

Sistemas de Numeração

Conceitos de Aritmética Computacional

1. Introdução

- O homem, através dos tempos, sentiu necessidade da utilização de sistemas numéricos. Existem vários, tais como decimal, binário, octal e hexadecimal.
- Ao longo da história, vários sistemas de numeração foram criados, muitos com formatos totalmente diferentes do que temos hoje.
- O sistema decimal é sem dúvidas o mais importante. Mas na computação, sistemas com base binária e hexadecimal também possuem grande relevância.

2. Sistema Binário

- O sistema binário possui apenas dois algarismos, o 0 (zero) e 1. Todos os números são formados apenas por estes dois algarismos.
- Mas fica a pergunta, como representar o número 2, se temos apenas 0 e 1?
- A resposta é simples, no sistema decimal também temos números de 0 até 9. Representamos o 10 juntando 1 (que representa o agrupamento de 1 dezena) e o 0 (agrupamento de 0 unidades).
- No sistema binário, é a mesma lógica, dois é representado por 10 (1 é o grupo de dois elementos e 0 representa nenhuma unidade).

2. Sistema Binário

- A tabela abaixo mostra a representação de alguns números binários:

DECIMAL	BINÁRIO
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

3. Conversão do Sistema Binário para Decimal

- Para explicar como funciona a conversão vamos utilizar um exemplo com o sistema decimal, no caso o 594.

$$5 \times 100 + 9 \times 10 + 4 \times 1 = 594$$

↓ ↓ ↓
centena dezena unidade

↑ ↑ ↑

$$5 \times 10^2 + 9 \times 10^1 + 4 \times 10^0 = 594$$

Esquematicamente, temos:

100	10	1
5	9	4

→
$$5 \times 100 + 9 \times 10 + 4 \times 1 = 594$$

10^2	10^1	10^0
5	9	4

→
$$5 \times 10^2 + 9 \times 10^1 + 4 \times 10^0 = 594$$

3. Conversão do Sistema Binário para Decimal

- Para conversão de binário, utilizamos o mesmo processo. Por exemplo, o binário 101 é 5 em decimal:

$$\begin{array}{c|c|c} 2^2 & 2^1 & 2^0 \\ \hline 1 & 0 & 1 \end{array}$$

$\underbrace{\hspace{10em}}$

$$\begin{array}{ccc} 1 \times 2^2 & + & 0 \times 2^1 & + & 1 \times 2^0 \\ \updownarrow & & \updownarrow & & \updownarrow \\ 1 \times 4 & + & 0 \times 2 & + & 1 \times 1 = 5 \end{array}$$

3. Conversão do Sistema Binário para Decimal

- Exercícios - Converta os números binários a seguir para decimal:

1. 1001

2. 01110

3. 1010

4. 1100110001

3. Conversão do Sistema Binário para Decimal

- Resolução:

2^3	2^2	2^1	2^0
1	0	0	1

$$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 =$$
$$1 \times 8 + 1 \times 1 = 9_{10} \therefore 1001_2 = 9_{10}$$

2^3	2^2	2^1	2^0
1	1	1	0

$$1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 =$$
$$8 + 4 + 2 + 0 = 14_{10} \therefore 1110_2 = 14_{10}$$

2^3	2^2	2^1	2^0
1	0	1	0

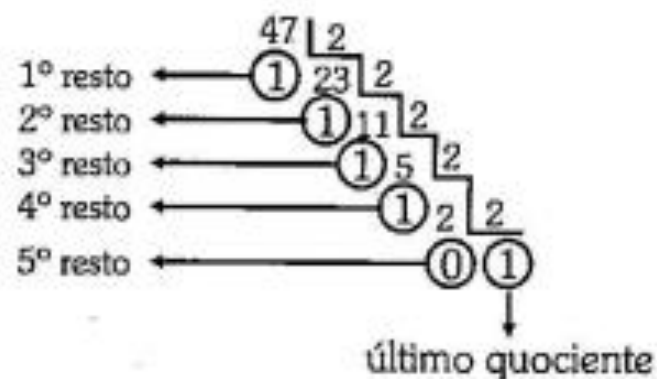
$$1 \times 2^3 + 1 \times 2^1 = 10_{10} \therefore 1010_2 = 10_{10}$$

