

FUNDAMENTOS PARA COMPUTAÇÃO

CAPÍTULO 4 - COMO AS INFORMAÇÕES SÃO REPRESENTADAS E INTERPRETADAS PELO SISTEMA COMPUTACIONAL?

Fábio Tavares Arruda

INICIAR

Introdução

Você já deve ter percebido que em todos os lugares podemos encontrar computadores, mas você já parou para pensar como a informação é representada no computador? E como o computador entende as informações? Independentemente do tipo de computador que você utiliza, seja um *notebook*, um *smartphone*, um *tablet* ou um computador *desktop*, essas ferramentas são sistemas eletrônicos com fios elétricos que precisam representar e interpretar informações. Então, vamos estudar de que maneira as informações são representadas e interpretadas por um sistema computacional.

Para iniciar este estudo, você fará algumas reflexões importantes para que possa integrar os conceitos de como as informações são representadas e interpretadas pelo sistema computacional. Como podemos definir um Sistema de Numeração? Qual a relação dos sistemas de numeração com o computador? Como podemos realizar conversões entre os sistemas de numeração? Como podemos realizar operações aritméticas com os sistemas de numeração?

Você também estudará o conceito de sistema de numeração decimal, binário, octal e hexadecimal, além de como realizar conversão entre os sistemas decimal e binário e entre os sistemas decimal e octal. Além disso, será abordada a aplicação de aritmética de números binários, realizando as operações de soma, subtração, multiplicação e divisão. Por fim, serão abordados os conceitos de conversão entre os sistemas binário e octal e conversão entre os sistemas binário e hexadecimal.

4.1 Sistema de numeração decimal, binário, octal e hexadecimal

Iniciamos nosso estudo considerando como a informação é representada e interpretada pelos computadores. Nos sistemas digitais (computação), é frequente utilizar os diferentes sistemas de numeração para realizar a representação da informação digital. Nesse contexto, você sabe quais são os tipos de sistemas de numeração? O que são os sistemas de numeração posicional e não posicional? Como podemos decompor para obter números na base 10 e na base 2? Quais as diferenças entre os sistemas de numeração?

Existem diversos sistemas de numeração como, por exemplo, o binário, o decimal, o octal e o hexadecimal. Assim, eles podem ser classificados em sistemas de numeração posicionais e não posicionais. Nos posicionais, alterar a posição ocupada por cada algarismo em um número influencia no seu valor, ou seja, o valor de cada algarismo depende da sua posição na composição do número. Um exemplo disso é o sistema decimal. Já nos não posicionais, cada símbolo, independentemente da posição, representa um valor, como exemplo temos o sistema romano.

Os sistemas de numeração são definidos através de suas bases, assim sua base é uma quantidade de algarismos disponíveis na representação de um número. O sistema de numeração decimal tem a base igual a 10, ou seja, possui 10 algarismos;

já o binário tem a base igual a 2; o octal tem a base igual a 8; e o hexadecimal tem a base igual a 16.

O decimal é o mais utilizado e é formado pelos algarismos 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Como dito anteriormente, é um sistema de numeração posicional e é o que utilizamos no nosso dia a dia para representar quantidade, valor monetário etc. É chamado também de sistema de numeração indo-arábico, pois foi concebido pelos hindus e divulgado no ocidente pelos árabes.

No sistema de numeração decimal, podemos chamar cada algarismo de uma ordem e deve ser lido da direita para a esquerda, a cada três ordens temos uma classe. Assim, as classes podem ser utilizadas para ler números muito grandes, colocando-se um ponto para separá-las.

| | Classe dos bilhões | | | Classe dos milhões | | | Classe dos milhares | | | Classe das unidades | | |
|---------|--------------------|-----------------|-----------------|--------------------|----------------|----------------|---------------------|----------------|----------------|---------------------|----------------|----------------|
| Ordem | 12 ^a | 11 ^a | 10 ^a | 9 ^a | 8 ^a | 7 ^a | 6 ^a | 5 ^a | 4 ^a | 3 ^a | 2 ^a | 1 ^a |
| Exemplo | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Tabela 1 - Como um número pode ser representado mostrando suas classes e ordens. Fonte: Elaborada pelo autor, 2018.

No sistema de numeração decimal, as quantidades também podem ser agrupadas de 10 em 10 algarismos, sendo representadas da seguinte forma:

- 10 unidades = 1 dezena;
- 10 dezenas = 1 centena;
- 10 centenas = 1 unidade de milhar;
- e assim por diante.

O sistema de numeração binário é um sistema de numeração posicional que possui apenas dois algarismos, o zero e o um (0 e 1). O sistema binário é o sistema mais utilizado por computadores, uma vez que os sistemas digitais trabalham internamente com dois estados (ligado/desligado, verdadeiro/falso, aberto/fechado).

