EAD

EDUCAÇÃO A DISTÂNCIA

UNIDADE I

Circuitos Lógicos Digitais

Prof. Me. Roberto Leminski

Sistemas numéricos

- O sistema de numeração que utilizamos é o sistema decimal, que utiliza, por sua vez, os algarismos arábicos.
- Este sistema de numeração, com dez algarismos, é tão amplamente difundido que é fácil pensarmos que se trata de algo natural, não de uma convenção.
- Entendermos como funciona o sistema decimal, que é o sistema numérico que usamos no nosso cotidiano, nos permitirá entender os outros sistemas numéricos.
- O sistema decimal, na verdade, é uma forma de representar os valores como uma somatória de potências de 10, que é a base do sistema.

Sistemas numéricos

- Para entendermos como funciona o sistema decimal, pegaremos um número real qualquer, por exemplo: 4382,19.

Agora, faremos uma decomposição deste número:

- $4382{,}19 = 4000 + 300 + 80 + 2 + 0{,}1 + 0{,}09 =$

$$= (4 \cdot 1000) + (3 \cdot 100) + (8 \cdot 10) + (2 \cdot 1) + (1 \cdot 0{,}1) + (9 \cdot 0{,}01) =$$

$$= (4 \cdot 10^3) + (3 \cdot 10^2) + (8 \cdot 10^1) + (2 \cdot 10^0) + (1 \cdot 10^{-1}) + (9 \cdot 10^{-2})$$

- Outros sistemas, como o binário, o octal e o hexadecimal, utilizam-se da mesma notação, sendo diferente a base da potência.

Sistemas numéricos

O número de algarismos presentes em cada sistema é igual à base do sistema:

- Sistema decimal (10 algarismos): 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9;
- Sistema binário (2 algarismos): 0 e 1;
- Sistema octal (8 algarismos): 0, 1, 2, 3, 4, 5, 6 e 7;
- Sistema hexadecimal (16 algarismos): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F.

No sistema hexadecimal, as letras representam valores numéricos:

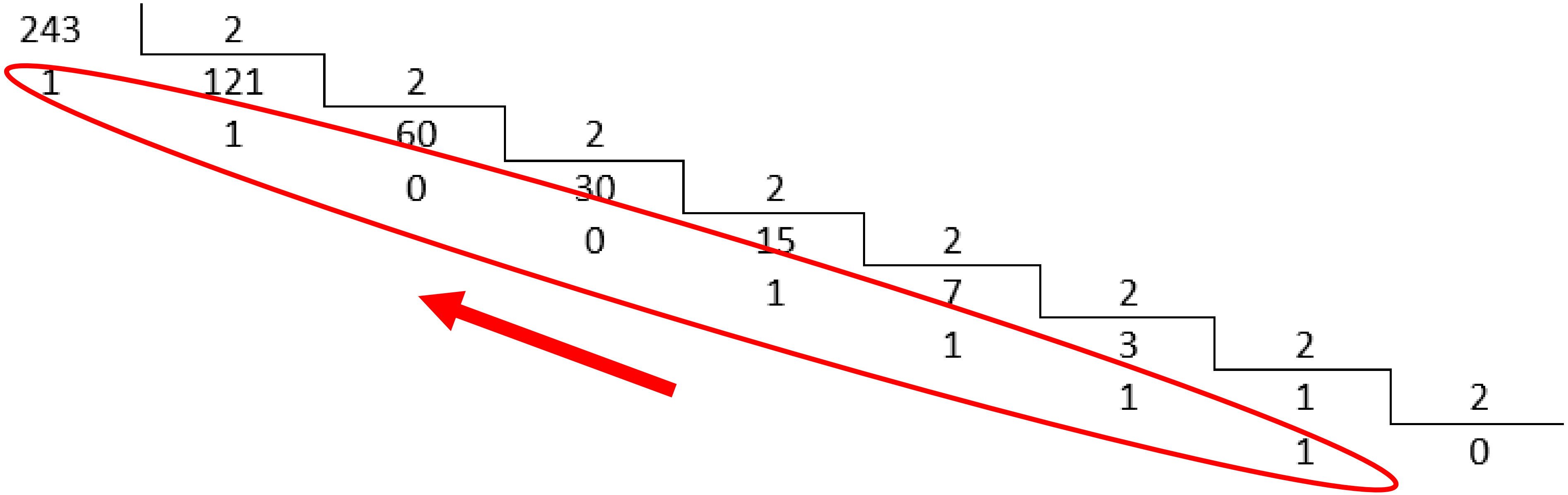
Sistema hexadecimal	Sistema decimal
A	10
B	11
C	12
D	13
E	14
F	15