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	0	1	1	0	0	0	1

$$1 \times 2^9 + 1 \times 2^8 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^0 =$$
$$1 \times 512 + 1 \times 256 + 1 \times 32 + 1 \times 16 + 1 \times 1 = 817_{10}$$

4. Conversão do Sistema Decimal para Binário

- Para a conversão basta dividir o número decimal por 2 até que não seja mais possível, como mostra o exemplo abaixo com o número 47:



O último quociente será algarismo mais significativo e ficará colocado à esquerda. Os outros algarismos seguem-se na ordem até o 1º resto:

1	0	1	1	1	1
↑	↑	↑	↑	↑	↑
último	5º	4º	3º	2º	1º
quociente	resto	resto	resto	resto	resto

$$\therefore 101111_2 = 47_{10}$$

4. Conversão do Sistema Decimal para Binário

- Exercícios - Converta os números decimais em binário:

1. 400

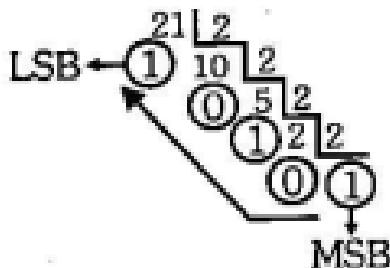
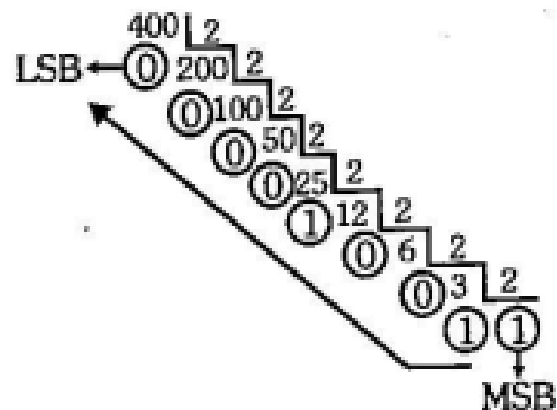
2. 21

3. 552

4. 715

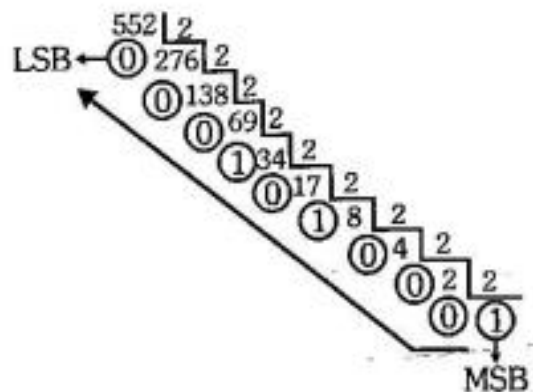
4. Conversão do Sistema Decimal para Binário

- Resolução:



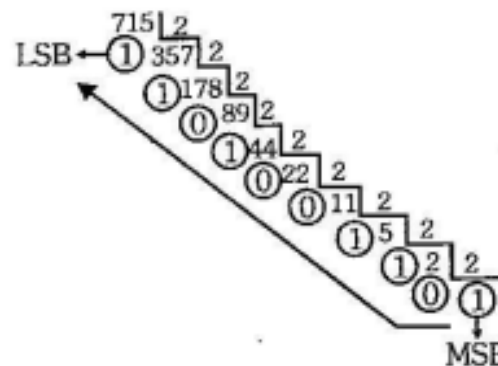
$$\therefore 21_{10} = 10101_2$$

$$\text{Verificação: } 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^0 = 21$$



$$\therefore 1000101000_2 = 552_{10}$$

$$\text{Verificação: } 2^9 + 2^5 + 2^3 = 512 + 32 + 8 = 552_{10}$$



$$\therefore 715_{10} = 1011001011_2$$

$$\begin{aligned} \text{Verificação: } 2^9 + 2^7 + 2^6 + 2^3 + 2^1 + 2^0 = \\ 512 + 128 + 64 + 8 + 2 + 1 = 715_{10} \end{aligned}$$

5. Conversão de Binário Fracionário em Decimal

- Até agora, trabalhamos apenas com números inteiros. Para entender como funciona um número fracionário, vamos usar como exemplo um número decimal 10,5:

