

Sistemas de Numeração

Bases numéricas:

decimal

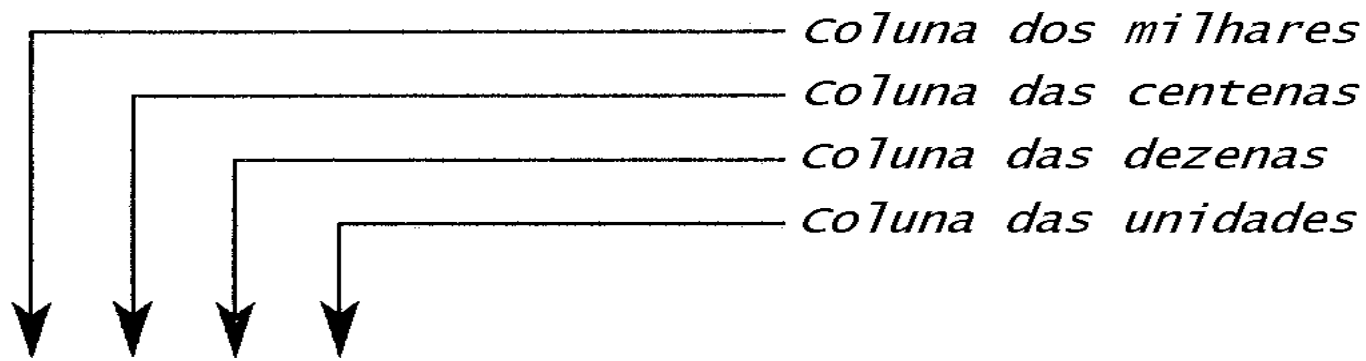
binária

hexadecimal

octal

outras...

Sistema Decimal (base 10)



4 = *Quatro*
4 0 = *Quarenta*
4 0 0 = *quatrocentos*
4 0 0 0 = *quatro mil*

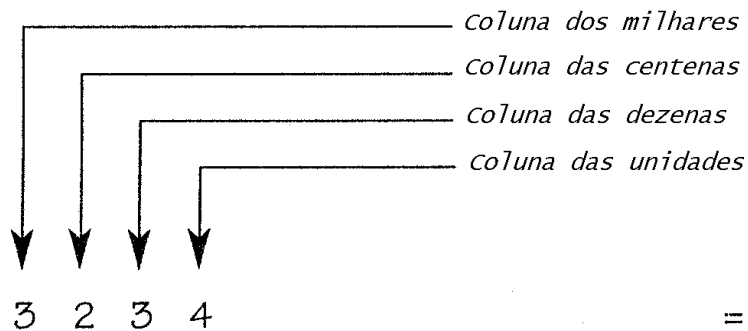
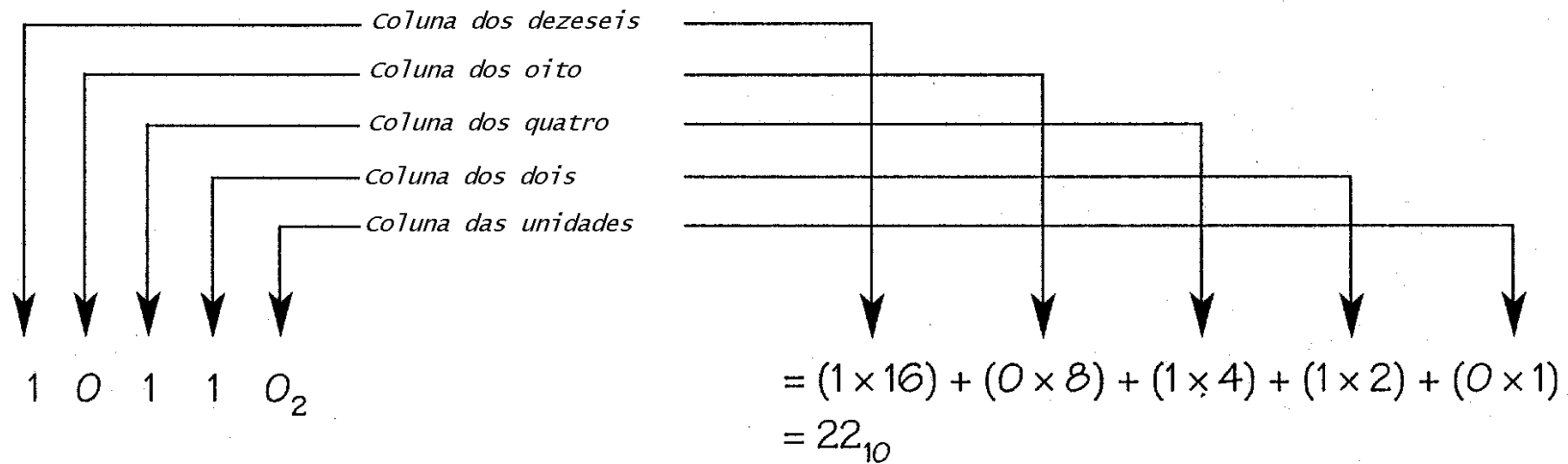


