

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SANTA CATARINA
Campus Araranguá**

ELETRÔNICA DIGITAL

CURSO TÉCNICO EM ELETROMECCÂNICA

Prof. Eng. Eletricista João Francisco Veremzuk Xavier

AGRADECIMENTO a Unidade São José por gentilmente ter cedido este material.

1. Sistemas Analógico e Digital

1.1 Introdução

Eletrônica digital é a parte da eletrônica que trabalha com sinais discretos. Em contrapartida existe a eletrônica analógica que trabalha com sinais analógicos ou também contínuos.

Ambas as eletrônicas, digital e analógica usam os mesmos componentes ou sejam resistores, diodos, transistores, fios condutores, etc.

Sinais Analógicos: São sinais contínuos no tempo. No sinal analógico a passagem de uma condição para outra se dá de forma suave, sem descontinuidade.

O mundo físico real é essencialmente analógico, onde os sinais, que representam informações, aparecem de modo contínuo.

Sinais Digitais: São sinais discretos no tempo, de tal forma que sempre existe uma descontinuidade entre uma condição e outra.

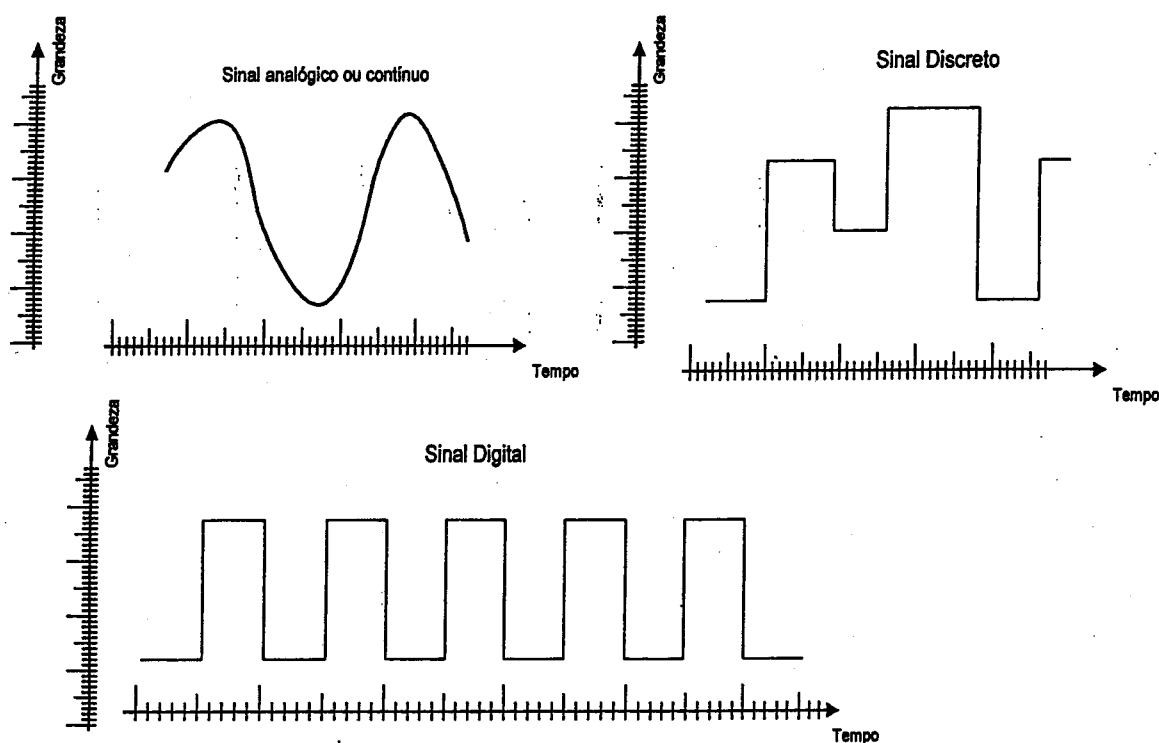


Figura 1 – Formas de onda de sinais analógicos, discretos e digitais

Na Figura 1, observe que o sinal contínuo pode assumir qualquer valor. O sinal discreto pode assumir somente alguns valores possíveis e o **sinal discreto digital somente dois valores**.

Um bom exemplo de sinais contínuos e sinais discretos é o mostrador de um termômetro analógico, aquele que usa uma coluna de mercúrio, e um termômetro de mostrador digital. Note bem que o mostrador pode ser analógico ou digital mas a grandeza física que eles representam sempre serão analógicos. No termômetro analógico você pode fazer infinitas leituras entre as marcações 22 graus e 23 graus, mas um termômetro de marcador digital ou ele indica 22 graus ou 23 graus, ele não mostra valores intermediários. Se quisermos um termômetro digital com melhor resolução nos valores mostrados, teremos que comprar um mais caro que por exemplo marque décimos de grau. Assim podemos ler: 22,0 22,1; 22,2 ... Mas não

poderemos ler entre, por exemplo 22,1 e 22,2. No termômetro com mostrador por coluna de mercúrio colocando-se uma lupa no mostrador pode-se ler valores entre décimos ou até milésimos.

Como já mencionado, o sinal discreto digital poderá assumir somente dois valores que são expressos por afirmações declarativas (cada valor está associado a um significado) e são mutuamente exclusivos.

As vantagens dos sinais digitais são as seguintes :

- Mais Fáceis de Projetar
- Facilidade de Armazenamento da Informação Digital
- Operação Programada
- Menos Suscetível a Ruídos
- Maior Integração dos Circuitos Integrados(CIs)

Como o mundo físico real os sinais são analógicos e com a eletrônica digital queremos lidar com sinais discretos, terá que haver uma transformação de sinais analógico para sinal digital. Primeiramente trabalharemos com os sistemas de numeração, observando e aprendendo como representar os números com apenas dois valores possíveis.

2. Sistemas de Numeração

2.1 Introdução – Os Números

Acredita-se que a necessidade de criação de números veio com a necessidade de contar. Seja o número de animais, alimentos, ou coisas do tipo. Como a evolução nos legou algumas características, como os cinco dedos em cada mão e cinco dedos em cada pé, seria muito natural que os primeiros sistemas de numeração fizessem uso das bases 10 (decimal) e 20 (vigesimal).

Em eletrônica e Computação, as bases mais utilizadas para sistemas de numeração são:

- Decimal (Base 10)
- Binária (Base 2)
- Octal (Base 8)
- Hexadecimal (Base 16)

2.2 Sistema de Numeração Decimal

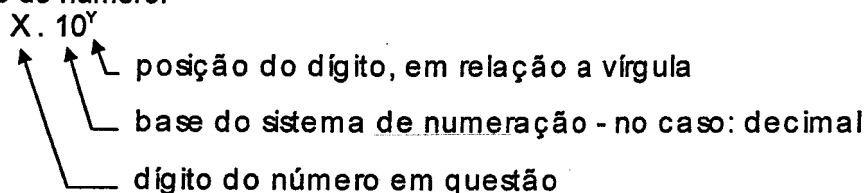
O sistema de numeração normalmente utilizado, o sistema decimal, apresenta dez dígitos (algarismos), são eles: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. No sistema decimal, 10 é a base do sistema.

Obs.: para um sistema de base N, os dígitos vão de 0 à N-1.

$$\begin{aligned}\text{Ex.: } 328451_{10} &= 3 \times 10^5 + 2 \times 10^4 + 8 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 1 \times 10^0 \\ &= 300000 + 20000 + 8000 + 400 + 50 + 1 \\ &= 328451_{10}\end{aligned}$$

Descrição do número:

$X \cdot 10^Y$



2.3 Sistema de Numeração Binário

Este sistema de numeração, como o próprio nome sugere, apresenta base 2. Os números 0 e 1 são os dígitos deste sistema.

Para representarmos a quantidade zero, utilizamos o algarismo (0), para representarmos a quantidade um utilizamos o algarismo (1). E para representarmos a quantidade dois, se nós não possuímos o algarismo (2) nesse sistema? Basta lembrar-se de como é obtido o número dez no sistema de numeração decimal, onde os dígitos vão de 0 a 9.

Representamos a quantidade de uma dezena utilizando o algarismo 1 (um) seguido do algarismo 0 (zero). Neste caso, o algarismo 1 (um) significa que temos um grupo de uma dezena e o algarismo 0 (zero) nenhuma unidade, o que significa dez.

No sistema binário agimos da mesma forma, para representarmos a quantidade dois, utilizamos o algarismo (1) seguido do algarismo (0). Sendo assim, a numeração em binário vai tornar-se:

Decimal	Binário
0	0
1	1
2	10
3	11
4	100
5	101
.	.
.	.
.	.

O sistema binário é de grande importância, pois apresenta correspondência direta com os estados de um sistema digital. Por exemplo: para o dígito 0 pode-se atribuir o valor de tensão 0 V e para o dígito 1 pode-se atribuir o valor de tensão de 5 V.

$$\begin{aligned}\text{Ex.: } 1001101_2 &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 64 + 0 + 0 + 8 + 4 + 0 + 1 \\ &= 77_{10}\end{aligned}$$

2.3.1 Conversão de um número no sistema binário para o equivalente no sistema decimal.

Regra geral: multiplica-se cada dígito pelo valor da base elevada a uma dada potência, definida pela posição do dígito, e finalmente realiza-se a soma.

$$\begin{aligned}\text{Ex.: } 11001101_2 &= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 128 + 64 + 0 + 0 + 8 + 4 + 0 + 1 \\ &= 205_{10}\end{aligned}$$

2.3.2 Conversão de decimal para binário.

Ex.: Conversão do número 23_{10} para binário.

$$\begin{array}{r} 23 \overline{) 2} \\ 1 \quad 11 \end{array} \longrightarrow 23 = 2 \times 11 + 1$$

$$\begin{array}{r} 11 \overline{) 2} \\ 1 \quad 5 \end{array} \longrightarrow 23 = 2 \times (2 \times 5 + 1) + 1 = 5 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$\begin{array}{r} 5 \overline{) 2} \\ 1 \quad 2 \end{array} \longrightarrow 23 = (2 \times 2 + 1) \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$\begin{array}{r} 2 \overline{) 2} \\ 0 \quad 1 \end{array} \longrightarrow 23 = (1 \times 2) \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$= 23_{10}$$

Regra prática:

$$\begin{array}{r} 23 \overline{) 1} \\ \textcircled{1} \quad 11 \overline{) 2} \\ \textcircled{1} \quad 5 \overline{) 2} \\ \textcircled{1} \quad 2 \overline{) 2} \\ \textcircled{0} \quad 1 \overline{) 2} \\ \textcircled{1} \quad 0 \end{array}$$

$$10111_2 = 23_{10}$$

2.3.3 Conversão de números fracionários

Regra de formação:

$$\text{Decimal: } 197,526_{10} = 1 \times 10^2 + 9 \times 10^1 + 7 \times 10^0 + 5 \times 10^{-1} + 2 \times 10^{-2} + 6 \times 10^{-3}$$

$$\text{Binário: } 101101,101 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

2.3.3.1 Conversão de binário para decimal

$$\begin{aligned} 1101,111_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ &= 8 + 4 + 0 + 1 + 0,5 + 0,25 + 0,125 \\ &= 13 + 0,875 \\ &= 13,875_{10} \end{aligned}$$

2.3.3.2 Conversão de decimal para binário

$$35,625_{10} = ?_2$$

$$35,625_{10} = \underbrace{35_{10}}_{\text{parte inteira}} + \underbrace{0,625_{10}}_{\text{parte fracionária}}$$

A conversão da parte inteira segue o procedimento já descrito:

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

A conversão da parte fracionária segue a seguinte regra prática:

- Multiplica-se a parte fracionária pelo valor da base.
- O número resultante a esquerda da vírgula é o dígito (0 ou 1) procurado.
- Se o dígito à esquerda for 0 (zero) continuar a multiplicação pela base.
- Se o dígito à esquerda for 1 este é retirado e prossegue-se a multiplicação.
- O processo continua até obter-se 0 (zero) como resultado ou atingir-se a resolução estabelecida, no caso de dízima.
- A leitura dos dígitos, ao contrário do caso da parte inteira, é feita de cima para baixo.

$$\begin{array}{r}
 0,625_{10} \\
 \times 2 \quad \leftarrow \text{base do sistema} \\
 \hline
 \textcircled{1},250 \\
 \\
 0,25_{10} \\
 \times 2 \\
 \hline
 \textcircled{0},50 \\
 \\
 \times 2 \\
 \hline
 \textcircled{1},00
 \end{array}$$

$$0,625_{10} = 0,101_2$$

$$35,625_{10} = 100011,101_2$$

Exercícios:

1. Converter para decimal:

- 11010101010101₂
- 11011010101₂
- 11000001₂
- 1,01010101₂
- 1001,010101₂
- 1101₂
- 11,010101₂
- 11010101,010101₂
- 11010,1101₂

2. Converter para binário com 5 casas após a vírgula:

- 234,435₁₀
- 945,3445₁₀
- 5235₁₀

- d) $12,234_{10}$
- e) $23,3457_{10}$
- f) $94,345_{10}$
- g) $43,956_{10}$
- h) $9,4567_{10}$

2.4 Sistema Octal de Numeração

A base de um sistema numérico é igual o número de dígitos que ela usa. Portanto, o sistema octal, que apresenta base 8, tem 8 dígitos a saber: 0, 1, 2, 3, 4, 5, 6, 7 (base $N = 8 \rightarrow$ dígitos $0 \rightarrow N-1 = 7$).

Sua utilidade nos sistemas digitais vem do fato de que, associando-se os algarismos de um número binário (bits) em grupos de três, obtém-se uma correspondência direta com os dígitos do sistema octal. Observaremos nitidamente este mais adiante.

