

Computador

Um computador elétrico digital é uma máquina que basicamente processa e armazena dados através do sistema binário, o menor sistema numérico possível. As quatro funções de um computador digital são: **processamento** de dados, **armazenamento** de dados, **movimentação** de dados e **controle**.

Existem vários tipos de computadores digitais, entre eles dois que são opostos: os computadores de **propósito geral** e os **sistemas embarcados**.

Um computador de **propósito geral** é uma máquina projetada para atender a vários propósitos, não apenas um fim específico. Como exemplo podemos tomar os computadores pessoais (PCs) ou até mesmo o ENIAC (o primeiro computador elétrico de propósito geral) que inicialmente foi desenvolvido para ser usado em cálculos balísticos mas por ter ficado pronto apenas após o fim da Segunda Guerra Mundial foi utilizado para realizar cálculos que ajudaram a descobrir a viabilidade da bomba de hidrogênio, o que mostrou sua natureza de computador de propósito geral.

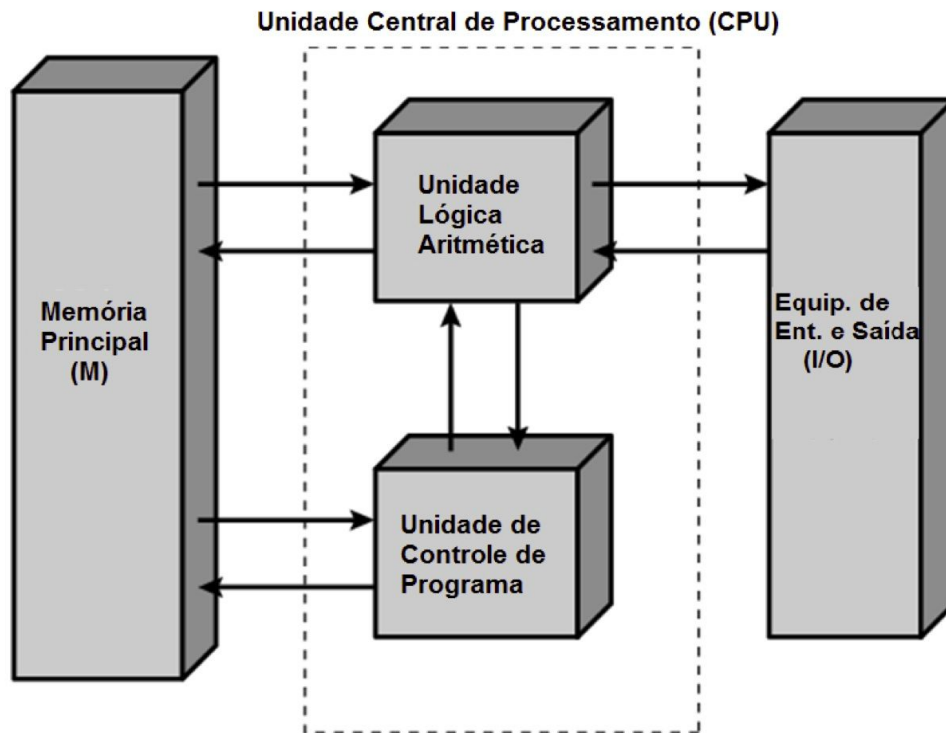
Já um **sistema embarcado** ao contrário de um computador de propósito geral é projetado para atender uma finalidade específica. Um sistema embarcado é uma combinação de hardware e software dedicada a executar uma tarefa. Muitas vezes ele faz parte de um outro sistema maior.

Máquina de von Neumann

Um computador é chamado de von Neumann se possuir os seguintes requisitos:

- Possuir uma **memória principal** que armazena tanto dados quanto instruções para a máquina;
- Uma **unidade lógica aritmética (ULA)** capaz de operar dados binários;
- Uma **unidade de controle** para ler e executar instruções que estão na memória principal e

- Um **dispositivo de entrada e saída (E/S)** sendo operado pela unidade de controle.



Arquitetura e Organização

A **arquitetura** de computadores tem como função **arquitetar** as funções do computador visíveis ao programa, tais como o conjunto de instruções e a quantidade de bits usados para representar uma variável.

Já a **organização** de computadores é voltada a **organizar o hardware** do computador a fim de atender as especificações da arquitetura.

Um exemplo é que a tarefa de decidir se um computador terá ou não uma instrução para realizar multiplicação é responsabilidade da arquitetura, enquanto a maneira que isso será implementado pelo hardware (por uma unidade física específica para isso ou por uma sequência de somas, por exemplo) é responsabilidade da organização.

