Sistemas de Numeração

Bases numéricas:

decimal

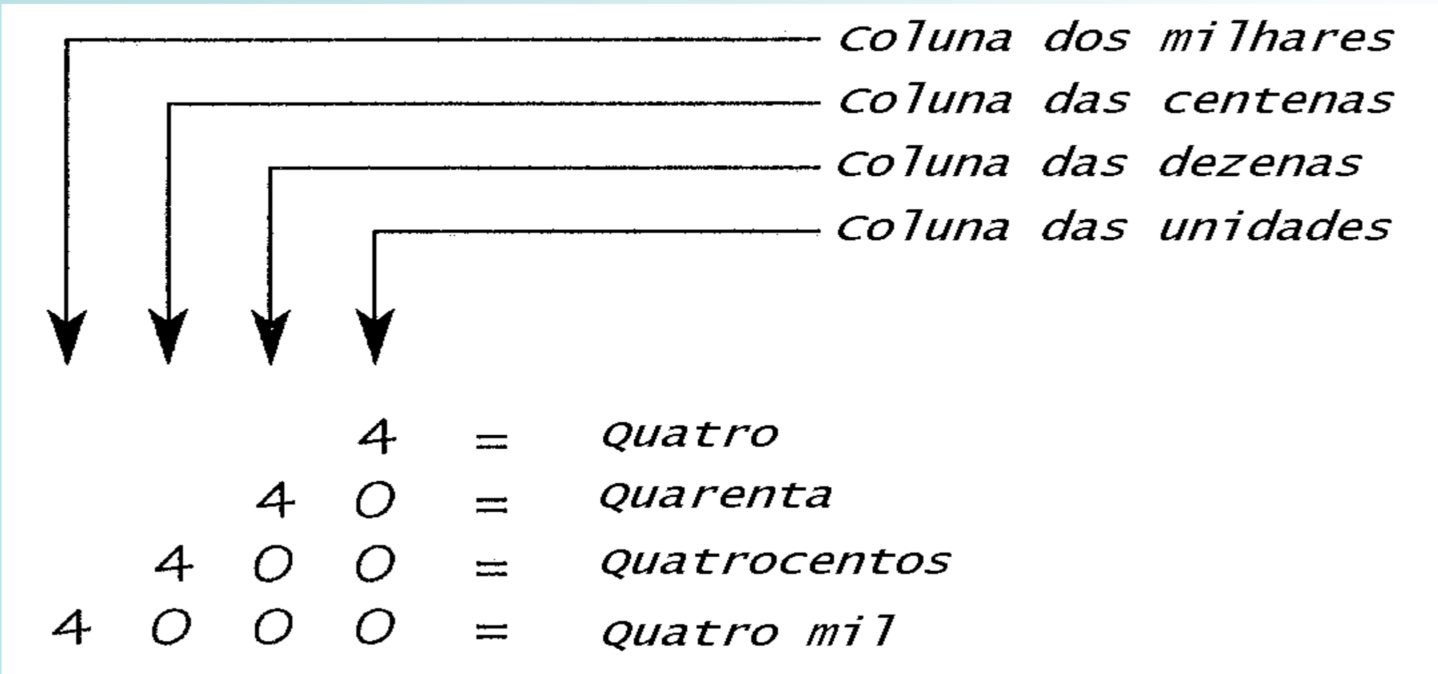
binária

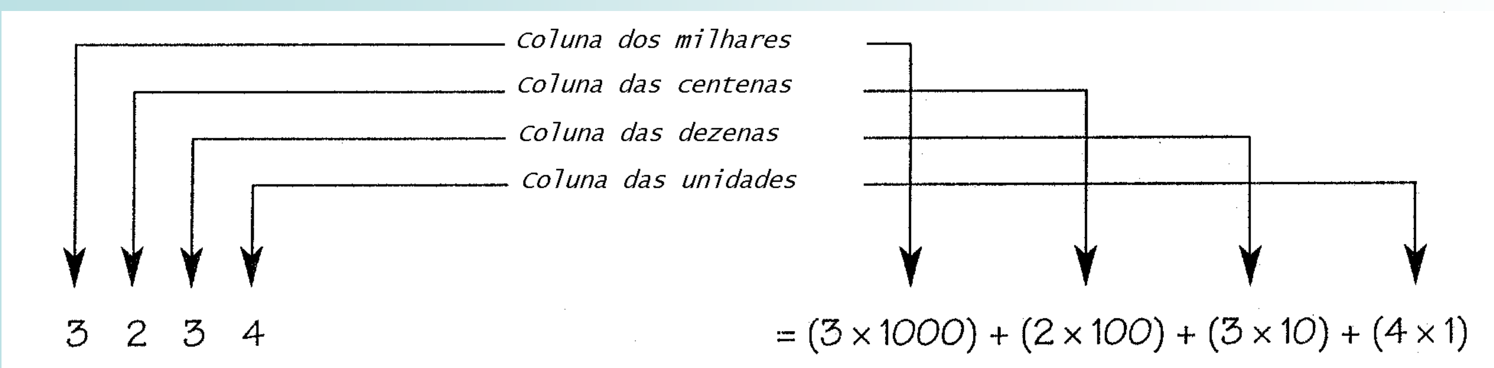
hexadecimal

octal

outras...