Diagram illustrating the expansion of the number 3234 into its place value components:

$$= (3 \times 1000) + (2 \times 100) + (3 \times 10) + (4 \times 1)$$

Sistema Binário (base 2)



0_2 (0)	1000_2 (8)	10000_2 (16)	etc.
1_2 (1)	1001_2 (9)	10001_2 (17)	
10_2 (2)	1010_2 (10)	10010_2 (18)	
11_2 (3)	.	.	
100_2 (4)	.	.	
101_2 (5)	1101_2 (13)	11101_2 (29)	
110_2 (6)	1110_2 (14)	11110_2 (30)	
111_2 (7)	1111_2 (15)	11111_2 (31)	

Sistema octal (base 8) e hexadecimal (base 16)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

<i>Decimal</i>	<i>Binário</i>	<i>Octal</i>	<i>Hexadecimal</i>
0	00000000	000	00
1	00000001	001	01
2	00000010	002	02
3	00000011	003	03
4	00000100	004	04
5	00000101	005	05
6	00000110	006	06
7	00000111	007	07
8	00001000	010	08
9	00001001	011	09
10	00001010	012	0A
11	00001011	013	0B
12	00001100	014	0C
13	00001101	015	0D
14	00001110	016	0E
15	00001111	017	0F
16	00010000	020	10
17	00010001	021	11
18	00010010	022	12
:	:	:	:
etc.	etc.	etc.	etc.

Sistema de numeração

Um sistema de numeração é composto por:

Base - b

e.g. $B = 16$

Alfabeto Ordenado - conjunto de b símbolos distintos (dígitos)

e.g. $[0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F]$

Número - corresponde a uma sequência de dígitos.

e.g. $N_{(b)} \triangleleft \dots d_2 d_1 d_0 d_{-1} d_{-2} \dots$

Valor do Dígito (peso) - função do símb. e da pos. na sequência.

e.g. $p_2 = d_2 b^2$

► Exemplo:

S.N. : Decimal

28886_{10}

Binário

10101110_2

Octal

5270_8

Hexadecimal

$A32C_{16}$

Mudança no Sistema de Numeração

CONVERSÃO DE BASES ($b_1 \neq 10$ para a base $b_2 = 10$)

A conversão de um número numa base diferente de 10 para a base decimal reduz-se a representar esse número como um polinómio e de seguida determinar o equivalente decimal (ver Determinação do Equivalente Decimal)

CONVERSÃO DE BASES ($b_1 = 10$ para a base $b_2 \neq 10$)

A conversão de um número na base 10 para uma base diferente realiza-se em duas fases:

- (1) A parte inteira é convertida segundo o método das divisões sucessivas.
- (2) A parte fraccionária é convertida segundo o método das multiplicações sucessivas.

Mudança no Sistema de Numeração

► Exemplo:

S.N. : Decimal

$20,35_{(10)}$



Binário

$10100, \dots_{(2)}$

Hexadecimal

$14, \dots_{(16)}$

O número a converter e os quocientes sucessivos são divididos pela base.

A sequência de restos constitui o resultado da conversão.