2.4.1 Conversão de Octal em Decimal

$$1247,235_8 = ?_{10}$$

$$1 \times 8^3 + 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} + 3 \times 8^{-2} + 5 \times 8^{-3}$$

$$512 + 128 + 32 + 7 + 1/8 + 3/64 + 5/512$$

$$1247,235_8 = 679,1816406_{10}$$

2.4.2 Conversão de Decimal para Octal

Converter o número 223 da base decimal para a octal.

$$\begin{array}{r} 223 \div 8 \\ \hline 27 \text{ R } 7 \\ 27 \div 8 \\ \hline 3 \text{ R } 3 \\ 3 \div 8 \\ \hline 0 \text{ R } 3 \end{array}$$

$223_{10} = 337_8$

Converter o número fracionário 381,796 da base decimal para octal (4 casas decimais após a vírgula).

$$381,796_{10} = 381_{10} + 0,796_{10}$$

Parte inteira:

$$\begin{array}{r} 381 \div 8 \\ \hline 47 \text{ R } 5 \\ 47 \div 8 \\ \hline 5 \text{ R } 7 \\ 5 \div 8 \\ \hline 0 \text{ R } 5 \end{array}$$

$381_{10} = 575_8$

Parte fracionária:

$$\begin{array}{r}
 0,796_{10} \\
 \times 8 \\
 \hline
 \textcircled{6},368 \\
 0,368_{10} \\
 \times 8 \\
 \hline
 \textcircled{2},944 \\
 \times 8 \\
 \hline
 \textcircled{7},952 \\
 \times 8 \\
 \hline
 \textcircled{4},416
 \end{array}$$

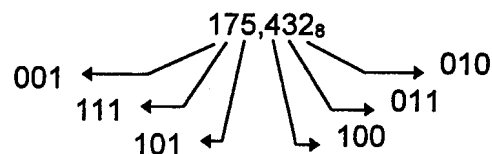
$$0,796_{10} \approx 0,6274_8 \text{ (aproximado)}$$

2.4.3 Conversão de Octal em Binário

Para converter um número expresso em uma determinada base é normal convertermos o primeiro para um número na base 10 e, em seguida, fazer a conversão para a base desejada. Entretanto, como já foi dito, no caso do octal para o binário (e vice-versa) podemos fazer a conversão diretamente, sem passar pelo sistema decimal, já que, 8 é terceira potência de 2 e, portanto, são múltiplos e tem correspondência direta um com o outro.

Regra: Cada dígito octal, a partir da vírgula, é representado pelo equivalente a três dígitos binários. A tabela de equivalência é mostrada a seguir.

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



$$175,432_8 = 001\ 111\ 101,100\ 011\ 010_2$$

2.4.4 Conversão de Binário em Octal

Agrega-se os dígitos binários, a partir da vírgula, em grupos de três e converte-se para o equivalente em octal. Caso os dígitos extremos, da direita ou esquerda, não formarem um grupo completo de três, adiciona-se zeros até que isto ocorra.

Converter os seguintes números de binário para octal.

$$101110,011101_2 = \frac{101}{5} \frac{110}{6} , \frac{011}{3} \frac{101}{5}_8$$

$$1011,11101_2 = \frac{001}{1} \frac{011}{3}, \frac{111}{7} \frac{010}{2}_8$$

Converter o número 677_{10} para binário.

1ª alternativa: dividir 677_{10} sucessivamente por 2. Solução bastante extensa.

2ª alternativa: converter 677_{10} para octal e, em seguida, converter para binário. Solução menos trabalhosa).

$$677_{10} = 1245_8 = 1010100101_2$$

2.5 Sistema de Numeração Hexadecimal

Este sistema apresenta base igual a 16. Portanto 16 dígitos distintos. São usados os dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Como no sistema de numeração octal, o hexadecimal apresenta equivalência direta entre seus dígitos e grupos de quatro dígitos binários. A tabela a seguir mostra esta equivalência.

Decimal	Binário	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

2.5.1 Conversão de Hexadecimal para Decimal

A regra é a mesma da conversão de qualquer sistema de numeração para o decimal.

$$AFC0,7D_{16} = ?_{10}$$

$$A \times 16^3 + F \times 16^2 + C \times 16^1 + 0 \times 16^0 + 7 \times 16^{-1} + D \times 16^{-2}$$

$$10 \times 16^3 + 15 \times 16^2 + 12 \times 16^1 + 0 \times 16^0 + 7 \times 16^{-1} + 13 \times 16^{-2}$$

$$44992,48828_{10}$$

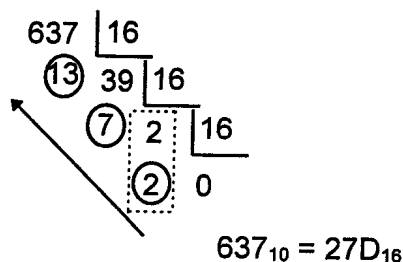
2.5.2 Conversão de Decimal para Hexadecimal

A regra é a mesma da conversão do decimal para qualquer sistema de numeração.

$$637,33_{10} = ?_{16}$$

$$637,33_{10} = 637_{10} + 0,33_{10}$$

Parte inteira



0,33₁₀

x 16

5,28

0,28₁₀

x 16

4,48

x 16

7,68

x 16

10,88

0,33₁₀ ≈ 0,547A₁₆ (aproximado)

Parte Fracionária

2.5.3 Conversão de Hexadecimal em Binário

Da mesma forma que no sistema octal, não é necessário converter o número para o sistema decimal e depois para binário. Basta representar cada dígito hexadecimal, a partir da vírgula, em grupos de quatro dígitos binários equivalentes. A base 16 é a quarta potência da base 2. A tabela de equivalência é a que foi apresentada acima.

FACA, CACA₁₆ = ?₂

F A C A , C A C A ₁₆
1111 1010 1100 1010 , 1100 1010 1100 1010 ₂

FACA, CACA₁₆ = 1111101011001010,1100101011001010₂

2.5.4 Conversão de Binário para Hexadecimal

Como no caso da conversão de binário para octal, agrega-se os dígitos binários, a partir da vírgula, em grupos de quatro e converte-se para o equivalente em hexadecimal. Caso os dígitos extremos, da direita ou esquerda, não formarem um grupo completo de quatro, adiciona-se zeros até que isto ocorra.

100101010,00111₂ = ?₁₆

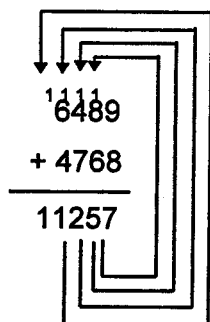
0001 0010 1010 , 0011 1000₂
1 2 A , 3 8₁₆

100101010,00111₂ = 12A,38₁₆

2.6 Operações Aritméticas

2.6.1 Adição e Subtração no Sistema de Numeração Decimal

Adição:

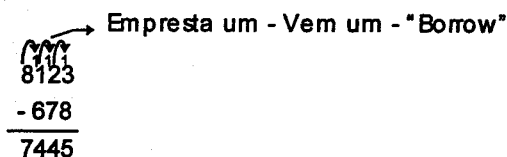


Vai um - Transporte - "Carry"

Obs.: Toda vez que a soma de dígitos for maior ou igual ao valor da base (10) ocorre um vai um.

Subtração:

$$\begin{array}{r} 9 \\ -8 \\ \hline 1 \end{array}$$



Empresta um - Vem um - "Borrow"

2.6.2 Adição e Subtração no Sistema de Numeração Binário

Procede-se da mesma forma que nas operações no sistema decimal, só que, neste caso, tem-se apenas dois dígitos.

Adição:

Regras:

$$0 + 0 = 0_2$$

$$0 + 1 = 1_2$$

$$1 + 0 = 1_2$$

$$1 + 1 = 10_2$$

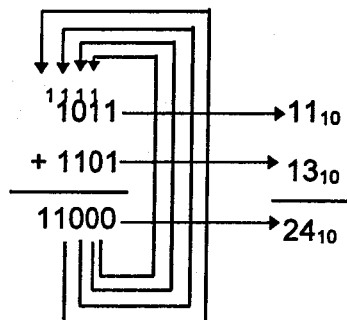
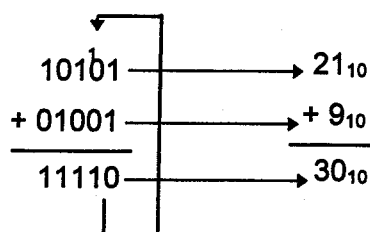
$$1 + 1 + 1 = 11_2$$

Ex.: $1_2 + 1_2 = 2_{10}$ ou

$$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$$

Obs.: Soma maior ou igual ao valor da base ocorre um vai um.

Ex.:



Subtração:

Regras:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ e empresta um - "Borrow"}$$

Exemplos:

$$\begin{array}{r} \overset{\text{Borrow}}{\uparrow} \uparrow \\ 100 \longrightarrow 4 \\ - 1 \longrightarrow -1 \\ \hline 011 \longrightarrow 3 \end{array}$$

$$\begin{array}{r} \overset{\text{Borrow}}{\uparrow} \uparrow \\ 1101 \longrightarrow 13 \\ - 111 \longrightarrow -7 \\ \hline 0110 \longrightarrow 6 \end{array}$$

2.6.3 Adição e Subtração no Sistema de Numeração Octal e Hexadecimal

A forma mais rápida e prática de efetuar uma operação aritmética em um número octal ou hexadecimal é transformá-lo em binário, efetuar a operação e depois reconvertê-lo para octal ou hexadecimal.

Exemplos:

$$\begin{array}{r} 147_8 \\ + 654_8 \\ \hline 1023_8 \end{array}$$
$$\begin{array}{r} 001100111_2 \\ + 110101100_2 \\ \hline 1000010011_2 \\ \hline \underbrace{1}_{1} \underbrace{000}_0 \underbrace{010}_2 \underbrace{011}_{3_8} \end{array}$$

Exemplos: Transformar os números octais para binário e verificar se o resultado da operação é correto:

$$\begin{array}{r} 147 \\ - 121 \\ \hline 26 \end{array} \quad \begin{array}{r} 100 \\ + 37 \\ \hline 41 \end{array} \quad \begin{array}{r} 6234 \\ - 2351 \\ \hline 3663 \end{array}$$

Exemplos: Transformar os números hexadecimais para binário e verificar se o resultado da operação está correto:

$$\begin{array}{r} F0FC \\ + A73 \\ \hline FB6F \end{array} \quad \begin{array}{r} 900 \\ + CA1 \\ \hline 15A1 \end{array} \quad \begin{array}{r} F731 \\ - 11 \\ \hline F720 \end{array} \quad \begin{array}{r} BEBE \\ + 62DEB \\ \hline 6ECA9 \end{array}$$

Uma relação entre as formas de numeração apresentadas pode ser visualizada na tabela a seguir :

Decimal	Binária	Octal	Hexadecimal
00	00000	00	00
01	00001	01	01
02	00010	02	02
03	00011	03	03
04	00100	04	04
05	00101	05	05
06	00110	06	06
07	00111	07	07
08	01000	10	08
09	01001	11	09
10	01010	12	0A

11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

3. Circuitos aritméticos

3.1 Introdução

Já sabemos que os circuitos digitais funcionam utilizando como base os números binários. microprocessadores e microcomputadores todas as operações aritméticas são realizadas através somas e subtrações destes números binários.

Cabe então perguntar como serão os circuitos digitais que realizarão as operações de soma e subtração?

Este capítulo responde esta questão. Primeiro veremos como representar números positivos e negativos através de seqüências de bits, em seguida veremos os circuitos que permitem somar e subtrair estes números.

3.2 Representação de números em seqüências de bits.

Até agora para representar números positivos ou negativos em binário temos utilizado os sinais + e -, isto é:

se escrevermos + 2 em uma seqüência de 4 bits teremos +0010 (ou 0010).

se escrevermos -2 em uma seqüência de 4 bits teremos -0010.

Porém, nos equipamentos digitais (computadores, microprocessadores) todas as informações são representadas somente por variações de seqüências de zeros e uns (números binários). Temos então que utilizar uma forma codificada de representar os sinais "+" e "-" em zero e um.

3.2.1 Representação Sinal Magnitude

Nesta representação o bit mais significativo define se o número é positivo ou negativo e demais bits definem a magnitude do número.

Exemplo:

2	0010
-2	1010
4	0100
-4	1100

Esta representação limita a representação da magnitude do número ao número total de bits menos 1. Veja o exemplo abaixo:

Para uma seqüência de quatro bits os números possíveis de serem representados serão:

+0	0000
+1	0001
+2	0010
+3	0011
+4	0100

-0	1000
-1	1001
-2	1010
-3	1011
-4	1100

+5	0101
+6	0110
+7	0111
+8	1000 (erro pois o bit mais significativo representa o sinal, não podemos representar o número 8 em quatro bits)

-5	1101
-6	1110
-7	1111
-8	1000 (erro pois o bit mais significativo representa o sinal, não podemos representar o número -8 em quatro bits)

Exercício: Utilizando a representação sinal magnitude faça as operações abaixo. Verifique também se resultado da operação pode ser representado em sinal – magnitude, numa sequência de bits.

- a) $00011 + 00100 =$
- b) $01100 + 10010 =$
- c) $01101 - 10111 =$
- d) $01101 + 01000 =$
- e) $01101 - 11111 =$

3.2.2 Representação em Complemento de Dois.

Outra forma de representar números positivos e negativos em seqüências binárias é complemento de Dois. Nesta representação o número positivo continua sendo representado da mesma forma que o número positivo sinal-magnitude (bit mais significativo igual a zero, indicando que número é positivo).

Porém, o número negativo é representado pelo complemento de dois do respectivo número positivo.