10^1	10^0	10^{-1}
1	0	5

da tabela resulta: $1 \times 10^1 + 0 \times 10^0 + 5 \times 10^{-1} = 10,5$

Para números binários agimos da mesma forma. Para exemplificar vamos transformar em decimal o número $101,101_2$:

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
1	0	1	1	0	1

podemos escrever:

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

$$= 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times \frac{1}{2} + 0 \times \frac{1}{4} + 1 \times \frac{1}{8} =$$

$$4 + 1 + 0,5 + 0,125 = 5,625_{10}$$

$$\therefore 101,101_2 = 5,625_{10}$$

5. Conversão de Binário Fracionário em Decimal

- Exercícios - Converta os número binários fracionários em decimal:

1. 10,8125

2. 111,001

3. 100,11001

5. Conversão de Binário Fracionário em Decimal

- Resolução:

2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
1	0	1	0	1	1	0	1

$$1 \times 2^3 + 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} =$$

$$1 \times 8 + 1 \times 2 + 1 \times \frac{1}{2} + 1 \times \frac{1}{4} + 1 \times \frac{1}{16} =$$

$$8 + 2 + 0,5 + 0,25 + 0,0625 = 10,8125_{10}$$

$$\therefore 1010,1101_2 = 10,8125_{10}$$

Converta o número binário $111,001_2$ em decimal.

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
1	1	1	0	0	1

$$1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} =$$

$$4 + 2 + 1 + 0,125 = 7,125_{10} \therefore 111,001_2 = 7,125_{10}$$

Converta o número $100,11001_2$ em decimal.

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
1	0	0	1	1	0	0	1

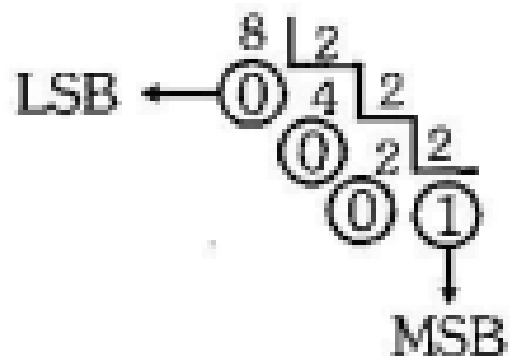
$$1 \times 2^2 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-5} =$$

$$4 + 0,5 + 0,25 + 0,03125 = 4,78125_{10}$$

$$\therefore 100,11001_2 = 4,78125_{10}$$

6. Conversão de Decimal Fracionário em Binário

- Essa conversão ocorre em duas etapas: Primeiro calcula a parte inteira, em seguida a parte decimal. Como exemplo, vamos converter 8,375 em binário:



$$\therefore 8_{10} = 1000_2$$

6. Conversão de Decimal Fracionário em Binário

- Para parte fracionária, faremos sucessivas multiplicações pela base (2), até atingir zero:

$$\begin{array}{rcl} 0,375 & \rightarrow & \text{parte fracionária} \\ \times 2 & \rightarrow & \text{base do sistema} \\ \hline \text{primeiro algarismo} \leftarrow \boxed{0},750 & & \\ \times 2 & & \\ \hline \boxed{1},500 & & \\ \downarrow & \rightarrow & \text{segundo algarismo} \end{array}$$

Quando atingirmos o número 1, e a parte do número após a vírgula não for nula, separamos esta última e reiniciamos o processo:

$$\begin{array}{rcl} 0,500 & & \\ \times 2 & & \\ \hline \text{terceiro algarismo} \leftarrow \boxed{1},000 & \rightarrow & \text{o processo pára aqui, pois a parte do} \\ & & \text{número depois da vírgula é nula.} \end{array}$$

Assim sendo, podemos escrever: $0,011_2 = 0,375_{10}$

Para completarmos a conversão, efetuamos a composição da parte inteira com a fracionária:

$$1000,011_2 \therefore 8,375_{10} = 1000,011_2$$

6. Conversão de Decimal Fracionário em Binário

- Exercícios - Converta os números decimais fracionários para binário:

1. 4,8

2. 3,380

3. 57,3

6. Conversão de Decimal Fracionário em Binário

- Resolução: 4,8

O primeiro passo é transformar a parte inteira do número: $4_{10} = 100_2$.