1º resto = dígito menos significativo

$$\begin{array}{r} 20 \overline{) 2} \\ 00 \overline{) 10} \overline{) 2} \\ \quad 0 \overline{) 5} \overline{) 2} \\ \quad \quad 1 \overline{) 2} \overline{) 2} \\ \quad \quad \quad 0 \overline{) 1} \overline{) 2} \\ \quad \quad \quad \quad 1 \overline{) 0} \end{array}$$

$$\begin{array}{r} 20 \overline{) 16} \\ \textcircled{4} \overline{) 1} \overline{) 16} \\ \quad \textcircled{1} \overline{) 0} \end{array}$$

Mudança no Sistema de Numeração

● MUDANÇA DE SISTEMA DE NUMERAÇÃO

CONVERSÃO DE BASES ($b_1 = 10$ para a base $b_2 \neq 10$) (cont.)

► Exemplo: (cont)

S.N. : Decimal

$20,35_{(10)}$



Binário

$10100,0101 \dots_{(2)}$

Hexadecimal

$14,59 \dots_{(16)}$

$$\begin{array}{r} 0,35 \\ \times 2 \\ \hline \textcircled{0} 70 \\ \times 2 \\ \hline \textcircled{1} 40 \\ \times 2 \\ \hline \textcircled{0} 80 \\ \times 2 \\ \hline \textcircled{1} 60 \end{array}$$

$$\begin{array}{r} 0,35 \\ \times 16 \\ \hline 210 \\ 35 \\ \hline \textcircled{5} 60 \\ \times 16 \\ \hline 360 \\ 60 \\ \hline \textcircled{9} 60 \end{array}$$

Mudança no Sistema de Numeração

CONVERSÃO DE BASES ($b_1 = 2^t$ para a base $b_2 = 2$ e vice-versa)

Atendendo às propriedades das potências facilmente se infere que:

- (1) Na conversão da base 2^t para a base 2, transforma-se cada dígito da base 2^t em t bits da base 2.
- (2) Na conversão da base 2 para a base 2^t , transforma-se cada t bits da base 2 num dígito da base 2^t .

► **Exemplo:**

Binário			
0001 0100, 0101 ₍₂₎			
↕	↕	↕	
1	4	,	5 ₍₁₆₎
Hexadecimal			

Códigos Binários

● CÓDIGOS BINÁRIOS

CÓDIGO BINÁRIO

No presente contexto, por **código binário**, entende-se o código que estabelece a correspondência entre palavras escritas num qualquer sistema de numeração e palavras constituídas por caracteres binários.

$$\text{e.g. } 12_{(10)} \Leftrightarrow 1100_{(2)}$$

CÓDIGO BINÁRIO NATURAL (CBN)

Código ponderado, gerado pelo sistema de numeração de base 2, em que os pesos das colunas são sucessivamente $2^{n-1}, 2^{n-2}, \dots, 2^1, 2^0$.