Dentro dos computadores atuais, a informação é codificada por meio de padrões de 0s e 1s. Esses dígitos são chamados de *bits* (*binary digits* – dígitos binários). Você pode pensar em associar esses *bits* a valores numéricos, já que são representados por números (0s e 1s), mas eles são apenas símbolos cujo significado depende de

onde eles estão sendo utilizados. Algumas vezes, padrões de *bits* são usados para representar valores numéricos; outras vezes, eles representam caracteres e pontuações em um alfabeto; e outras eles representam imagens. (BOOKSHEAR, 2013)



Figura 1 - Comando do usuário são traduzidos para padrões de 0s e 1s para que o hardware entenda.

Fonte: Creativa Images, Shutterstock, 2018.

O sistema de numeração octal utiliza 8 símbolos para a representação de quantidade. No ocidente, esses símbolos são os algarismos arábicos: 0, 1, 2, 3, 4, 5, 6, 7. Inicialmente, o sistema de numeração octal era bastante utilizado nos computadores como uma alternativa ao sistema de numeração binário, depois passou-se a utilizar mais o hexadecimal como alternativa ao binário (WEBER, 2000).

O hexadecimal também é um sistema de numeração posicional com 16 símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Como ele possui 16 símbolos, foi necessário incluir letras para completar o sistema, já que existem apenas dez números (de 0 a 9). Ele é muito utilizado para representar números binários de forma mais compacta. Devido a esse fato, o sistema hexadecimal é muito utilizado em microprocessadores, programação, representação de cores e entre outros. (WEBER, 2000)

VOCÊ SABIA?

Que a empresa Samsung era a maior fabricante de processadores (*chips*) do mundo em 2017? A Intel liderava o *ranking* desde 1993. Com o fim de dezembro de 2017 e o balanço anual das empresas sendo divulgado, os números deixaram o cenário bem claro: enquanto a Intel vendeu US\$ 63 bilhões, a Samsung arrecadou US\$ 69 bilhões com a venda de *chips*. Ficou curioso para saber mais? Acesse: <https://www.tecmundo.com.br/mercado/126707-samsung-confirma-maior-fabricante-chips-mundo-2017.htm> (<https://www.tecmundo.com.br/mercado/126707-samsung-confirma-maior-fabricante-chips-mundo-2017.htm>)>.

Quando consideramos as atividades internas de um computador, vamos nos deparar com padrões de *bits*, aos quais podemos chamar de uma cadeia de *bits* e que podem ser bastante longos. Uma longa cadeia de *bits* é comumente chamada de um fluxo, entretanto os fluxos são difíceis de entender e podem levar a erros ao manipular. Para simplificar a representação desses padrões de *bits*, usamos normalmente uma notação abreviada chamada de notação hexadecimal, que tira proveito do fato de que padrões de *bits* dentro de uma máquina tendem a ter tamanhos em múltiplos de quatro. A notação hexadecimal usa um único símbolo para representar um padrão de quatro *bits*. Por exemplo, uma cadeia de doze *bits* pode ser representada por três símbolos hexadecimais. (BOOKSHEAR, 2013)

VOCÊ SABIA?

Que podemos representar cores através do sistema de numeração hexadecimal? Quando queremos usar cores em uma página Web (um documento HTML) ou estamos utilizando *softwares* para a criação e edição de imagens, precisamos usar os valores das cores em hexadecimais. Os *softwares* podem suportar outras formas de identificar cores, mas o HTML só suporta esse formato para cores, exceto algumas cores que, além do hexadecimal, também aceita com o nome em inglês. Ficou curioso para saber mais? Acesse: <http://www.linhadecodigo.com.br/artigo/3640/cores-em-html-saiba-o-codigo-hexadecimal-de-sua-cor-preferida.aspx> (<http://www.linhadecodigo.com.br/artigo/3640/cores-em-html-saiba-o-codigo-hexadecimal-de-sua-cor-preferida.aspx>)>.

A figura a seguir apresenta o sistema de numeração hexadecimal. A coluna esquerda descreve todos os padrões de *bits* possíveis de tamanho quatro e a coluna direita descreve o símbolo usado em notação hexadecimal para representar o padrão de *bits* à sua esquerda (BOOKSHEAR, 2013). Como exemplo de utilização do sistema de numeração hexadecimal, temos o padrão de *bits* 11110101, que é representado como F5. Para chegar a esse valor, podemos separar o padrão de *bits* em subcadeias de tamanho quatro e representar cada subcadeia por seu equivalente no sistema de numeração hexadecimal, onde 1111 é representado por F e 0101 é representado por 5.

| Padrão de bits | Representação 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 |

Tabela 2 - Cada padrão de 4 bits possui uma representação hexadecimal. Fonte: BOOKSHEAR, 2013, p. 26.