Sistemas numéricos – Conversão

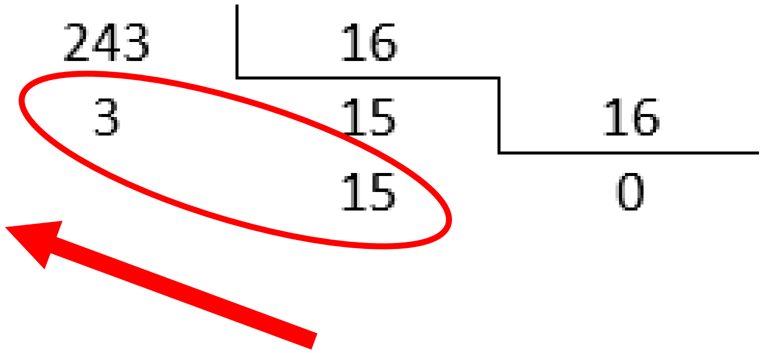
- A conversão do sistema decimal para outros sistemas numéricos é feita por meio de sucessivas divisões com resto pela base desejada, até que não seja mais possível dividir (até que o dividendo seja igual a zero).
- Os restos da divisão, na ordem inversa em que foram obtidos, serão o valor no outro sistema numérico.

Exemplo: converter $\underline{(243)_{10}}$ para os sistemas binário e hexadecimal:

Sistemas numéricos – Conversão



$(243)_{10} = (11110011)_2$



$(243)_{10} = (F3)_{16}$

Sistemas numéricos – Conversão

- Para realizar a conversão de outro sistema numérico para o sistema decimal, divide-se o número nas suas respetivas potências e calcula-se os valores.

- A posição das potências é sempre a mesma em todos os sistemas numéricos: a casa imediatamente à esquerda da vírgula é a base elevada a zero, aumentando-se em 1 para cada dígito à esquerda.

- No caso de números inteiros, as potências começam com o expoente igual a zero no dígito mais à direita, crescendo para a esquerda.

Sistemas numéricos – Conversão

Exemplos: converter os números $(100111)_2$ e $(AB3C)_{16}$ para o sistema decimal:

- $(100111)_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$
  $= 1 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 32 + 0 + 0 + 4 + 2 + 1 =$
  $= \underline{(39)_{10}}$

- $(AB3C)_{16} = A \cdot 16^3 + B \cdot 16^2 + 3 \cdot 16^1 + C \cdot 16^0 =$
  $= 10 \cdot 4096 + 11 \cdot 256 + 3 \cdot 16 + 12 \cdot 1 =$
  $= \underline{(43836)_{10}}$

Sistemas numéricos – Uso do sistema binário

- O uso do sistema binário, na Ciência da Computação, se deve ao fato deste possuir, apenas, dois algarismos (0 e 1).
- Assim, um dígito de um valor numérico, representado neste sistema, pode ser representado por dois estados elétrica e eletronicamente: com carga elétrica (1) ou sem carga elétrica (0), ligado (1) ou desligado (0), magnetizado ou desmagnetizado (0).
- Na Ciência da Computação, a menor unidade de armazenamento possível é o *bit*, que pode armazenar um dígito binário (um ou zero).
- Um conjunto de oito *bits* forma um *byte*. Assim sendo, um *byte* pode representar 256 (2^8) valores diferentes.

Sistemas numéricos – Uso do sistema binário

- Como os computadores utilizam-se do sistema binário, os prefixos utilizados para indicar a ordem de grandeza são diferentes dos utilizados no Sistema Internacional de Medidas (SI).
- Assim, em vez de corresponder a uma variação de 1000 (10^3) vezes, cada prefixo corresponde a uma variação de 1024 (2^{10}) vezes.
- Dessa forma, por exemplo, 1 *kbyte* não corresponde a 1000 *bytes*, mas sim a 1.024 *bytes*.

