Sistemas de numeração

Destaque para decimal, binário, octal e hexadecimal.

Definições :
bit : binary digit - a menor porção binária
nibble : conjunto de 4 bits
byte : agrupamento de 8 bits
MSB : Most significant bit
LSB : Least significant bit
Carry : e a operação matemática também conhecida como “vai um”.

Conversão entre sistemas:

Decimal para binário (valores inteiros)
Sucessivas divisões por 2 até atingir a impossibilidade de nova divisão. Faz-se o agrupamento de baixo para cima dos números.

Decimal para binário (contendo também valores fracionários)
A parte inteira procede-se como acima descrito.
A parte fracionária procede-se realizando-se sucessivas multiplicações por 2 até obtermos como valor final 0 (zero) ou então uma dízima periódica.

Binário para decimal (valores inteiros)
Aplicamos os pesos exponenciais da base 2, multiplicado por 0 ou 1 da posição associada. Somamos os valores subtotais, para obter o decimal.

Binário para decimal (contendo também valores fracionários)
O processo é idêntico para valores inteiros, já que estamos trabalhando com base e seu exponencial, que representa a posição no bit.

Octal para decimal
Processo é idêntico binário, exceto que a base de trabalho é 8.

Decimal para octal
Divisões sucessivas por 8 até tornar a divisão impossível. A ordenação é como na decimal para binária, agrupando os valores de baixo para cima.

Binário para octal
Realizamos agrupamentos sucessivos (da esquerda para a direita) de 3 bits e convertemos para octal.

Octal para binário
Cada número octal deve ser convertido para o seu respectivo binário.

Hexadecimal para decimal
Idêntico ao binário, exceto que a base é 16

Decimal para Hexadecimal
Sucessivas divisões por 16,até o limite, com ordenação de baixo para cima.

Hexadecimal para binário
Cada número (inclusive as letras) são convertidas individualmente por seus respectivos binários.

Binário para hexadecimal
agrupamentos sucessivos da esquerda para direita de 4 em 4, e posterior conversão.

Operações aritméticas no sistema binário.

- Soma

```
   0          0          1          1
 +0         +1         +0         +1
-------    --------   ---------  ---------
   0          1          1          1 (e vai um)
```

- Multiplicação

```
   0          0          1          1
  *0         *1         *0         *1
-------    --------   ---------  ---------
   0          0          0          1
```

- Subtração

```
   0          0                 1          1
  -0         -1                -0         -1
-------    --------         ---------  ---------
   0          1 (e foi 1)       1          1
```

Notação de números positivos e negativos
Pode-se adotar os sinais + e -.
Na prática, usa-se o bit de sinal, posto a esquerda do MSB - É a notação Sinal-módulo.

Ex : $35_{10} = 100011_2 \Rightarrow \mathbf{0}100011_2$
Ex : $-35_{10} = 100011_2 \Rightarrow \mathbf{1}100011_2$

Outra forma de representação é através de complemento 2 (A2)
Deve-se aplicar primeiro o complemento 1 (A1) - vide abaixo:
Deve-se complementar o número em questão:
10011011 => 01100100

Para a conversão para A2, soma-se 1 a resultado de A1 !!!

01100100 + 1 = 01100101

Utilização do complemento de 2 em operações aritméticas (soma e subtração).
Quando estivermos sob a situação de número subtraindo outro.
Aplicamos o complemento A2 do segundo termo, e somamos ao primeiro.
Os números devem conter o mesmo número de casas (vide negrito).

111010111 - **00**100101= 111010111 + 11011010 = <u>1</u>10110010

Observe o número sublinhado ... **ele deve ser ignorado.**

A vantagem deste processo reside em que os sistemas digitais podem utilizar um mesmo circuito somador para efetuar as operações com números negativos e positivos, simplificando a quantidade de componentes.

Álgebra de Boole

Nota importante : Considere A\ como se houvesse um travessão por sobre "A". Note que o travessão a direita se refere a variável "A".

Postulados

- Se A=0, logo A\ = 1 e se A=1 logo A\ = 0 (inversor)
  Bloco lógico executor : **INVERSOR**

- A notação A\\ = A
  Bloco lógico executor : **DUPLO INVERSOR**

0 + 0 = 0	A + 0 = A	
0 + 1 = 1	A + 1 = 1	Postulado da soma
1 + 0 = 1	A + A = A	Bloco executor : **OR**
1 + 1 = 1 (e vai 1)	A + A\ = A	

0 . 0 = 0	A . 0 = 0	
0 . 1 = 0	A . 1 = A	Postulado da multiplicação
1 . 0 = 0	A . A = A	Bloco executor : **AND**
1 . 1 = 1	A . A\ = 0	

Propriedades

Comutativa (tanto faz a ordem)
A + B = B + A
A . B = B . A

Associativa (agrupando ...)
A + (B+C) = (A+B) + C = (A+C) +B
A . (B.C) = (A.B) . C = (A.C) . B

Distributiva (chuva ...)
A . (B + C) = AB + AC

Teorema de DeMorgan

1) (A.B) = A\ + B\   2) (A + B)\ = A\ . B\

Identidades Auxiliares

A + AB = A
(A+B) . (A+C) = A + BC
A + A\B = A + B
A\ + AB\ = A\ + B\

Demonstrações :

A+AB = A (1 + B) = A
A + A\B = A (B + B\) + A\B = AB + AB\ + A\B = A + B   (Porta OR)