Podemos ainda aplicar os princípios da decomposição para obter números em base 10 e em base 2. Para decompor um número para transformar em base 10, podemos fazer como no exemplo a seguir. Temos o número 7.329 e vamos decompor a soma dos valores relativos a seus algarismos:

- $7.329 = 7000 + 300 + 20 + 9$
- $7000 = 7 \times 1000 = 7 \times 10 \times 10 \times 10 = 7 \times 10^3$
- $300 = 3 \times 10 \times 10 = 3 \times 10^2$
- $20 = 2 \times 10 = 2 \times 10^1$
- $9 = 9$

4.1.1 Conversão entre os sistemas decimal e binário

É possível converter números do sistema decimal para o binário e vice-versa. Vamos começar convertendo números do sistema de numeração binário para o decimal. Para realizar essa conversão, cada algarismo do número binário irá representar uma potência de 2 de acordo com sua posição. Sendo assim, multiplicamos cada dígito do número binário pela potência de sua posição e depois somamos os resultados (WEBER, 2000). Vamos ver um exemplo: temos o número 1010 no sistema de numeração binário e queremos convertê-lo para o sistema de numeração decimal. Veja abaixo como podemos fazer a conversão.

- Número em binário: 1010
- Então fazemos: $(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$
- Calculando a expressão anterior temos: $8 + 0 + 2 + 0$
- Resultado = 10

VOCÊ QUER VER?

A trilogia *Matrix* (1999, 2003, 2003), de Lana Wachowski e Lilly Wachowski, conta a história de Thomas A. Anderson, programador durante o dia e *hacker* à noite, quando invade sistemas de computador ilegalmente usando o apelido de Neo. Na busca por entender o que é a Matrix, descobre que o mundo que conhece até então não existe, é apenas parte de uma simulação interativa neural, chamada Matrix. Os humanos libertados da Matrix possuem a capacidade de entrar e sair dela quando quiserem, conectando seus cérebros e percorrendo a simulação. Assim, a realidade do mundo é que a humanidade vive em um futuro pós-apocalíptico onde as máquinas possuem Inteligência Artificial e entraram em guerra com os humanos quando estes as tentaram destruir. Diante da derrota eminente, a humanidade cobriu o céu de nuvens negras permanentes de forma a bloquear o acesso das máquinas ao Sol, a sua principal fonte de energia.

Agora vamos ver como podemos converter um número do sistema decimal para o binário. Para isso, fazemos a divisão sucessiva por 2, que é a base do sistema binário. Após esse procedimento teremos o último quociente e o agrupamento dos restos de divisão, estes elementos vão representar o número binário correspondente (WEBER, 2000). Vamos ver um exemplo: temos o número 39 no sistema binário e queremos convertê-lo para o sistema decimal. Veja abaixo como podemos realizar a conversão.

- $39 / 2 = 19$ e resta 1
- $19 / 2 = 9$ e resta 1
- $9 / 2 = 4$ e resta 1
- $4 / 2 = 2$ e resta 0
- $2 / 2 = 1$ e resta 0
- Resultado = 100111

Observe que o último resultado também será computado para o resultado final da conversão, logo o número começa com 1 e segue dos restos de baixo para cima.

4.1.2 Conversão entre os sistemas decimal e octal

É possível converter números do sistema octal para o decimal e vice-versa. Assim, para converter números do sistema de numeração octal para o decimal é bem semelhante à conversão do sistema binário para o decimal, a diferença é que invés de utilizar a base 2 utilizamos a base 8. Então, para realizar a conversão cada algarismo vai representar uma potência de 8 de acordo com sua posição, que é a base do sistema octal. Após essa etapa devemos multiplicar cada número pela potência de sua posição e, no final, somar os resultados (WEBER, 2000). Vamos ver um exemplo, temos o número 0723 no sistema octal e vamos convertê-lo para o sistema decimal. Veja abaixo como podemos fazer a conversão.

- Número em octal: 0723
- Então fazemos: $(0 \times 8^3) + (7 \times 8^2) + (2 \times 8^1) + (3 \times 8^0)$
- Calculando a expressão anterior temos: $0 + 448 + 16 + 3$
- Resultado = 467 decimal.

Podemos também converter números do sistema decimal para o octal ao realizar a divisão sucessiva por 8 (base do sistema octal), obtendo um resultado e um resto. O resultado da conversão será dado pelo último quociente e o agrupamento dos

restos de divisão será o número octal. Vamos ver um exemplo: temos o número 407 no sistema decimal e queremos convertê-lo para o sistema octal. Veja abaixo como podemos realizar a conversão.

- $407 / 8 = 50$ e resta 7
- $50 / 8 = 6$ e resta 2
- Resultado = 627 octal

4.2 Aritmética digital

Agora que você refletiu sobre quais sistemas de numeração existem e como o computador realiza a representação da informação digital, pense em quais operações aritméticas podemos realizar com números do sistema de numeração binário e como realizar essas operações.

Assim, um dos principais sistemas de numeração para os sistemas computacionais é o sistema de numeração binário. Você sabia da existência dos números binários? Por que eles são tão importantes? Você já se perguntou como o computador utiliza os números binários?