Outro exemplo é o fato de um computador com arquitetura de 64 bits exigir que o hardware esteja preparado para operar de tal forma. Então a decisão de

trabalhar com um barramento de 64 bits ou com um barramento de 32 bits e enviar os bits duas vezes é uma decisão de organização de computadores.

Organização de Computadores Digitais

Conforme mencionado previamente, um computador é uma máquina que basicamente movimenta, processa e armazena dados. Para isso a essência de um computador são a **unidade de processamento central UPC (CPU da sigla em inglês)**, **memória principal**, **entrada e saída (E/S)** e o **sistema de interconexão**.

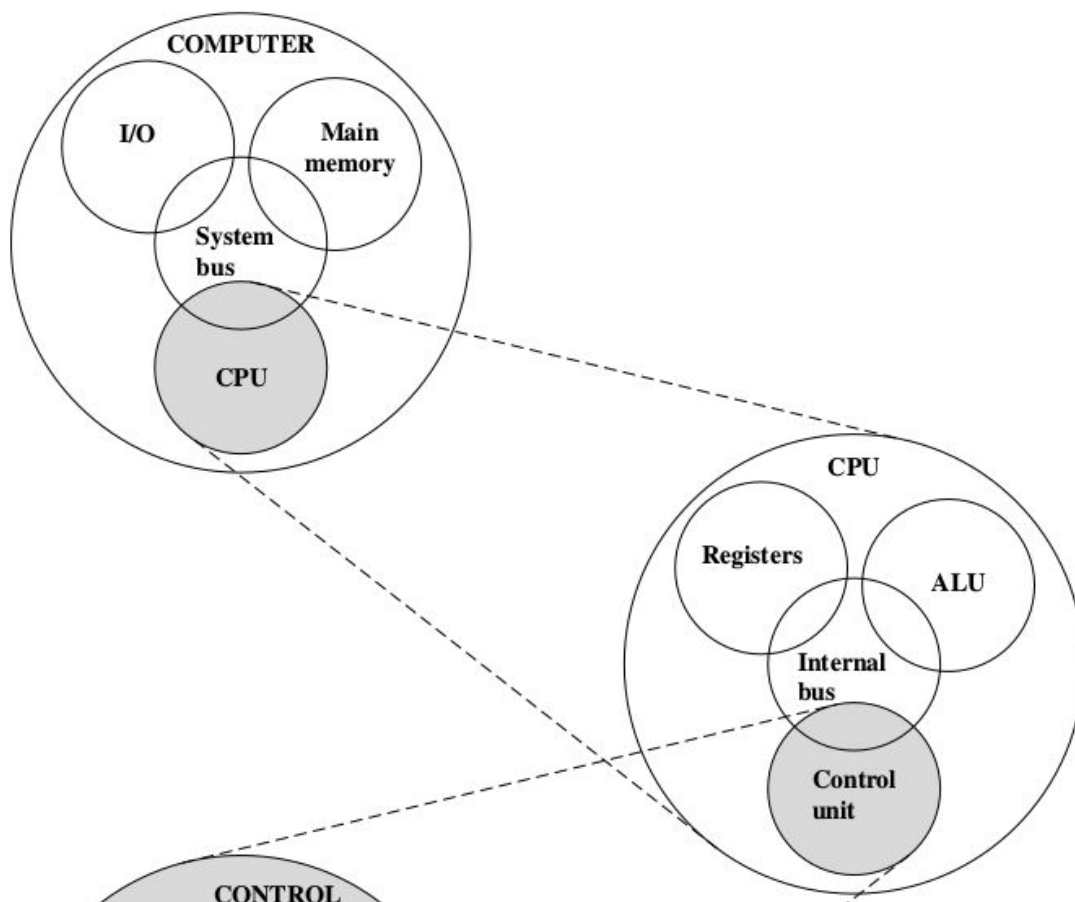
- **Unidade de processamento central (UPC):** processa dados e realiza o controle da máquina;
- **Memória principal:** armazena dados e instruções de programas;
- **Entrada e saída (E/S):** move dados entre o computador e o ambiente externo;
- **Sistema de interconexão:** interliga os três componentes anteriores.

Um exemplo de sistema de interconexão é o sistema de barramento.

O processo pelo qual dados são recebidos ou enviados para um dispositivo fora do computador, mas diretamente conectado a ele, se chama entrada-saída (input-output). É interessante notar que tal dispositivo é chamado de **periférico** e que mouse e teclado são dois exemplos de periféricos.

Por sua vez **unidade de processamento central** é composta por **registradores**, **unidade lógica aritmética (ULA)**, **unidade de controle** e **sistema de barramento interno**.