O próximo passo é converter a parte fracionária utilizando a regra já explicada:

$$\begin{array}{r} 0,8 \\ \times 2 \\ \hline \text{primeiro algarismo} \leftarrow \overline{1},6 \end{array}$$

Por atingir o valor, separamos a parte posterior à vírgula e reiniciamos o processo:

$$\begin{array}{r} 0,6 \\ \times 2 \\ \hline \text{segundo algarismo} \leftarrow \overline{1},2 \end{array}$$

Novamente, reiniciamos o processo:

$$\begin{array}{r} 0,2 \\ \times 2 \\ \hline \text{terceiro algarismo} \leftarrow \overline{0},4 \end{array}$$

$$\begin{array}{r} 0,4 \\ \times 2 \\ \hline \text{quarto algarismo} \leftarrow \overline{0},8 \end{array}$$

Podemos notar que o número 0,8 tornou a aparecer, logo se continuarmos o processo, teremos a mesma sequência já vista até aqui. Este é um caso equivalente a uma dízima.

Temos, então:

$$0,8_{10} = (0, \underbrace{1100}_{\text{seqüência calculada}} \underbrace{11001100 \dots}_{\text{repetições}})_2$$

$$\text{logo: } 4,8_{10} = (100, 1100110011001100 \dots)_2$$

6. Conversão de Decimal Fracionário em Binário

- Resolução: 3,380

• Converta número 3,380 em binário.

Conversão da parte inteira:

$$3_{10} = 11_2$$

Conversão da parte fracionária:

$$\begin{array}{rcl} & 0,38 & \\ & \times 2 & \\ 1^\circ \leftarrow & \overline{0},76 & \\ & \times 2 & \\ 2^\circ \leftarrow & \overline{1},52 & \\ & 0,52 & \\ & \times 2 & \\ 3^\circ \leftarrow & \overline{1},04 & \\ & 0,04 & \\ & \times 2 & \\ 4^\circ \leftarrow & \overline{0},08 & \\ & \times 2 & \\ 5^\circ \leftarrow & \overline{0},16 & \\ & \times 2 & \\ 6^\circ \leftarrow & \overline{0},32 & \\ & \times 2 & \\ 7^\circ \leftarrow & \overline{0},64 & \\ & \times 2 & \\ 8^\circ \leftarrow & \overline{1},28 & \\ & 0,28 & \\ & \times 2 & \\ 9^\circ \leftarrow & \overline{0},56 & \end{array}$$

No caso, temos:

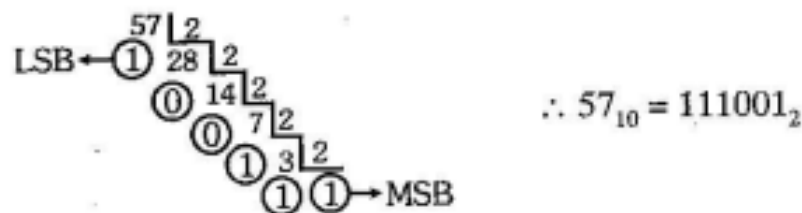
$$0,011000010_2 = 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-8} = 0,37890625_{10}$$

Se aproximarmos o número decimal em duas casas, teremos 0,38, logo, para uma precisão de duas casas decimais é suficiente que tenhamos seguido o método até aí. Podemos escrever, então:

$$0,38_{10} = 0,01100001_2 \therefore 3,38_{10} = 11,01100001_2$$

6. Conversão de Decimal Fracionário em Binário

- Resolução: 57,3



Conversão da parte fracionária:

$ \begin{array}{r} 0,3 \\ \times 2 \\ \hline 0,6 \\ \times 2 \\ \hline 1,2 \\ 0,2 \\ \times 2 \\ \hline 0,4 \\ \times 2 \\ \hline 0,8 \\ \times 2 \\ \hline 1,6 \end{array} $	$ \begin{array}{r} 0,6 \\ \times 2 \\ \hline 1,2 \\ 0,2 \\ \times 2 \\ \hline 0,4 \\ \vdots \end{array} $	$ \left. \begin{array}{c} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{trecho} \\ \text{repetitivo} \end{array} $