Como vimos, em sistemas computacionais, pode-se trabalhar com quatro bases: a decimal, para a entrada e saída dos dados (já que nossa sociedade é baseada no sistema decimal); a binária, para os cálculos internos do computador; a hexadecimal, como forma compactada de representação interna; e a octal, pelo mesmo motivo. Entretanto, a escolha das bases 8 e 16 não é aleatória: as transformações entre as bases 2, 8 e 16 podem ser feitas facilmente pelo método da substituição direta. Apesar de a base hexadecimal ter uma representação mais complexa por utilizar letras e números, ela é preferida sobre a base octal por requerer menos espaço para representar os resultados. (WEBER, 2000)

Como sabemos, o sistema binário é constituído por dois símbolos, 0 e 1. Esse sistema numérico é muito importante, pois é utilizado para expressar todas as operações realizadas em um sistema digital/sistema computacional. A arquitetura de um computador é composta por um sistema digital complexo, uma vez que possui processador(es), memórias, circuitos de entrada e saída, entre outros componentes que processam sinais puramente digitais. (PAIXÃO, 2014)

De certa forma, pode-se afirmar que os microprocessadores são formados por inúmeros blocos lógicos complexos, compostos por portas lógicas básicas, cujo funcionamento está apoiado nos postulados da eletrônica digital. Um desses postulados determina que a eletrônica digital opere somente com dois níveis de tensão bem definidos e distintos. Em um sistema digital, independentemente da quantidade de níveis de tensão usada nesse mesmo sistema, há somente dois níveis lógicos (0 e 1). (PAIXÃO, 2014).

Nesse contexto, a informação dentro dos computadores é codificada por meio de padrões de 0s e 1s, e esses dígitos são chamados de *bits*. Temos também o *byte*, que é formado por um conjunto de oito *bits*. Por exemplo: o número 11101010 é composto por 8 *bits*, ou seja, um *byte*. Assim, os computadores podem realizar diversas operações aritméticas nos números binários. Vamos ver agora como realizar as operações aritméticas de adição e subtração com os números binários através da aritmética digital.

4.2.1 Adição e subtração de números binários

Como vimos anteriormente, o computador manipula os dados (números) através do sistema de numeração binário, assim é possível realizar operações aritméticas com esses números, como soma e subtração. Para realizar a soma de números binários, podemos utilizar como base a figura a seguir, a qual nos mostra como somar dois números binários. Então, para realizar a soma de dois números binários colocamos os dois números um abaixo do outro, semelhante a quando vamos fazer a soma de dois números no sistema de numeração decimal. Caso o número de cima seja 0 e o de baixo também seja 0, o resultado da soma é 0. Caso o número de cima seja 0 e o de baixo seja 1, o resultado da soma é 1. Caso o número de cima seja 1 e o de baixo seja 0, o resultado da soma é 1. E caso o número de cima seja 1 e o de baixo também seja 1, o resultado da soma é 0 e vai mais um número 1 para o próximo (da direita para esquerda).

| a | c | d = a + c |
|---|---|--------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 e “vai-um” |

Figura 2 - Como calcular a soma de

números binários. Fonte: WEBER, 2000, p. 13.

Vamos ver um exemplo: temos o número 011100 e queremos somar com o número 011010. Veja abaixo como podemos fazer a soma.

$$\begin{array}{r}
 011100 \\
 + 011010 \\
 \hline
 110110
 \end{array}$$

Podemos observar que o passo-a-passo da soma de 011100 e 011010, lembrando que as operações devem ser realizadas da direita para esquerda, é o seguinte:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$ e vai-um para o próximo
- $1 + 1 = 0$ e vai-um para o próximo, $0 + 1 = 1$
- $1 + 0 = 1$, $1 + 0 = 1$

Para realizar a subtração de números binários, podemos utilizar como base a figura a seguir, a qual nos mostra como somar dois números binários. Então, para realizar a subtração de dois números binários, colocamos esses números um abaixo do outro, semelhante a quando vamos fazer a subtração de dois números no sistema de numeração decimal. Caso o número de cima seja 0 e o de baixo também seja 0, o resultado da subtração é 0. Caso o número de cima seja 0 e o de baixo seja 1, o resultado da subtração é 1 e vem um do próximo. Caso o número de cima seja 1 e o de baixo seja 0, o resultado da subtração é 1. E caso o número de cima seja 1 e o de baixo também seja 1, o resultado da subtração é 0.

| a | c | d = a - c |
|---|---|---------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 e “pede-um” |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Figura 3 - Como calcular a subtração

de números binários. Fonte: WEBER, 2000, p. 27.