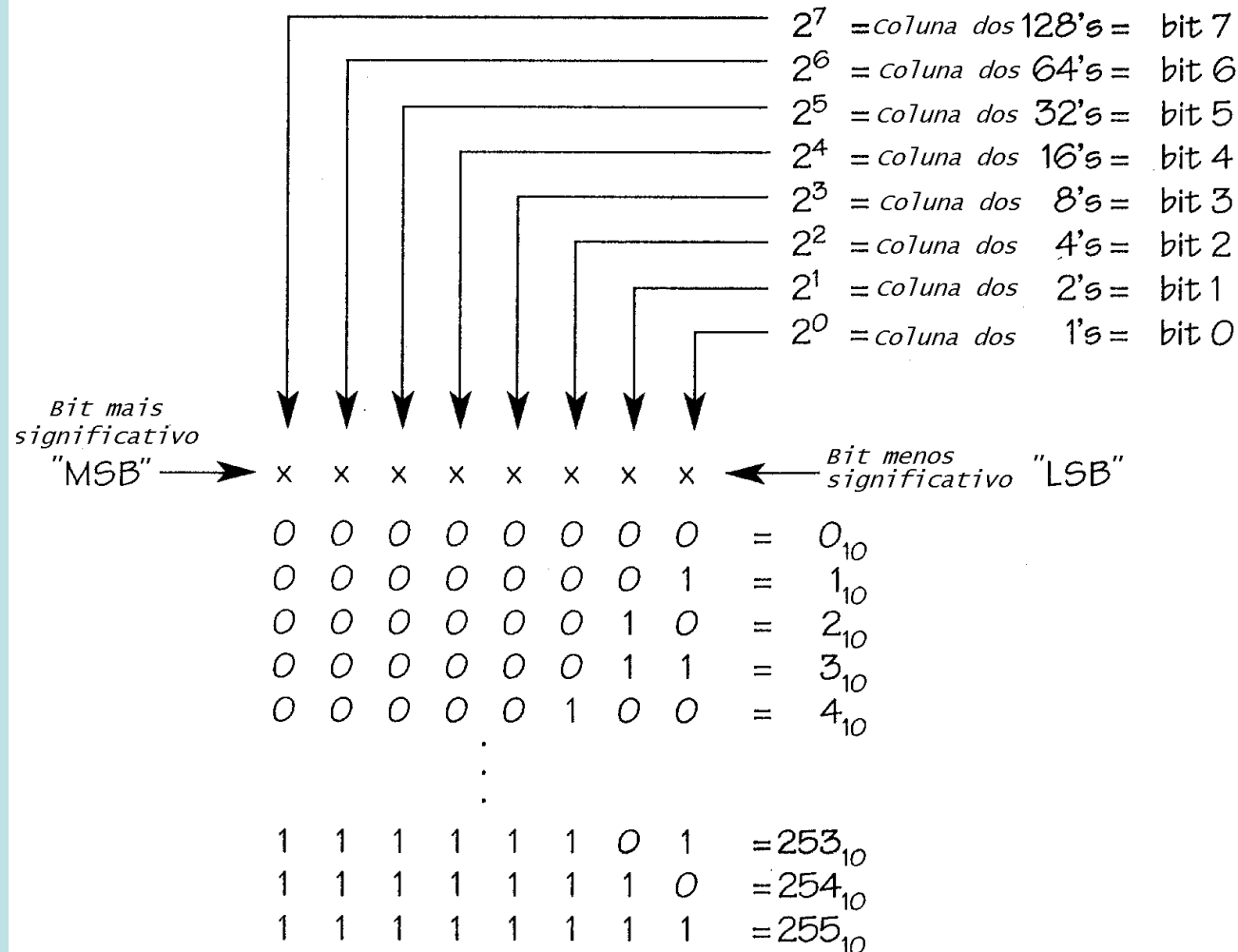
CÓDIGO BINÁRIO REFLECTIDO (CBR) ou CÓDIGO DE GRAY

Código não ponderado, obtido do CBN por troca de símbolos do alfabeto binário, i.e., na primeira coluna temos 01 10 em vez de 01 01 do CBN, na segunda coluna temos 00 11 11 00 em vez de 00 11 00 11 do CBN etc., daí a designação de CB reflectido.

	CBN	CBR
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Aritmética Binária

Números binários sem sinal



Aritmética Binária

Adição binária

$$\begin{array}{r} 0\ 0\ 1\ 1\ 1\ 0\ 0\ \boxed{1} \\ +\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ \boxed{0} \\ \hline = 1 \end{array}$$

(a) Bit 0, $1 + 0 = 1_2$

$$\begin{array}{r} 0\ 0\ 1\ 1\ 1\ 0\ \boxed{0}\ 1 \\ +\ 0\ 0\ 0\ 1\ 1\ 0\ \boxed{1}\ 0 \\ \hline = 1\ 1 \end{array}$$

(b) Bit 1, $0 + 1 = 1_2$

$$\begin{array}{r} 0\ 0\ 1\ 1\ 1\ \boxed{0}\ 0\ 1 \\ +\ 0\ 0\ 0\ 1\ 1\ \boxed{0}\ 1\ 0 \\ \hline = 0\ 1\ 1 \end{array}$$

