

**Curso Superior de Tecnologia em Gestão da Produção Industrial**  
**Disciplina: Introdução à Computação - 1º Período**  
**Professor: José Maurício S. Pinheiro**

**AULA 3: Bases Numéricas**

O computador é um equipamento eletroeletrônico e como tal funciona através de pulsos elétricos. Os circuitos eletrônicos que formam os computadores digitais atuais são capazes de distinguir apenas dois níveis de tensão: Entre 0 e +0,5 Volts, associado ao valor binário "0", e de +2.5 a +3.5 Volts, associado ao valor binário "1". Cada sinal elétrico que o computador processa é chamado de BIT (Binary digiT) e é representado simbolicamente por "0" ou "1", daí a importância do sistema binário. Para ilustrar, lembramos que:

- **Leibniz** (séc. XVIII) introduz a numeração binária que utiliza dois símbolos (0 e 1), um sistema posicional de base 2.
- **Boole** (séc. XIX) estuda a simbologia do pensamento humano, emprestando-lhe um sentido determinista, introduzindo a lógica binária com uma álgebra própria.

As informações binárias são representadas em computadores digitais por meio de sinais elétricos. Estes sinais podem ser representados pelo nível de tensão, que vai especificar um estado entre dois possíveis. Por exemplo, se um sinal digital apresenta um nível de tensão de +5v, considera-se que o mesmo representa o valor digital 1. Da mesma forma, se o sinal digital apresenta um nível de tensão de -5v, então este representa o valor digital 0 (Fig. 1).

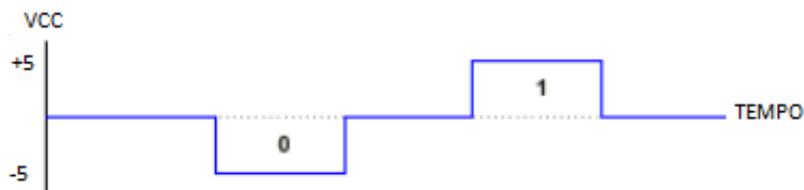


Figura 1 - Representação de sinais digitais

## 1. Unidades de Informação

A menor unidade de informação armazenável em um computador é o algarismo binário ou dígito binário, conhecido como bit (*binary digit*). O bit assume dois valores: 0 e 1. O menor grupo ordenado de bits representando uma informação útil e inteligível para o ser humano é o caractere. Cada sistema poderá definir como cada conjunto de bits irá representar um determinado caractere, quantos bits e como se organizam.

Ao conjunto de 4 bits denominamos *nibble*. Já o conjunto de 8 bits foi definido pela primeira vez pela IBM e é chamado de *Byte* (BYnary TErm). A representação de cada um dos símbolos, incluindo letras, algarismos e sinais

de pontuação, é feita através de grupos bits. Cada grupo de 8 (oito) bits é denominado Byte.

A palavra é o conjunto de bits que representa uma informação útil e está relacionada com o armazenamento e a transferência de informações entre a Memória Principal e a CPU. Portanto, palavra é um conjunto de Bytes. Um computador com uma palavra de 32 bits tem 4 Bytes/palavra.

As operações de armazenamento e recuperação de dados na memória são feitas Byte a Byte ou palavra a palavra, sendo comum mencionar o tamanho de uma memória em termos de Bytes. Para representar valores maiores, utilizamos o sistema métrico de grandeza, com algumas adaptações para os computadores, conforme mostra a Tabela 1.

**Tabela 1 - Grandezas de medida**

Sufixo	Quantidade
Kilo (K)	1 Kilobytes ou 1 KB = $1024 = 2^{10}$
Mega (M)	1 Megabytes ou 1 MB = $1.048.576 = 2^{20}$
Giga (G)	1 Gigabytes ou 1 GB = $1.073.741.824 = 2^{30}$
Tera (T)	1 Terabytes ou 1 TB = $1.099.511.627.776 = 2^{40}$
Peta (P)	1 Petabytes ou 1 PB = $1.125.899.906.843.624 = 2^{50}$
Exa (E)	1 Exabytes ou 1 EB = $1.152.921.504.607.870.976 = 2^{60}$

### 1.1. Representação de Texto

A informação na forma de texto normalmente é representada por meio de um código, no qual se atribui a cada um dos diferentes símbolos do texto (letras do alfabeto e caracteres de pontuação) um único padrão de bits. O texto fica então representado como uma longa cadeia de bits, na qual, padrões sucessivos representam os símbolos sucessivos no texto original.