Não esqueça que se subtrai as colunas da direita para a esquerda, semelhante a uma subtração decimal. Vamos ver um exemplo: temos o número 111100 e queremos subtrair com o número 011010. Veja abaixo como podemos fazer a subtração.

$$\begin{array}{r}
 111100 \\
 - 011010 \\
 \hline
 10010
 \end{array}$$

Podemos observar que o passo-a-passo da subtração de 111100 e 011010, lembrando que as operações devem ser realizadas da direita para esquerda, é o seguinte:

- $0 - 0 = 0$
- $0 - 1 = 1$ e vai-um para o próximo
- $1 - 1$ (veio da operação anterior) $= 0$, e $0 - 0 = 0$
- $1 - 1 = 0$
- $1 - 0 = 1$

4.3 Aritmética digital

Após entender como o computador utiliza os números binários e como realizar as operações de soma e subtração com esses números, você acha que podemos realizar outras operações aritméticas? Como podemos pensar na estrutura de um

programa ao nos deparar com a solicitação de um cliente? O processo de desenvolvimento de *software* não é semelhante à criação de qualquer outro produto, como um carro, uma peça de roupa, ou uma televisão.

Vimos que os *bytes*, ou o conjunto de 8 *bits*, é muito importante, no entanto não constitui nada além disso. Normalmente, as sequências de *bits* usadas têm um número de *bits* múltiplo de 8 e, por conseguinte, de *bytes*, sendo a vantagem de utilizá-los como unidade mínima de manipulação, em vez de *bits* individuais, facilitar a referência e a manipulação dos números binários. (DELGADO; RIBEIRO, 2017)

O *byte* foi definido para servir de elemento de referência para a construção e funcionamento dos dispositivos de armazenamento e também como referência para os processos de transferência de dados entre periféricos de entrada e saída. Por essa razão, em anúncios de computadores menciona-se que ele possui “8 *Gibabytes* de memória” ou “2 *Gbytes* de memória”, por exemplo. (MONTEIRO, 2007)

VOCÊ QUER LER?

Qual a diferença entre as tecnologias de armazenamento HD e SSD? Você sabia que o HD, ou disco rígido, é utilizado há muitos anos como forma de armazenar informações de forma persistente? E que o SSD surgiu como uma alternativa mais rápida e silenciosa? Então, leia o texto "SSD e HD: entenda a diferença entre as tecnologias" e descubra. Acesse: <<http://www.techtudo.com.br/artigos/noticia/2011/06/qual-diferenca-entre-hd-e-ssd.html> (http://www.techtudo.com.br/artigos/noticia/2011/06/qual-diferenca-entre-hd-e-ssd.html)>.

Com a rápida evolução dos computadores e com o progressivo aumento da capacidade de armazenamento dos computadores, criaram-se mais dois múltiplos das unidades de medidas, para abreviar valores mais elevados: *Terabytes*, para representar um valor igual a 1024 *Gibabytes*, e *Petabytes* para representar 1024 *Terabytes*. (MONTEIRO, 2007)

Os primeiros computadores eram programados de forma direta utilizando números do sistema de numeração binário, além de interruptores e ligações físicas com cabos. Os resultados eram visualizados num conjunto de lâmpadas: uma lâmpada acesa significava 1 e apagada significava 0. (DELGADO; RIBERO, 2017)

Hoje já não é mais assim. Atualmente, os programadores utilizam linguagens de alto nível que, além de serem mais próximas da linguagem humana, são mais simples de serem utilizadas se comparadas com a programação de baixo nível, que são mais próximas da linguagem de máquina. O computador pode executar muitos programas, inclusive de forma “simultânea”.

VOCÊ QUER LER?

Qual a diferença entre as linguagens de programação de alto nível e de baixo nível? Você sabia que existem várias linguagens de programação? E que cada linguagem de programação foi criada com algum objetivo, como por exemplo, facilidade de escrita, facilidade de manutenção, melhora da *performance*?

Acesse: <https://www.infoescola.com/engenharia-de-software/linguagem-de-programacao-de-alto-nivel/> (https://www.infoescola.com/engenharia-de-software/linguagem-de-programacao-de-alto-nivel/)>.

Para entender como os *bits* são armazenados e manipulados dentro de um computador, é conveniente imaginar que o *bit* 0 representa o valor falso e o *bit* 1 representa o valor verdadeiro, pois isso nos permite pensar na manipulação de *bits* como a manipulação de valores verdadeiro/falso. Operações que manipulam valores de verdadeiro ou falso são chamadas de operações booleanas, em homenagem ao matemático George Boole, que foi um pioneiro na área da matemática chamada de lógica. Três das operações booleanas básicas são AND (e), OR (ou) e XOR (ou exclusivo). Em contraste com as operações aritméticas, as operações booleanas combinam valores verdadeiro e falso ao invés de valores numéricos (BROOKSHEAR, 2013).

VOCÊ O CONHECE?

Matemático e filósofo britânico, George Boole foi o criador da álgebra booleana, que foi fundamental para o desenvolvimento da computação moderna. Atualmente, quase todas as linguagens de programação têm um tipo de dados lógico chamado boolean como referência ao seu nome, que pode conter os valores 1 (verdadeiro) ou 0 (falso).

Vamos ver agora como realizar operações aritméticas de multiplicação e divisão com os números binários através da aritmética digital.