(c) Bit 2, $0 + 0 = 0_2$

$$\begin{array}{r} 0\ 0\ 1\ 1\ \boxed{1}\ 0\ 0\ 1 \\ +\ 0\ 0\ 0\ 1\ \boxed{1}\ 0\ 1\ 0 \\ \hline = 0\ 0\ 1\ 1 \end{array}$$

(d) Bit 3, $1 + 1 = 10_2$

$$\begin{array}{r} 0\ 0\ 1\ \boxed{1}\ 1\ 0\ 0\ 1 \\ +\ 0\ 0\ 1\ \boxed{1}\ 1\ 0\ 1\ 0 \\ \hline = 1\ 0\ 0\ 1\ 1 \end{array}$$

(e) Bit 4, $1 + 1 + \text{vai_um} = 11_2$

$$\begin{array}{r} 0\ 0\ \boxed{1}\ 1\ 1\ 0\ 0\ 1 \\ +\ 0\ 1\ \boxed{0}\ 1\ 1\ 0\ 1\ 0 \\ \hline = 0\ 1\ 0\ 0\ 1\ 1 \end{array}$$

(f) Bit 5, $1 + 0 + \text{vai_um} = 10_2$

$$\begin{array}{r} 0\ \boxed{0}\ 1\ 1\ 1\ 0\ 0\ 1 \\ +\ 0\ \boxed{1}\ 0\ 1\ 1\ 0\ 1\ 0 \\ \hline = 1\ 0\ 1\ 0\ 0\ 1\ 1 \end{array}$$

(g) Bit 6, $0 + 0 + \text{vai_um} = 1_2$

$$\begin{array}{r} \boxed{0}\ 0\ 1\ 1\ 1\ 0\ 0\ 1 \\ +\ \boxed{0}\ 1\ 0\ 1\ 1\ 0\ 1\ 0 \\ \hline = 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1 \end{array}$$

(h) Bit 7, $0 + 0 = 0_2$

$$\begin{array}{r} 57_{10} \\ +\ 26_{10} \\ \hline =\ 83_{10} \end{array}$$

Aritmética Binária

Subtração binária

Subtração Padrão

$$\begin{array}{r} 647 \\ - 283 \\ \hline = 364 \end{array}$$

Equivalente em complemento de nove

$\begin{array}{r} 999 \\ - 283 \\ \hline = 716 \end{array}$	\rightarrow	$\begin{array}{r} 647 \\ + 716 \\ \hline = 1363 \\ \hline 364 \end{array}$	$\left. \begin{array}{l} \text{Vai-um do} \\ \text{MSB para o} \\ \text{LSB} \end{array} \right\} \text{"End-around-carry"}$
$\underbrace{}_{\text{Tomando o complemento de nove}}$		$\begin{array}{r} 363 \\ \rightarrow 1 \\ \hline 364 \end{array}$	

subtração padrão

$$\begin{array}{r} 647 \\ - 283 \\ \hline = 364 \end{array}$$

Equivalente em complemento de dez

$\begin{array}{r} 1000 \\ - 283 \\ \hline = 717 \end{array}$	\rightarrow	$\begin{array}{r} 647 \\ + 717 \\ \hline = 1364 \\ \hline 364 \end{array}$	$\left. \begin{array}{l} \text{Elimina} \\ \text{qualquer} \\ \text{vai-um} \end{array} \right\}$
$\underbrace{}_{\text{Tomando o complemento de Dez}}$		364	

Aritmética Binária

Subtração binária

Subtração padrão

$$\begin{array}{r} 00111001 \\ - 00011110 \\ \hline = 00011011 \end{array}$$

$$57_{10} - 30_{10} = 27_{10}$$

Equivalente em complemento de um

$\begin{array}{r} 11111111 \\ - 00011110 \\ \hline = 11100001 \end{array}$ <p style="text-align: center;">Tomando o complemento de um</p>	→	$\begin{array}{r} 00111001 \\ + 11100001 \\ \hline 100011010 \\ \hline 00011011 \end{array}$	} Vai-um do MSB para o LSB "End-around-carry"
<p style="text-align: center;"> </p>			