Prefixo	Valor	Valor exato	Valor do prefixo no SI
Quilo (k)	2^{10}	1.024	10^3 (um mil)
Mega (M)	2^{20}	1.048.576	10^6 (um milhão)
Giga (G)	2^{30}	1.073.741.824	10^9 (um bilhão)
Tera (T)	2^{40}	1.099.511.627.776	10^{12} (um trilhão)

Interatividade

O valor numérico $(479)_{10}$, quando convertido para os sistemas binário, octal e hexadecimal, resulta, respectivamente, em:

a) $(1\ 1101\ 1011)_2$, $(735)_8$ e $(1DF)_{16}$.
b) $(1\ 1101\ 1111)_2$, $(737)_8$ e $(1DF)_{16}$.
c) $(1\ 1101\ 1111)_2$, $(735)_8$ e $(1DE)_{16}$.
d) $(1\ 1101\ 1011)_2$, $(737)_8$ e $(1DE)_{16}$.
e) $(1\ 1101\ 1111)_2$, $(737)_8$ e $(1DE)_{16}$.

Resposta

O valor numérico $(479)_{10}$, quando convertido para os sistemas binário, octal e hexadecimal, resulta, respectivamente, em:

a) $(1\ 1101\ 1011)_2$, $(735)_8$ e $(1DF)_{16}$.
b) $(1\ 1101\ 1111)_2$, $(737)_8$ e $(1DF)_{16}$.
c) $(1\ 1101\ 1111)_2$, $(735)_8$ e $(1DE)_{16}$.
d) $(1\ 1101\ 1011)_2$, $(737)_8$ e $(1DE)_{16}$.
e) $(1\ 1101\ 1111)_2$, $(737)_8$ e $(1DE)_{16}$.

Sistemas numéricos – Conversão

- Os valores 8 e 16, que são as bases dos sistemas octal e hexadecimal, respectivamente, são potências de 2: $8 = 2^3$ e $16 = 2^4$.

- Assim, é possível realizar as conversões do sistema binário para estes outros dois e vice-versa, sem a necessidade de passar pelo sistema decimal.

- Para converter um número binário para uma destas outras bases, separamos o número binário em grupos de dígitos (grupos de 3 dígitos para converter para o sistema octal e de 4 dígitos para o hexadecimal), da direita para a esquerda e, simplesmente, substituímos pelo valor equivalente no outro sistema.

Sistemas numéricos – Conversão

- Exemplo: converter o número $(1010011)_2$ do sistema binário para os sistemas octal e hexadecimal.

Para octal: separamos o número em grupos de três dígitos:

$(1 \quad 010 \quad 011)_2$
$(1)_2 = (1)_8$
$(010)_2 = (2)_8$
$(011)_2 = (3)_8$
$\mathbf{(1 \;\; 010 \;\; 011)_2 = (1\ 2\ 3)_8}$

Para hexadecimal: separamos o número em grupos de quatro dígitos:

$(101 \;\; 0011)_2$
$(101)_2 = (5)_{16}$
$(0011)_2 = (3)_{16}$
$\mathbf{(101 \;\; 0011)_2 = (5\ 3)_{16}}$

Sistemas numéricos – Conversão

- Para converter um número dos sistemas octal ou hexadecimal para o sistema binário, fazemos o oposto: cada dígito do número nestes sistemas será convertido em um conjunto de números binários (novamente, grupos de 3 dígitos para converter do sistema octal e de 4 dígitos do hexadecimal).

<u>Exemplo: $(3F4)_{16}$</u>:

$(3)_{16} = (0011)_2$
$(F)_{16} = (1111)_2$
$(4)_{16} = (0100)_2$

$\mathbf{(3F4)_{16} = (001111110100)_2}$

Sistemas numéricos – Conversão de números reais

- Para converter um número real não inteiro do sistema decimal para os demais sistemas, a conversão é feita em duas partes: a parte inteira do número é convertida como mostrado anteriormente.

- A parte fracionária do número é convertida multiplicando-a pelo valor da base, sucessivamente, até chegar em um número cuja parte fracionária seja zero. A parte fracionária do número convertido corresponderá às partes inteiras dos resultados das multiplicações, na ordem em que foram obtidas.

    - Um detalhe importante é que apenas a parte fracionária do produto anterior é multiplicada, novamente, pela base.