Temos, então: $0,3_{10} = (0,0100110011001...)_{21}$

$\therefore 57,3_{10} = (111001,0100110011001...)_{21}$

7. Sistema Octal

- O sistema Octal (base 8) é formado pelos algarismos 0,1,2,3,4,5,6 e 7. A lógica octal é similar a binária, entretanto ele é pouco utilizado na prática. A tabela abaixo mostra alguns exemplos:

DECIMAL	OCTAL
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20

8. Conversão de Octal para Decimal

- A conversão é muito similar ao caso do sistema binário. Como exemplo, vamos converter 144 (na base octal) para decimal:

8^2	8^1	8^0
1	4	4

$$1 \times 8^2 + 4 \times 8^1 + 4 \times 8^0 =$$

$$1 \times 64 + 4 \times 8 + 4 \times 1 = 64 + 32 + 4 = 100_{10}$$

$$\therefore 144_8 = 100_{10}$$

9. Conversão de Decimal para Octal

- A conversão também é por divisão sucessiva por 8. Como exemplo, vamos converter 92 (base decimal) para octal:

1ª resto ← 4 92 | 8
2ª resto ← 3 11 | 8
último quociente ← 1 1

$\therefore 92_{10} = 134_8$

10. Conversão de Octal para Binário

- É uma conversão bastante simples. Por exemplo, vamos converter 27 (base octal) para binário. Para isso, basta converter os algarismos separadamente e depois fazer o agrupamento:

$$\begin{array}{c} 2 \\ \hline 010 \end{array} \quad \begin{array}{c} 7 \\ \hline 111 \end{array}$$

$$\therefore 27_8 = 10111_2$$

11. Conversão de Binário para Octal

- Neste caso, vamos aplicar o processo inverso. Basta separar os binários da direita para esquerda em grupos de 3 bits, e fazer a conversão individual para octal, como mostra o exemplo de conversão do número binário 110010 para octal:

110 010

Efetuando, agora, a conversão de cada grupo de bits diretamente para o sistema octal, temos:

$\begin{array}{c} \underline{110} \\ 6 \end{array} \quad \begin{array}{c} \underline{010} \\ 2 \end{array}$

O número convertido será composto pela união dos algarismos obtidos.

$$\therefore 110010_2 = 62_8$$

12. Sistema Hexadecimal de Numeração

- Este sistema possui 16 algarismos, sendo sua base igual a 16. Ela é formada pelos algarismos 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E e F. Observe que A representa 10, B representa 11, e assim sucessivamente, até F que representa 15.
- Este sistema é muito utilizado na área de microprocessadores e também no mapeamento de memórias de sistemas digitais. Atualmente é um dos sistemas mais importantes da computação.

DECIMAL	HEXADECIMAL
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

10	A
11	B
12	C
13	D
14	E
15	F
16	10
17	11
18	12
19	13
20	14

12. Conversão de Hexadecimal para Decimal

- A conversão é similar as demais bases numéricas. Como exemplo, vamos converter 3F (base hexadecimal) para decimal:

16^1		16^0
3		F

$$3 \times 16^1 + F \times 16^0 =$$

sendo $F_{16} = 15_{10}$, substituindo temos:

$$3 \times 16^1 + 15 \times 16^0 = 3 \times 16 + 15 \times 1 = 63_{10}$$

$$\therefore 3F_{16} = 63_{10}$$

13. Conversão de Decimal para Hexadecimal

- A conversão é feita por meio de divisões sucessivas por 16. Como exemplo, vamos converter 1000 (base decimal) para hexadecimal:



Sendo $14_{10} = E_{16}$, temos: $3E8_{16}$

$$\therefore 1000_{10} = 3E8_{16}$$

14. Conversão de Hexadecimal para Binário

- A conversão é similar a do sistema octal para binário, mas neste caso precisamos de 4 bits para representar cada algarismo. Como exemplo, vamos converter C13 (base hexadecimal) para binário:

$$\begin{array}{c} \text{C} \\ \hline 1100 \end{array} \Rightarrow (C_{16} = 12_{10}) \quad \begin{array}{c} 1 \\ \hline 0001 \end{array} \quad \begin{array}{c} 3 \\ \hline 0011 \end{array}$$

$$\therefore C13_{16} = 110000010011_2$$

15. Conversão de Binário para Hexadecimal

- Análogo a conversão de binário para octal, mas o agrupamento é de 4bits (ao invés de 3 como no octal). Como exemplo, vamos converter o binário 10011000 para hexadecimal:

$$\begin{array}{cc} \underline{1001} & \underline{1000} \\ 9 & 8 \end{array} \quad \therefore 10011000_2 = 98_{16}$$

