Organização e Arquitetura de Computadores

Aula 04

Professora Dra. Mayara dos Santos Amarante Lima

Objetivo do Conteúdo

Compreender o sistema de numeração utilizado pelos sistemas computacionais.

Bases e sistemas de numeração

Os computadores e a matemática tem uma ligação íntima - sem o sistema numérico, os computadores não existiriam. E, como visto no capítulo anterior, os computadores atuais evoluíram das calculadoras - seu principal uso ainda é facilitar cálculos que levariam muito tempo para uma pessoa comum. Então vamos construir a base numérica necessária para compreender os componentes internos de um processador.

Sistemas numéricos posicionais

Vamos entender o que exatamente é um sistema numérico posicional. Em um sistema posicional, um mesmo símbolo ou algarismo pode assumir valores diferentes, de acordo com a sua posição.

Então, para se saber o valor de qualquer número que esteja escrito em um sistema posicional, é necessário conhecer o valor posicional de cada símbolo. Com esse propósito, considere os números 4664 e XXXVII.

Agora, observe algumas das propriedades mais importantes em ambas as ocasiões:

Bases e sistemas de numeração

No número arábico decimal 4664, o valor do primeiro algarismo 4 é diferente do valor do último algarismo 4. Enquanto o primeiro indica 4 mil, o último indica 4 unidades. Caso semelhante ocorre com o algarismo 6.

No número romano, cada um dos X vale 10, independentemente de sua posição, acontecendo o mesmo com o I.

Bases e sistemas de numeração

A partir dessas observações, conclui-se que, no sistema arábico, o valor de um determinado símbolo depende diretamente de sua posição, sendo, portanto, um sistema posicional. Já o mesmo não se pode dizer do sistema romano, que, portanto, não é posicional. O cálculo do valor de um determinado símbolo ou algarismo é feito a partir da fórmula do valor posicional, que vale para qualquer base:

Bases e sistemas de numeração

$$V = A * B^P$$

Onde:

V é o valor posicional do símbolo. Exemplo: o valor posicional do símbolo 4 no número decimal 345 é 40.

A é o valor absoluto do símbolo. Exemplo: o valor do símbolo 4 no sistema decimal é 4.

B é a base do sistema numérico, ou seja, é a quantidade de símbolos que se tem para escrever os números, sendo que, no sistema decimal, tem-se um total de 10 símbolos (0 1 2 3 4 5 6 7 8 9); portanto, a base desse sistema é 10.

P é a posição em que o símbolo se encontra no número. Essa posição é definida da direita para esquerda e iniciada em zero.

Exemplos: A posição do símbolo 5 no número 345 é 0 (zero). Já a posição do símbolo 4 no número 345 é 1. E a posição do símbolo 3 no número 345 é 2.

Sistema Decimal

$$V = A * B^P$$

Após saber quanto que um algarismo vale em determinado número, basta fazer o somatório de todos os algarismos para, então, determinar o valor total do número.

Exemplos no sistema decimal:

$$(347)_{10} = 3 * 10^2 + 4 * 10^1 + 7 * 10^0$$

$$(32)_{10} = 3 * 10^1 + 2 * 10^0$$

$$(555)_{10} = 5 * 10^2 + 5 * 10^1 + 5 * 10^0$$

$$(4232)_{10} = 4 * 10^3 + 2 * 10^2 + 3 * 10^1 + 2 * 10^0$$

Sistema binário

Para nós, seres humanos, o sistema numérico decimal é bastante natural: temos dez dedos nas mãos e usamos dez dígitos para contar. No entanto, para o computador, esse sistema não é muito prático, já que os dados precisam ser interpretados pelo estado da corrente elétrica (ligada/desligada, alta/baixa). Para resolver esse impasse, é usual que máquinas digitais como o computador adotem o sistema de numeração binária, que usa apenas dois dígitos: ZERO e UM em sua representação.

Esses dois dígitos podem então ser empregados como correspondentes aos estados desligado e ligado dos diversos componentes digitais.

Sistema binário

Exemplos do sistema binário:

$$(11)_2 = 1 \times 2^1 + 1 \times 2^0 = (3)_{10}$$

$$(111)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (7)_{10}$$

$$(10111)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (23)_{10}$$

Sistema hexadecimal

Normalmente, os computadores usam como unidade mais básica de memória o byte, que é a junção de 8 bits. Com um byte, é possível se representar $2^8 = 256$ valores, além de ficar mais simples trabalhar com o sistema hexadecimal quando usando tal quantidade de informações.

