

# Circuitos Digitais

Thiago Figueiredo Marcos

22 de maio de 2024

## 1 Introdução

Um computador digital pode ser descrito como aquilo que computa, ou aquilo que processa informação digital. A informação que é processada por um circuito digital é aquela que é **quantizada** ou **discretada** (?).

No mundo comum, as informações são analógicas, ou seja, a onda que representa aquela informação possui uma gama de valores diferenciados. Já os sinais digitais podem ser representados apenas por dois valores, ou seja, uma lógica binária.(?)

Os circuitos digitais são construídos por componentes eletrônicos e tem como entradas e saídas sinais digitais. Para usarmos informações do mundo analógico é preciso discretar essas informações, afim de convertelas à binária e geralmente é usado uma medida em Volts para determinar se uma informação é ligada ou desligada. Após convertida a informação e processada no circuito digital é preciso converter o sinal de saída do circuito que é digital, em analógico novamente.(?)

## 2 Conversão Analógico - Digital (Discretização)

Sinais discretos são frequências descontinuas no tempo, ou seja, definida apenas para determinados instantes. Representa aproximadamente o mundo real, entretanto, podem ser utilizadas várias técnicas para melhorar a representação, como as de processamento de sinais digitais.

Aqui também vale ressaltar que o processo de discretização de alguma informação está consequentemente ligada a perda de determinadas informações.

### 2.1 Principais propriedades da discretização

1. **Amostragem:** Discretização do sinal analógico no tempo.
2. **Quantização:** Discretização da amplitude do sinal amostrado em níveis.
3. **Codificação:** Atribuição de códigos, onde geralmente são binários às amplitudes do sinal quantizado.

### 3 Conversão Digital - Analógico (Linearização)

Se refere a o processo que transforma um sinal modelado por eventos discretos em um sinal contínuo, ou seja, o processo de integração de vários sinais discretos para simular um evento contínuo.

### 4 Processamento

O processamento de informação se refere a diversas operações realizadas por um circuito digital para transformar a entrada de dados em uma saída significativa de interesse. Isso pode ser calculos, manipular dados como agregação, separação e classificação ou ainda filtragem, entre outros. Além disso, no circuito digital é simplificado o armazenamento de informações bem como possui uma menor probabilidade de interferências.

### 5 Sistemas de numeração

Número nos remete a ideia de quantidade, já o numeral é a representação desta ideia, na prática, nos referimos a palavra número para qualquer tipo de representação numeral.

**Exemplo:** A quantidade **Quarenta e dois** é representada pelo numeral 42.

Sem o conhecimento da organização posicional dos números, como podemos representar todos eles? Poderíamos pensar em um simbolo para cada número, porém, existe uma infinidade de quantidades.

Há cerca de 3.000 anos atrás os **Egípcios** desenvolveram um sistema de numeração, entre esse sistema esta a base 10, na qual utilizamos até hoje. Os números representados por hieróglifos eram mais usados em monumentos e templos, pintados ou talhados em pedra. Há sete símbolos, representando os números 1, 10, 100, 1000, 10 000, 100 000 e 1 000 000.(?)

Algarismos é um conjunto finito de símbolos numéricos que usamos para representar quantidades reais. Todo e qualquer número pode ser representado por uma combinação de algarismos, os mais conhecidos são os indo-arábicos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Sistema de numeração é a forma de atribuir uma representação única para cada número. O sistema de numeração posicional atribui valor ao algarismo conforme a sua posição, mais a esquerda ou mais a direita.

No sistema decimal de numeração posicional possuímos 10 algarismos, 0 .. 9, um cada um deles representa seu valor absoluto, ou seja, o valor 0 representa o nada, o valor 1 representa uma única unidade e assim por diante. Dependendo da posição que o valor estiver, seu valor absoluto pode variar, por exemplo: Imagine um número com 4 casas decimais, - - - -, o número que estiver na "casa" mais a esquerda não representara seu valor absoluto e sim o seu valor absoluto multiplicado por um milhão.

**Exemplo:**  $4237 = 4*1000 + 2*100 + 3*10 + 7*1$ , observe o seguinte:

$$\begin{aligned}
4 * 1000 &= 4.000 \\
2 * 100 &= 200 \\
3 * 10 &= 30 \\
7 * 1 &= 7
\end{aligned}$$

A soma de todos os valores resulta no valor original 4237.