4.3.1 Multiplicação e divisão de números binários

Além das operações de soma e subtração, podemos também realizar as operações de multiplicação e divisão em números binários. Para realizar a multiplicação de números binários, podemos utilizar como base os cálculos abaixo, os quais nos mostram como multiplicar dois números binários.

| Multiplicação | Resultado |
|---------------|-----------|
| 0 x 0 | 0 |
| 0 x 1 | 0 |
| 1 x 0 | 0 |
| 1 x 1 | 1 |

Tabela 3 - Como calcular a multiplicação de números

binários. Fonte: Elaborada pelo autor, 2018.

Então, para realizar a multiplicação de dois números binários, os colocamos um abaixo do outro. A operação é semelhante a quando vamos fazer a multiplicação de dois números no sistema de numeração decimal. Caso o número de cima seja 0 e o de baixo também seja 0, o resultado da multiplicação é 0. Caso o número de cima seja 0 e o de baixo seja 1, o resultado da multiplicação é 0. Caso o número de cima seja 1 e o de baixo seja 0, o resultado da multiplicação é 0. E caso o número de cima seja 1 e o de baixo também seja 1, o resultado da multiplicação é 1 (DELGADO; RIBEIRO, 2017). Vamos ver um exemplo: temos o número 101 e queremos multiplicar pelo número 110. Veja abaixo como podemos fazer a multiplicação.

$$\begin{array}{r}
 101 \\
 \times 110 \\
 \hline
 000 \\
 101 \\
 +101 \\
 \hline
 11110
 \end{array}$$

Podemos observar que a multiplicação de números binários é semelhante à multiplicação de números decimais, lembrando que as operações devem ser realizadas da direita para a esquerda. Primeiro multiplica-se cada dígito do segundo número por todos os dígitos do primeiro número e depois soma-se as três linhas resultantes da etapa anterior.

Além disso, podemos ainda realizar a operação de divisão com números binários. Para realizar a divisão de números binários não existe tabela de referência. A divisão funciona de forma semelhante a quando vamos fazer a divisão de dois números no sistema de numeração decimal. Assim, para realizar a divisão de números binários primeiro devemos verificar se o valor do dividendo é maior ou igual ao do divisor, se for maior ou igual, escrevemos o número 1 no quociente. Depois, colocamos o número que está no divisor em baixo do dividendo e fazemos a subtração dos dois números. Em seguida, baixamos o próximo *bit* e fazemos a verificação novamente, se for maior ou igual, escrevemos o número 1 no quociente, se não for, baixamos o próximo *bit* e adicionamos o valor 0 no quociente (DELGADO; RIBEIRO, 2017). Fazemos essas operações até atingir o valor zero, no caso da divisão exata.

Vamos ver um exemplo: temos o número 110111 e queremos a divisão com o número 101. Veja abaixo como podemos fazer a divisão.

$$\begin{array}{r}
 110111 \overline{)101} \\
 \underline{101} \\
 00111 \\
 \underline{101} \\
 0101 \\
 \underline{101} \\
 0000
 \end{array}$$

Podemos observar que o passo-a-passo da divisão de 110111 e 101 é o seguinte:

- observamos que o divisor tem 3 *bits* então vamos pegar os 3 primeiros *bits* do dividendo, mas observamos que não é suficiente, então ao invés de 3 vamos

utilizar 4 *bits* do dividendo;

- depois, repetimos o divisor abaixo dos 4 *bits* que selecionamos e fazemos a subtração deles;
- o resultado dessa subtração é 0011;
- verificamos se o número 0011 é menor ou igual a 101 e identificamos que 0011 é menor que 101. Então, colocamos o número 0 no quociente e baixamos o próximo *bit* do dividendo, que no caso é 1;
- verificamos se o número 00111 é menor ou igual a 101 e identificamos que 00111 é maior que 101. Então, colocamos o número 1 no quociente e repetimos o divisor abaixo do número 00111 para realizar a subtração, obtendo o número 0101;
- verificamos se o número 0101 é menor ou igual a 101 e identificamos que é igual. Então, colocamos o número 1 no quociente e repetimos o divisor abaixo do número 0101 para realizar a subtração, obtendo o número 0000;
- o resultado da divisão é dado pelo número que está no quociente, ou seja, o número binário 1011.

4.4 Relações entre as representações numéricas

Até agora pensamos em como o computador representa e interpreta as informações através dos sistemas de numeração, além de entender como podemos realizar as operações aritméticas de adição, subtração, multiplicação e divisão. Vimos também como converter números do sistema decimal para o binário e do decimal para o octal. Você já se perguntou como podemos converter números do sistema binário para o octal? E do binário para o hexadecimal?

