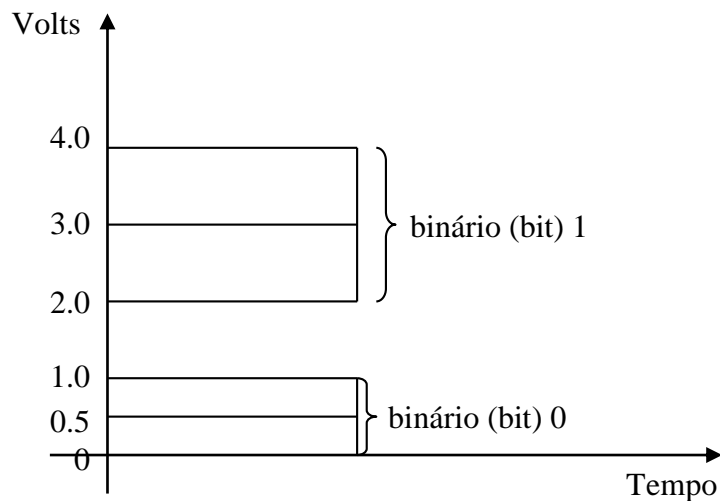


Conceitos da Lógica Digital

Computador digital – máquina projetada para armazenar e manipular informações representadas apenas por algarismos ou dígitos e que só podem assumir dois valores distintos: 0 ou 1.

A informação binária é representada em um sistema digital por quantidades físicas, sinais elétricos, que são gerados e mantidos internamente ou recebidos de elementos externos, em dois níveis de intensidade, cada um correspondente a um valor binário (pode-se armazenar, também, como campos magnéticos e sinais óticos).

Ex.: se escolhermos um sinal de +3V para representar o bit 1 e +0,5V para representar o bit 0, existirá uma faixa de tolerância, pois nenhum sinal é absolutamente preciso em seu valor.



Internamente, um computador é fabricado com circuitos eletrônicos que precisam armazenar os sinais binários e realizar certos tipos de operações com eles. Estes circuitos são chamados de circuitos digitais e são formados de pequenos elementos capazes de manipular grandezas apenas binárias. Estes elementos são chamados de portas (*gates*), pois permitem ou não a passagem de sinais. Os circuitos que contêm as portas lógicas são chamados de circuitos lógicos.

Portanto, um computador digital é fabricado contendo uma infinidade de circuitos lógicos ou portas, convenientemente distribuídos e organizados, de modo que alguns servirão para armazenamento de valores, outros permitirão e controlarão o fluxo de sinais entre os componentes e outros servirão para realizar operações matemáticas.

Para que os computadores consigam compreender as interações humanas, é necessário traduzir essas interações para uma linguagem de máquina, em números binários.

Os números binários são constituídos por 0 e 1 e representam qualquer informação que está salva ou sendo processada pelos computadores. Cada número desses é considerado um bit; portanto, um bit pode ser 0 ou 1.

Os números são representados no sistema decimal, mas os computadores utilizam o sistema binário. Embora empreguem símbolos distintos, os dois sistemas formam números a partir das mesmas regras e podem ser facilmente convertidos entre si. De fato, os números podem ser representados em qualquer base maior ou igual a dois, e essas representações podem ser facilmente convertidas de uma frase para outra.

Com o agrupamento de caracteres, é possível formar palavras. Nas linguagens humanas, as palavras possuem tamanhos diferentes, mas, nos computadores, todas as palavras possuem o mesmo número de caracteres, representados pela sequência de bits (zeros e uns). O tamanho de uma palavra representa a capacidade de dados que podem ser manipulados por determinada arquitetura, pois, quanto maior for o número de bytes, melhor será sua eficiência por meio do aumento do número de instruções inteligíveis. As arquiteturas mais antigas são conhecidas com o sistema de 8 bits (1 byte) porque as palavras eram compostas por um byte. Posteriormente, com a evolução do hardware, os sistemas foram passando para palavras de 16 bits e 32 bits, até chegarem aos modelos atuais de 64 bits.

Sistemas de numeração

Desde o início, o homem vem adotando formas e métodos específicos para representar números, tornando possível, com eles, contar objetos e efetuar operações aritméticas.