O complemento de dois de um número binário é obtido da seguinte forma:

Exemplo: para o número -7

- complementa-se o número positivo respectivo bit a bit:

número positivo respectivo 7, em binário

0111

complemento do número acima

1000 (chamado complemento de 1)

- soma-se mais 1 ao complemento obtido no item anterior

$$\begin{array}{r} 1000 \\ + 0001 \\ \hline 1001 \end{array}$$
 (número -7 em complemento de 2)

Exemplos:

$-3 > 0011 > 1100 > 1101$
 $-6 > 0110 > 1001 > 1010$

3.2.3 Regra prática para obter o complemento de dois

A partir do número positivo, da direita pra a esquerda manter os zeros (se existirem) encontrar o primeiro um. Mantenha este um e complemente os demais bits (se existirem).

3.2.4 Soma e subtração com complemento de dois

A grande vantagem do uso do complemento de dois para representação de números negativos é a facilidade de somar e subtrair números com este formato. observe o exemplo abaixo:

Soma e subtração de dois números em sinal-magnitude (seqüência de quatro bits)	Soma e subtração de dois números em complemento de dois (seqüência de quatro bits)
a) $2 + 2$	b) $2 + 2$
$0010 + 0010 = 0100$	$0010 + 0010 = 0100$
b) $4 - 3$	b) $4 - 3$
$0100 - 1011 = 0001$ (temos que desconsiderar o bit mais significativo que representa somente o sinal)	$0100 + 1101 = 0001$ (o resultado é obtido pela soma direta dos dois números binários)

Devido a esta facilidade a representação complemento de dois passou a ser utilizada por microprocessadores e microcomputadores.

3.2.5 – Over Flow

Over Flow é a mudança no sinal do resultado devido a realização de operações com números que levam ao estouro da capacidade do registrador (seqüência de bits). Esta situação ocorre quando realiza-se operações equivalentes de soma de dois números positivos ou de dois números negativos

Exemplos: Utilizando um registrador de 4 bits, considerando representação em complemento de dois

a) $3+2$	c) $-3 -2$
$\begin{array}{r} 0011 \\ + 0010 \\ \hline 0101 \end{array}$ (5 resultado correto)	$\begin{array}{r} 1101 \\ + 1110 \\ \hline 1011 \end{array}$ (-5, resultado correto)
b) $5+4$	d) $-5 -4$
$\begin{array}{r} 0101 \\ + 0100 \\ \hline 1001 \end{array}$ (resultado errado. Número um no bit mais significativo indica número negativo, portanto pela representação de complemento de dois o resultado obtido foi -7.)	$\begin{array}{r} 1011 \\ + 1100 \\ \hline 0111 \end{array}$ (resultado errado. Número zero no bit mais significativo indica número positivo portanto o resultado obtido foi +7)

Exercícios:

1) Realize as operações abaixo utilizando representação de complemento de dois e registradores de bits. Para cada uma verifique se não ocorreu overflow.

- a) $10 + 3$
- b) $10 - 20$
- c) $7 + 8$
- d) $-3 - 8$
- e) $-9 + 30$
- f) $2 + 15$
- g) $5 - 22$

2. Funções Lógicas

2.1 Introdução

No início da era eletrônica, todos os problemas eram resolvidos por sistemas analógicos, também conhecidos por sistemas lineares, onde uma quantidade é representada por um sinal elétrico proporcional ao valor da grandeza medida. As quantidades analógicas podem variar em uma faixa contínua de valores.

Com o avanço da tecnologia, esse mesmos problemas começaram a ser solucionados através da eletrônica digital, onde uma quantidade é representada por um arranjo de símbolos chamados dígitos. Este ramo da eletrônica emprega em suas máquinas, tais como: computadores, calculadoras, sistemas de controle e automação, codificadores, decodificadores, etc., apenas um pequeno grupo de circuitos lógicos (funções lógicas) básicos, que são conhecidos como portas OU, E, NÃO e flip-flops.

Como já visto, um circuito digital emprega um conjunto de funções lógicas, onde função é a relação existente entre as variável independente e a variável dependente (função). Para cada valor possível da variável independente determina-se o valor da função.

O conjunto de valores que uma variável pode assumir depende das restrições ou especificações do problema a ser resolvido. Esta variável é, normalmente, conhecida como variável independente.

Para o momento, nosso interesse está no comportamento de um sistema lógico (descrito por George Boole em meados do século passado). Nestes sistemas as variáveis independentes são conhecidas como variáveis lógicas e as funções, como funções lógicas (variável lógica dependente).

As variáveis lógicas (dependentes ou independentes) possuem as seguintes características:

- Pode assumir somente um de dois valores possíveis;
- Os seus valores são expressos por afirmações declarativas (cada valor está associado a um significado);
- Os dois valores possíveis das variáveis são mutuamente exclusivos.

Uma variável lógica A pode assumir um valor verdadeiro ($A=V$) ou o valor falso ($A=F$).

Em geral, usa-se uma faixa de tensão para representar o valor falso ou verdadeiro de uma variável lógica.

Lógica Positiva - a tensão mais positiva representa o valor V (1) e a mais negativa o valor F (0).

Lógica Negativa - o valor V é representado pela tensão mais negativa (1) e F pela tensão mais positiva (0).

Lógica Mista - no mesmo sistema, usam-se as lógicas positiva e negativa.

2.2 Funções Lógicas Básicas

O passo seguinte na evolução dos sistemas digitais foi a implementação dos sistemas lógicos (funções lógicas Booleanas), utilizando-se dispositivos eletrônicos (circuitos digitais), obtendo-se assim, rapidez na solução dos problemas (descritos pela álgebra de Boole). Nos circuitos digitais tem-se somente dois níveis de tensão, que apresentam correspondência com os possíveis valores das variáveis lógicas. Exemplo: lógica TTL ("Transistor Transistor Logic") positiva - $0V \rightarrow 0$ lógico - $5V \rightarrow 1$ lógico.

Um sistema lógico pode ser implementado utilizando-se funções lógicas básicas. Pode-se citar: NÃO (NOT), E (AND), OU (OR), NÃO-E (NAND), NÃO-OU (NOR), OU EXCLUSIVO (XOR) e flip-flop.

2.2.1 Função Lógica NÃO (NOT)

É normalmente denominado de inversor, pois se a entrada tem um valor a saída apresentará o outro valor possível.

Símbolo:



$$Y = f(A) = \bar{A}$$

Tabela da Verdade:

É uma tabela que mostra todas as possíveis combinações de entrada e saída de um circuito lógico.

$$Y = \bar{A}$$

Entrada $\rightarrow$ Saída

A	Y
0	1
1	0

2.2.2 Função Lógica E (AND)

A função lógica "AND" de duas entradas realiza a seguinte operação de dependência.

$$Y = f(A, B) = A \cdot B = B \cdot A \quad (\text{produto l\u00f3gico})$$

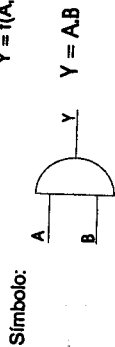
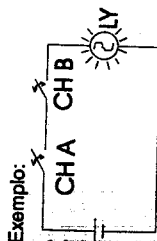


Tabela da Verdade

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Conven\u00e7\u00e3o:
 CH A aberta = 0
 CH A fechada = 1
 CH B aberta = 0
 CH B fechada = 1
 L\u00e2mpada apagada = 0
 L\u00e2mpada acesa = 1



Se analisarmos todas as situa\u00e7\u00f5es poss\u00edveis das chaves verifica-se que a l\u00e2mpada acende somente quando as chaves A e B estiverem fechadas (assume 1 somente quando todas as entradas forem 1).

2.2.2.1 Fun\u00e7\u00e3o l\u00f3gica AND com mais de duas vari\u00e1veis de entrada.

$$Y = A \cdot B \cdot C = B \cdot A \cdot C = C \cdot A \cdot B = (A \cdot B) \cdot C = A \cdot (B \cdot C)$$

Comutatividade Associatividade

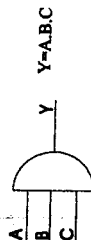
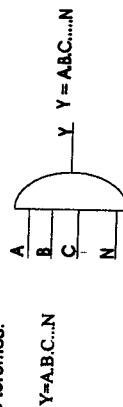


Tabela da Verdade
 (3 var. $\rightarrow 2^3$ combina\u00e7\u00f5es)

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Se tivermos N entradas teremos:



$$Y = A \cdot B \cdot C \cdot \dots \cdot N$$

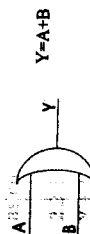
A tabela da verdade ter\u00e1 2^N combina\u00e7\u00f5es na entrada e Y ser\u00e1 1 somente quando todas as entradas forem 1.

2.2.3 Fun\u00e7\u00e3o L\u00f3gica OU (OR)

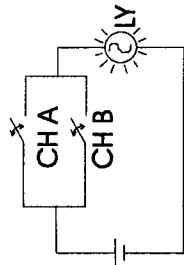
A fun\u00e7\u00e3o l\u00f3gica OR de duas vari\u00e1veis realiza a seguinte opera\u00e7\u00e3o de depend\u00eancia:
 $Y = f(A, B) = A + B$ (soma l\u00f3gica)

Tabela da Verdade

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



Exemplo:



Utiliza-se as mesmas conven\u00e7\u00f5es adotadas para a porta AND. Ao analisar-se todas as situa\u00e7\u00f5es que as chaves podem assumir verifica-se que a l\u00e2mpada acende quando CH A OU CH B OU ambas estiverem ligadas (a sa\u00edda assume 0 somente quando todas as entradas forem 0).

2.2.3.1 Fun\u00e7\u00e3o l\u00f3gica OR de mais de duas vari\u00e1veis de entrada

$$Y = A + B + C = C + B + A = B + C + A = A + (B + C) = (A + B) + C$$

Comutatividade

Associatividade

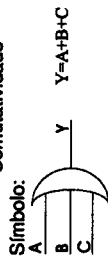
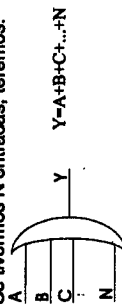


Tabela da Verdade

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Se tivermos N entradas, teremos:



$$Y = A + B + C + \dots + N$$

2.2.4 Fun\u00e7\u00e3o L\u00f3gica N\u00c3O E (NAND)

Como o pr\u00f3prio nome diz esta fun\u00e7\u00e3o \u00e9 uma combina\u00e7\u00e3o das fun\u00e7\u00f5es AND e INVERSOR, onde \u00e9 realizada a fun\u00e7\u00e3o E invertida.

$$Y = f(A, B) = \overline{A \cdot B}$$

S\u00edmbolo:



Tabela da Verdade

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Tabela da Verdade

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

2.2.5 Fun\u00e7\u00e3o L\u00f3gica N\u00c3O OU (NOR)

Como o pr\u00f3prio nome diz esta fun\u00e7\u00e3o \u00e9 uma combina\u00e7\u00e3o das fun\u00e7\u00f5es OR e INVERSOR, onde \u00e9 realizada a fun\u00e7\u00e3o OU invertida.

$$Y = f(A, B) = \overline{A + B}$$

Símbolo:



Exercícios:

1. Representar portas NOR e NAND com mais de duas entradas (símbolo, função e tabela da verdade).
2. Pesquisar sobre a porta OU-EXCLUSIVO.
3. Pesquisar sobre a porta COINCIDÊNCIA.

BLOCOS LÓGICOS BÁSICOS

PORTA	Símbolo Usual	Tabela da Verdade	Função Lógica	Expressão															
E AND		<table> <tr><th>A</th><th>B</th><th>S</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1	Função E: Assume 1 quando todas as variáveis forem 1 e 0 nos outros casos.	$S = A \cdot B$
A	B	S																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OU OR		<table> <tr><th>A</th><th>B</th><th>S</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1	Função E: Assume 0 quando todas as variáveis forem 0 e 1 nos outros casos.	$S = A + B$
A	B	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NÃO NOT		<table> <tr><th>A</th><th>S</th></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	S	0	1	1	0	Função NÃO: Inverte a variável aplicada à sua entrada.	$S = \bar{A}$									
A	S																		
0	1																		
1	0																		
NE NAND		<table> <tr><th>A</th><th>B</th><th>S</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0	Função NE: Inverso da função E.	$S = \overline{(A \cdot B)}$
A	B	S																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
NOU NOR		<table> <tr><th>A</th><th>B</th><th>S</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0	Função NOR: Inverso da função OU.	$S = \overline{(A + B)}$
A	B	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
OU Exclusivo		<table> <tr><th>A</th><th>B</th><th>S</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0	Função OU Exclusivo: Assume 1 quando as variáveis assumirem valores diferentes entre si.	$S = A \oplus B$ $S = \bar{A} \cdot B + A \cdot \bar{B}$
A	B	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
Coincidência		<table> <tr><th>A</th><th>B</th><th>S</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	1	Função Coincidência: Assume 1 quando houver coincidência entre os valores das variáveis.	$S = \bar{A} \cdot \bar{B} + A \cdot B$
A	B	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

2.3 Interligação entre Expressões, Circuitos e Tabela da Verdade

Todo circuito lógico, por mais complexo que seja, é formado pela combinação de portas lógicas básicas.

2.3.1 Expressões Booleanas Obtidas de Circuitos Lógicos

Todo o circuito lógico executa uma função booleana e, por mais complexo que seja, é formado pela interligação das portas lógicas básicas. Assim, pode-se obter a expressão booleana que é executada por um circuito lógico qualquer.

Para exemplificar, será obtida a expressão que o circuito da Fig 2.17 abaixo executa.

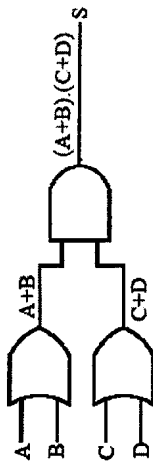


Figura 2.17 – Circuito lógico.

Para facilitar, analisa-se cada porta lógica separadamente, observando a expressão booleana que cada uma realiza, conforme ilustra o exemplo da Fig. 2.17.