Nos primórdios da computação muitos códigos diferentes foram projetados e utilizados em associação com diferentes partes do equipamento computacional, acarretando a proliferação de problemas de comunicação. Dentre os códigos de representação mais difundidos se destacam:

- ASCII (American Standard Code for Information Interchange): Código de um byte utilizado pela maioria dos microcomputadores e em alguns periféricos de equipamentos de grande porte;
- UNICODE: Código que utiliza dois bytes e consiste em 65.536 diferentes padrões de bits – o suficiente para representar os mesmos símbolos que o código ASCII, além daqueles mais comuns dos idiomas chinês e japonês.

### 1.2. Sistemas de Numeração

Os sistemas de numeração têm por objetivo prover símbolos e convenções para representar quantidades, de forma a registrar a informação quantitativa e

tornar possível processá-la. A representação de quantidades se faz com os números.

Na antiguidade, destacaram-se dois sistemas de numeração: O egípcio, onde, por exemplo, eram usados os símbolos # para representar uma centena, & para uma dezena e @ representando uma unidade, desse modo usaríamos ###&&@ para representar 321. O sistema Romano tornou-se mais conhecido, nele eram usados símbolos (letras) que representam as quantidades, como por exemplo: I (valendo 1), V (valendo 5), X (valendo 10). Desse modo, XV = 10 + 5 = 15.

Nos dois sistemas acima, cada símbolo tem um valor que independe de sua posição no número, essa característica dificulta a representação e as operações aritméticas com essa representação. Assim, foram mais desenvolvidos os sistemas numéricos posicionais, neles o valor representado no número depende de sua posição na representação. Nesses sistemas, o número de símbolos representa a sua base, e a posição ocupada por cada algarismo em um número altera seu valor de uma potência da base para cada casa à esquerda.

A base de um sistema é o número de símbolos diferentes, ou algarismos, necessários para representar um número qualquer.

O sistema decimal, utilizado de forma universal, utiliza dez símbolos diferentes (ou dígitos) para representar um número. É, portanto, um sistema numérico na base 10. Já os computadores utilizam a base 2 (sistema binário) e os programadores, por facilidade, usam em geral uma base que seja uma potência de 2, tal como  $2^4$  (base 16 ou sistema hexadecimal) ou eventualmente ainda  $2^3$  (base 8 ou sistema octal).

Se na base 10, dispomos de 10 algarismos para a representação do número: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Na base 2, seriam apenas 2 algarismos: 0 e 1. Na base 16, seriam 16: os 10 algarismos aos quais estamos acostumados, mais os símbolos A, B, C, D, E, F, representando respectivamente 10, 11, 12, 13, 14 e 15 unidades. Generalizando, temos que uma base  $b$  qualquer disporá de  $b$  algarismos, variando entre 0 e  $(b-1)$ .

### 1.3. Sistema Decimal

Os símbolos ou dígitos utilizados são os algarismos 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Os elementos são agrupados de dez em dez e, por essa razão, os números podem ser expressos por intermédio de potência de dez e recebem o nome de sistema de numeração decimal. Ex.  $486 = 400 + 80 + 6 = 4 \times 100 + 8 \times 10 + 6 \times 1 = 4 \times 10^2 + 8 \times 10^1 + 6 \times 10^0$ , ou seja,  $486 = 4 \times 10^2 + 8 \times 10^1 + 6 \times 10^0$ .