- **Registradores:** são usado para armazenada interno dentro da UPC;
- **Unidade lógica aritméticas (ULA):** realiza operações lógicas e aritméticas, isto é, realiza o processamento dos dados;
- **Unidade de controle:** controla as operações da UPC e do computador;
- **Sistema de barramento interno:** conecta os três componentes listados anteriormente.



Gerações de Computadores

A **primeira geração** de computadores foi marcada pelas **válvulas eletrônicas** (tubos de vácuo). As válvulas eletrônicas consumiam muita eletricidade, esquentavam de mais e queimavam frequentemente, por isso os **transistores** foram a marca da **segunda geração**. O fato de eles serem mais baratos (pois são feitos silício, material de baixo custo e abundante), menores, e emissores de menos calor do que um válvula eletrônica foi fundamental para o desenvolvimento do poder computacional.

Já a **terceira geração** fez o uso dos transistores em **sistemas integrados**, isto é, ao invés de produzir componentes discretos (individuais), como o transistor, para depois montá-los em um circuito inteiro, o circuito inteiro era impresso em um bloco de silício. Isso possibilitou colocar mais componentes dentro de um chip.

Houveram **outras gerações**, tais como as que trouxeram o **microprocessador** (um único chip que contém todos os componentes de uma UCP).

Lei de Moore e Gargalo de von Neumann

Em 1965 Gordon Moore observou que a cada ano o número de transistores dentro de um chip dobraria, e ele estava certo. Em 1970 foi alterado para 18 meses o prazo para o número de transistores dobrarem, mas isso se mantém até hoje.

Os chips foram ficando cada vez menores e por isso o tempo de percorrimto do sinal no barramento interno do chip ficava cada vez menor. O gasto energético também diminuía, embora enquanto a densidade do chip aumentava o calor produzido também aumentava.

O problema, chamado de Gargalo de von Neumann, é que enquanto o poder de processamento cresce de forma exponencial, a velocidade de outros componentes importantes como a memória primária, por exemplo, não cresce no mesmo ritmo, e isso vem aumentando cada vez mais a distância entre a velocidade do processador e a velocidade de tais componentes.

Aumentar o número de bits é uma solução interessante, o problema é que não dá para abusar dessa solução pois isso pode ocupar muito espaço, já que em um barramento cada fio não pode ficar muito perto dos outros pois caso isso aconteça um fio pode induzir corrente no outro.

A memória CACHE é bem rápida, porém cara. O uso desse tipo de memória está aumentando dentro de locais estratégicos no computador (inclusive dentro do processador), porém está chegando no limite.

Quando o assunto são os periféricos o uso da memória CACHE também é uma solução, assim como o aumento da velocidade dos barramentos, pois o problema consiste em mover os dados em alta velocidade.

Fim da Lei de Moore

Embora a lei de Moore esteja sendo obedecida por décadas está cada vez mais difícil dobrar a quantidade de resistores dentro do chip. Entre os motivos está a dificuldade de diminuir o tamanho do resistor, a quantidade de calor gerada pelo chip que está aumentando devido ao aumento da densidade e o limite físico dos fios, pois quanto mais fino for o fio maior é a resistência, o que diminui a velocidade da transmissão dos elétrons. Colocar fios muito próximos uns dos outros também é algo problemático já que um pode induzir correndo no outro.

A solução está na organização e na arquitetura, pois os limites físicos estão cada vez mais próximos.

Bases Numéricas

“Podemos considerar, a fim de simplificação, que base numérica é um conjunto de símbolos (ou algarismos) com o qual podemos representar uma certa quantidade ou número.” (<http://www.dainf.cefetpr.br/~robson/prof/aulas/common/bases.htm>)

Os seguintes dígitos são usados nas seguintes bases:

- **Decimal:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (10 algarismos);
- **Hexadecimal:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (16 algarismos); {**A** = 10, **B** = 11, **C** = 12, **D** = 13, **E** = 14, **F** = 15}
- **Octal:** 0, 1, 2, 3, 4, 5, 6, 7 (8 algarismos);
- **Binária:** 0, 1 (2 algarismos);

Podemos saber o correspondente V_{10} (na base 10) de um número V_B (na base B) usamos a seguinte fórmula:

$$V_{10} = \sum_{i=1}^d f(i) \cdot B^{i-1}$$

Onde representa d o número de dígitos do número V_B , i é a posição do dígito da direita para a esquerda, $f(i)$ é o dígito na posição i , e B o número de algarismos da base.