Como no sistema hexadecimal são necessários 16 algarismos para representar um número e levando em conta que a base decimal possui dez algarismos, foi necessário recorrer ao alfabeto para preencher as lacunas. Sendo assim, os 16 dígitos do sistema hexadecimal são 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.:

Sistema hexadecimal

Exemplos do sistema hexadecimal:

$$(12C)_{16} = 1 \times 16^2 + 2 \times 16^1 + C \times 16^0 = (300)_{10}$$

$$(BF)_{16} = B \times 16^1 + F \times 16^0 = (191)_{10}$$

$$(54CC)_{16} = 5 \times 16^3 + 4 \times 16^2 + C \times 16^1 + C \times 16^0 = (21708)_{10}$$

Sistema octal

Inicialmente, o sistema octal foi muito utilizado na computação como uma alternativa mais compacta ao sistema binário na programação em linguagem de máquina. Embora ainda possa ser empregado, é muito mais comum, atualmente, o emprego do sistema hexadecimal. No octal, a base é 8, ou seja, utiliza oito símbolos para a representação de quantidade: 0,1,2,3,4,5,6,7.

Sistema octal

Exemplos do sistema octal:

$$(502)_8 = 5 \times 8^2 + 0 \times 8^1 + 2 \times 8^0 = (322)_{10}$$

$$(22)_8 = 2 \times 8^1 + 2 \times 8^0 = (18)_{10}$$

$$(71)_8 = 7 \times 8^1 + 1 \times 8^0 = (57)_{10}$$

Conversão entre sistemas numéricos

Embora o sistema binário resolva o problema das máquinas eletrônicas, para nós o sistema decimal continua sendo mais fácil de ler e interpretar. E, em muitos locais, usamos o sistema hexadecimal. Então, o que fazer para que possamos conviver harmoniosamente com todos os sistemas numéricos? Convertendo as bases onde e quando necessário.

Para a conversão, lançamos mão de várias técnicas:

- Método das Divisões – de decimal para qualquer base.
- Método das Subtrações – de decimal para qualquer base.
- Método Polinomial – de qualquer base para decimal.
- Método da Substituição Direta – apenas entre binário/octal/hexa.

Método das divisões

Esta técnica consiste em dividir-se sucessivamente o valor, em decimal, pelo quociente correspondente à base desejada. Depois, lê-se os restos (as sobras de cada divisão) do último para o primeiro, para se ter o número.

A Figura 1 mostra a conversão do número 342_{10} para o sistema binário, que resulta no número 101010110_{2}.