O exemplo da Fig. 2.18 visa evidenciar um símbolo de negação muito utilizado e que muitas vezes é esquecido e não considerado. Ele pode ser utilizado na saída de uma porta lógica (O---), como na porta NAO E abaixo, e na entrada de algumas portas, como será visto mais adiante (-----O).

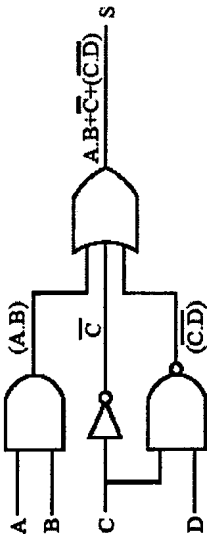


Figura 2.18 – Circuito lógico.

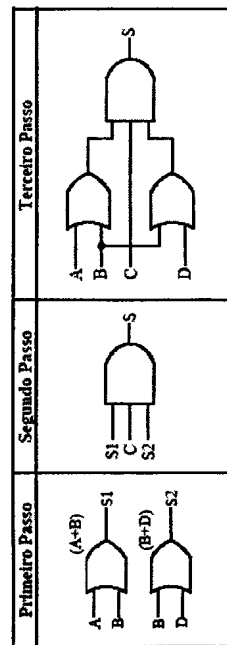
2.3.2 Circuitos Lógicos Obtidos de Expressões Booleanas

Será visto neste tópico que é possível desenhar um circuito lógico que executa uma função booleana qualquer, ou seja, pode-se desenhar um circuito a partir de sua expressão característica.

O método para a resolução consiste em se identificar as portas lógicas na expressão e desenhá-las com as respectivas ligações, a partir das variáveis de entrada. Deve-se sempre respeitar a hierarquia das funções da aritmética elementar, ou seja, a solução inicia-se primeiramente pelos parênteses.

Para exemplificar, será obtido o circuito que executa a expressão $S = (A + B) \cdot C \cdot (B + D)$.

Para o primeiro parêntese tem-se uma soma booleana $A + B$, logo o circuito que o executa será uma porta OU. Para o segundo, tem-se outra soma booleana $B + D$, logo o circuito será uma porta OU. Posteriormente tem-se a multiplicação booleana de dois parênteses juntamente com a variável C, sendo o circuito que executa esta multiplicação uma porta E. Para finalizar, unem-se as respectivas ligações obtendo o circuito completo.



Exercício. Esboce os circuitos obtidos a partir das seguintes expressões:

1. $S = (A.B + C.D)$
2. $S = (A + \overline{B} + C).(\overline{A} + C + \overline{D})$
3. $S = (A + \overline{B}).C.(A + C).B$
4. $S = ((A + \overline{B}).C) + (\overline{B}.D).(\overline{A} + (B.D)))$

2.3.3 Tabelas da Verdade obtidas de Expressões Booleanas

Uma maneira de se fazer o estudo de uma função booleana é a utilização da tabela da verdade. Para extrair a tabela da verdade de uma expressão deve-se seguir alguns procedimentos:

- 1ª) Montar o quadro de possibilidades;
- 2ª) Montar colunas para os vários membros da equação;
- 3ª) Preencher estas colunas com os seus resultados;
- 4ª) Montar uma coluna para o resultado final e
- 5ª) Preencher esta coluna com os resultados finais.

Para exemplificar este processo, utiliza-se a expressão:

$$S = A.B.C + A.\overline{D} + \overline{A}.B.D$$

A expressão contém 4 variáveis: A, B, C e D, logo, existem 24=16 possibilidades de combinação de entrada. Desta forma, monta-se o quadro de possibilidades com 4 variáveis de entrada, três colunas auxiliares, sendo uma para cada membro da expressão, e uma coluna para o resultado final.

Variáveis de entrada				1º membro	2º membro	3º membro	Resultado final
A	B	C	D	$A.B.C$	$A.\overline{D}$	$\overline{A}.B.D$	
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	0	1
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	1
1	1	0	0	0	0	1	0
1	1	0	1	0	0	1	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1

2.3.4 Expressões Booleanas Obtidas de Tabelas da Verdade

Neste item, será estudada a forma de obter expressões e circuitos a partir de tabelas da verdade, sendo este o caso mais comum de projetos práticos, pois, geralmente, necessita-se representar situações através de tabelas da verdade e a partir destas, obter a expressão booleana e consequentemente, o circuito lógico. Para demonstrar este procedimento, será obtida a expressão da seguinte tabela:

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

Na tabela, analisa-se onde S=1 e monta-se a expressão adequada.

- Em (a), S=1 se $S = \overline{A}.B.C$
- Em (b), S=1 se $S = A.\overline{B}.C$
- Em (c), S=1 se $S = A.B.\overline{C}$
- Em (d), S=1 se $S = A.B.C$

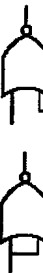
Para se obter a expressão basta realizar a soma booleana de cada termo acima:

$$S = \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

Nota-se que o método permite obter, de qualquer tabela, uma expressão padrão formada sempre pela soma de produtos. No próximo capítulo, relativo a álgebra de Boole, será estudado o processo de simplificação de expressões booleanas, possibilitando a obtenção de circuitos reduzidos.

2.3.5 Equivalência Entre Blocos Lógicos

As portas lógicas podem ser montadas de forma que possam realizar as mesmas tarefas, ou seja, ter as saídas funcionando de maneira igual a uma outra já conhecida. Estas equivalências são muito importantes na prática, ou seja, na montagem de sistemas digitais, pois possibilitam maior otimização na utilização dos circuitos integrados comerciais, assegurando principalmente a redução de componentes e a consequente minimização do custo do sistema.

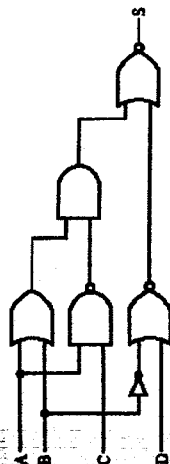
BLOCO LÓGICO	BLOCO EQUIVALENTE
	 
	 $S = \overline{A+B}$
	 $S = \overline{A.B}$
	 $S = \overline{A}.B$
	 $S = \overline{A+B}$

Todos os Blocos lógicos e expressões podem ser verificadas utilizando-se a tabela da verdade.

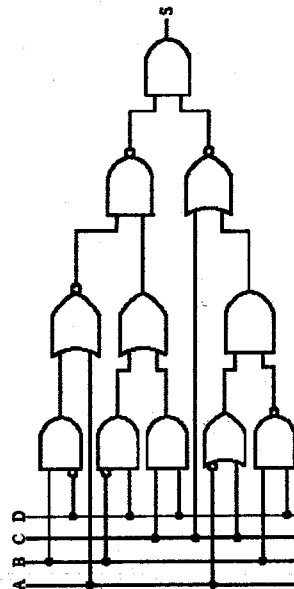
2.3.6 Exercícios do Capítulo II

Determine as expressões dos circuitos abaixo:

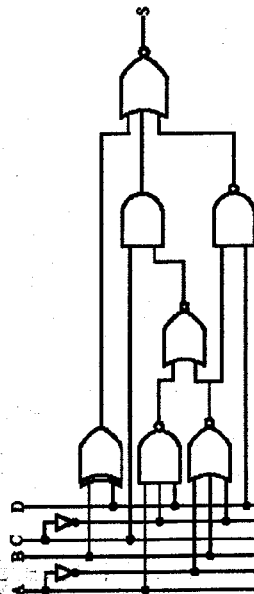
a) Circuito 1



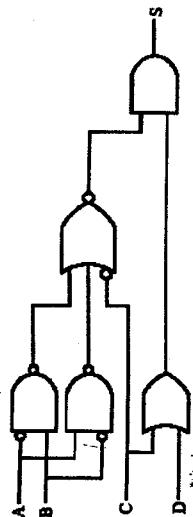
b) Circuito 2



c) Circuito 3



d) Circuito 4



2.14.2) Desenhe o circuito que executa as seguintes expressões:

a) $S = \overline{[(A+B) + (C+D)]} \cdot \overline{D}$

b) $S = \overline{A} \cdot [\overline{B} \cdot C + A \cdot (C+D)] + B \cdot \overline{C} \cdot D + B \cdot \overline{D}$

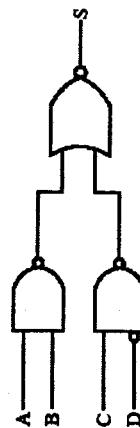
c) $S = (A \odot B) \cdot [A \cdot \overline{B} + (B+D) + C \cdot \overline{D} + \overline{(B \cdot C)}] + \overline{A} \cdot B \cdot \overline{C} \cdot D$

2.14.3) Levante a tabela verdade das seguintes expressões:

a) $S = \overline{C} \cdot [A \cdot \overline{B} + B \cdot (A+C)]$

b) $S = (B \odot D) \cdot [\overline{A+B} \cdot (C+D) + A \cdot \overline{B} \cdot \overline{C}]$

2.14.4) Escreva a expressão característica do circuito abaixo e levante sua respectiva tabela verdade.



2.14.5) Determine a expressão booleana a partir das seguintes tabelas:

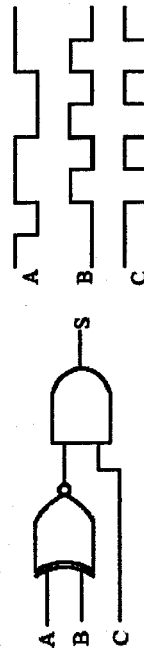
a) Tabela 1

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

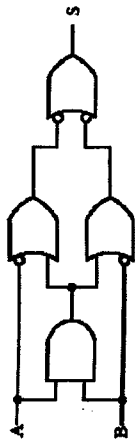
b) Tabela 2

A	B	C	D	S
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

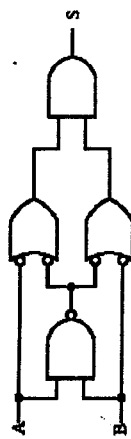
2.14.6) Desenhe o sinal de saída do circuito abaixo:



2.14.7) Mostre que o circuito abaixo é um OU Exclusivo.



2.14.8) Mostre que o circuito é um circuito Celadência.



2.14.9) Prove que:

$$A \oplus (B \oplus C) = A \oplus (B \oplus C)$$

2.14.10) Levante a tabela da verdade e esquematize o circuito que executa a seguinte expressão:

$$S = \{[A \cdot B + C] \oplus [A + B]\} \oplus C$$

2.14.11) Esquematize o circuito Celadência usando apenas porta NOU.

2.14.12) Esquematize o circuito OU Exclusivo, utilizando somente 4 portas NE.

2.14.13) Esquematize o circuito Celadência, utilizando apenas 4 portas NOU.

2.14.14) Desenhe o circuito que executa a expressão do exercício 2.14.2 letra b, usando somente portas NE.

2.14.15) Desenhe o circuito que executa a expressão do exercício 2.14.2 letra c, usando somente portas NOU.

2.14.16) Levante a tabela da verdade e, a partir desta, desenhe o circuito somente com portas NE.

$$S = (B \oplus C) \cdot [D + A \cdot C + D \cdot (A + B + C)]$$

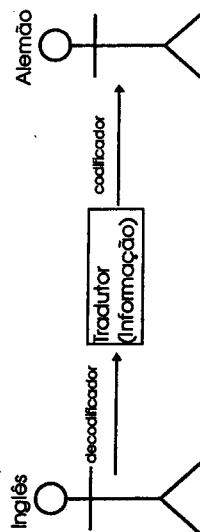
2.14.17) Desenhe novamente o circuito do exercício 2.14.1, circuito 3, utilizando apenas portas NOU.



Circuitos Combinacionais

1.1. Decodificadores e Codificadores

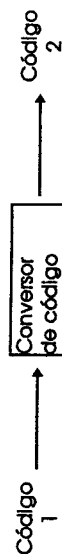
Estes termos diferenciam-se por uma questão de referência. O termo decodificador significa obter informação a partir de um código. O termo codificador significa gerar um código a partir de uma informação. O exemplo a seguir ilustra a diferença.



O tradutor fez o papel de um decodificador para a pessoa que entende alemão, logo esse elemento irá encará-lo como um decodificador, pois, está passando uma mensagem qualquer de um código desconhecido (o inglês) para um código conhecido (o alemão).

Porém para a pessoa do idioma inglês o tradutor faz o papel de um codificador, pois está transformando uma linguagem conhecida sua para uma outra.

Estas mesmas observações podem ser feitas quando deseja-se transformar as informações representadas em um determinado código para outro código. Por exemplo o teclado da calculadora converte um número em código decimal para uma número em código binário.



Conversor de código = decodificador + codificador.

1.2. Códigos Binários

São as diversas formas padronizadas empregadas para representar informações em binário. Dependendo de cada situação um código apresenta vantagens em relação aos

outros. Existem códigos muito difundidos para representar caracteres alfanuméricos tais como ASCII e EBCDIC. Neste capítulo serão abordados somente os códigos mais utilizados.

1.3.Código BCD 8421

A sigla BCD representa as iniciais de "Binary-Coded Decimal", que significa decimal codificado em binário. Os dígitos 8421 representam o valor em decimal do correspondente dígito em binário ($2^3 = 8$, $2^2 = 4$, $2^1 = 2$, $2^0 = 1$). A tabela a seguir mostra o código.

Decimal	BCD 8421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

1.4.Código ASCII

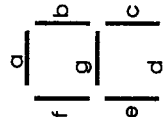
Os caracteres do PC, e nos computadores mais modernos, ocupam um byte de 8 bits, de forma que pode haver 2^8 , ou seja, 256 caracteres diferentes. A figura abaixo mostra cada um destes caracteres, e os seus códigos numéricos em decimal e hexadecimal.