Subtração padrão

$$\begin{array}{r} 00111001 \\ - 00011110 \\ \hline = 00011011 \end{array}$$

Equivalente em complemento de dois

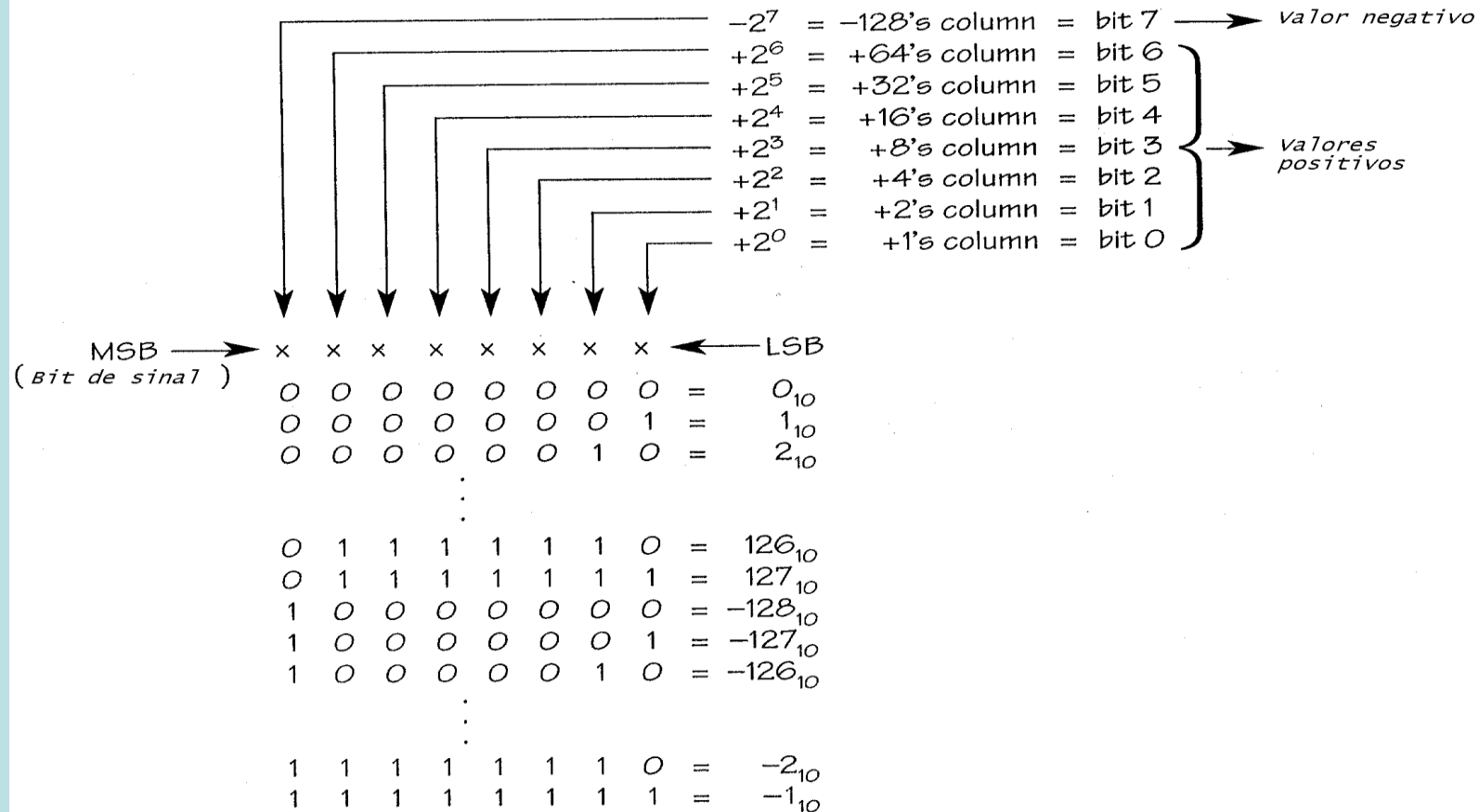
$\begin{array}{r} 10000000 \\ - 00011110 \\ \hline = 11100010 \end{array}$ <p style="text-align: center;">Toma o complemento de dois</p>	→	$\begin{array}{r} 00111001 \\ + 11100010 \\ \hline 100011011 \\ \hline 00011011 \end{array}$	} Elimina qualquer vai-um do MSB
<p style="text-align: center;"> </p>			