Na figura a seguir, podemos visualizar a representação dos números em cada um dos sistemas de numeração: o binário, o decimal, o octal e o hexadecimal.

| binário | decimal | octal | hexadecimal |
|---------|---------|-------|-------------|
| 0000 | 0 | 00 | 0 |
| 0001 | 1 | 01 | 1 |
| 0010 | 2 | 02 | 2 |
| 0011 | 3 | 03 | 3 |
| 0100 | 4 | 04 | 4 |
| 0101 | 5 | 05 | 5 |
| 0110 | 6 | 06 | 6 |
| 0111 | 7 | 07 | 7 |
| 1000 | 8 | 10 | 8 |
| 1001 | 9 | 11 | 9 |
| 1010 | 10 | 12 | A |
| 1011 | 11 | 13 | B |
| 1100 | 12 | 14 | C |
| 1101 | 13 | 15 | D |
| 1110 | 14 | 16 | E |
| 1111 | 15 | 17 | F |

Figura 4 - Números nos sistemas de numeração binário, decimal, octal e hexadecimal. Fonte: WEBER, 2000, p. 12.

O nosso sistema visual distingue melhor as variações acentuadas entre elementos como, por exemplo, entre B, 3, D, 4, 5, 6, do que diferenças mínimas, tais como apenas 0s e 1s. Por esse motivo, distinguimos melhor as diferenças entre objetos coloridos do que os mesmos objetos em preto e branco, apenas distinguidos por tons diferentes de preto. Nesse contexto, apesar de internamente nos computadores o sistema numérico ser essencialmente binário, costuma-se empregar outros sistemas de numeração para representar externamente os valores armazenados ou manipulados pelos computadores. (MONTEIRO, 2007)

Assim, utiliza-se com frequência os sistemas de numeração octal e hexadecimal por serem bases maiores, além de a conversão da base 2 para elas, e vice-versa, ser mais rápida do que para o sistema decimal. Atualmente, o sistema de numeração hexadecimal é a base mais usada para representar os valores que estão

internamente no computador em binário (MONTEIRO, 2007). Com esse propósito, vamos apresentar como podemos converter os números entre os sistemas binário e octal e entre os binário e hexadecimal.

4.4.1 Conversão entre os sistemas binário e octal

É possível converter números do sistema octal para o binário e vice-versa. Assim, a forma mais simples de converter números do sistema octal para o binário é utilizando a tabela de equivalências entre os números dos dois sistemas, ilustrada na tabela a seguir.

| Número em octal | Número em binário |
|-----------------|-------------------|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

Tabela 4 - Números nos sistemas de numeração octal e binário. Fonte: Elaborada pelo autor, 2018.

Para realizar a conversão, basta separar os algarismos do número octal e substituir pelo seu correspondente no sistema de numeração binário. Vamos ver um exemplo: temos o número 1772 no sistema octal e vamos convertê-lo para o sistema binário. Veja abaixo como podemos fazer a conversão.

- Número em octal: 1772
- Separamos os dígitos do número octal: 1 7 7 2
- Substituímos cada número pelo seu correspondente em binário: 1 = 001; 7 = 111; 7 = 111; 2 = 010
- Resultado: 001 111 111 01

Podemos também converter números do sistema binário para o octal. A forma mais simples é também utilizando a tabela de equivalências entre os números dos dois sistemas, ilustrada na tabela anterior. Então, para realizar a conversão basta separar os algarismos do número binário em grupos de três, da direita para a esquerda, e depois substituir cada trio pelo seu correspondente no sistema de numeração octal. Vamos ver um exemplo: temos o número 1010111100 no sistema binário e queremos convertê-lo para o octal. Veja abaixo como podemos realizar a conversão.

- Número em decimal: 1010111100
- Separamos os dígitos do número decimal em grupos de três: 001 010 111 100
- Substituímos cada trio pelo seu correspondente em octal: 001 = 1; 010 = 2; 111 = 7; 100 = 4
- Resultado: 1274 octal

4.4.2 Conversão entre os sistemas binário e hexadecimal

É possível converter números do sistema octal para o binário e vice-versa. Assim, a forma mais simples de converter números do sistema octal para o binário é utilizando a tabela de equivalências entre os números dos dois sistemas, ilustrada na tabela a seguir.

| Número em hexadecimal | Número em binário |
|-----------------------|-------------------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

Tabela 5 - Números nos sistemas de numeração

hexadecimal e binário. Fonte: Elaborada pelo autor, 2018.

Então, para realizar a conversão, basta separar os algarismos do número hexadecimal e substituir pelo seu correspondente no sistema de numeração binário. Vamos ver um exemplo: temos o número F13B no sistema hexadecimal e vamos convertê-lo para o binário. Veja abaixo como podemos fazer a conversão.

- Número em hexadecimal: F13B
- Separamos os dígitos do número hexadecimal: F 1 3 B
- Substituímos cada número pelo seu correspondente em binário: F = 1111; 1 = 0001; 3 = 0011; B = 1011
- Resultado: 1111000100111011 binário.

CASO

Você precisa ir até a faculdade para a aula, mas percebe que está atrasado e tem a ideia de alugar uma bicicleta para fazer o trajeto. Aproveitando que a bicicleta possui um hodômetro, você o utiliza para verificar a distância percorrida até o destino final e entre determinados pontos do percurso.