0	00	32 20	64 40 @	96 60	128 80 Ç	160 A0 á	192 C0	224 E0 □
1	01 □	33 21	65 41 A	97 61 a	129 81 ü	161 A1 í	193 C1	225 E1 □
2	02 ■	34 22 □	66 42 B	98 62 b	130 82 é	162 A2 ó	194 C2	226 E2 □
3	03 □	35 23	67 43 C	99 63 c	131 83 â	163 A3 ú	195 C3	227 E3 □
4	04 □	36 24 \$	68 44 D	100 64 d	132 84 ä	164 A4 ñ	196 C4	228 E4 □
5	05 □	37 25	69 45 E	101 65 e	133 85 à	165 A5 Ñ	197 C5	229 E5 □
6	06 □	38 26	70 46 F	102 66 f	134 86 å	166 A6 ¢	198 C6	230 E6 □
7	07 □	39 27	71 47 G	103 67 g	135 87 ç	167 A7 °	199 C7	231 E7 □
8	08 ■	40 28	72 48 H	104 68 h	136 88 é	168 A8 ¿	200 C8	232 E8 □
9	09 □	41 29	73 49 I	105 69 i	137 89 è	169 A9	201 C9	233 E9 □
10	0A	42 2A	74 4A J	106 6A j	138 8A è	170 AA ~	202 CA	234 EA □
11	0B ¢	43 2B	75 4B K	107 6B k	139 8B ï	171 AB ½	203 CB	235 EB □
12	0C ¢	44 2C	76 4C L	108 6C l	140 8C î	172 AC ¼	204 CC	236 EC □
13	0D ¢	45 2D -	77 4D M	109 6D m	141 8D ï	173 AD	205 CD	237 ED □

14	OE	8	46	2E	78	4E	N	110	6E	n	142	8E	Ä	174	AE	206	CE	238	EE	□	
15	OF	□	47	2F	/	79	4F	O	111	6F	o	143	8F	Å	175	AF	207	CF	239	EF	□
16	10	▲	48	30	0	80	50	P	112	70	p	144	90	É	176	B0	208	D0	240	F0	□
17	11	◄	49	31	1	81	51	Q	113	71	q	145	91	æ	177	B1	209	D1	241	F1	□
18	12	□	50	32	2	82	52	R	114	72	r	146	92	Æ	178	B2	210	D2	242	F2	□
19	13	□	51	33	3	83	53	S	115	73	s	147	93	ß	179	B3	211	D3	243	F3	□
20	14	◄	52	34	4	84	54	T	116	74	t	148	94	ð	180	B4	212	D4	244	F4	□
21	15	§	53	35	5	85	55	U	117	75	u	149	95	ð	181	B5	213	D5	245	F5	□
22	16	□	54	36	6	86	56	V	118	76	v	150	96	û	182	B6	214	D6	246	F6	□
23	17	□	55	37	7	87	57	W	119	77	w	151	97	ü	183	B7	215	D7	247	F7	□
24	18	□	56	38	8	88	58	X	120	78	x	152	98	ÿ	184	B8	216	D8	248	F8	□
25	19	□	57	39	9	89	59	Y	121	79	y	153	99	ÿ	185	B9	217	D9	249	F9	□
26	1A	□	58	3A	:	90	5A	Z	122	7A	z	154	9A	ÿ	186	BA	218	DA	250	FA	□
27	1B	□	59	3B	:	91	5B	[	123	7B	{	155	9B	ç	187	BB	219	DB	251	FB	□
28	1C	□	60	3C	<	92	5C	\	124	7C		156	9C	£	188	BC	220	DC	252	FC	□
29	1D	□	61	3D	=	93	5D	]	125	7D	}	157	9D	¥	189	BD	221	DD	253	FD	□
30	1E	▲	62	3E	>	94	5E	^	126	7E	~	158	9E	□	190	BE	222	DE	254	FE	□
31	1F	▼	63	3F	?	95	5F	_	127	7F	□	159	9F	f	191	BF	223	DF	255	FF	□

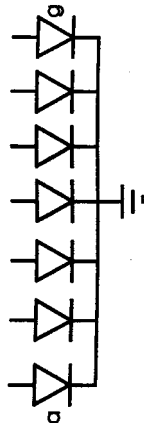
1.5.Conversor BCD 8421 para "Display" de 7 Segmentos

Este conversor é comumente chamado decodificador BCD para 7 segmentos. O display de 7 segmentos permite à escrita de números de 0 a 9 e alguns símbolos que podem ser letras ou sinais. A seguir mostra-se uma unidade de display com a identificação dos segmentos.



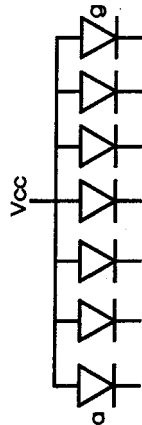
Será suposto que para acender qualquer dos segmentos é necessário ativar o referido segmento. Assim tem-se dois tipos de display:

1.6.Com cátodo comum - lógica positiva ou ativo alto.



1.7.Com ânodo comum:

Se para acender qualquer um dos segmentos é necessário aplicar nível lógico 0 no referido segmento, tem-se então os displays com ânodo comum - lógica negativa ou ativo baixo.



1.8.Exemplo de Conversor

Um conversor de código que passe de BCD 8421 para display de 7 segmentos é mostrado na tabela a seguir.

BCD 8421 ABCD	Decimal	Código para 7 segmentos a b c d e f g						
0000	0	1	1	1	1	1	0	0
0001	1	0	1	1	0	0	0	0
0010	2	1	1	0	1	1	0	1
0011	3	1	1	1	1	0	0	1
0100	4	0	1	1	0	0	1	1
0101	5	1	0	1	1	0	1	1
0110	6	0	0	1	1	1	1	1
0111	7	1	1	1	0	0	0	0
1000	8	1	1	1	1	1	1	1
1001	9	1	1	1	1	0	1	1
	N.T.	X	X	X	X	X	X	X
	N.T.	X	X	X	X	X	X	X
	N.T.	X	X	X	X	X	X	X
	N.T.	X	X	X	X	X	X	X
	N.T.	X	X	X	X	X	X	X
	N.T.	X	X	X	X	X	X	X

Obs.: Display com cátodo comum.

As expressões simplificadas são:

$$a = A + C + B.D + \overline{B}. \overline{D}$$

$$b = B + \overline{C}. \overline{D} + C.D$$

$$c = \overline{C} + B + D$$

$$d = A + \overline{B}. \overline{D} + C. \overline{B} + C. \overline{D} + B. \overline{C}. D$$

Como será o circuito?

$$e = \overline{B}. \overline{D} + C. \overline{D}$$

$$f = A + \overline{C}. \overline{D} + \overline{C}. B + B. \overline{D}$$

$$g = A + B. \overline{C} + \overline{B}. C + C. \overline{D}$$

1.9. Decodificador Binário (ou simplesmente decodificador)

É um decodificador que admite todas as possíveis combinações binárias da palavra de entrada. É comumente denominado simplesmente por decodificador. Para uma palavra com dois bits este decodificador apresenta a seguinte tabela da verdade:

A	B	S ₀	S ₁	S ₂	S ₃
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$S_0 = \overline{A} \cdot \overline{B}$$

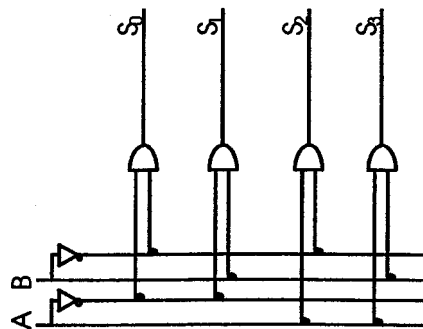
$$S_1 = \overline{A} \cdot B$$

$$S_2 = A \cdot \overline{B}$$

$$S_3 = A \cdot B$$

Da tabela acima obtemos as seguintes expressões:

O circuito então fica sendo:



O decodificador mostrado comporta-se de acordo com a lógica ativa alta ou positiva. O decodificador binário com lógica negativa é dado conforme a tabela a seguir:

A	B	S ₀	S ₁	S ₂	S ₃
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

Encontrar a expressão e o circuito para este decodificador.

Exemplo de circuitos utilizando portas lógicas:

- A) Uma campainha que toca (saída) se o motorista der a partida no motor do carro (entrada) sem estar com o cinto de segurança afivelado (entrada). Se a ignição for ACIONADA (1) e o cinto estiver DESAFIVELADO (1), a campainha é ACIONADA (1). Caso contrário, a campainha não toca

Tabela Verdade:

Ignição	Cinto	Campainha
0	0	0
0	1	0
1	0	0
1	1	1

Basta incluir uma porta AND.

- B) Detector de incêndio com vários sensores (entradas) e uma campainha para alarme (saída). Se QUALQUER UM dos sensores for acionado (significando que um dos sensores detectou sinal de incêndio), a campainha é ACIONADA.

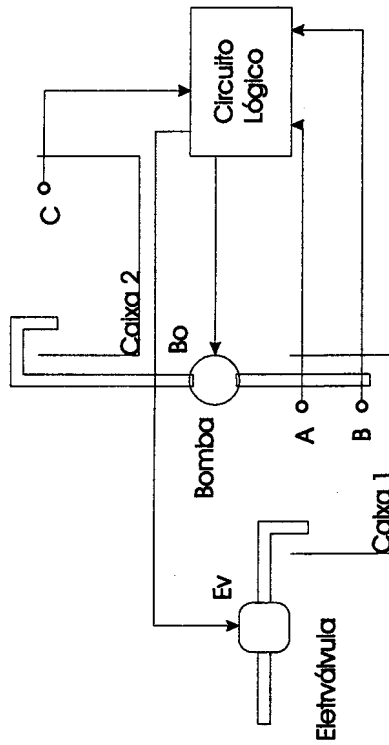
Tabela verdade:

Sensor 1	Sensor 2	Campainha
0	0	0
0	1	1
1	0	1
1	1	1

Basta incluir uma porta OR.

EXERCÍCIOS

1. Elaborar um circuito lógico que permita controlar uma bomba para encher uma caixa d'água no alto de um edifício a partir de outra, como reservatório, colocada no térreo, conforme mostra a figura. O circuito, através da informação de eletrodos (A, B, C), convenientemente dispostos nas caixas, deve atuar na bomba e numa eletroválvula ligada à canalização de entrada.



2. Uma indústria possui 4 máquinas de alta potência, podendo ser ligadas, no máximo, 2 delas simultaneamente. Elaborar um circuito lógico para efetuar este controle, respeitando a prioridade de funcionamento da máquina 1 sobre a 2, da 2 sobre a 3 e da 3 sobre a 4. Cada máquina possui um botão para ligá-la.

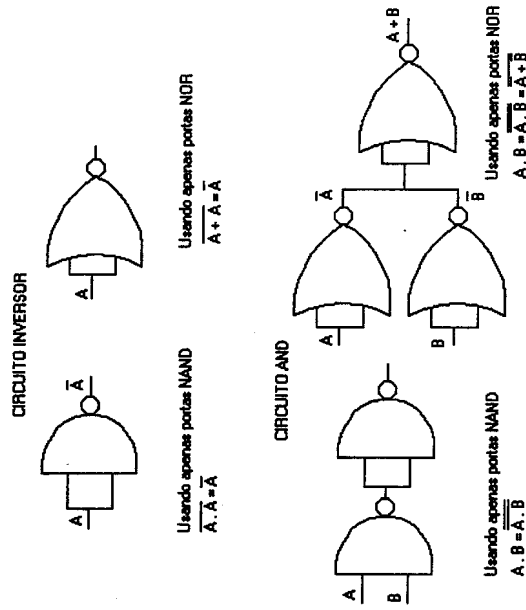
3. Elaborar um circuito lógico para resolver o seguintes problema: quatro juízes participam de um programa de calouros e cada um tem a sua disposição, uma chave "On/Off" (liga/desliga) correspondendo ao julgamento do candidato (On - aprovado, Off - reprovado). Na saída existem três lâmpadas, correspondentes a três resultados: aprovado (pela maioria), reprovado (pela maioria) ou empate.

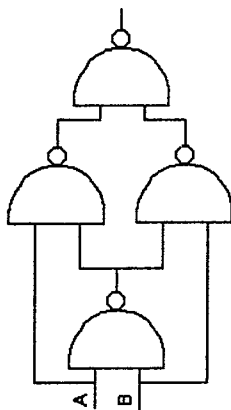
4. Um depósito pode armazenar quatro tipos de produtos químicos (A, B, C e D). Devido à natureza dos produtos torna-se perigoso armazenar num mesmo depósito os produtos B e C, a menos que o produto A esteja presente. O mesmo ocorre com os produtos C e D. Elaborar um circuito lógico que identifique a presença de uma combinação perigosa no depósito.

1.10. Representação De Circuitos Com As Funções Nand E Nor

Todo e qualquer circuito pode ser representado usando exclusivamente as função NAND ou as função NOR.

Para que serviria tal artimanha, além da dor de cabeça aos estudantes? Há neste caso uma razão maior que a comodidade ou a aparente dificuldade: a razão econômica. Por diversas razões construídas, fica mais barato construir TODOS os circuitos de um equipamento digital usando APENAS UM ÚNICO TIPO DE CIRCUITO. Sabe também que os CI's vem com número e tipo de portas determinados, sendo assim, como seria possível construir-se um porta arbitrário a partir das que o CI contém? Por exemplo, suponha que você dispõe apenas de um inversor, e lembrando sempre que isso é possível pela aplicação das Leis de De Morgan, bastaríamos fazer:





Escreva a expressão do circuito abaixo e simplifique.

$$\begin{aligned} C &= \overline{A \cdot \overline{B}} \cdot \overline{B \cdot \overline{A}} = \overline{A \cdot \overline{B} + B \cdot \overline{A}} = \overline{A \cdot \overline{B}} + \overline{B \cdot \overline{A}} = \\ &= \overline{A \cdot \overline{B}} \cdot (A + B) = (\overline{A \cdot \overline{B}}) \cdot (A + B) = \overline{A \cdot \overline{B}} \cdot A + \overline{A \cdot \overline{B}} \cdot B = \\ &= \overline{A} \cdot B + \overline{B} \cdot A = A \oplus B \end{aligned}$$

10

CAPÍTULO I - Flip-Flop

1.1- Definições

Os circuitos digitais podem ser classificados como circuitos combinacionais ou sequenciais. Entende-se por circuitos combinacionais àqueles cuja saída em um determinado instante de tempo depende somente das entradas naquele mesmo instante de tempo (circuitos sem memória). Por outro lado, entende-se por circuitos sequenciais àqueles cuja saída em um determinado instante de tempo não depende apenas das entradas naquele instante de tempo, mas também das entradas anteriores e da sequência como elas foram aplicadas. As figuras 1.a e 1.b ilustram em diagramas de blocos os circuitos combinacionais e sequenciais.