Sistemas numéricos – Conversão de números reais

Exemplo: converter o valor $(0{,}6875)_{10}$ para os sistemas binário, octal e hexadecimal:

- Binário: $0{,}6875 \cdot 2 = 1{,}375 = 1 + 0{,}375$
  $0{,}375 \cdot 2 = 1{,}5 = 1 + 0{,}5$
  $0{,}5 \cdot 2 = 1{,}0 = 1 + 0{,}0$

$(0{,}6875)_{10} = (0{,}111)_2$

- Octal: $0{,}6875 \cdot 8 = 5{,}5 = 5 + 0{,}5$
  $0\ 5 \cdot 8 = 4{,}0 = 4 + 0{,}0$

$(0{,}6875)_{10} = (0{,}54)_8$

- Hexadecimal: $0{,}6875 \cdot 16 = 11{,}0 = 11 + 0{,}0$

$(0{,}6875)_{10} = (0{,}B)_{16}$

Sistemas numéricos – Conversão de números reais

- Como o número 10, que é a base do sistema decimal, não é uma potência de 2, muitos números ao serem convertidos para outros sistemas numéricos se tornam dízimas periódicas.

Por exemplo, ao converter o número $(0,7)_{10}$ para o sistema binário, uma dízima é obtida:

$$
\begin{aligned}
0,7 \cdot 2 &= 1,4 = 1 + 0,4 \\
0,4 \cdot 2 &= 0,8 = 0 + 0,8 \\
0,8 \cdot 2 &= 1,6 = 1 + 0,6 \\
0,6 \cdot 2 &= 1,2 = 1 + 0,2 \\
0,2 \cdot 2 &= 0,2 = 0 + 0,4 \\
0,4 \cdot 2 &= 0,8 = 0 + 0,8 \\
0,8 \cdot 2 &= 1,6 = 1 + 0,6 \\
0,6 \cdot 2 &= 1,2 = 1 + 0,2 \\
0,2 \cdot 2 &= 0,2 = 0 + 0,4 \\
&\ldots
\end{aligned}
$$

Sistemas numéricos – Conversão de números reais

- Caso ocorra uma dízima, ao armazenar um número real convertido para o sistema binário, como cada número possui um limite de *bytes* em um computador, teremos que eliminar uma parte das infinitas casas decimais.

- Assim, o simples fato de armazenarmos um número na memória de um computador pode resultar em um erro.

- Na Matemática, o erro significa uma diferença entre um valor exato e um valor que está sendo utilizado em determinado cálculo.

Sistemas numéricos – Conversão de números reais

- A conversão de um número real de um outro sistema para o sistema decimal é feita da mesma forma que a conversão de um número inteiro.

- Porém os dígitos à esquerda da vírgula contarão como expoentes negativos, decrescendo da esquerda para a direita.

- Relembrando: em qualquer sistema de numeração, o dígito imediatamente à esquerda da vírgula corresponderá à base elevada à zero; o expoente cresce de um em um para a esquerda e decresce de um em um para a direita.

Sistemas numéricos – Conversão de números reais

Exemplos: converter os números para o sistema decimal:

- $(1011{,}011)_2$

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

$= 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 + 0 \cdot 0{,}5 + 1 \cdot 0{,}25 + 1 \cdot 0{,}125 =$

$= 8 + 0 + 2 + 1 + 0 + 0{,}25 + 0{,}125 =$

$= \underline{(11{,}375)_{10}}$

- $(D3{,}2E)_{16} =$

$= D \cdot 16^1 + 3 \cdot 16^0 + 2 \cdot 16^{-1} + E \cdot 16^{-2} =$

$= 13 \cdot 16 + 3 \cdot 1 + 2 \cdot 0{,}0625 + 15 \cdot 0{,}00390625 =$

$= 208 + 3 + 0{,}125 + 0{,}05859375 =$

$= \underline{(211{,}18359375)_{10}}$

Interatividade

O valor numérico $(101\ 010\ 011{,}1011)_2$, quando convertido para o sistema decimal, irá resultar em um número com quantos dígitos?

a) 5 dígitos.
b) 6 dígitos.
c) 7 dígitos.
d) 8 dígitos.
e) 9 dígitos.

Resposta

O valor numérico $(101\ 010\ 011{,}1011)_2$, quando convertido para o sistema decimal, irá resultar em um número com quantos dígitos?

a) 5 dígitos.
b) 6 dígitos.
c) 7 dígitos.
d) 8 dígitos.
e) 9 dígitos.

$(101\ 010\ 011)_2 = (339)_{10}$

$(0{,}1011)_2 = (0{,}6875)_{10}$

$(101\ 010\ 011{,}1011)_2 = (153{,}6875)_{10}$

Operações com números binários

- A Ciência da Computação se baseia em operações aritméticas utilizando os números binários, que estão armazenados em *bits*.

- As operações são as mesmas operações aritméticas com as quais estamos acostumados no sistema decimal.