Notação posicional

É a forma mais utilizada para representação numérica. Os algarismos assumem valores diferentes dependendo de sua posição no número. O valor total é, então, a soma dos valores relativos de cada algarismo. Ou seja, é a posição do algarismo ou dígito que determina seu valor.

A formação dos números e as operações que efetuadas dependem da quantidade de algarismos diferentes disponíveis no referido sistema.

Sistema decimal

A quantidade de algarismos disponíveis em um sistema de numeração é chamada de base. Assim, o sistema que possui dez algarismos diferentes: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9, é chamado de decimal. Se existem somente dois algarismos, como por exemplo, 0 e 1, o sistema é chamado de binário. Tem-se também o sistema octal e hexadecimal.

Exemplo para o sistema decimal:

$$1\ 3\ 0\ 3_{10} = 1*10^3 + 3*10^2 + 0*10^1 + 3*10^0 = 1000 + 300 + 0 + 3 = 1303.$$

Assim, num sistema qualquer de numeração posicional, um número N é expresso da seguinte forma:

$$N = (d_{n-1}\ d_{n-2}\ d_{n-3}\ \dots\ d_1\ d_0)_b$$

onde:

d indica cada algarismo do número

n-1, n-2, ..., 1, 0 indicam a posição de cada algarismo

b indica a base de numeração

n indica o número de dígitos inteiros

O valor do número é:

$$N = d_{n-1}*b^{n-1} + d_{n-2}*b^{n-2} + \dots + d_1*b^1 + d_0*b^0$$

Base binária

Um computador digital representa internamente as informações em algarismos binários, ou seja, trabalha em base 2. Como os números representados em base 2 são

muito extensos e difícil de manipulação visual, costuma-se representar externamente os valores binários em outras bases de valor mais elevado, como a hexadecimal.

O termo dígito binário é chamado bit, contração de *binary digit*.

Conversão binário-decimal

$$N = d_{n-1} * b^{n-1} + d_{n-2} * b^{n-2} + \dots + d_1 * b^1 + d_0 * b^0$$

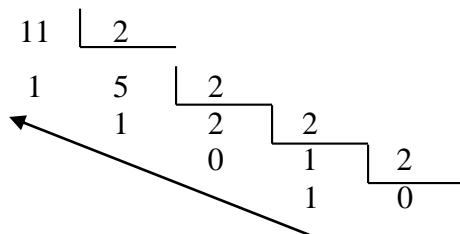
Ex.: $1\ 0\ 1\ 1 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 8 + 0 + 2 + 1 = 11$

Conversão decimal-binário

Algoritmo:

Enquanto o quociente for diferente de 0:

- Fazer sucessivas divisões pela base.



Assim, $11_{10} = 1011_2$

Obs.: sempre que um número for ímpar na notação decimal, deverá terminar com 1 na notação binária.

Sistema Octal

A base utilizada neste sistema é a base 8. Os algarismos utilizados são 0, 1, 2, 3, 4, 5, 6 e 7.

Conversão octal-decimal

Para se converter da base octal para decimal, basta aplicar a fórmula já vista anteriormente.

$$N = d_{n-1} * b^{n-1} + d_{n-2} * b^{n-2} + \dots + d_1 * b^1 + d_0 * b^0$$

Ex.: $743_8 = ()_{10}$

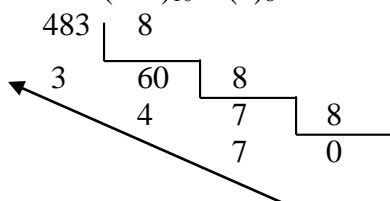
$n = 3$

$N = 7 * 8^2 + 4 * 8^1 + 3 * 8^0 = 7 * 64 + 4 * 8 + 3 * 1 = 448 + 32 + 3 = 483$

Conversão decimal-octal

Para se converter um número decimal para octal basta que se façam divisões sucessivas por 8, até que o valor do quociente seja 0.