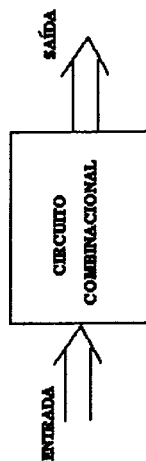


Fig. 1.a - Circuito Combinacional

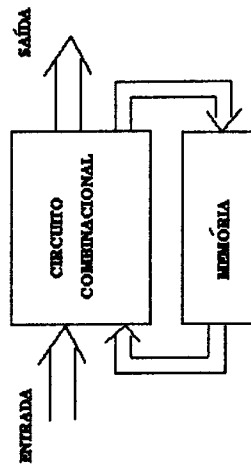


Fig. 1.b - Circuito Sequencial

Como os circuitos sequenciais também dependem das entradas anteriores torna-se evidente a necessidade de dispositivos de memória (atraso) nestes circuitos. Os circuitos de memória mais utilizados em circuitos sequenciais tratam-se dos Flip-Flops. O flip-flop (também denominado "LATCH") é um dispositivo biestável (dois estados estáveis 0,1) que permanece em um dos dois estados até a ocorrência de algum

evento que o possibillite assumir o outro estado estável. O fato do flip-flop manter uma informação ao longo do tempo o caracteriza como um dispositivo dotado de memória.

A figura 2 mostra a representação esquemática de um flip-flop. Convencionou-se que as saídas Q e $\bar{Q}$ de um flip-flop devem ser complementares.



Fig. 2 - Representação esquemática de um flip-flop

1.2 - Elemento Básico de Armazenamento

Uma unidade de armazenamento (flip-flop) básica pode ser obtida através do esquema mostrado a seguir.

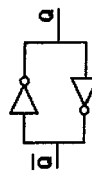


Fig. 3 - Unidade básica de armazenamento

Esta estrutura apresenta as seguintes características:

- circuito estável que permanece indefinidamente com o mesmo conteúdo (memória).
- as saídas Q e $\bar{Q}$ são complementares.
- para programá-lo é necessário conectá-lo temporariamente a uma fonte externa.

A grande desvantagem desta unidade de memória reside no fato de não apresentar flexibilidade de programação (manipulação).

1.3 - Flip-Flop RS Básico

São introduzidas ao "LATCH" apresentado em 1.2 entradas de controle adicionais que facilitem acessá-lo. A sua implementação pode ser feita tanto utilizando-se portas "nand" quanto portas "nor".

Inicialmente faz-se necessário alguns esclarecimentos, a saber:

-a letra S (set) refere-se a entrada utilizada para ativar o flip-flop ($Q=1$).

-a letra R (reset) refere-se a entrada utilizada para desativar o flip-flop ($Q=0$).

-o termo Q^n refere-se ao estado atual (presente) do flip-flop, que foi definido devido a ocorrência de eventos no instante de tempo n .

-o termo Q^{n+1} refere-se ao estado que o flip-flop assumirá (estado seguinte, estado futuro) caso ocorram determinados eventos no instante de tempo $n+1$. O diagrama de tempo mostrado a seguir esclarece o significado dos termos estados presente e futuro.

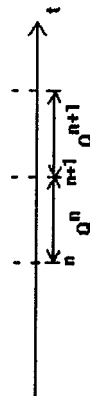


Fig. 4 - Diagrama de estados presente e futuro

1.3.1 - Flip-Flop RS Básico Implementado com Portas "NOR"

Um flip-flop RS implementado com portas "NOR" é mostrado a seguir:

TABELA VERDADE

S	R	Q^{n+1}	$\bar{Q}^{n+1}$
0	0	Q^n	$\bar{Q}^n$
0	1	0	1
1	0	1	0
1	1	0	0

(proibido)

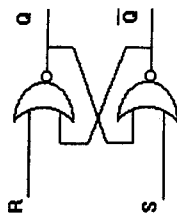


Fig. 5 - Flip-flop SR com portas NOR

A tabela verdade mostra que para $S=R=1$ a estrutura apresenta saídas incompatíveis com a definição de flip-flop, pois estas devem ser complementares.

1.3.2- Flip-Flop RS Básico Implementado com Portas "NAND"

O esquema para este dispositivo é mostrado a seguir.

TABELA VERDADE

S	R	Q^{n+1}	$\bar{Q}^{n+1}$
0	0	Q^n	$\bar{Q}^n$
0	1	0	1
1	0	1	0
1	1	1	1

(proibido)

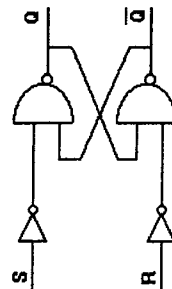


Fig. 6 - Flip-flop SR com portas NAND

A tabela verdade também mostra que esta estrutura apresenta saídas incompatíveis com a definição de flip-flop quando $S=R=1$.

1.4- Flip-Flop SR Sincronizado

Como já visto, as entradas S e R determinam o que será armazenado no flip-flop. Desta forma, pode-se dizer que as mesmas constituem-se em entradas de dados. Entretanto, em inúmeras situações práticas (sistemas sincronizados) é desejável que os dados, definidos pelas entradas S e R, sejam adquiridos somente em determinados intervalos. Para facilitar este processo é introduzido, conforme mostram as figuras 7.a e 7.b, um sinal de sincronismo ("clock") controlando a entrada de dados através de duas portas AND.

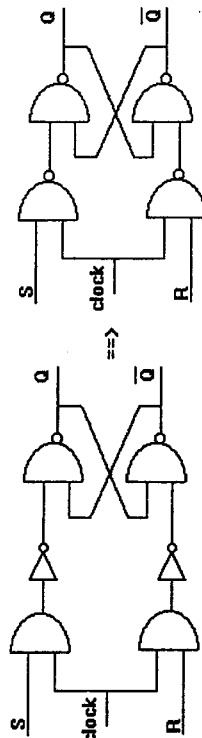


TABELA VERDADE

S	R	CLK	Q^{n+1}	$\bar{Q}^{n+1}$
X	X	0	Q^n	$\bar{Q}^n$
0	0	1	Q^n	$\bar{Q}^n$
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1

(proibido)

Fig. 7.a - Flip-flop SR sincronizado com portas NAND

TABELA VERDADE

S	R	CLK	Q ⁿ⁺¹	Q ⁿ	Q ⁿ⁺¹	Q ⁿ
X	X	0	0	1	0	1
0	0	1	1	0	1	0
0	1	1	1	0	1	0
1	0	1	1	0	1	0
1	1	1	0	0	0	0

(proibido)

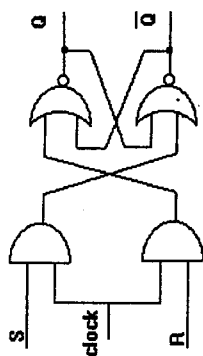


Fig. 7.b - Flip-flop SR Sincronizado com Portas NOR

O processo de aquisição e armazenamento descrito é ilustrado no diagrama de tempo mostrado na figura 8.

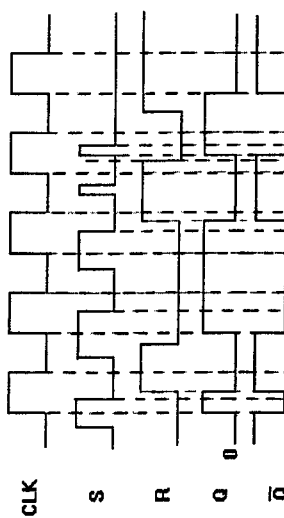


Fig. 8 - Aquisição e Armazenamento

1.5- Flip-Flop Tipo D Sincronizado

Em algumas situações as entradas (linhas) de dados S e R podem oferecer dificuldades adicionais. Por exemplo, para se armazenar os dados provenientes de algum dispositivo através de linhas é necessário estabelecer S=1 e R=0 para se obter Q=1, e S=0 e R=1 para se obter Q=0, ou seja manipular as duas entradas. Tal fato, pode ainda levar a ocorrência, por descuido, do estado proibido. Com o objetivo de facilitar o armazenamento de dados o flip-flop tipo D apresenta-se como uma possível solução,

pois o mesmo requer somente uma entrada de dados. O esquema da figura 9 mostra o flip-flop tipo D.

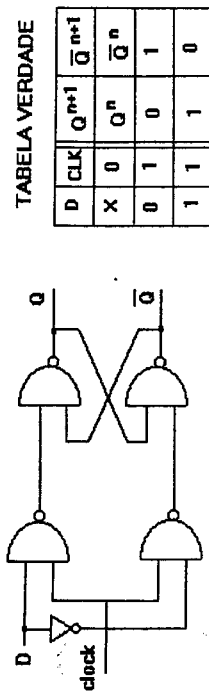


TABELA VERDADE

D	CLK	Q ⁿ⁺¹	Q ⁿ	Q ⁿ⁺¹	Q ⁿ
X	0	0	1	0	1
0	1	0	1	0	1
1	1	1	1	0	0

Fig. 9 - Flip-flop tipo D sincronizado pelo nível

1.6-Flip-Flop com Entradas Assíncronas

Entradas assíncronas são aquelas que quando ativadas atuam independentemente do sinal de clock, ou seja verifica-se o seu efeito imediatamente. Quando as entradas assíncronas estão desativadas permitem a operação normal do flip-flop. Estas entradas são normalmente utilizadas para se inicializar (reinicializar) o flip-flop. A figura 10 mostra o flip-flop SR sincronizado com entradas assíncronas PRESET (PR) e CLEAR (CLR).

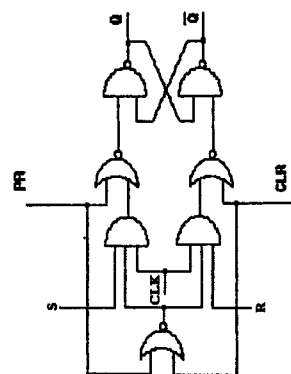


TABELA VERDADE

PR	CLR	S	R	CLK	Q ⁿ⁺¹	Q ⁿ
1	1	X	X	X	1	1
1	0	X	X	X	1	0
0	1	X	X	X	0	1
0	0	X	X	0	Q ⁿ	Q ⁿ
0	0	0	1	1	Q ⁿ	Q ⁿ
0	0	1	0	1	0	1
0	0	1	1	1	1	0
0	0	1	1	1	1	1

(proibido)

Fig 10 - Flip-flop SR sincronizado com entradas assíncronas

obs:

- a entrada PR é utilizada para ativar o flip-flop.
- a entrada CLR é utilizada para desativar o flip-flop.

1.7 - Aquisição de dados na borda do "clock"

A aquisição de dados durante a transição de clock é muitas vezes desejável, pois em um pequeno intervalo de tempo reduz-se a probabilidade da influência de entradas não desejadas e/ou também pela necessidade de sincronismo. A figura 13 ilustra uma possível solução.

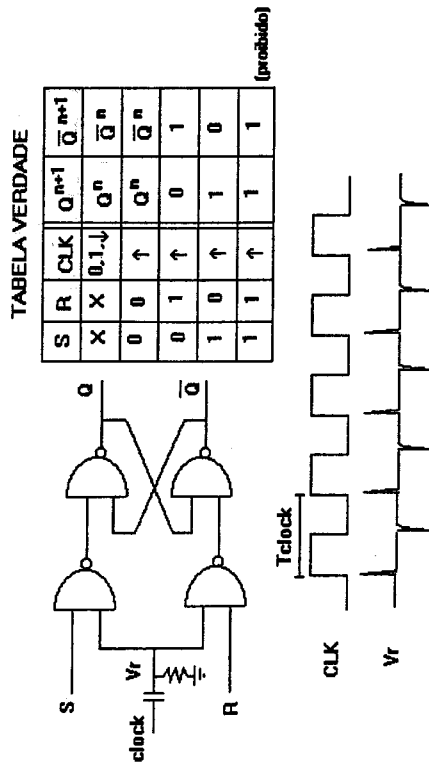


Fig. 11 - Aquisição de dados na borda do relógio

obs: a entrada de dados ocorre no flanco positivo do clock

Como mostrado, somente é permitida a entrada de dados durante pequenos intervalos de tempo (pulsos) dados pela saída (Vr) da rede RC. É necessário, para isto, garantir-se que a constante de tempo RC seja muito menor que o período do clock (RC << Tclock).

A implementação de capacitores em circuitos integrados requer áreas relativamente grandes, o que tem tornado inviável a utilização de redes RC para a geração de pulsos. A solução encontrada consiste em utilizar outras filosofias de

projeto (ex:acoplamento direto) para a construção de flip-flops que permitam a entrada dos dados somente nas transições do clock (flip-flop JK - item 1.9.1)

1.8- Influência do Tempo de Atraso de Propagação

O tempo de atraso de propagação é uma característica inerente a qualquer tipo de circuito, e portanto sua influência também deve ser conhecida quando do estudo de flip-flop. Para isto, na figura 12, o diagrama de tempo da figura 8 é novamente analisado considerando-se o tempo de atraso de propagação (tap) de cada porta.