- O princípio das operações aritméticas é o mesmo para qualquer base numérica: para podermos realizá-las, devemos entender como são as “tabuadas” das operações aritméticas no sistema binário.

Operações com números binários – Soma

No caso dos números binários, como o maior algarismo é 1, a "tabuada da soma" é a seguinte:

0 + 0 = 0
1 + 0 = 1
1 + 1 = 10
10 + 1 = 1 + 1 + 1 = 11

- Lembremos que $(1)_2 = (1)_{10}$, $(10)_2 = (2)_{10}$ e $(11)_2 = (3)_{10}$.
- Assim, a soma é realizada da mesma forma que no sistema decimal. Os números são alinhados a partir do dígito correspondente à potência 2^0; então, realizamos a soma da direita para a esquerda.
- Quando a soma de dois algarismos resultar em um valor com dois dígitos, "vai um" para o algarismo seguinte.

Operações com números binários – Soma

- Exemplo: efetuar a soma $(1101{,}101)_2 + (1110{,}11)_{2.}$

- Alinhamos os números a partir da vírgula que separa a parte inteira da fracionária.

Executa-se a soma da direita para a esquerda:

	1	1	1	1				
	1	1	0	1	,	1	0	1
+	1	1	1	0	,	1	1	
1	1	1	0	0		0	1	1

- Assim, $(1101{,}101)_2 + (1110{,}11)_2 = (11100{,}011)_{2.}$

Operações com números binários – Subtração

No caso dos números binários, como o maior algarismo é 1, a “tabuada da subtração” é a seguinte:

0 - 0 = 1 - 1 = 0
1 - 0 = 1
10 - 1 = 1
11 - 10 = 1

- Assim, a subtração é realizada da mesma forma que no sistema decimal. Os números são alinhados a partir do dígito correspondente à potência 2^0; então, realizamos a subtração da direita para a esquerda.
- Não iremos abordar, aqui, as operações que resultem em números negativos. Para estes casos, existe uma notação específica denominada Complemento de 2.

Operações com números binários – Subtração

<u>Exemplo</u>: efetuar a subtração $(11101,11)_2 – (1110,1)_{2}$:

- Alinhamos os números a partir da vírgula que separa a parte inteira da fracionária.

Executa-se a subtração da direita para a esquerda:

	(-1)	(-1)(+10)	(-1)(+10)	(+10)				
	1	1	1	0	1	,	1	1
-		1	1	1	0	,	1	
	0	1	1	1	1	,	0	1

- Assim, $(11101,11)_2$ - $(1110,1)_2 = (1111,01)_{2.}$

Operações com números binários – Multiplicação

No caso dos números binários, como o maior algarismo é 1, a “tabuada da multiplicação” é a seguinte:

0 x 0 = 0
1 x 0 = 0
1 x 1 = 1

- Assim, a multiplicação é realizada da mesma forma que no sistema decimal. Os números são alinhados a partir do dígito correspondente à potência 2^0; então, realizamos a multiplicação, da direita para a esquerda.
- Ao multiplicar os números de diversos dígitos, o produto é feito por partes, e cada resultado é deslocado uma casa para a esquerda.

Operações com números binários – Multiplicação

<u>Exemplo</u>: efetuar a multiplicação $(11011)_2$ x $(101)_{2:}$

- Executa-se o produto da direita para a esquerda, cada produto sendo colocado abaixo, deslocado um dígito para a esquerda.

			1	1	0	1	1
x					1	0	1
			1	1	0	1	1
		0	0	0	0	0	
+	1	1	0	1	1		
1	0	0	0	0	1	1	1

- Assim, $(11011)_2$ x $(101)_2 = (1000111)_{2.}$

Operações com números binários

- Não será abordada, aqui, a divisão.
- Na realidade, todas as operações realizadas em um computador são somas: somas com números negativos (subtração), somas sucessivas (multiplicação) e somas sucessivas com números negativos (divisão).
- As operações aritméticas mantêm as suas mesmas propriedades das operações utilizadas com o sistema decimal (comutatividade, associatividade, distributividade).
- As operações de soma e multiplicação de um único bit correspondem ao funcionamento de Portas Lógicas (OR e AND, respectivamente), as quais serão abordadas logo mais, na disciplina.

Interatividade

A expressão aritmética 11101 + 10001 x 1101, sendo todos os números no sistema binário, resulta em qual valor?

a) 1111 1011.
b) 1110 1011.
c) 1110 1010.
d) 1111 1110.
e) 1111 1010.