Método das divisões

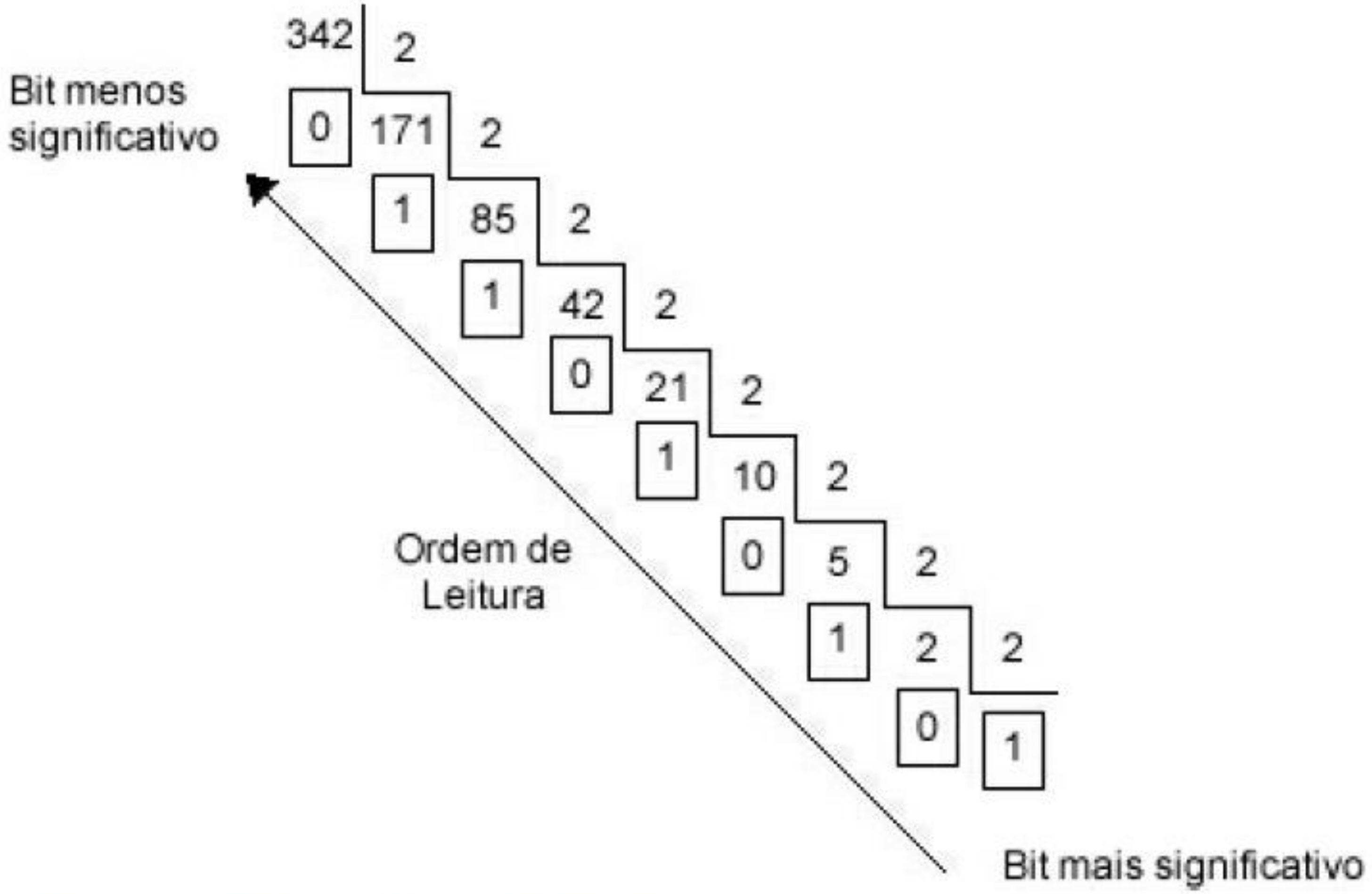

Figura 1 – Conversão de decimal para binário.

Método das subtrações

É o inverso do polinomial, ou seja, procura-se o maior coeficiente com a maior potência possível e subtrai-se o valor do total.

$$
\begin{array}{rrl}
680_{10} = & 680 - 1 \times 2^9 & = 680 - 512 = 168 \\
& 168 - 0 \times 2^8 & \\
& 168 - 1 \times 2^7 & = 168 - 128 = 40 \\
& 40 - 0 \times 2^6 & \\
& 40 - 1 \times 2^5 & = 40 - 32 = 8 \\
& 8 - 0 \times 2^4 & \\
& 8 - 0 \times 2^3 & = 8 - 8 = 0 \\
& 0 \times 2^2 & \\
& 0 \times 2^1 & \\
& 0 \times 2^0 &
\end{array}
$$

Método polinomial

Com este método, basta reescrever os números considerando a base a ser convertida elevada ao expoente da ordem (posição do algarismo no número, começando em zero). A Figura 2 traz um exemplo de conversão do número 10110_2 em binário para a base decimal.

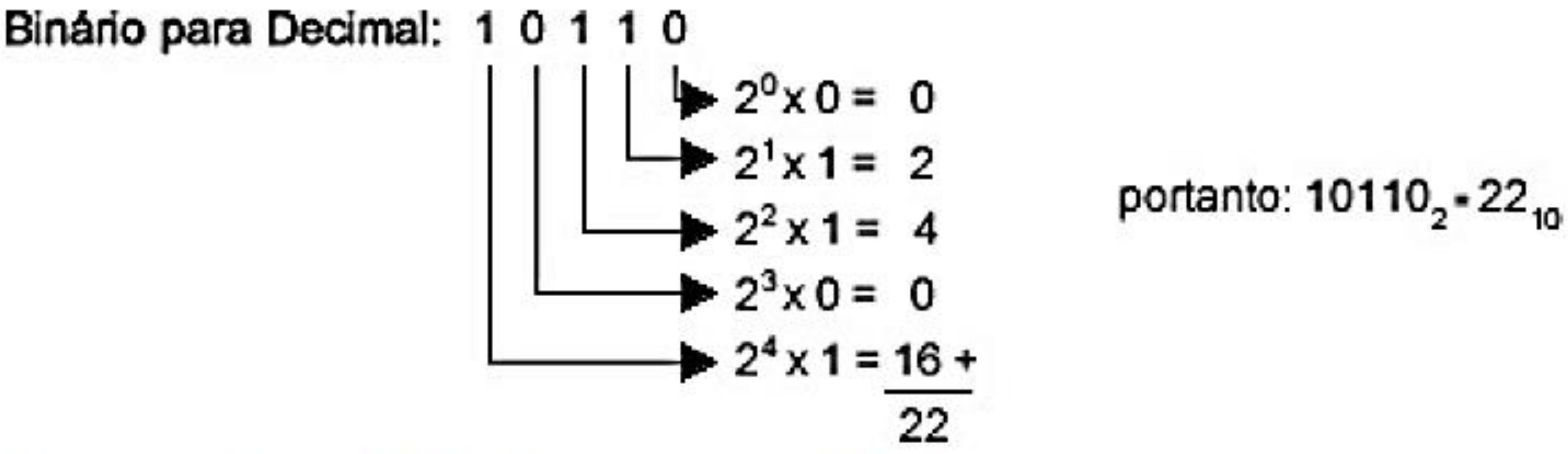

Figura 2 – Conversão de qualquer base para decimal.
Fonte: elaborada pelos autores.