### 1.4. Sistema Binário

Como o próprio nome já indica tem base 2, pois utiliza apenas dois símbolos ou algarismos: 0 e 1. Assim, a cada posição de cada algarismo corresponde uma potência de 2. Os dígitos binários dependendo do posicionamento o algarismo ou bit terá um peso; o da extrema esquerda será o bit mais significativo (*most significant bit* – MSB) e o da extrema direita o bit menos significativo (*least significant bit* – LSB). Ex.  $10111_{(2)} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 0 + 4 + 2 + 1 = 23$ , ou seja,  $10111_{(2)} = 23_{(10)}$  ou  $10111_{(2)} = 23$

### 1.4.1. Conversão de Decimal em Binário

Na conversão decimal-binário, podem ser utilizados dois métodos: o primeiro que é mais geral, dito das divisões sucessivas, consiste em dividir sucessivamente o número por 2 até obtermos o cociente 0 (zero). O resto dessa divisão colocado na ordem inversa corresponde ao número binário correspondente ao decimal dado (Fig. 2).

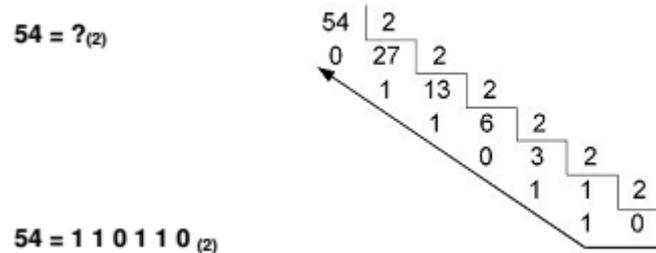


Figura 2 - Conversão decimal para binário – método das divisões sucessivas

O segundo método de conversão consiste em, começando como número decimal a ser convertido, extrair a maior potência de 2 (menor ou igual) possível. Repetindo este processo para o resto dessa subtração até que o resto seja zero. Concluindo, marque com o dígito 1 os expoentes utilizados e com o dígito zero os expoentes não utilizados (Fig. 3).

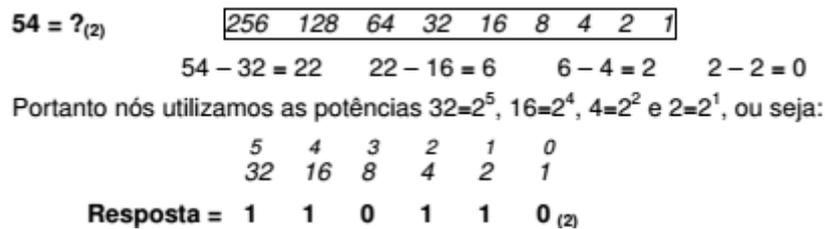


Figura 3 - Conversão decimal para binário – método de conversão

### 1.5. Sistema Octal

O sistema octal ou base 8 é composto por oito símbolos ou dígitos: 0, 1, 2, 3, 4, 5, 6, e 7. Os números binários, são mais apropriados para as máquinas ou computadores, mas para seres humanos são muito trabalhosos. Se considerarmos três dígitos binários, o maior que pode ser expresso por esses três dígitos é  $111_{(2)}$  ou em decimal 7. Como o 7 é também o algarismo mais significativo do sistema octal, conclui-se que com a combinação de três dígitos binários pode-se ter o algarismo octal correspondente; daí também se pode dizer que os números octais têm um terço do comprimento de um número binário e fornecem a mesma informação.

### 1.5.1. Conversão de Binário em Octal

É feita pela combinação de três dígitos binários, podendo assim ter todos os algarismos octais (Fig. 4):

$$\begin{aligned} 11011011_{(2)} &= 11 \ 011 \ 011 = 3 \ 3 \ 3_{(8)} & \rightarrow & 11011011_{(2)} = 333_{(8)} \\ 1011101_{(2)} &= 1 \ 011 \ 101 = 1 \ 3 \ 5_{(8)} & \rightarrow & 1011101_{(2)} = 135_{(8)} \end{aligned}$$

Figura 4 - Conversão de binário para octal

### 1.5.2. Conversão de Octal em Binário

A conversão de uma base em outra é bastante simples, uma vez que se trata da operação inversa, ou seja, basta converter individualmente cada dígito octal em três binários (Fig. 5).