Sistema Decimal (base 10)





Sistema Binário (base 2)

Coluna dos dezeseis

Coluna dos oito

Coluna dos quatro

Coluna dos dois

Coluna das unidades

$1\ 0\ 1\ 1\ 0_2$

$= (1 \times 16) + (0 \times 8) + (1 \times 4) + (1 \times 2) + (0 \times 1)$

$= 22_{10}$

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

Decimal	*Binário*	*Octal*	*Hexadecimal*
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. $N_{(b)} <> \dots d_2\ d_1\ d_0,\ d_{-1}\ d_{-2} \dots$

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

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

Exemplo:

S.N. : Decimal	Binário	Octal	Hexadecimal
28886_{10}	10101110_2	5270_8	$A32C_{16}$

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		**Binário**	**Hexadecimal**
$20,35_{(10)}$	$\rightarrow$	$10100,..._{(2)}$	$14,..._{(16)}$

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

A sequência de restos constitui o resultado da conversão.

1° resto = dígito menos significativo

```
20 |2
00 10 |2
    0  5 |2
       1  2 |2
          0  1 |2
             1  0
```

```
20 |16
 4  1 |16
    1  0
```

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		Binário	Hexadecimal
$20,35_{(10)}$	→	$10100,0101..._{(2)}$	$14,59..._{(16)}$

Binário	Hexadecimal
0,35	0,35
X 2	X 16
0,70	210
X 2	35
1,40	5,60
X 2	X 16
0,80	360
X 2	60
1,60	9,60

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

$0001\ 0100,0101_{(2)}$

$1\quad 4\ ,\ 5_{(16)}$

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.**

e.g. $\mathbf{12}_{(10)} <> \mathbf{1100}_{(2)}$

CÓDIGO BINÁRIO NATURAL (CBN)

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

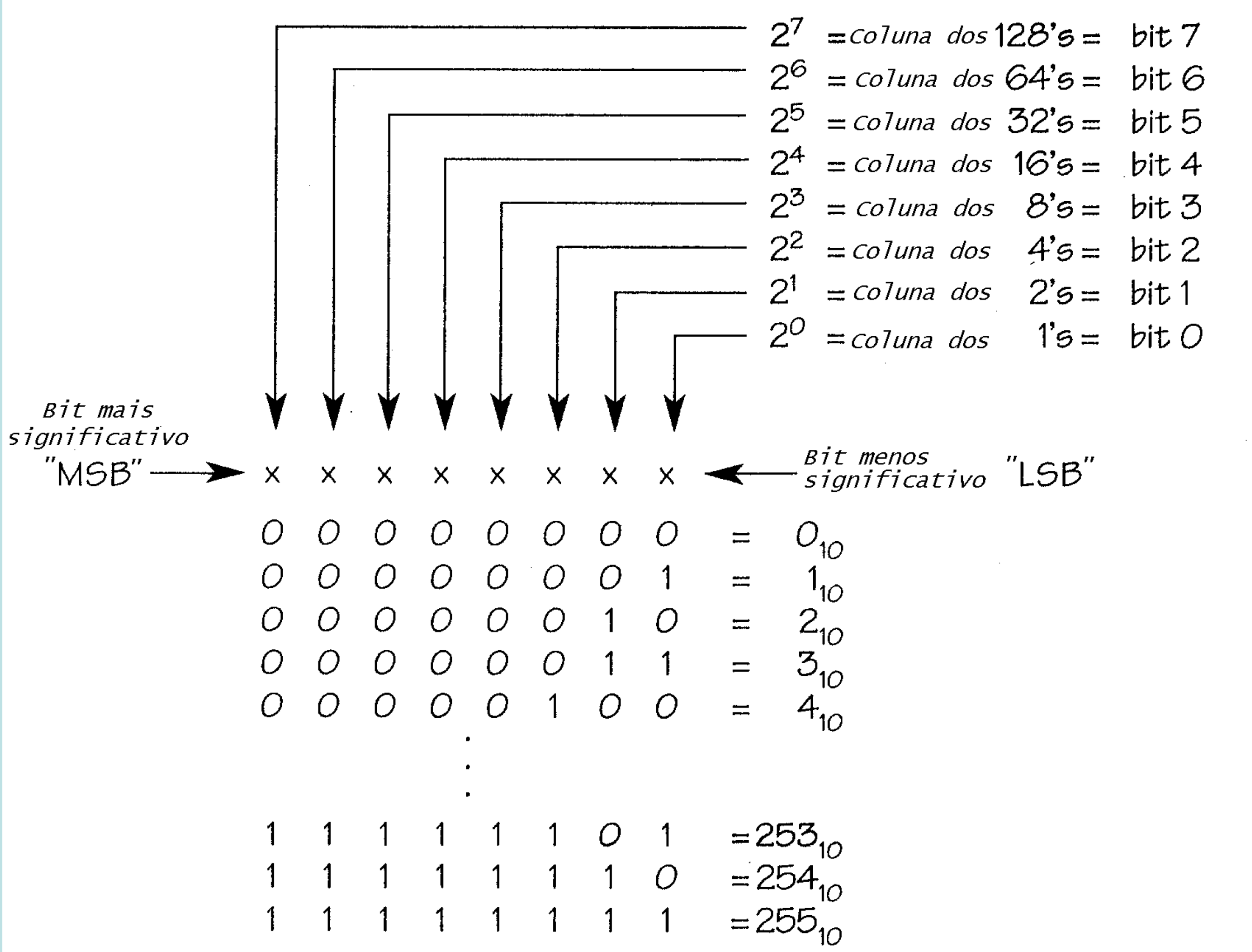
CÓDIGO BINÁRIO REFLECTIDO (CBR) ou CÓDIGO DE GRAY