Resposta

A expressão aritmética 11101 + 10001 x 1101, sendo todos os números no sistema binário, resulta em qual valor?

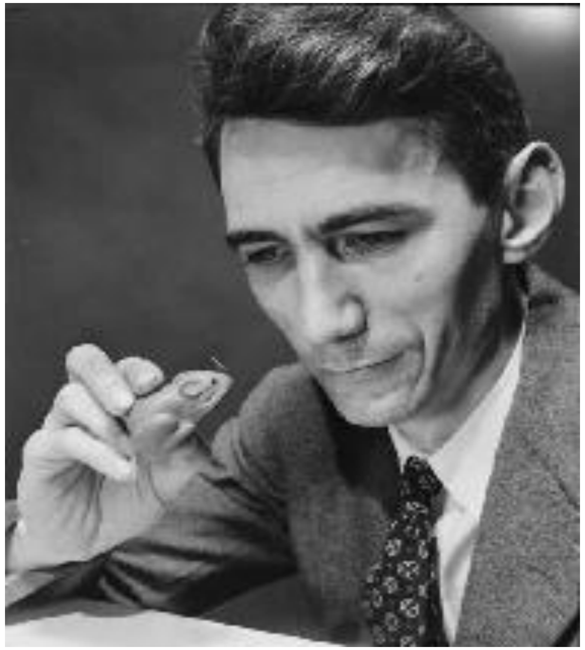
a) 1111 1011.
b) 1110 1011.
c) 1110 1010.
d) 1111 1110.
e) 1111 1010.

10001 x 1101 = 1101 1101

1101 1101 + 11101 = 1111 1010

Portas Lógicas

- Portas Lógicas são dispositivos que representam os operadores lógicos da Lógica Matemática, e recebem uma ou mais entradas lógicas de entrada, para produzir uma única saída, de acordo com a operação lógica correspondente.
- Portas Lógicas são usadas em circuitos eletrônicos para representar as operações lógicas correspondentes.
- Foi a partir de um trabalho do matemático norte-americano Claude Elwood Shannon, de 1937, que foram desenvolvidas as Portas Lógicas, que constituem a base dos Circuitos Lógicos Digitais.



Fonte: https://commons.wikimedia.org/wiki/File:Claude_Shannon_1776.jpg

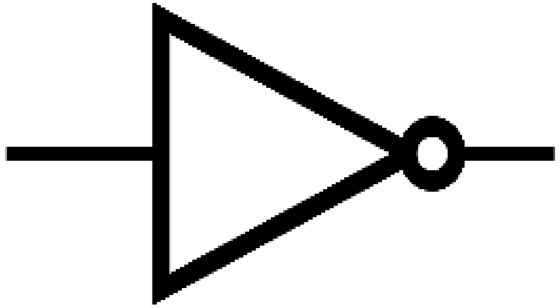
Portas Lógicas

- Existem <u>sete</u> Portas Lógicas.
- Em Circuitos Lógicos Digitais, os sinais possíveis serão 0 (falso) ou 1 (verdadeiro).
- As Portas Lógicas são, geralmente, indicadas por seu nome em inglês – em alguns, por uma abreviação da operação lógica correspondente.
- As Portas Lógicas serão apresentadas com a sua simbologia pela norma <u>ANSI</u> (*American National Standards Institute*), que é a mais utilizada. Embora existam outras simbologias, esta será a empregada ao longo de todo o curso.
- Os terminais à esquerda de cada Porta Lógica são as entradas, e o terminal à direita representa a saída única de cada uma delas.
- Cada Porta Lógica <u>terá, sempre, uma única saída</u>, independentemente da quantidade de entradas que tiver.
- Portas Lógicas são os elementos constituintes de qualquer Circuito Lógico Digital.

Portas Lógicas – Porta NOT

- O operador de negação inverte o valor lógico de uma entrada. Ou seja, 0 virará 1 e 1 virará 0.
- A Porta Lógica NOT representa esta operação.
- Em Circuitos Lógicos Digitais, usamos um traço sobre a entrada ou a associação de entradas: assim, a notação $\overline{A}$, que será adotada de agora em diante, será para indicar a negação da entrada A.

