations expressed in 2s complement notation.

Connection Diagram

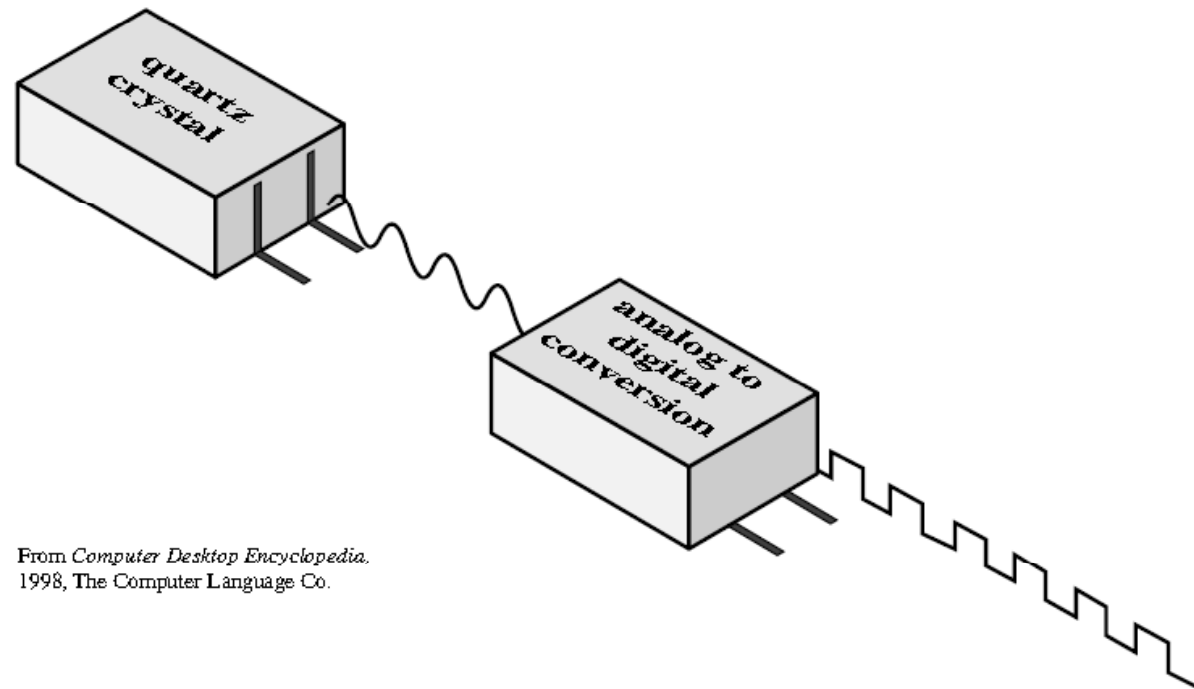


Pin Descriptions

| Pin Names | Description |
|-------------------------|-------------------------------------|
| $\bar{A}0$ – $\bar{A}3$ | Operand Inputs (Active LOW) |
| $\bar{B}0$ – $\bar{B}3$ | Operand Inputs (Active LOW) |
| S0–S3 | Function Select Inputs |
| M | Mode Control Input |
| C _n | Carry Input |
| $\bar{F}0$ – $\bar{F}3$ | Function Outputs (Active LOW) |
| A = B | Comparator Output |
| \bar{G} | Carry Generate Output (Active LOW) |
| \bar{P} | Carry Propagate Output (Active LOW) |
| C _{n+4} | Carry Output |

Computador: Componentes Básicos

- Todo computador possui um clock para sincronizar as atividades de seus componentes;
- O número fixo de ciclos de clock é necessário para executar uma dada operação ou transferências de dados;



From *Computer Desktop Encyclopedia*.
1998, The Computer Language Co.