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.

$$\begin{array}{ccccc} & & \textbf{CBN} & & \textbf{CBR} \\ \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix} & \Leftrightarrow & \begin{bmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{bmatrix} & & \begin{bmatrix} 000 \\ 001 \\ 011 \\ 010 \\ 110 \\ 111 \\ 101 \\ 100 \end{bmatrix} \end{array}$$

Aritmética Binária

Números binários sem sinal



Aritmética Binária

Adição binária

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline = & & & & & & & & 1 \end{array}$$

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

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline = & & & & & & & 1 & 1 \end{array}$$

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

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline = & & & & & & 0 & 1 & 1 \end{array}$$

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

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 0 & {}_{1}1 & 1 & 0 & 1 & 0 \\ \hline = & & & & & 0 & 0 & 1 & 1 \end{array}$$

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

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & {}_{1}0 & {}_{1}1 & 1 & 0 & 1 & 0 \\ \hline = & & & & 1 & 0 & 0 & 1 & 1 \end{array}$$

(e) Bit 4, $1 + 1 +$ *vai_um* $= 11_2$

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & {}_{1}0 & {}_{1}0 & {}_{1}1 & 1 & 0 & 1 & 0 \\ \hline = & & & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

(f) Bit 5, $1 + 0 +$ *vai_um* $= 10_2$

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & {}_{1}0 & {}_{1}0 & {}_{1}1 & 1 & 0 & 1 & 0 \\ \hline = & & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

(g) Bit 6, $0 + 0 +$ *vai_um* $= 1_2$

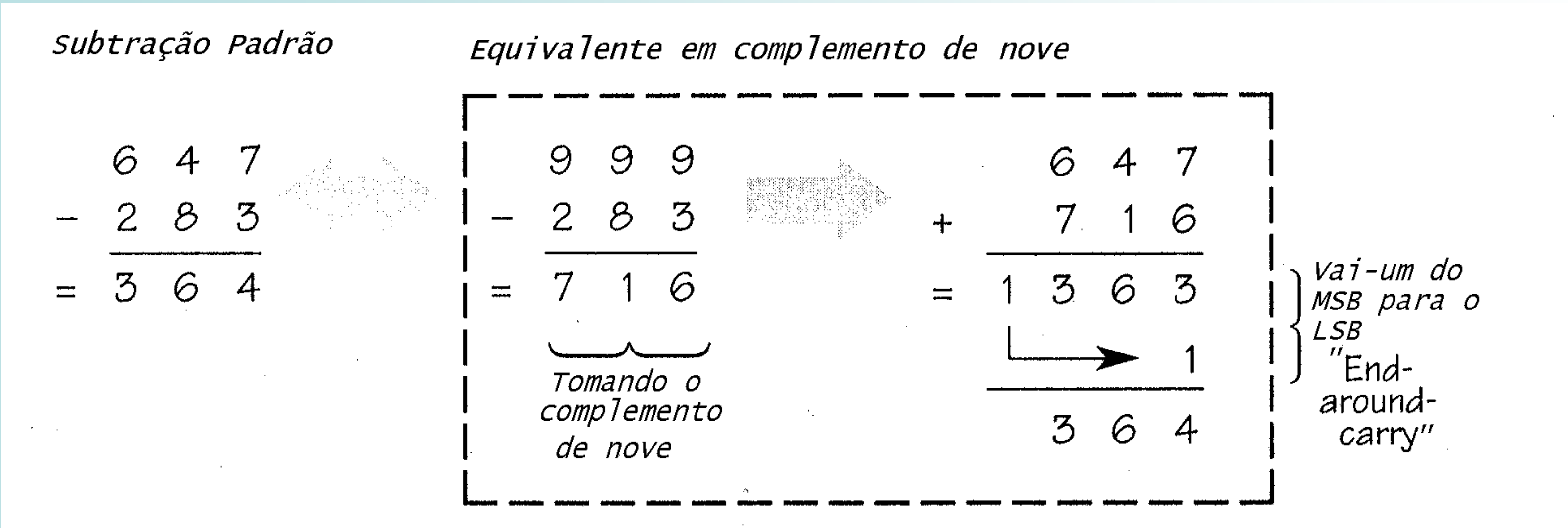
$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & {}_{1}0 & {}_{1}0 & {}_{1}1 & 1 & 0 & 1 & 0 \\ \hline = & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