The diagram shows the bit reversal process for the binary value 00011110. It consists of three rows:

- Initial Value:** 0 0 0 1 1 1 1 0. The leftmost bit (0) is labeled "MSB" (Most Significant Bit) and the rightmost bit (0) is labeled "LSB" (Least Significant Bit).
- Step 1:** A downward arrow indicates the first step. The result shown is dashes followed by 1 0: - - - - - 1 0. This represents copying the bits from the LSB up to the first 1.
- Step 2:** Another downward arrow indicates the second step. The final result is 1 1 1 0 0 0 1 0. This represents inverting all bits except the first one.

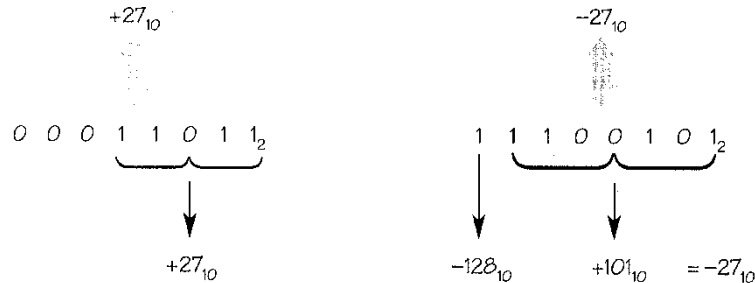
Aritmética Binária

Números binários com bit de sinal



Aritmética Binária

Números binários com bit de sinal



Representação
sinal-magnitude
de números decimais

Binário com sinal

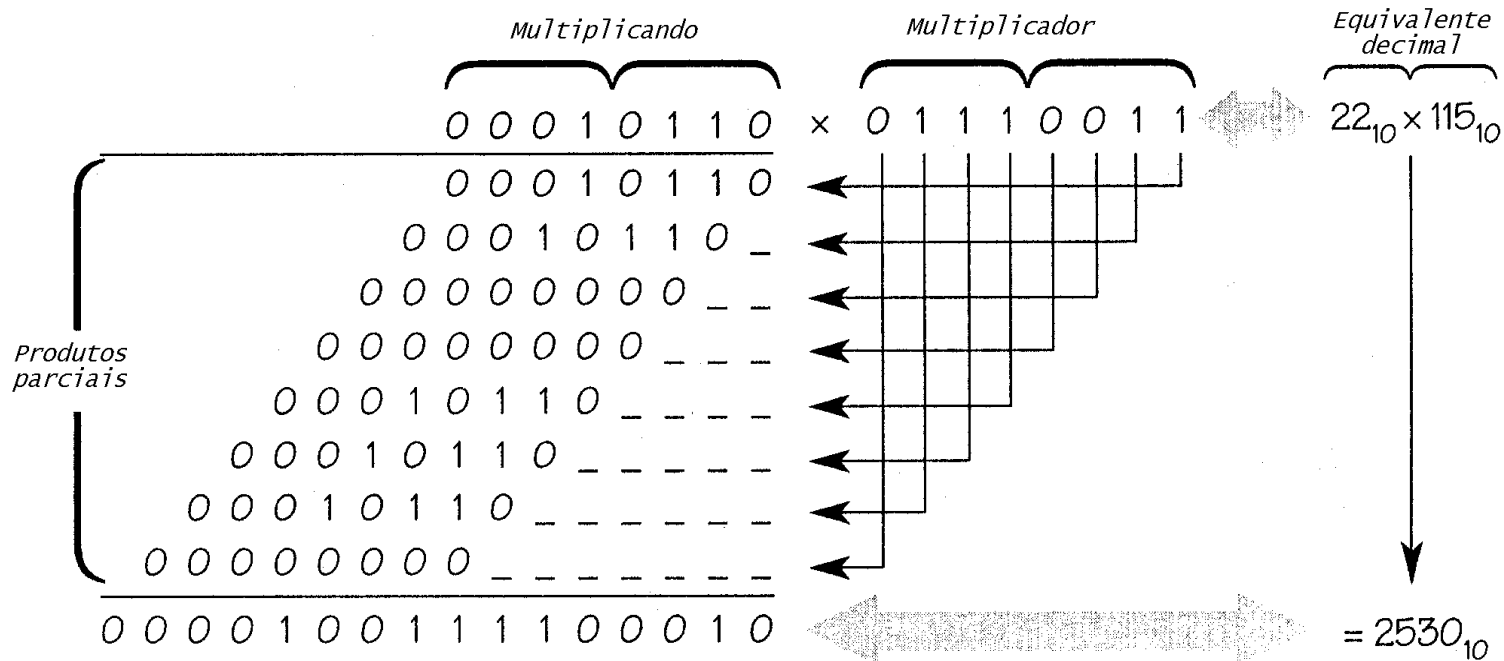
$$\begin{array}{r}
 57 \\
 + 30 \\
 \hline
 = 87
 \end{array}
 \longleftrightarrow
 \begin{array}{r}
 00111001 \\
 + 00011110 \\
 \hline
 01010111
 \end{array}$$