Agora podemos sistematizar isso matematicamente, sabemos que um número inteiro A no sistema decimal é apresentado por N dígitos assim:

$$A_{n-1}A_{n-2}\dots A_2A_1A_0$$

Cada  $a_i$  é um algarismo decimal.

$$a_{n-1} * 10^{n-1} + a_{n-2} * 10^{n-2} + \dots + a_2 * 100 + a_1 * 10 + a_0 * 1$$

ou seja,

$$\sum_{i=0}^{n-1} a_i * 10^i$$

Usando a mesma lógica, podemos representar os números racionais no sistema decimal

$$\sum_{i=0}^{n-1} a_i * 10^i + \sum_{i=1}^{\infty} a_{-i} * 10^{-i}$$

## 6 Truncamento

Vemos que à medida que caminhamos mais à direita depois da vírgula, o valor relativo a cada algarismo se torna cada vez menor, podemos fazer uma representação aproximada do número gerado, limitando o número de algarismos após a vírgula por uma constante M.

Essa aproximação chama-se truncamento. Com o truncamento há também um erro de aproximação que pode ser obtido com a diferença do número original com o número truncado.

Ao truncarmos um número com uma constante M para qualquer número real com n algarismos à esquerda da vírgula, e M algarismos à direita, assim:

$$a_{n-1}a_{n-2}\dots a_1a_0, a_{-1}a_{-2}\dots a_M$$

então temos que o erro de aproximação de qualquer número N será:

$$\text{err} < 10^{-M}$$

ou seja, aumentar M implica em diminuir o erro.

## 7 Bases não decimais

A quantidade de algarismos usados em um sistema de numeração posicional é chamada de base, por exemplo: O sistema decimal tem 10, o sistema binário tem 2 e assim por diante.

Acima fizemos uma representação matemática dos números posicionais racionais, a questão é que, em um sistema posicional em uma determinada base  $\mathbf{d}$ , pode ser representada da seguinte forma:

$$\sum_{i=0}^{n-1} a_i * d^i + \sum_{i=1}^{\infty} a_{-i} * d^{-i}$$

Para indicar a base em que um número está representado é comum a seguinte notação:

$$(a_{n-1}a_{n-2}\dots a_1a_0, a_{-1}a_{-2}\dots a_M)_d$$

## 8 Conversão de bases

Para converter um número  $\mathbf{n}_{10}$  para  $\mathbf{n}_d$  faremos divisões sucessivas entre o quociente e a base  $\mathbf{d}$ .

Existem algumas conversões que são triviais de realizar, como por exemplo da base 2 para base 16, ou ainda, da base 10 (até o 15) para base 16.

Da base 2 para base 16 por exemplo, pode-se seguir o padrão de contagem de 4 em 4 bits e ver qual a representação.

## 9 Operações Binárias

A base 2 é a base numérica mais utilizada na computação hoje em dia, em conjunto com a base 8, 16 e 64. Veremos futuramente que o computador realiza operações aritméticas na sua Unidade Lógica Aritmética (ULA), entenderemos agora como essas operações são realizadas na base binária.

### 9.1 soma

Antes de falarmos sobre a soma binária, precisamos relembrar o algoritmo da soma decimal, considere dois números A e B, sendo eles:

$$\begin{aligned} A &= a_{n-1}a_{n-2}\dots a_1a_0 \\ B &= b_{n-1}b_{n-2}\dots b_1b_0 \end{aligned}$$

O resultado da soma de A com B:

$$C = c_n c_{n-1} c_{n-2} \dots c_1 c_0 \text{ com } n + 1 \text{ algarismos.}$$

O mesmo método se aplica a somas binárias, exemplificado abaixo:

x	Y	Resultado	Vai um
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

## 9.2 Overflow

Agora considere a seguinte soma,  $(111001)_2 + (110011)_2 = (1101100)_2$  repare que a soma de dois números binários de 6 bits, resultou em um número binário de 7 bits, se nossa memória fosse apenas de 6 bits, ocasionária um **overflow**.

Há também a possibilidade de fazer a representação em uma quantidade maior de bits, basta adicionar o zero a esquerda do número.

## 9.3 Subtração

Para realizar as subtrações binárias, usaremos o complemento de 1 e complemento de 2, para evitar, detalhamentos adicionais.

**Complemento de 1:** O complemento de 1, basicamente, é o processo de inversão do bit, se for 1 fica 0 e vice versa.

**Complemento de 2:** É o complemento de 1 de uma sequência de bit, adicionando uma unidade ao final, exemplificando:

$\bar{B}$  : Complemento de 1

$\bar{B} + 1$  : Complemento de 2