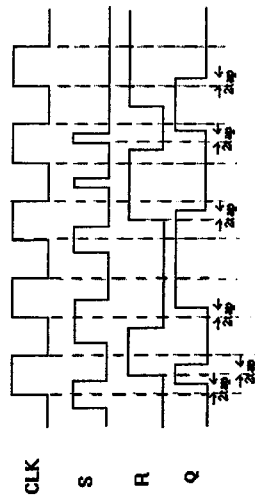


Fig. 12 - Diagrama considerando o tempo de propagação (tap)

Obs:

-os tempos de atraso de propagação de todas as portas são considerados iguais;

-a saída responde 2tap após as mudanças nas entradas;

1.9 - Flip-Flop JK

O flip-flop JK tem por objetivo eliminar a incompatibilidade surgida quando efetua-se, com clk=1, S=R=1 no flip-flop SR. Uma análise mais detalhada da tabela verdade do flip-flop SR mostra que a única condição de estado futuro não contemplada trata-se da possibilidade deste ser o complemento do estado atual. Com o intuito de se alcançar este objetivo a primeira alternativa proposta, mostrada na figura 13, utiliza as

informações dadas pelas saídas atuais (Q e $\bar{Q}$) na entrada, ou seja um processo de realimentação. As entradas S e R passam a chamar-se, respectivamente de J e K.

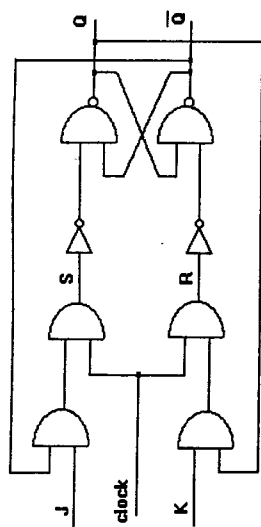


TABELA VERDADE

J	K	Q^n	CLK	S	R	Q^{n+1}
X	X	X	0	0	0	Q^n
0	0	0	1	0	0	0
0	0	1	1	0	0	1
0	1	0	1	0	0	$Q^n - 0$
0	1	1	1	0	1/0	0
1	0	0	1	1/0	0	1
1	0	1	1	0	0	$Q^n - 1$
1	1	0	1	1-0... 0-1...	1-0... 0-1...	oscila
1	1	1	1	0-1... 1-0...	0-1... 1-0...	oscila

Fig. 13 - Proposta de flip-flop JK realimentado

A análise da tabela verdade gerada mostra que esta alternativa não apresenta incompatibilidade quando as entradas J e K assumem nível lógico alto com CLK=1, mas introduz oscilações nas saídas nesta mesma situação. A instabilidade (oscilações) surge devido ao fato da saída assumir o novo estado enquanto o sinal de clock (CLK) está em nível lógico alto. O novo estado, devido a realimentação, é levado para a entrada provocando uma nova mudança de estado na saída, e assim sucessivamente.

Embora esta alternativa não se apresente como satisfatória ela aponta na direção da solução. A eliminação deste inconveniente (instabilidade) pode ser feita através de duas maneiras que serão abordadas nos itens 1.9.1 e 1.9.2. As duas soluções partem do princípio que o novo estado não pode ser levado a saída enquanto o sinal de clock estiver em nível lógico alto.

1.9.1-Flip-Flop JK Disparado pela Borda (Transição)

Este modo consiste em fazer os pulsos de clock mais estreitos que a soma dos tempos de atraso de propagação das portas (2 τ_p), de forma que quando a saída assumir o novo estado as portas "and" controladas pelo clock já estejam desabilitadas, evitando assim que a saída mude mais de uma vez enquanto o clock está alto (é também importante perceber que o pulso de clock deve ser ligeiramente maior que τ_p para permitir a entrada inicial dos dados, e também que os tempos de atraso de propagação das portas podem ser diferentes).

Na prática, como já citado, a geração de pulsos estreitos com tais restrições é inviável. Assim, aqui também a solução consiste em utilizar filosofias de projeto de flip-flops que permitam a entrada de dados somente durante a transição do clock. Estes flip-flops incluem-se na categoria dos que são disparados pela borda do relógio. Neste curso, não serão abordadas as técnicas de projeto pela borda e sim, tão somente, utilizar os resultados encontrados nas suas tabelas funcionais. A seguir são mostrados o símbolo e a tabela verdade para flip-flops disparados pela borda positiva e negativa.

TABELA VERDADE

J	K	CLK	Q^{n+1}	$\bar{Q}^{n+1}$
X	X	0,1, $\downarrow$	Q^n	$\bar{Q}^n$
0	0	$\uparrow$	Q^n	$\bar{Q}^n$
0	1	$\uparrow$	0	1
1	0	$\uparrow$	1	0
1	1	$\uparrow$	$\bar{Q}^n$	Q^n

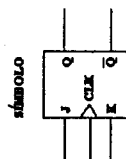


TABELA VERDADE

J	K	CLK	Q^{n+1}	$\bar{Q}^{n+1}$
X	X	0,1, $\uparrow$	Q^n	$\bar{Q}^n$
0	0	$\downarrow$	Q^n	$\bar{Q}^n$
0	1	$\downarrow$	0	1
1	0	$\downarrow$	1	0
1	1	$\downarrow$	$\bar{Q}^n$	Q^n

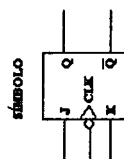


Fig. 14 - Símbolos e tabelas funcionais para os flip-flops disparados pela borda

obs:

-os flip-flops disponíveis comercialmente também apresentam entradas assíncronas.

1.9-Flip-Flop Tipo T ("toggle")

É um flip-flop com uma única entrada, onde J e K são conectados em um único ponto denominado de entrada T. O símbolo e a tabela verdade deste flip-flop são mostrados na figura 18.

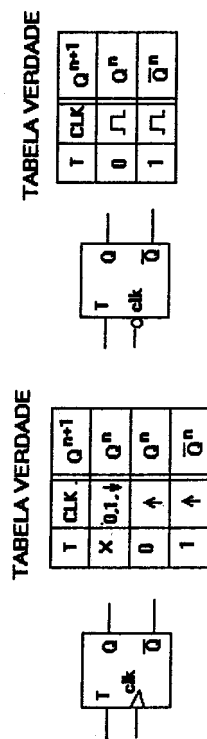


Fig. 18 - Flip-flop tipo T

Se a entrada T for levada a 1 este flip-flop opera como um divisor de frequência.

exercício: Traçar o diagrama de tempo com $T=1$ para o flip-flop disparado pela borda e para o flip-flop mestre-escravo.

1.10-Tempo de Ajuste e Tempo de Conservação

tempo de ajuste: quantidade mínima de tempo que a entrada deve estar disponível antes da ocorrência da transição de clock.

tempo de conservação: quantidade mínima de tempo que o dado deve permanecer após a ocorrência da transição de clock.

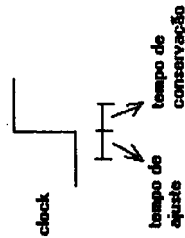


Fig. 19 - Tempo de ajuste e tempo de conservação

A) 74LS74

(TOP VIEW)

1	CLR	14	VCC
2	1D	13	2CLR
3	CLK	12	2D
4	PRE	11	2CLK
5	1Q	10	2PRE
6	1Q	9	2Q
7	GND	8	2Q

B) 74LS76

1	CLK	1	16	1K
2	PRE	2	15	1Q
3	CLR	3	14	1Q
4	1J	4	13	GND
5	VCC	5	12	2K
6	CLK	6	11	2Q
7	PRE	7	10	2Q
8	CLR	8	9	2J

^f This configuration is nonstable; that is, it will not persist when either preset or clear returns to its inactive (high) level.

INPUTS			Q
PRE	CLR	CLK	D
L	H	X	X
H	L	X	X
L	L	X	X
H	H	↑	H
H	H	↑	L
H	H	L	X

^f The output levels in this configuration are not guaranteed to meet the minimum levels in V_{OH} if the lows at preset and clear are near V_{IL} maximum. Furthermore, this configuration is nonstable; that is, it will not persist when either preset or clear returns to its inactive (high) level.

FUNCTION TABLE					OUTPUTS	
INPUTS					Q	$\bar{Q}$
PRE	CLR	CLK	J	K		
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H [†]	H [†]
H	H	↓	L	L	Q ₀	$\bar{Q}_0$
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	TOGGLE	
H	H	X	X	X	Q ₀	$\bar{Q}_0$

^f This configuration is nonstable; that is, it will not persist when either preset or clear returns to its inactive (high) level.

C) 74LS107

1J	1	14	VCC
1Q	2	13	1CLR
1Q	3	12	1CLK
1K	4	11	2K
2Q	5	10	2CLR
2Q	6	9	2CLK
GND	7	8	2J

D) 74LS109

1CLR	1	16	VCC
1J	2	15	2CLR
1K	3	14	2J
1CLK	4	13	2K
1PRE	5	12	2CLK
1Q	6	11	2PRE
1Q	7	10	2Q
GND	8	9	2Q

f The output levels in this configuration are not guaranteed to meet the minimum levels for V_{OH} if the lows at prestat and clear are near V_{IL} maximum. Furthermore, this configuration is nonstable; that is, it will not persist when prestat or clear return to their inactive (high) level.

INPUTS				OUTPUTS	
CLR	CLK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	$\uparrow$	L	L	Q_0	$\bar{Q}_0$
H	$\uparrow$	H	L	H	L
H	$\uparrow$	L	H	L	H
H	$\uparrow$	H	H	TOGGLE	
H	H	H	X	Q_0	$\bar{Q}_0$

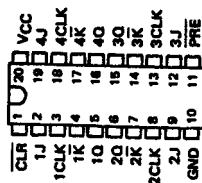
FUNCTION TABLE (each flip-flop)

INPUTS				OUTPUTS	
PRE	CLR	CLK	J	K	Q
L	H	X	X	X	H
H	L	X	X	X	L
L	L	X	X	X	H [†]
H	H	↑	L	L	L
H	H	↑	H	L	L
H	H	↑	L	H	Q ₀
H	H	↑	H	H	L
H	H	L	X	X	Q ₀

14

E) 74LS276

SN74276 ... J PACKAGE
SN74276 ... N PACKAGE
(TOP VIEW)



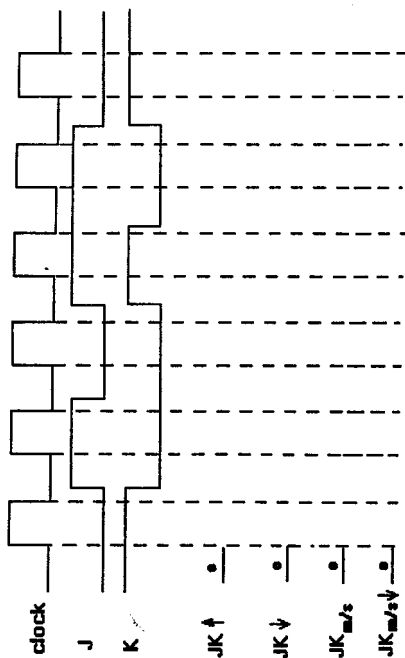
FUNCTION TABLE (EACH FLIP-FLOP)

COMMON INPUTS		INPUTS			OUTPUT	
PRE	CLR	CLK	J	K	Q	Q
L	H	X	X	X	H	H
H	L	X	X	X	L	L
L	L	X	X	X	H [†]	H [†]
H	H	L	L	H	Q ₀	Q ₀
H	H	L	H	H	H	H
H	H	L	L	L	L	L
H	H	L	H	L	TOGGLE	TOGGLE
H	H	H	X	X	Q ₀	Q ₀

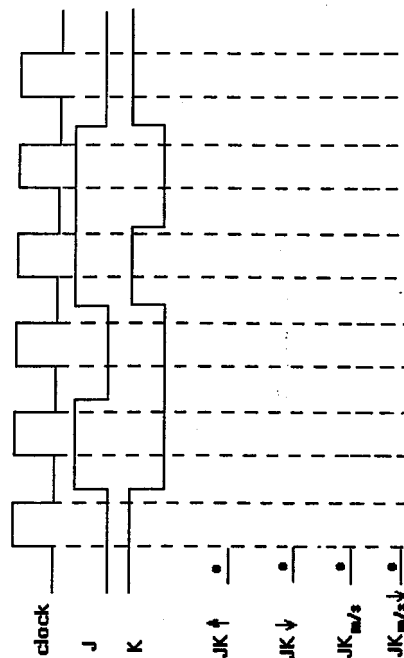
[†] This configuration is nonstable; that is, it may not persist when preset and clear return to their inactive (high) level.

1.13 - Exercícios:

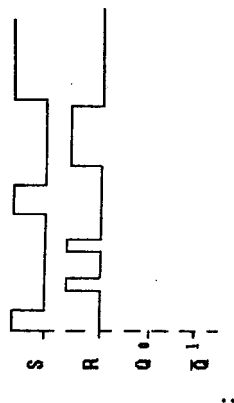
1)- dado o conjunto de entradas J e K mostradas indicar o comportamento da saída para os flip-flops JK disparado pela borda positiva, JK disparado pela borda negativa e JK mestre-escravo.



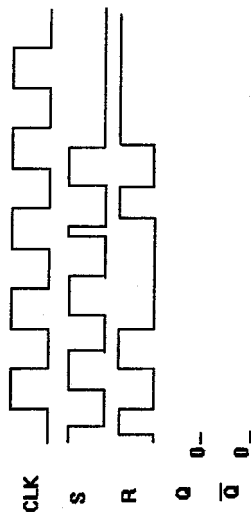
2)- repetir o exercício anterior levando-se em consideração o atraso de propagação dos flip-flops.