	B\	B
A\	0	1
A	1	1

A + AB\ = A (1 + B\)= A
A + A\B\ = A(B\ + B) + A\B\ = AB\ + AB + A\B\ = A + B\

	B\	B
A\	1	0
A	1	1

A\ + A\B = A\ (1 + B\) = A\
A\ + AB = A\ (B\ + B) + AB = A\B\ + A\B + AB = A\ + B

	B\	B
A\	1	1
A	0	1

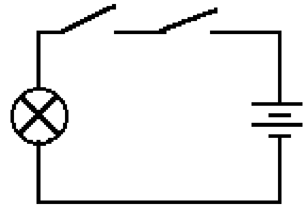
A\ + A\B\ = A\ (1 + B\) = A\
A\ + A B\ = A\ (B + B\) + AB\ = A\B + A\B\ + AB\ = A\ + B\   (Porta NAND)

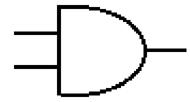
	B\	B
A\	1	1
A	1	0

Tabela Verdade   Expressão Boleana   Circuito com chaves   Simbolo

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

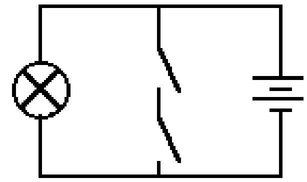
AND
A * B = S

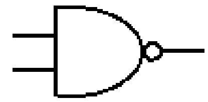




A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

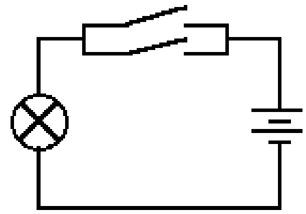
NAND
(A * B)\ = S

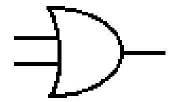




A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

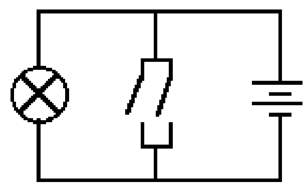
OR
A + B = S

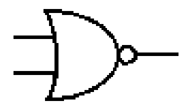




A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

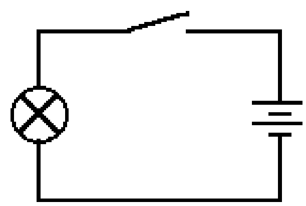
NOR
(A + B) \ = S

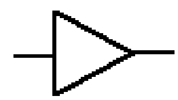




A	S
0	0
1	1

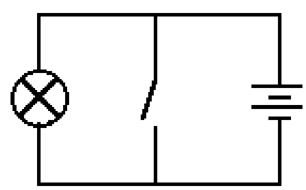
Buffer
S = A

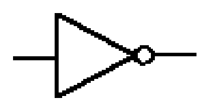




A	S
0	1
1	0

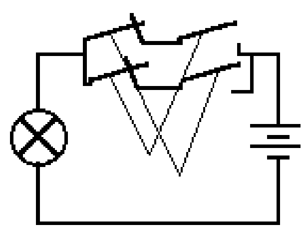
Inversor
S = A\

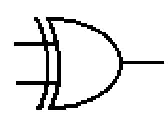




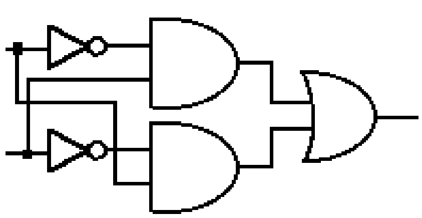
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

OR_EX
AB\ + A\B = S
A (+) B = S
Por intertravamento





Outra Maneira de representação :



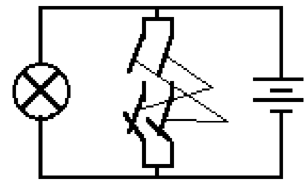
Forma Boleana

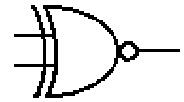
AB\ + A\B = S

Nota : Não existe Ex Or com mais de 2 entradas.

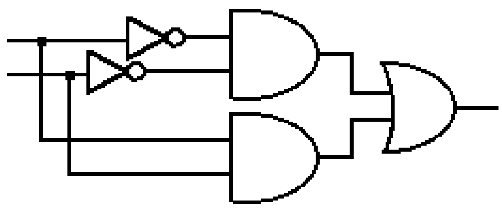
A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

NOR_EX
A\B\ + AB = S
(A (+) B)\ = S
Por intertravamento





Outra Maneira de representação : Forma Boleana



A\B\ + AB = S

Nota : Não existe Ex Nor com mais de 2 entradas.

Casos que não admitem simplicação

	B\	B
A\	0	1
A	1	0

AB\ + A\B = S

A\B\ + AB = S

	B\	B
A\	1	0
A	0	1

Nota : Não existe OREX e NOREX de mais de duas entradas. No entanto, de acordo com a necessidade pode-se montar um circuito lógico que execute tal função. Da expansão das variáveis tem-se que se o número de variáveis for :

Par : $\overline{\text{OREX}}$ = NOREX
Impar OREX = NOREX

Complete você agora a tabela de correlação entre os sistemas.

Decimal	Binário	Hexadecimal	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	
4	0100	4	
5	0101	5	
6		6	
7		7	
8		8	
9		9	
10		A	
11		B	
12		C	
13		D	
14		E	
15		F	