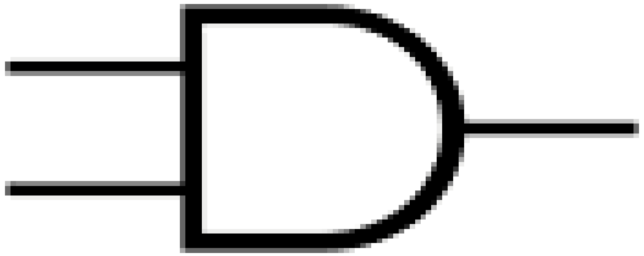
A	$\overline{A}$
0	1
1	0



Portas Lógicas – Porta AND

- A Porta Lógica AND representa o operador lógico E, que associa duas ou mais entradas na forma A · B (lê-se “A e B”).
- Esta operação somente terá um valor lógico verdadeiro quando todas as entradas forem verdadeiras.

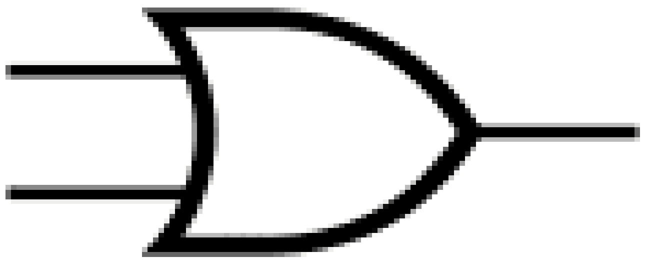
A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1



Portas Lógicas – Porta OR

- A Porta Lógica OR representa o operador lógico OU, que associa duas ou mais entradas na forma A + B (lê-se “A ou B”).
- Esta operação terá um valor lógico verdadeiro quando, ao menos, uma das entradas for verdadeira, sendo falsa, somente, quando todas as entradas forem falsas.

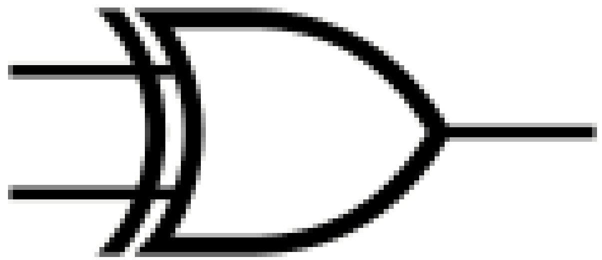
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1



Portas Lógicas – Porta XOR

- A Porta Lógica XOR representa o operador lógico OU EXCLUSIVO, que associa duas entradas na forma $A \oplus B$ (lê-se “A ou exclusivo B”, ou “ou A ou B”).
- Para duas entradas, esta operação é <u>verdadeira quando, apenas, uma das entradas for verdadeira, e é falsa quando ambas ou nenhuma das entradas forem verdadeiras</u>.

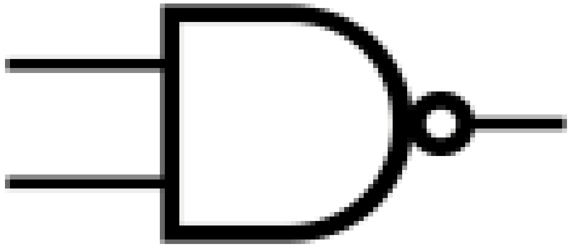
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



Portas Lógicas – Porta NAND

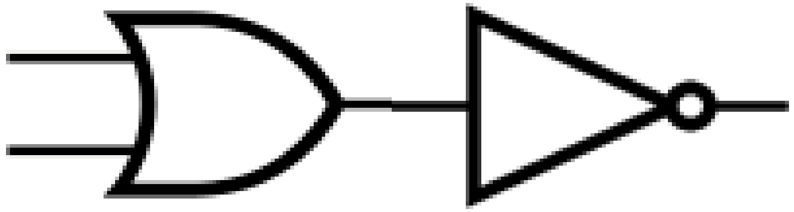
- A Porta Lógica NAND representa a negação da Porta Lógica AND.
- Assim, a operação é falsa, apenas, quando todas as entradas forem verdadeiras, e é verdadeira quando, pelo menos, uma ou todas as entradas forem falsas.