$$\begin{array}{r}
 57 \\
 + -30 \\
 \hline
 = 27
 \end{array}
 \longleftrightarrow
 \begin{array}{r}
 00111001 \\
 + 11100001 \\
 \hline
 00011011
 \end{array}$$

$$\begin{array}{r}
 -57 \\
 + 30 \\
 \hline
 = -27
 \end{array}
 \longleftrightarrow
 \begin{array}{r}
 11000111 \\
 + 00011110 \\
 \hline
 11100101
 \end{array}$$

$$\begin{array}{r}
 -57 \\
 + -30 \\
 \hline
 = -87
 \end{array}
 \longleftrightarrow
 \begin{array}{r}
 11000111 \\
 + 11100001 \\
 \hline
 10101001
 \end{array}$$

Multiplicação binária - algoritmo de Booth



Funções Lógicas/Portas Lógicas

George Boole (1815-1864)

Claude Shannon (~ 1930)

Operações Lógicas Básicas


AND		
X	Y	$X.Y$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X	Y	$X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

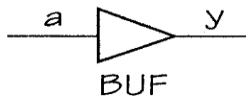

NOT	
X	\bar{X}
0	1
1	0

Funções Lógicas/Portas Lógicas


$$y = a$$



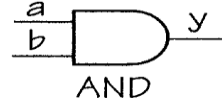

a	y
0	0
1	1




$$y = a \& b$$



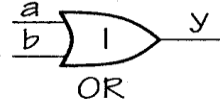

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1




$$y = a \mid b$$




a	b	y
0	0	0
0	1	1
1	0	1
1	1	1




$$y = a \wedge b$$



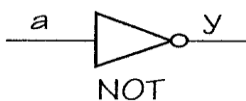

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0




$$y = \bar{a}$$




a	y
0	1
1	0




$$y = \overline{a \& b}$$




a	b	y
0	0	1
0	1	1
1	0	1
1	1	0




$$y = \overline{a \mid b}$$




a	b	y
0	0	1
0	1	0
1	0	0
1	1	0






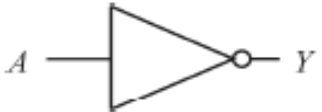



$$y = \overline{a \wedge b}$$



a	b	y
0	0	1
0	1	0
1	0	0
1	1	1



Funções Lógicas/Portas Lógicas

Função Lógica Básica	Símbolo Gráfico da Porta	Equação Booleana
AND		$Y = A \cdot B$
OR		$Y = A + B$
XOR		$Y = A \oplus B$
NOT		$Y = \overline{A}$
NAND		$Y = \overline{A \cdot B}$
NOR		$Y = \overline{A + B}$
XNOR		$Y = \overline{A \oplus B}$