$137_{(8)} = ?_{(2)}$   
 O número 1 equivale a  $001_{(2)}$ , o número 3 igual a  $011_{(2)}$  e o número 7 vale  $111_{(2)}$ .  
 Portanto:  
 $137_{(8)} = 001011111_{(2)}$ , ou seja,  $137_{(8)} = 1011111_{(2)}$ .

Figura 5 - Conversão de octal para binário

### 1.5.3. Conversão de Octal em Decimal

Esta conversão passa pela conversão em binário e posteriormente em decimal, conforme a Figura 6:

$17_{(8)} = ?$   
 $17_{(8)} \rightarrow 001 \ 111_{(2)} \rightarrow 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0 \rightarrow 8 + 4 + 2 + 1 = 15.$

Figura 6 - Conversão de octal em decimal

### 1.5.4. Conversão de Decimal em Octal

Neste caso, devemos passar pelo sistema binário e depois para octal (Fig.7).

$22 = ?_{(8)}$   
 $22 \rightarrow 10110_{(2)} \rightarrow 10 \ 110_{(2)} \rightarrow 26_{(8)}$ , ou seja,  $22 = 26_{(8)}$ .

Figura 7 - Conversão de decimal em octal

## 1.6. Sistema Hexadecimal

O sistema hexadecimal foi adotado por alguns fabricantes para abreviar os códigos binários que ocupavam um espaço enorme e eram difíceis de ser lidos. Dessa maneira foi criada uma linguagem de baixo nível (mais próxima do aspecto eletrônico), chamada ASSEMBLY que utilizava o código hexadecimal. Se considerarmos quatro dígitos binários, o maior número que pode ser expresso por esses quatro dígitos é 1111 ou em decimal 15, da mesma forma que 15 é o algarismo mais significativo do sistema hexadecimal, portanto com a combinação de 4 bits ou dígitos binários pode-se ter o algarismo hexadecimal correspondente. Assim, com esse agrupamento de 4 bits ou dígitos, podem-se definir 16 símbolos, de 0 até 15. Contudo, como não existem símbolos dentro do sistema arábico que possam representar os números decimais entre 10 e 15 sem repetir os símbolos anteriores, foram usadas as letras A, B, C, D, E, F. Portanto, o sistema hexadecimal é formado por 16 símbolos alfanuméricos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

### 1.6.1. Conversão de Hexadecimal em Binário

Basta converter cada dígito hexadecimal em seu similar binário, ou seja, cada dígito em hexa equivale a um grupo de 4 bits (Fig. 8).

$$\begin{array}{l}
 \mathbf{B15}_{(16)} = ?_{(2)} \qquad \mathbf{B}_{(16)} \rightarrow 11 \rightarrow 1011_{(2)} \\
 \qquad \qquad \qquad \qquad \mathbf{1}_{(16)} \rightarrow 1 \rightarrow 0001_{(2)} \\
 \qquad \qquad \qquad \qquad \mathbf{5}_{(16)} \rightarrow 5 \rightarrow 0101_{(2)} \\
 \text{Logo, } \mathbf{B15}_{(16)} = \mathbf{101100010101}_{(2)}
 \end{array}$$

Figura 8 - Conversão hexadecimal em binário

### 1.6.2. Conversão de Binário em Hexadecimal

De maneira análoga, basta realizar o processo inverso de hexadecimal para binário (Fig. 9).

$$\begin{array}{l}
 \mathbf{10011011}_{(2)} = ?_{(16)} \qquad \mathbf{1001}_{(2)} \rightarrow 9 \rightarrow 9_{(16)} \\
 \qquad \qquad \qquad \qquad \mathbf{1011}_{(2)} \rightarrow 11 \rightarrow \mathbf{B}_{(16)} \\
 \text{Portanto, } \mathbf{10011011}_{(2)} = \mathbf{9B}_{(16)}
 \end{array}$$

Figura 9 - Conversão binário em hexadecimal

Na Tabela 2 temos a correspondência de conversão entre os sistemas.

**Tabela 2 - Conversão entre os diferentes sistemas numéricos**

TABELA DE CONVERSÃO			
Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000		8
9	1001		9
10	1010		A
11	1011		B
12	1100		C
13	1101		D
14	1110		E
15	1111		F