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

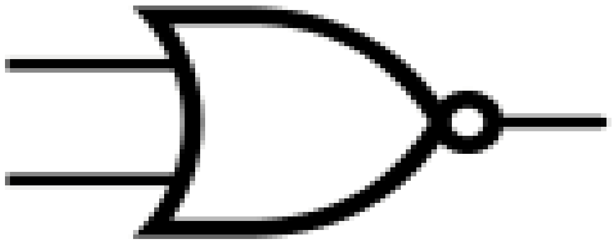


Portas Lógicas – Porta NOR

- A Porta Lógica NOR representa a negação da Porta Lógica OR.
- Assim, a operação é verdadeira, apenas, quando apenas todas as entradas forem falsas, e é falsa quando, pelo menos, uma ou todas as entradas forem verdadeiras.

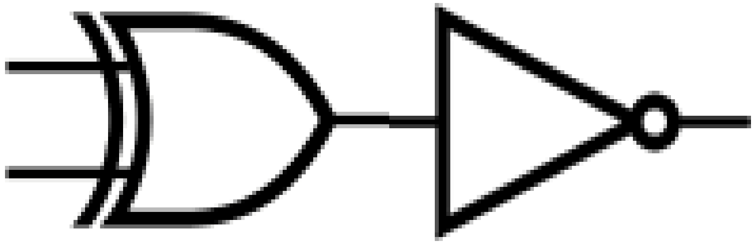


A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

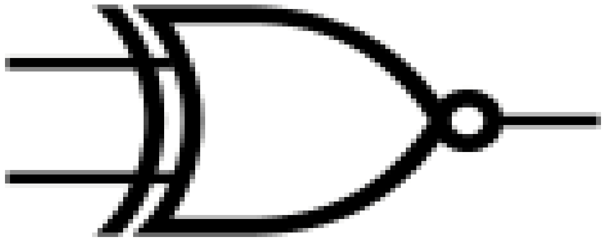


Portas Lógicas – Porta NXOR

- A Porta Lógica NXOR representa a negação da Porta Lógica XOR.
- Para duas entradas, esta operação é falsa, apenas, quando apenas uma das entradas for verdadeira, e é verdadeira quando ambas ou nenhuma das entradas forem verdadeiras.



A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1



Portas Lógicas com múltiplas entradas

- Uma Porta Lógica não está restrita a duas entradas (com exceção da Porta NOT, que terá sempre, apenas, uma entrada).
- Para as Porta Lógicas AND e OR, bem como para as suas negações (NAND e NOR), as regras de operação continuam as mesmas.
- A associação com a Porta XOR será <u>verdadeira, somente, quando o número de entradas com um valor lógico verdadeiro for ímpar</u>.
- A associação com a Porta NXOR será <u>verdadeira, somente, quando o número de entradas com um valor lógico verdadeiro for par</u>.

Interatividade

Em relação à operação das Portas Lógicas, qual das alternativas a seguir apresenta o funcionamento **incorreto** de uma Porta Lógica?

a) AND: somente terá um valor lógico verdadeiro quando todas as entradas forem verdadeiras.
b) OR: terá um valor lógico verdadeiro quando, ao menos, uma das entradas for verdadeira, sendo falsa, somente, quando uma das entradas for falsa.
c) XOR: é verdadeira quando, apenas, uma das entradas for verdadeira, e é falsa quando ambas ou nenhuma das entradas forem verdadeiras.
d) NAND: falsa, apenas, quando todas as entradas forem verdadeiras, e é verdadeira quando, pelo menos, uma ou todas as entradas forem falsas.
e) NXOR: a associação será verdadeira, somente, quando o número de entradas com um valor lógico verdadeiro for par.

Resposta

Em relação à operação das Portas Lógicas, qual das alternativas a seguir apresenta o funcionamento **incorreto** de uma Porta Lógica?

a) AND: somente terá um valor lógico verdadeiro quando todas as entradas forem verdadeiras.

b) OR: terá um valor lógico verdadeiro quando, ao menos, uma das entradas for verdadeira, sendo falsa, somente, quando uma das entradas for falsa.

c) XOR: é verdadeira quando, apenas, uma das entradas for verdadeira, e é falsa quando ambas ou nenhuma das entradas forem verdadeiras.

d) NAND: falsa, apenas, quando todas as entradas forem verdadeiras, e é verdadeira quando, pelo menos, uma ou todas as entradas forem falsas.

e) NXOR: a associação será verdadeira, somente, quando o número de entradas com um valor lógico verdadeiro for par.

Referências

- ALENCAR FILHO, E. *Iniciação à lógica matemática*. São Paulo: Nobel, 2008.
- IDOETA, I. V.; CAPUANO, F. G. *Elementos de eletrônica digital*. São Paulo: Érica, 1998.

ATÉ A PRÓXIMA!