No entanto, ao iniciar o percurso, você percebe que o hodômetro mostra a distância através do sistema de numeração hexadecimal. Como você resolveria esse problema?

Esse cenário é bem comum, pois, assim como com o computador, às vezes, podem surgir situações onde seja necessário realizar conversões entre os sistemas de numeração binário, decimal, octal e hexadecimal.

Em um determinado trecho, o hodômetro hexadecimal mostra o número C80. Após rodar alguns quilômetros e chegar ao destino final, o hodômetro apresenta a seguinte leitura: 157C. Quanto foi percorrido em cada um dos trechos (utilizando o sistema de numeração binário)? Qual a distância total percorrida (no sistema de numeração decimal)?

Para resolver esse problema, temos de realizar a conversão dos números hexadecimais em números binários verificando na tabela o que cada algarismo do número hexadecimal representa no sistema de numeração binário. Assim, na primeira medição temos o número binário 110010000000, na segunda temos o número binário 1010101111100 e a distância total percorrida foi 8700 em decimal.

Podemos também converter números do sistema binário para o hexadecimal. A forma mais simples é também utilizando a tabela de equivalências entre os números dos dois sistemas, ilustrada na tabela anterior. Então, para realizar a conversão, basta separar os algarismos do número binário em grupos de quatro dígitos, da direita para a esquerda, e depois substituir cada grupo pelo seu correspondente no sistema de numeração hexadecimal. Vamos ver um exemplo: temos o número 11001010100100011111 no sistema binário e queremos convertê-lo para o sistema hexadecimal. Veja abaixo como podemos realizar a conversão.

- Número em decimal: 11001010100100011111
- Separamos os dígitos do número decimal em grupos de quatro: 1100 1010 1001 0001 1111
- Substituímos cada grupo pelo seu correspondente em hexadecimal: 1100 = C; 1010 = A; 1001 = 9; 0001 = 1; 1111 = F
- Resultado: CA91F hexadecimal.

Síntese

Você concluiu os estudos sobre como as informações são representadas e interpretadas pelo sistema computacional. Com essa discussão, esperamos que você consiga definir um sistema de numeração, os sistemas de numeração existentes, aritmética digital e conversão de números entre os diferentes sistemas de numeração.

Neste capítulo, você teve a oportunidade de:

- compreender os sistemas de numeração decimal, binário, octal e hexadecimal;
- entender como um sistema computacional representa as informações;
- definir como podemos converter números entre os sistemas decimal e binário, decimal e octal, binário e octal, binário e hexadecimal;
- identificar como realizar operações aritméticas de adição, subtração, multiplicação e divisão com números binários.

Bibliografia

AMARIZ, L. C. Linguagem de Programação de Alto Nível. **InfoEscola**, [s/d]. Disponível em: <<https://www.infoescola.com/engenharia-de-software/linguagem-de-programacao-de-alto-nivel/>> (<https://www.infoescola.com/engenharia-de-software/linguagem-de-programacao-de-alto-nivel/>)>. Acesso em: 05/04/2018.

ARRIGONI, R. **Linha de Código**. [s/d]. Disponível em: <<http://www.linhadecodigo.com.br/artigo/3640/cores-em-html-saiba-o-codigo-hexadecimal-de-sua-cor-preferida.aspx>> (<http://www.linhadecodigo.com.br/artigo/3640/cores-em-html-saiba-o-codigo-hexadecimal-de-sua-cor-preferida.aspx>)>. Acesso em: 05/04/2018.

BROOKSHEAR, J. G. **Ciência da computação**: uma visão abrangente. 11. ed. Porto Alegre: Bookman, 2013.

CIRIACO, Douglas. **Tecmundo**. 31/01/2018. Disponível em:
<<https://www.tecmundo.com.br/mercado/126707-samsung-confirma-maior-fabricante-chips-mundo-2017.htm>
(<https://www.tecmundo.com.br/mercado/126707-samsung-confirma-maior-fabricante-chips-mundo-2017.htm>)>. Acesso em: 05/04/2018.

DELGADO, J.; RIBEIRO, C. **Arquitetura de computadores**. Grupo Gen-LTC, 2017.

FREIRE, R. SSD e HD: entenda a diferença entre as tecnologias. **Tech Tudo**, 28/06/2011. Disponível em:
<<http://www.techtudo.com.br/artigos/noticia/2011/06/qual-diferenca-entre-hd-e-ssd.html>
(<http://www.techtudo.com.br/artigos/noticia/2011/06/qual-diferenca-entre-hd-e-ssd.html>)>. Acesso em: 05/04/2018.

MONTEIRO, M. A. **Introdução à Arquitetura de Computadores**. LTC, 2007.

PAIXÃO, R. R. **Arquitetura de computadores - PCs**. São Paulo: Érica, 2014.

WEBER, R. F. **Fundamentos de arquitetura de computadores**. Sagra Luzzatto, 2000.