Ex.: $(483)_{10} = ()_8$



Assim, $483_{10} = 743_8$

Conversão binário-octal

Para se converter um número que está na base binária para a base decimal deve-se:

- Separar, da direita para a esquerda, o número em grupos de 3 bits
- Se o grupo mais à esquerda não possuir 3 bits, completar com zeros à esquerda
- Substituir os grupos de acordo com a tabela a seguir:

Base Binária	Base Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Ex.: $(1010011111)_2 = ()_8$

001 010 011 111 = 1237

Conversão octal-binário

Para que seja realizada a conversão de octal para binário, basta que se converta cada um dos dígitos, de acordo com a tabela acima.

Ex.: $727_8 = (111\ 010\ 111)_2$

Sistema Hexadecimal

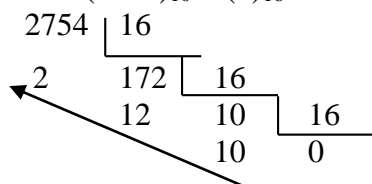
Como já foi dito anteriormente, a base binária possui uma visualização difícil, por isso utiliza-se o sistema hexadecimal para a comunicação com a máquina. Como o próprio nome diz, este sistema utiliza a base 16.

Conversão decimal-hexadecimal

Para se converter um número decimal para hexadecimal, basta que se façam divisões sucessivas por 16, até que o valor do quociente seja 0. Se o número obtido for maior do que 9, este deverá ser substituído de acordo com a tabela abaixo:

10	A
11	B
12	C
13	D
14	E
15	F

Ex.: $(2754)_{10} = ()_{16}$



Assim, $2754_{10} = AC2_{16}$

Conversão hexadecimal-decimal

Para se converter da base hexadecimal para decimal, basta aplicar a fórmula já vista anteriormente.

Ex.: $1EA_{16} = ()_{10}$

$n = 3$

$$N = 1 * 16^2 + 14 * 16^1 + 10 * 16^0 = 1 * 256 + 14 * 16 + 10 * 1 = 256 + 224 + 10 = 490$$

Conversão binário-hexadecimal

Para se converter um número que está na base binária para a base hexadecimal deve-se:

- Separar, da direita para a esquerda, o número em grupos de 4 bits
- Se o grupo mais à esquerda não possuir 4 bits, completar com zeros à esquerda
- Substituir os grupos de acordo com a tabela abaixo:

Base Binária	Base Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Ex.: $(1010011111)_2 = ()_{16}$

$$0010 \ 1001 \ 1111 = 29F$$

Conversão hexadecimal-binário

Para que seja realizada a conversão de hexadecimal para binário, basta que se converta cada um dos dígitos, de acordo com a tabela acima.

Ex.: $2DB_{16} = (0010 \ 1101 \ 1011)_2$

Exercícios

- 1) Converter os valores decimais abaixo para a base binária:
 - a) 37
 - b) 47
 - c) 128
 - d) 257
 - e) 1048

2) Converter os valores binários abaixo para a base decimal:

- a) 1111
- b) 1111 1111
- c) 1100 0001 1010 1111
- d) 1110 0010 0111

3) Converter para a base decimal:

- a) $(27)_8$
- b) $(101101)_2$
- c) $(1010)_2$
- d) $(9AC)_{16}$
- e) $(AFE2)_{16}$

4) Converter para a base binária:

- a) $(327)_8$
- b) $(45)_{10}$
- c) $(272D)_{16}$
- d) $(F50)_{16}$

5) Converter para a base hexadecimal:

- a) $(1111\ 1111)_2$
- b) $(0110\ 1011)_2$
- c) $(65)_{10}$
- d) $(2476)_{10}$
- e) $(45026)_{10}$

Respostas dos exercícios:

- 1) a) 0010 0101; b) 0010 1111; c) 1000 0000; d) 0001 0000 0001; e) 0100 0001 1000
- 2) a) 15; b) 255; c) 49583; d) 3623
- 3) a) 23; b) 45; c) 10; d) 2476; e) 45026
- 4) a) 011 010 111; b) 0010 1101; c) 0010 0111 0010 1101; e) 1111 0101 0000
- 5) a) FF; b) 6B; c) 41; d) 9AC; e) AFE2