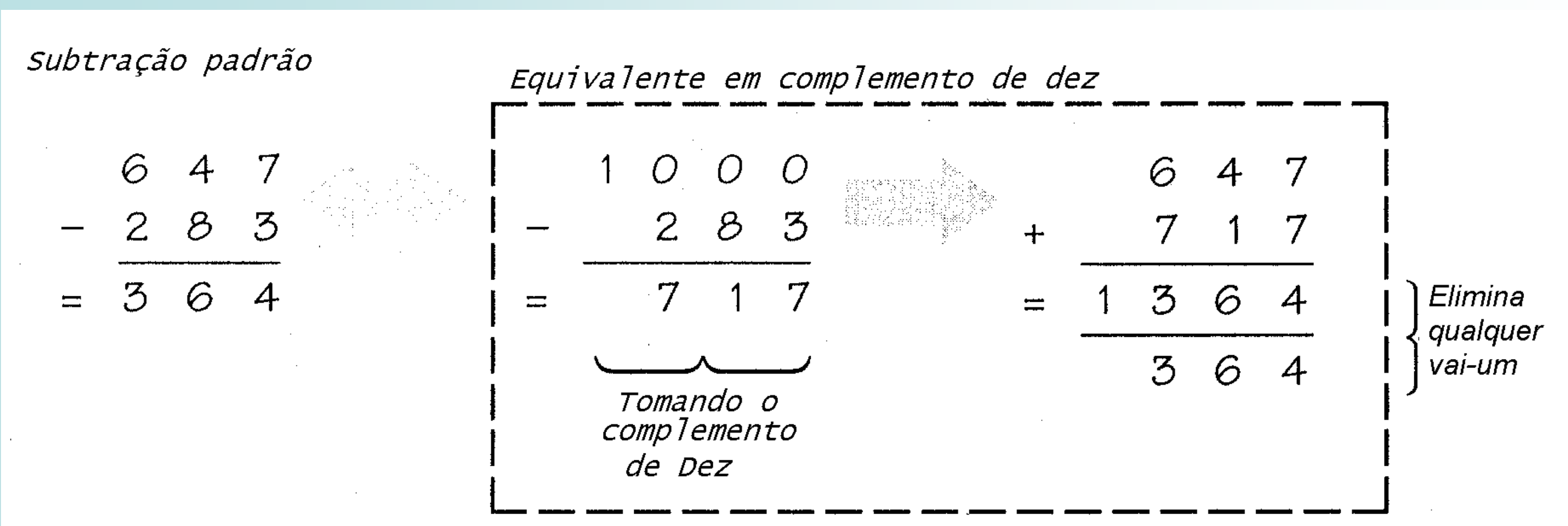
(h) Bit 7, $0 + 0 = 0_2$

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

Aritmética Binária

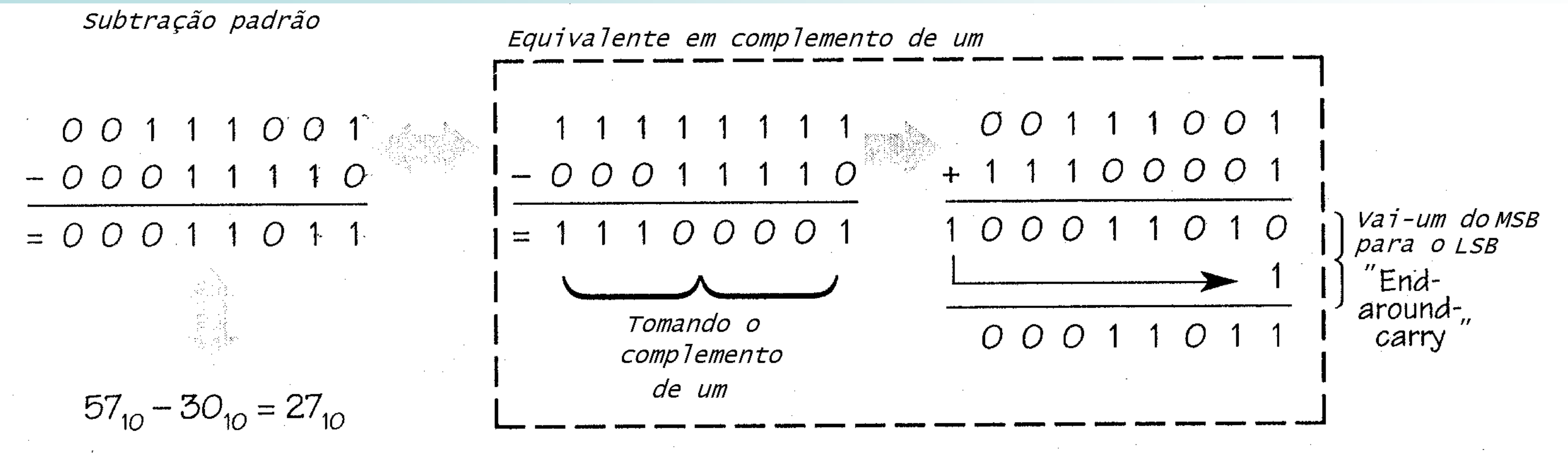
Subtração binária

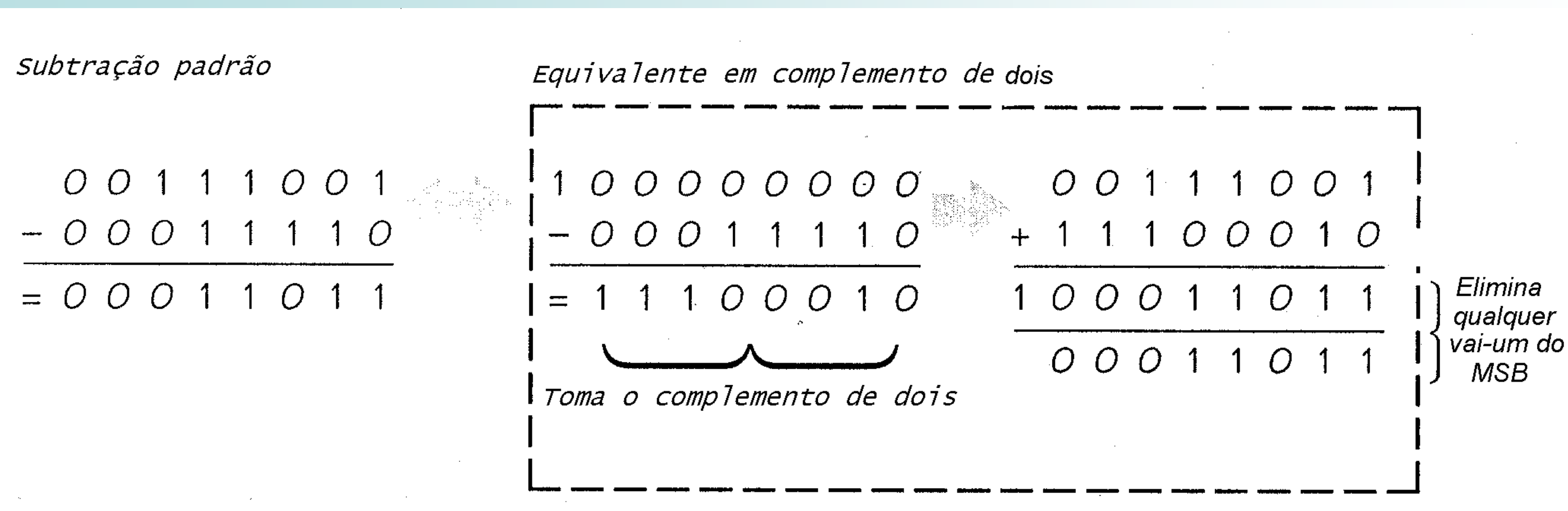




Aritmética Binária