Método da substituição direta

Para este método, faz-se necessário usar a Tabela 2, que traz a representação dos valores de cada base vista até aqui.

Tabela 2 – Conversão básica entre sistemas numéricos.

Tabela 2

Decimal	Binário 4	Binário 3	Hexa	Octal
0	0000	000	0	0
1	0001	001	1	1
2	0010	010	2	2
3	0011	011	3	3
4	0100	100	4	4
5	0101	101	5	5
6	0110	110	6	6
7	0111	111	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	

Fonte: elaborada pelos autores.

Método da substituição direta

Para converter um número binário em octal separa-se a notação binária em grupos de três dígitos e substitui-se cada grupo pelo seu valor octal correspondente.

Exemplo: 1111100101 binário para octal:

001 111 100 101
1 7 4 5

De binário para hexadecimal, apenas muda-se o tamanho de cada grupo para quatro dígitos:

0011 1110 0101
3 E 5

Método da substituição direta

De octal ou hexadecimal para binário, basta substituir pelo conjunto de 3 (octa) ou 4 (hexa) dígitos binários que o representa:

Exemplo 1: Converter o octal 5701 para binário.

5 7 0 1
101 111 000 001

Exemplo 2: Converter o hexadecimal C0Apara binário.

C 0 A
1100 0000 1010

Método da substituição direta

Lembre-se de que binário, octal, decimal e hexadecimal são apenas algumas bases comumente usadas na computação. Apesar de parecer difícil trabalhar com elas, é uma questão de costume — estamos usando diariamente bases numéricas bem diferentes sem nos dar conta.
Exemplos? Que tal 7 (dias em uma semana), 12 (dúzia), 24 (horas em um dia), 30 (dias em um mês), 60 (horas ou minutos), 365 (dias em um ano).
Você sabe por que a dúzia é tão usada em pequenos mercados?
Porque é muito mais fácil dividir — há mais divisores se comparado com decimal:

- Decimal: 1, 2, 5 e 10
- Dúzia: 1, 2, 3, 4, 6 e 12

Representando informações no computador

Como já mencionado, o computador trabalha com o sistema binário. Assim, chamamos a menor unidade de informação no computador de bit (BInary digiT). Devido à necessidade de se trabalhar com grandezas maiores, utiliza-se então unidades maiores, formadas a partir de um conjunto de bits.

Essa unidade maior precisa ter bits suficientes para representar uma gama de informações, tais como: dígitos numéricos, letras do alfabeto, sinais de pontuação, entre outros. No início, considerou-se que um espaço de 256 possibilidades seria o suficiente para tal representação - surge então o byte (BinarY TErm), formado por um grupo de oito bits.

Cada byte armazena o equivalente a um caractere de nossa linguagem.

Representando informações no computador

costuma-se empregar sufixos, que determinam as quantidades de bytes envolvidas em uma determinada grandeza, como pode ser visto na Tabela 3.

Tabela 3 – Grandezas computacionais

Byte	B	8 bits	1 byte
Quilobyte (ou Kilobyte)	KB	1.024 bytes	2^{10} = 1.024 bytes
Megabyte	MB	1.024 KB	2^{20} = 1.048.576 bytes
Gigabyte	GB	1.024 MB	2^{30} = 1.073.741.824 bytes
Terabyte	TB	1.024 GB	2^{40} = 1.099.511.627.776 bytes

Fonte: elaborada pelos autores.

Vídeos

Entenda o sistema binário e como ele funciona. (Informática Básica)
https://www.youtube.com/watch?v=q3xLvOsqhpo

Conversão de bases numéricas: Binário x Decimal. (Informática Básica)
https://youtu.be/VcNSBwQjVnQ?si=tS3ivSF4-cM29vht

Entenda o sistema hexadecimal e como ele funciona. (Informática Básica) (youtube.com)

Conversão de base numérica: HEXADECIMAL x DECIMAL x BINÁRIO. (Informática Básica) (youtube.com)