

Como funciona a numeração binária

Os Computadores, microprocessadores, microcontroladores operam utilizando lógica digital binária. Assim, a representação de todos os valores nesses equipamentos e componentes é feita com base na numeração binária. Veja como isso é feito e como ler números binários nas entradas e saídas dos circuitos digitais.

Os circuitos eletrônicos não possuem dedos. É evidente também que não seria muito fácil projetar circuitos que sejam capazes de reconhecer 10 níveis de uma tensão ou de outra grandeza elétrica sem o perigo de que, qualquer pequeno problema faça-os o causar confusão.

Uma pequena variação da tensão nestes circuitos pode mudar um 3 para 4 ou vice-versa, afetando os cálculos que ele tenha que realizar.

Muito mais simples para os circuitos eletrônicos é trabalhar com um sistema de numeração que esteja mais de acordo com o seu princípio de funcionamento e isso realmente é feito.

Um circuito eletrônico pode ter ou não ter corrente, pode ter ou não ter tensão, pode receber ou não um pulso elétrico. Também é muito mais fácil diferenciarmos dois estados de elementos indicadores como uma lâmpada acesa ou apagada, uma campainha em silêncio ou tocando.

Ora, os circuitos eletrônicos são mais apropriados para operar com sinais que tenham duas condições possíveis, ou seja, que representem dois dígitos ou algarismos. Também podemos dizer que as regras que regem o funcionamento dos circuitos que operem com apenas duas condições possíveis são muito mais simples.

Houve época em que se tentou trabalhar com as quantidades na forma original analógica, com a criação de computadores capazes de realizar cálculos complexos, mas com o tempo ficou claro que trabalhar com duas condições possíveis apenas para os circuitos, adotando uma lógica digital, era muito mais vantajoso, por diversos motivos.

Assim, o sistema adotado nos circuitos eletrônicos digitais modernos é o sistema binário ou de base 2 onde apenas dois dígitos são usados, correspondentes a duas condições possíveis de um circuito: 0 e 1.

A idéia básica é a mesma usada na representação de quantidades no sistema decimal: atribuir pesos aos dígitos conforme sua posição no número. Para entendermos melhor como tudo isso funciona, vamos tomar como exemplo o valor 1101 que em binário representa o número 13 decimal (veja nota do autor) e ver como isso ocorre.

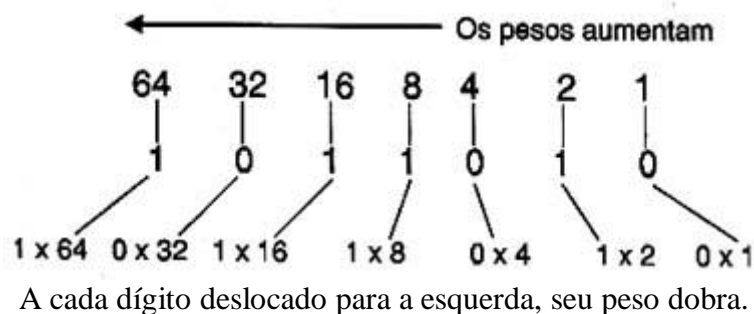
Para não fazermos confusões em relação ao tipo de base que está sendo usada para representar um número ou quantidade, é comum colocarmos ao lado, como índice, a base que está sendo usada.

Assim, ao falarmos em 1101 em binário, escrevemos simplesmente 1101₂ e para representar 13 em decimal, escrevemos 13₁₀.

O primeiro dígito da direita nos indica que temos uma vez o peso deste dígito ou 1. O zero do segundo dígito da direita para a esquerda indica que não temos nada com o peso 2.

Agora o terceiro dígito da direita para a esquerda e que tem peso 4 é um 1, o que indica que temos "uma vez quatro". Finalmente o primeiro dígito da esquerda que é um 1, e que está na posição de peso 8, nos diz que temos "uma vez oito". Somando uma vez oito, com uma vez quatro e uma vez um, temos o total que é justamente a quantidade que conhecemos em decimal como treze.

Basta saber que a cada dígito que nos deslocamos para a esquerda seu peso dobra.



Como não existe um limite para os valores dos pesos, isso significa que é possível representar qualquer quantidade em binário, por maior que seja, simplesmente usando a quantidade apropriada de dígitos.

Para 4 dígitos podemos representar números até 15; para 8 dígitos podemos ir até 255; para 16 dígitos podemos ir até 65 535 e assim por diante.

O leitor deve lembrar-se desses valores limites para 4, 8 e 16 dígitos de um número binário pois eles têm uma grande importância nas aplicações digitais modernas.

Tabela com potências de 2

Potência de 2	Valor
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024
2^{11}	2048
2^{12}	4096
2^{13}	8192
2^{14}	16 384
2^{15}	32 768
2^{16}	65 536
2^{17}	131 072
2^{18}	262 144
2^{19}	524 288
2^{20}	1 048 576

A seguir damos a representação binária dos números decimais até 17 para que o leitor tenha uma idéia de como tudo funciona:

decimal	Binário	decimal	binário
0	0	9	1001
1	1	10	1010
2	10	11	1011
3	11	12	1100
4	100	13	1101
5	101	14	1110
6	110	15	1111
7	111	16	10000
8	1000	17	10001

Para o leitor que pretende entender de eletrônica digital aplicada aos computadores, equipamentos industriais e mecatrônicos existem momentos em que é preciso saber converter uma indicação em binário para o decimal correspondente.