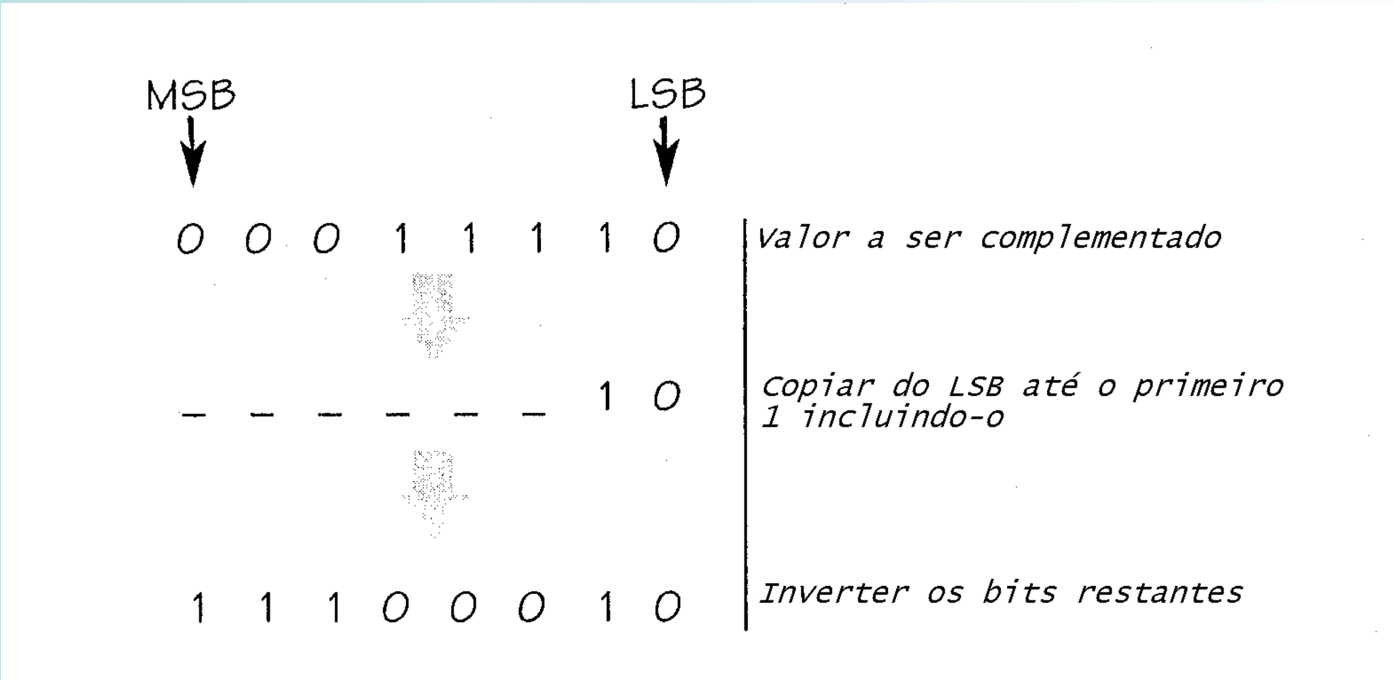
Subtração binária





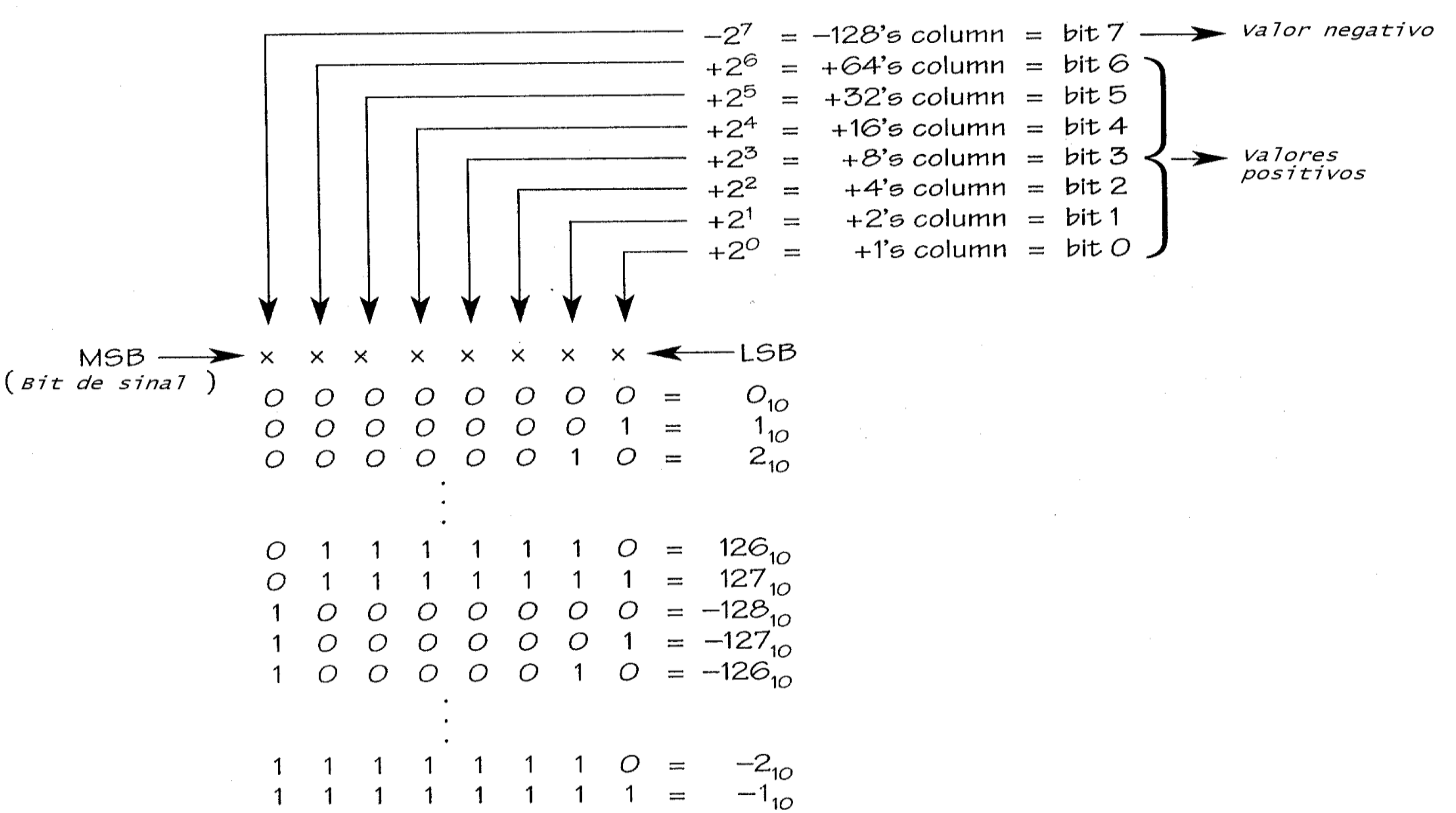
Aritmética Binária

Subtração binária-técnica



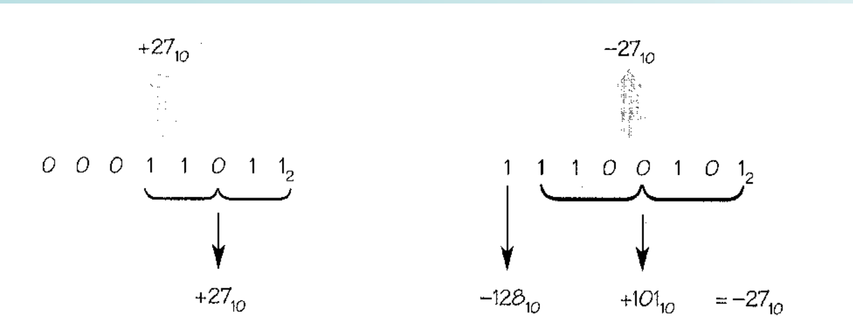
Aritmética Binária

Números binários com bit de sinal



Aritmética Binária