Exemplos de como transformar V_B em V_{10} :

$$\begin{aligned} \text{➤ } 1234_{10} &= 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 \\ &= 1000 + 200 + 30 + 4 = 1234_{10} \end{aligned}$$

$$\begin{aligned} \text{➤ } 45FA_{16} &= 4 \cdot 16^3 + 5 \cdot 16^2 + F \cdot 16^1 + A \cdot 16^0 \\ &= 4 \cdot 16^3 + 5 \cdot 16^2 + 15 \cdot 16^1 + 10 \cdot 16^0 \\ &= 16384 + 1280 + 240 + 10 \\ &= 17914_{10} \end{aligned}$$

$$\begin{aligned} \text{➤ } 1234_8 &= 1 \cdot 8^3 + 2 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0 \\ &= 512 + 128 + 24 + 4 \\ &= 668_{10} \end{aligned}$$

$$\begin{aligned} \text{➤ } 1001_2 &= 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 8 + 0 + 0 + 1 \\ &= 9_{10} \end{aligned}$$

Se para fazer a conversão de uma base B para a base decimal o processo de multiplicação é usado, então podemos observar que para fazer o processo inverso podemos usar a operação inversa da multiplicação: a divisão.

Se temos um número V_{10} e queremos saber o valor V_B , isto é, o valor de tal número na base B, basta seguir o processo prático dos exemplos abaixo.

Exemplos de como transformar V_{10} em V_B :

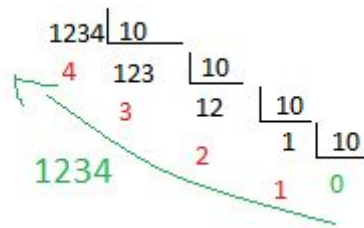
$$\text{➤ } 1234_{10} = 1234_{10}$$

$$4^\circ) 1234 \% 10 = 4$$

$$3^\circ) (1234 / 10) \% 10 = 123 \% 10 = 3$$

$$2^\circ) (123 / 10) \% 10 = 12 \% 10 = 2$$

$$1^\circ) 12 / 10 = 1$$



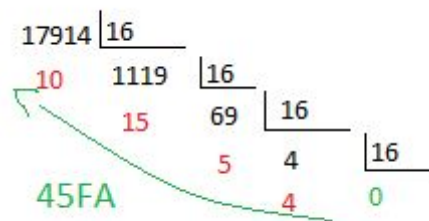
➤ $17914_{10} = 45FA_{16}$

4º) $17914 \% 16 = 10$

3º) $(17914 / 16) \% 16 = 1119 \% 16 = 15$

2º) $(1119 / 16) \% 16 = 69 \% 16 = 5$

1º) $(4 / 16) \% 16 = 4$



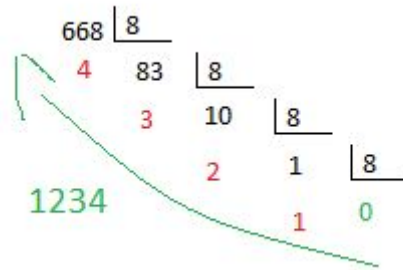
➤ $668_{10} = 1234_8$

4º) $668 \% 8 = 4$

3º) $(668 / 8) \% 8 = 83 \% 8 = 3$

2º) $(83 / 8) \% 8 = 10 \% 8 = 2$

1º) $(10 / 8) \% 8 = 1 \% 8 = 1$



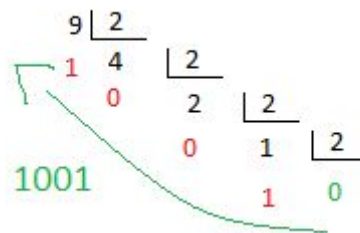
➤ $9_{10} = 1001_2$

4º) $9 \% 2 = 1$

3º) $(9 / 2) \% 2 = 4 \% 2 = 0$

2º) $(4 / 2) \% 2 = 2 \% 2 = 0$

1º) $(2 / 2) \% 2 = 1 \% 2 = 1$



Também é possível fazer uma conversão direta entre bases diferentes de 10. Por exemplo, vamos converter um número de binário diretamente para octal.

Vamos pegar o número 101100011101_2 . Sabemos que $8 = 2^3$, então a cada 3 dígitos binários nós fazemos a conversão para decimal.

$$101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 0 + 1 = 5$$

$$100 = 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 4 + 0 + 0 = 4$$

$$011 = 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 0 + 2 + 1 = 3$$

$$101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 0 + 1 = 5$$

101100011101

5 4 3 5 ==> 5435_8

Agora vamos pegar o mesmo número 101100011101_2 e vamos converter para hexadecimal. Sabemos que $16 = 2^4$, então a cada 4 dígitos binários nós fazemos a conversão para decimal.

$$1011 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = 11 = B$$

$$0001 = 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0 + 0 + 0 + 1 = 1 = 1$$

$$1101 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 0 + 1 = 13 = D$$

101100011101

B 1 D ==> $B1D_{16}$