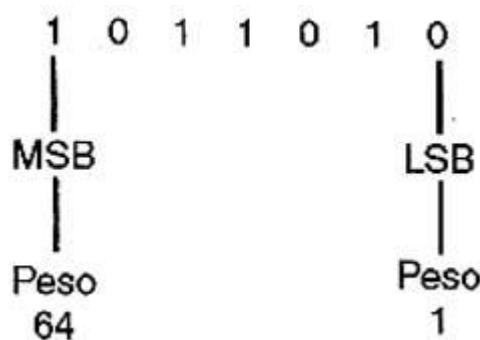
Podemos dar como exemplo o caso de certas placas que são usadas no diagnóstico de computadores e de máquinas industriais as quais possuem um conjunto de LEDs que acendem indicando um número que corresponde a um código de erros. Os LEDs apagados indicam o algarismo 0 e os LEDs acesos indicam o algarismo 1.

Vamos supor que num diagnóstico a sequência de acendimento dos LEDs seja 1010110. De início o leitor precisa saber por onde começar a leitura ou seja, se o de menor peso é o da direita ou da esquerda.

Nas indicações dadas por instrumentos ou mesmo na representação da valores binários, como por exemplo na saída de um circuito é preciso saber qual dos dígitos tem maior peso e qual ter menor peso.

Isso é feito com uma sigla que é adotada normalmente e que refere-se ao dígito, no caso denominado bit.

Assim, para o dígito de menor peso ou bit menos significativo é adotada a sigla LSB (Least Significant Bit) e para o mais significativo é adotada a sigla MSB (Most Significant Bit), conforme mostra a figura 3.



LSB = dígito de menor peso; MSB = dígito de maior peso.

O que fazemos é somar os valores dados pelos dígitos multiplicados pelo peso de sua posição. No caso do valor tomado como exemplo, 1010110, temos:

Dígito		Peso		Valor
1	x	64	=	64
0	x	32	=	0
1	x	16	=	16
0	x	8	=	0
1	x	4	=	4
1	x	2	=	2
0	x	1	=	0

Somando os valores temos:

$$64 + 16 + 4 + 2 = 86$$

O valor decimal de 1010110 é 86, ou usando a notação mais apropriada:

$$1010110_2 = 86_{10}$$

Observe que, para fazer uma conversão, tudo que o leitor tem de fazer é lembrar que a cada dígito que saltamos para a esquerda seu peso dobra na sequência 1, 2, 4, 8, 16, 32, 64, 128, etc.

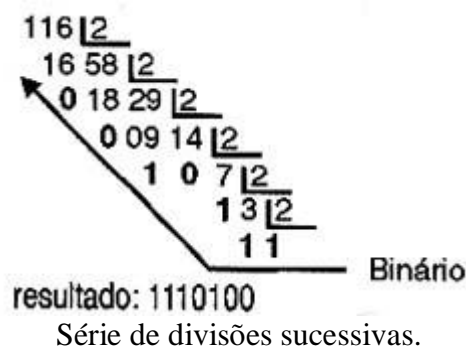
Na prática também pode ocorrer o problema inverso que é a transformação de um valor expresso em decimal (base 10) para a base 2 ou binário. Para esta transformação podemos fazer uso de algoritmo muito simples que memorizado pelo leitor pode ser de grande utilidade, dada sua praticidade.

Para os que não sabem, algoritmo nada mais é do que uma sequência de operações que seguem uma determinada regra e que permite realizar uma operação mais complexa.

Quando você soma os números um sobre o outro (da mesma coluna) ao fazer uma soma e passar para cima os dígitos que excedem o 10, fazendo o conhecido "vai um" você nada mais está fazendo do que usar um algoritmo.

Os microprocessadores dos computadores, máquinas industriais, equipamentos "inteligentes" baseados em DSPs e microprocessadores usam muitos tipos de algoritmos quando fazem suas operações.

Vamos ver então como usar o algoritmo para a conversão de um decimal para binário, como por exemplo o 125. O que fazemos é uma série de divisões sucessivas, conforme mostra a figura 4.



Vamos dividindo os números por 2 até o ponto em que chegamos a um valor menor que 2 e que portanto não mais pode ser dividido. O resultado desta última divisão, ou seja, seu quociente é então o primeiro dígito binário do número convertido. Os demais dígitos são obtidos lendo-se os restos da direita para a esquerda da série de divisões que realizamos. Tudo muito simples e rápido.

Quilo, Mega e Giga

Em eletrônica digital e principalmente a que envolve os computadores, assim como na eletrônica analógica tradicional é comum o uso de prefixos para representar quantidades muito grandes ou muito pequenas. Dessa forma, os prefixos Quilo (k) para representar milhares, Mega (M) para representar milhões e Giga (G) para representar bilhões também são usados em eletrônica digital.

No entanto, diferentemente, do que ocorre com a eletrônica analógica, esses prefixos representam quantidades "levemente" diferentes.

O que ocorre é que quando usamos quilo (k) para representar 1 000 (mil), na base 10, 1000 é um valor inteiro, o que não ocorre com a base 2. Assim, o valor mais próximo da base 2 é 1024. Isso significa que, em eletrônica digital, o valor quilo, na realidade representa 1024. Uma memória que tenha uma capacidade de armazenar 1 quilo bit ou 1 kb, na verdade, armazena 1024 bits. Da mesma forma, quando falamos que um programa de computador exige um espaço de memória de 1 Mega bytes, na verdade, ele exige um espaço de 1 048 576 bytes, já que a potência de 2 mais próxima de 1 milhão é 1 048 576. Em outras palavras, para informática e eletrônica digital o quilo (k) vale 1024, o mega (M) vale 1 048 576 e o Giga vale 1 073 741 824.