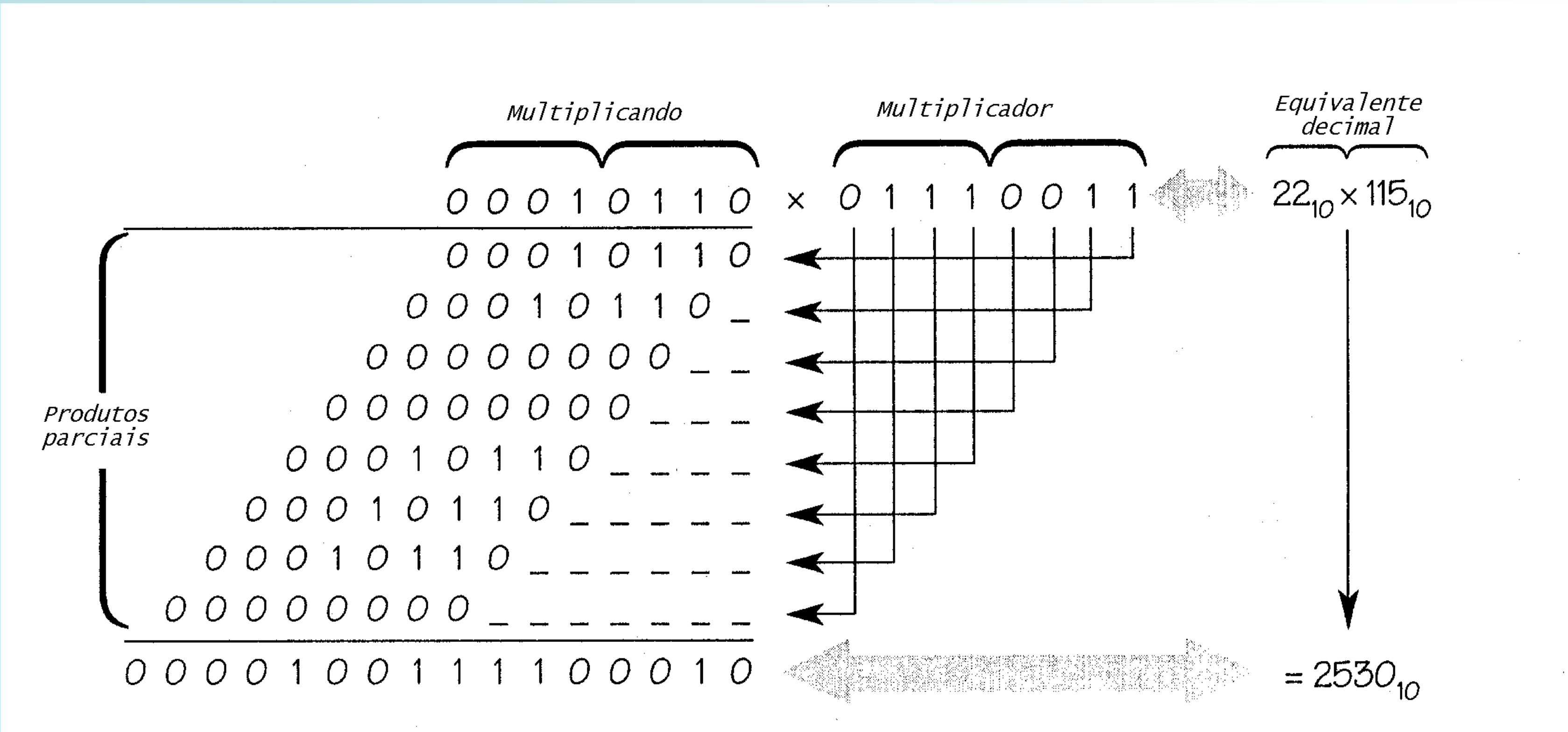
Números binários com bit de sinal



Representação sinal-magnitude de números decimais	Binário com sinal
5 7	0 0 1 1 1 0 0 1
+ 3 0	+ 0 0 0 1 1 1 1 0
= 8 7	0 1 0 1 0 1 1 1
5 7	0 0 1 1 1 0 0 1
+ −3 0	+ 1 1 1 0 0 0 1 0
= 2 7	0 0 0 1 1 0 1 1
−5 7	1 1 0 0 0 1 1 1
+ 3 0	+ 0 0 0 1 1 1 1 0
= −2 7	1 1 1 0 0 1 0 1
−5 7	1 1 0 0 0 1 1 1
+ −3 0	+ 1 1 1 0 0 0 1 0
= −8 7	1 0 1 0 1 0 0 1

Aritmética Binária

Multiplicação binária - algoritmo de Booth



Funções Lógicas/Portas Lógicas

George Boole (1815-1864)
Claude Shannon (~ 1930)