16. Operações Aritméticas no Sistema Binário

- Na área de microprocessadores, o estudo das operações aritméticas no sistema binário é muito importante para entendimento da arquitetura e de como os circuitos aritméticos funcionam.
- O processador faz todos os cálculos utilizando circuitos eletrônicos, principalmente a ULA. Cálculos mais complexos podem exigir processadores específicos para esta finalidade, ou a incorporação de coprocessadores aritméticos.

17. Adição no Sistema Binário

- Adição é muito similar ao sistema decimal.

Para efetuarmos a adição no sistema binário, devemos agir como numa adição convencional no sistema decimal, lembrando que, no sistema binário, temos apenas 2 algarismos. Temos, então:

0	0	1	1
+0	+1	+0	+1
0	1	1	10

Convém observar que no sistema decimal $1 + 1 = 2$ e no sistema binário representamos o número 2_{10} por 10_2 . Pela operação realizada, notamos a regra de transporte para a próxima coluna: $1 + 1 = 0$ e transporta 1 “vai um”.

A operação de transporte também é denominada **carry**, termo derivado do inglês.

17. Adição no Sistema Binário

- Como exemplo, vamos calcular as soma binárias $11 + 10$ e $110 + 111$ (ambos representados em binário):

$$\begin{array}{r} 1 \leftarrow \\ + 11 \\ 10 \\ \hline 101 \end{array} \quad \left. \begin{array}{l} 1+1=0 \text{ e transporta } 1 \end{array} \right\}$$

$$\therefore 11_2 + 10_2 = 101_2$$

Verificação: $(3_{10} + 2_{10} = 5_{10})$

Como outro exemplo, vamos efetuar $110_2 + 111_2$:

$$1+1+1 = 11 \quad \begin{array}{r} \Downarrow \Downarrow \\ 11 \\ + 110 \\ 111 \\ \hline 1101 \end{array}$$

$$\therefore 110_2 + 111_2 = 1101_2$$

Verificação: $(6_{10} + 7_{10} = 13_{10})$

18. Subtração no Sistema Binário

- A subtração tem algumas regras que precisam ser gravadas:

O método de resolução é análogo a uma subtração no sistema decimal.
Temos, então:

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ -0 \quad -1 \quad -0 \quad -1 \\ \hline 0 \quad 1 \quad 1 \quad 0 \end{array}$$

Observamos que para o caso 0-1, o resultado será igual a 1, porém haverá um transporte para a coluna seguinte que deve ser acumulado no subtraendo e, obviamente, subtraído do minuendo.

Para exemplificar, vamos efetuar a operação $111_2 - 100_2$:

$$\begin{array}{r} 1 \quad 1 \quad 1 \\ -1 \quad 0 \quad 0 \\ \hline 0 \quad 1 \quad 1 \end{array}$$

$$\therefore 111_2 - 100_2 = 11_2 \quad (7_{10} - 4_{10} = 3_{10})$$

18. Subtração no Sistema Binário

- Para entender na prática a questão do 0-1, vamos fazer o exemplo $1000 - 111$ (base binária):

$$\begin{array}{r} 1000 \\ - 111 \\ \hline 1 \end{array}$$

0-1=1 e transporta 1 para a coluna seguinte

$$\begin{array}{r} 1000 \\ - 111 \\ \hline 01 \end{array}$$

transporte anterior
0-1-1=0 e transporta 1 para a coluna seguinte

$$\begin{array}{r} 1000 \\ - 111 \\ \hline 001 \end{array}$$

0-1-1=0 e transporta 1 para a coluna seguinte

$$\begin{array}{r} 1000 \\ - 111 \\ \hline 0001 \end{array}$$

transporte anterior
1-1=0

$$\therefore 1000_2 - 111_2 = 0001_2$$

18. Subtração no Sistema Binário

- Exemplo: 11000-111

A handwritten diagram illustrating the binary subtraction of 111 from 11000. The numbers are aligned vertically: 11000 on top and 111 below it, with a minus sign to the left of the 111. A horizontal line is drawn under the 111. The result 10001 is written below the line. Dashed lines and arrows show the borrowing process: a dashed line connects the first 0 of the minuend to the first 1 of the subtrahend, and another dashed line connects the second 0 to the second 1. Arrows indicate the flow of the borrow from right to left.