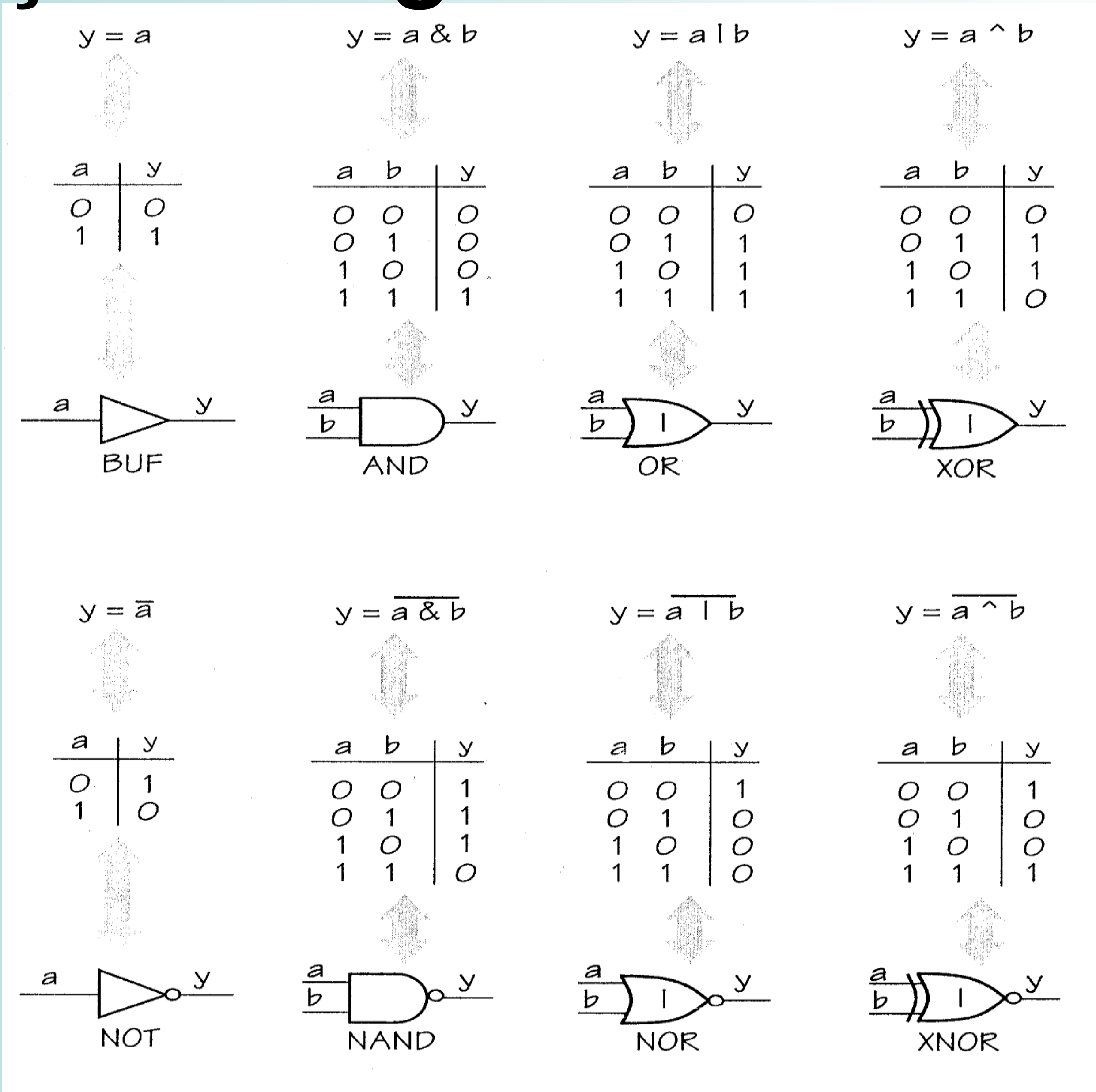
Operações Lógicas Básicas

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

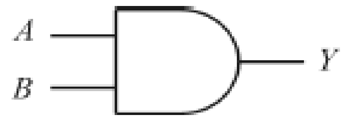
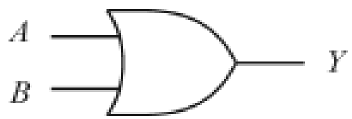
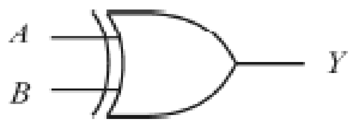
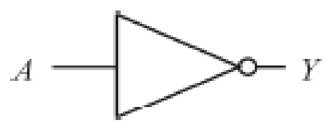
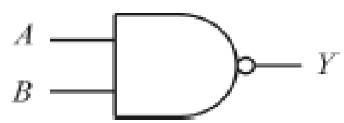
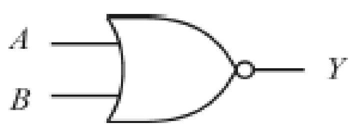
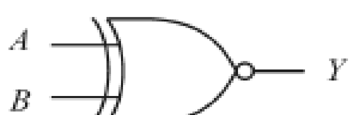
OR		
X	Y	X + Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	$\overline{X}$
0	1
1	0

Funções Lógicas/Portas Lógicas



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		$Y = \overline{A \oplus B}$