$$\begin{array}{r} 11000 \\ - 111 \\ \hline 10001 \end{array}$$

19. Multiplicação no Sistema Binário

- A imagem abaixo mostra os possíveis resultados das multiplicações:

Procede-se como em uma multiplicação no sistema decimal. Assim sendo, temos:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Para exemplificar, vamos efetuar a operação $11010_2 \times 10_2$:

$$\begin{array}{r} 11010 \\ \times 10 \\ \hline 00000 \\ 11010+ \\ \hline 110100 \end{array}$$

$$\therefore 11010_2 \times 10_2 = 110100_2$$

19. Multiplicação no Sistema Binário

- Exemplo: 1100 x 11:

$$\begin{array}{r} 1100 \\ \times 11 \\ \hline 1100 \\ 1100+ \\ \hline 100100 \end{array}$$

20. Representação de Binários Positivos e Negativos

- Temos duas formas de representar sinal: Utilizando um **bit de sinal**, ou **complemento de 2**. A primeira forma consiste em utilizar um bit de sinal mais à esquerda para representar se é negativo ou positivo. Se o bit de sinal for 0, é positivo. Se o bit de sinal for 1 é negativo. Essa forma de representação é chamada de **sinal módulo**.

Para exemplificar o exposto, vamos representar os números $+35_{10}$ e -73_{10} em binário utilizando a notação sinal-módulo:

$$35_{10} = 100011_2$$

$$\therefore +100011_2 = 0100011_2$$

└ bit de sinal (0 → indica número positivo)

$$73_{10} = 1001001_2$$

$$\therefore -1001001_2 = 11001001_2$$

└ bit de sinal (1 → indica número negativo)

20. Representação de Binários Positivos e Negativos

- Já o segundo método consiste em representar os números negativos determinando o complemento de 2 dos números. Para isso, deve ser determinado o complemento de 1 (inverter os bits) e somar 1. No exemplo, vamos representar o -155 em complemento de 2:
- O 155 em binário é 10011011. Vamos determinar o complemento de 1, de pois somar 1 para formar o complemento de 2 e determinar -155:

Número binário:	11001101
Complemento de 1:	00110010
	+ 1
Complemento de 2:	00110011

20. Representação de Binários Positivos e Negativos

- Observe na tabela abaixo que na notação de complemento de 2 os binários positivos são idênticos ao binário padrão. Apenas os negativos são diferentes (complemento de 1 + 1):

DECIMAL	-9	-8	-7	-6	-5	-4	-3	-2	-1
BINÁRIO	-1001	-1000	-0111	-0110	-0101	-0100	-0011	-0010	-0001
COMPL. DE 2	0111	1000	1001	1010	1011	1100	1101	1110	1111

DECIMAL	0	+1	+2	+3	+4	+5	+6	+7	+8	+9
BINÁRIO	0000	+0001	+0010	+0011	+0100	+0101	+0110	+0111	+1000	+1001
COMPL. DE 2	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

20. Representação de Binários Positivos e Negativos

- Outra informação importante é como fazer o processo contrário do complemento de 2 voltar para a representação binária. Para isso, basta fazer o complemento de 2 do número que está em complemento de 2.
- Para exemplificar, vamos voltar o 1011 (-5 na notação de complemento de 2) para binário:

$$\begin{array}{r} \phantom{\text{Complemento de 1:}} 1011 \\ \text{Complemento de 1: } 0100 \\ \phantom{\text{Complemento de 1:}} + 1 \\ \hline \text{Complemento de 2: } 0101 \end{array}$$

∴ Extraíndo novamente o complemento de 2 do número 1011_2 obtemos -
 0101_2 (-5_{10}), que é o mesmo número, porém, na notação normal.