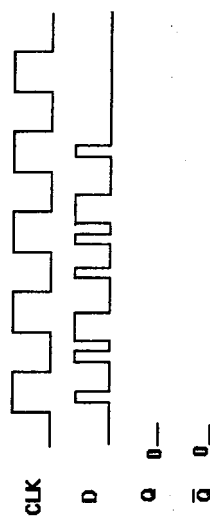
- 3) Completar o seguinte diagrama para o flip-flop SR implementado com portas NAND.



- 4) Complete o seguinte diagrama de tempo considerando o flip-flop SR sincronizado com portas NAND.

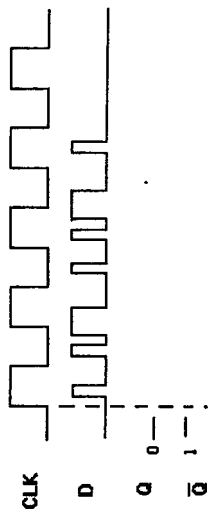


- 5) complete o seguinte diagrama de tempo considerando o flip-flop tipo D sincronizado implementado com portas NAND

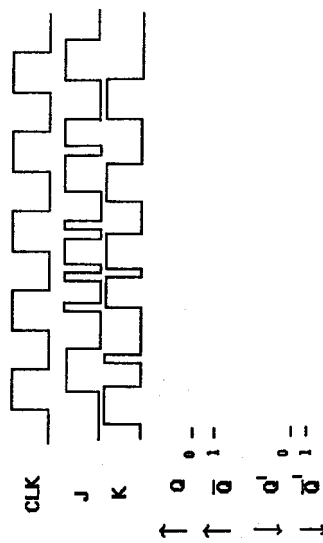


- 6) Introduza entradas assíncronas no flip-flop SR implementado com portas NOR. Monte a Tabela Verdade.

- 7) Complete o seguinte diagrama para o flip-flop tipo D implementado com portas NAND considerando o tempo de atraso de propagação.



- 8) Complete o seguinte diagrama de tempo:



- 9) É possível implementar utilizando flip-flops disparados pela borda um dispositivo que tenha comportamento semelhante (não transparente) ao do flip-flop mestre-escravo sincronizado pelo nível.

ANEXOS

FUNDAMENTOS DE ELETRÔNICA DIGITAL – PORTAS LÓGICAS E SUAS APLICAÇÕES

JONATHAS WILLIAM NARCISO DE OLIVEIRA
Aluno do curso "Montagem e Manutenção de Microcomputadores"

EDUARDO CAMPOS DOS SANTOS - ORIENTADOR
Professor e Coordenador

MICROCAMP – BHCENTRO
Av. João Pinheiro, 297. Centro.
CEP: 30130-180
e-mail: coordena@microcampbhcentro.com.br

RESUMO

O presente trabalho tem como objetivo mostrar a importância, funcionamento e aplicações das portas lógicas. Exibindo: explicações sucintas de componentes usados na construção das mesmas, símbolos, tabelas de verdade, algumas de suas variações e exemplos de sua utilização.

ÍNDICE

1. INTRODUÇÃO	1
2. COMPONENTES ELETRÔNICOS	1
2.1 Resistores	1
2.2 Capacitores	2
2.3 Componentes baseados em semicondutores	3
2.3.1. Uníões PN	5
2.3.2. Diodos	6
2.3.3. Transistores	6
3. CIRCUITOS ELETRÔNICOS	7
4. PORTAS LÓGICAS – CIRCUITOS ELETRÔNICOS DIGITAIS	7
4.1 PORTA E ou AND	8
4.2 PORTA OU ou OR	8
4.3 PORTA NÃO ou NOT (inversora)	9
4.4 PORTA NÃO E, NE ou NAND	9
4.5 PORTA NÃO OU, NOU ou NOR	9
4.6 PORTA EXCLUSIVE-OR(OU-EXCLUSIVAS)	10
5. APLICAÇÕES DE PORTAS LÓGICAS	10
5.1 Circuito Oscilador ou Circuito Gerador de Clock	11
5.2 Comparador de bits e o Comparador de palavras	12
5.3 Conversor decimal-binário	13
5.4 Conversor binário-decimal	13
6. PROJETOS FUTUROS	15
7. CONCLUSÃO	16
8. AGRADECIMENTOS	16
9. REFERÊNCIAS BIBLIOGRÁFICAS	16

1. INTRODUÇÃO

Em meados de 1800, o autodidata inglês George Boole após estudar várias teorias matemáticas, desenvolveu um sistema matemático de análise lógica que ficou conhecido como *Lógica Formal* ou *Álgebra de Boole*.

No início da Era Eletrônica, todos os problemas eram resolvidos por sistemas analógicos, também conhecidos por sistemas lineares.

Com o avanço da tecnologia, esses mesmos problemas começaram a ser solucionados através da eletrônica digital. Esse ramo da eletrônica emprega em suas máquinas, tais como: computadores, processadores de dados, sistemas de controle e de comunicação digital, apenas um grupo de circuitos lógicos básicos, que são conhecidos como *portas lógicas*.

As portas lógicas e suas aplicações em computação digital constitui o tema deste trabalho. Faremos inicialmente uma breve introdução sobre os componentes eletrônicos que utilizaremos nas construção de portas lógicas.

Como principais referências bibliográficas, citamos: Paixão e Honda (1999), Boylestad e Nashelsky (1984), Capuano e Idoeta (1985), Malvino (1985) e Carvalho (1980).

2. COMPONENTES ELETRÔNICOS

Se você observar, sob a ótica de um principiante, o interior de um sistema eletrônico tal como o rádio, um televisor, um equipamento de som ou um computador – nosso caso, é muito provável que se sinta intimidação e desmotivado a estudar eletrônica. À primeira vista, tudo parece muito complicado que se pode pensar que a eletrônica é só para gênios, cientistas ou intelectos superiores. Ainda bem que em eletrônica as coisas são bem mais fáceis do que aparentam ser. Isto acontece por duas razões fundamentais. Em primeiro lugar, mesmo que os sistemas eletrônicos consistam de uma grande quantidade de componentes ou peças, estas últimas se agrupam em número muito limitado de tipos básicos, cada qual com suas próprias variantes. Em segundo lugar, os componentes se agrupam formando circuitos que cumprem funções determinadas. Novamente, ainda que um sistema eletrônico possa consistir de muitos circuitos, estes pertencem a um número limitado de categorias básicas. A combinação de circuitos dá origem a sistemas, que são utilizados em comunicações, controle de potência, áudio, vídeo, entretenimento e outras aplicações.

Os componentes são os blocos básicos de construção dos sistemas eletrônicos. A função de um componente é manipular de alguma forma a corrente elétrica que circula através de um circuito, para, por exemplo, limitá-la, armazená-la, interrompê-la, amplificá-la, dirigi-la e transferi-la. A seguir estão os principais componentes utilizados nas *portas lógicas* como veremos mais adiante.

2.1 Resistores

As resistências são componentes eletrônicos projetados para oferecer certa oposição ou resistência à passagem de corrente elétrica. Fisicamente são feitas de carvão ou de metal. São utilizadas principalmente para limitar ou controlar a quantidade de corrente que circula através de um circuito. São os componentes mais abundantes nos equipamentos eletrônicos e os mais baratos (veja a figura 2.1). As resistências podem ser *fixas* ou *variáveis*, dependendo da quantidade de oposição que apresentam para a passagem da corrente, chamada de *resistência*. As resistências fixas também são denominadas de *resistores*.

Resistor

condutor

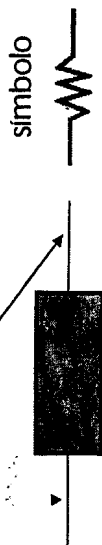


Figura 2.1

cavão/metal

2.2 Capacitores

Os capacitores são componentes projetados para armazenar temporariamente energia elétrica em forma de tensão e se opõem às variações de tensões elétricas. Fisicamente são formados por duas superfícies metálicas chamadas *placas*, separadas por um material dielétrico (isolante). Eles são, depois das resistências, os elementos mais abundantes na maioria dos circuitos eletrônicos, mas são mais caros.

Os capacitores, tal como os resistores, podem ser fixos ou variáveis dependendo de suas cargas elétricas ou eletricitidade. Sua capacidade de armazenamento de carga é medida por uma grandeza denominada *capacitância*. A unidade de medida da capacitância é o Faraday (F), sendo que os valores mais usuais são da ordem de micro-Faraday (uF) ou pico-Faraday (pF).

Capacitor



Figura 2.2

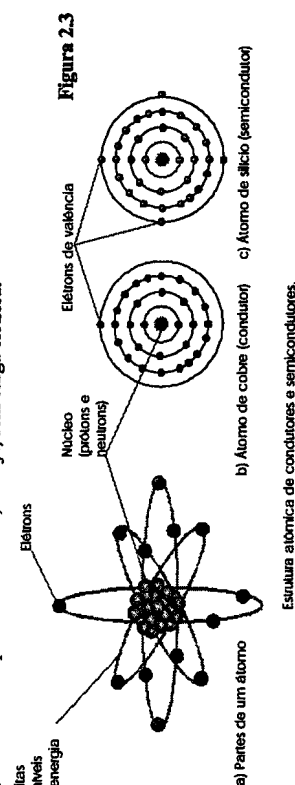
2.3 Componentes baseados em semicondutores

A grande revolução da eletrônica e sobretudo da microeletrônica só foi possível graças ao advento dos dispositivos baseados em semicondutores: os diodos, os transistores e os circuitos integrados.

Denominamos semicondutores uma classe de elementos químicos que apresentam características de condutibilidade bem particulares que os diferem de elementos classificados como condutores ou isolantes.

Os semicondutores, são componentes eletrônicos baseados na propriedade de certos materiais do mesmo nome como o silício, o germânio, o sulfeto de cádmio e o arsênio de gálio de comportarem-se indistintamente como condutores ou isolantes sob determinadas condições ou estímulos externos.

A característica principal dos semicondutores, diferentemente dos condutores e dos isolantes é sua estrutura atômica (veja a figura 2.5). Para compreender este conceito, é conveniente lembrar que no átomo, componente essencial da matéria, o núcleo, formado pelos prótons e os nêutrons, está rodeado pelos elétrons, os quais se distribuem em órbitas ou níveis de energia. São sete níveis, onde, cada nível é representado por uma letra sendo o mais próximo do núcleo o nível da camada K e o mais afastado o nível da camada Q determinando assim o nº máximo de elétrons: K=2; L=8; M=18; N=32; P=18; Q=2. Os prótons são partículas de carga positiva (+), os elétrons partículas de carga negativa (-) e os nêutrons partículas neutras, ou seja, sem carga elétrica.



Os elétrons mais distantes do núcleo, localizados no nível de energia mais externo, são denominados elétrons de valência e são muito importantes do ponto de vista eletrônico porque determinam a condutibilidade de um material, ou seja, sua maior ou menor facilidade para fornecer portadores de carga.

Os condutores, como o cobre, a prata e o ouro, tem pouquíssimos elétrons na faixa de valência, tipicamente 1, e os mesmos são fracamente atraídos pelo núcleo. Portanto sob a influência de uma força externa, esses elétrons de valência podem facilmente escapar do átomo, convertendo-se em elétrons livres que viajam ou migram através do material e participam na criação das correntes elétricas.

Os semicondutores, como o silício e o germânio, se caracterizam por serem tetravalentes, ou seja, por terem quatro elétrons de valência. Estes elétrons formam enlaces com os elétrons de valência dos átomos vizinhos produzindo um padrão tridimensional regular chamado rede cristalina ou cristal (veja a figura 2.6). Em um cristal ou semicondutor puro, cada átomo compartilha seus elétrons de valência com os átomos vizinhos até ficar quimicamente estável, ou seja, com 8 elétrons em sua órbita mais externa (Lei do octeto).

Rede cristalina ou Cristal

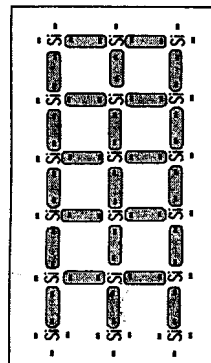


Figura 2.4

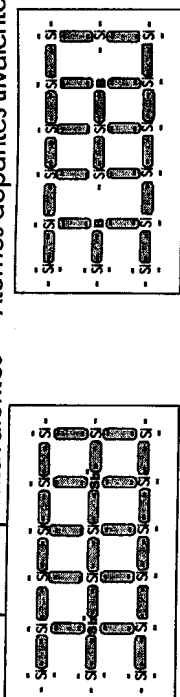
substrato

Os cristais semicondutores puros ou intrínsecos são raramente empregados em Eletrônica, devido a que, em seu estado natural, possuem muito poucos elétrons livres e precisam de grandes quantidades de energia para transportar correntes importantes. Na prática, os materiais semicondutores utilizados na fabricação de diodos e transistores, são dopados, ou seja, contém quantidades muito pequenas, mas controladas, de impurezas denominadas dopantes que são as que determinam suas características elétricas. Estes tipos de semicondutores são denominados extrínsecos.

Nota: O material semicondutor puro sobre o qual se realiza o processo de dopagem para obter um semicondutor extrínseco é conhecido como substrato. A maioria dos dispositivos semicondutores práticos utilizam silício como substrato por ser um elemento muito mais estável, econômico e abundante. Dopar um semicondutor é como adicionar um pouco de levedura a um montão de farinha, misturar bem e colocar em um forno: aparentemente não aconteceu nada mas os efeitos físicos são evidentes.

Os dopantes utilizados nos semicondutores extrínsecos são invariavelmente átomos de elementos químicos pentavalentes ou trivalentes, ou seja, com cinco ou três elétrons de valência. Devido à concentração de dopante, que é extremamente pequena, da ordem de um átomo de impureza por cada duzentos milhões de átomos de silício, estes últimos sempre envolverão aos primeiros e compartilharão seus átomos de valência como no cristal original, com a diferença que agora um átomo de silício foi substituído por um átomo de impureza.

Os substratos dopados com impurezas pentavalentes são denominados semicondutores tipo N. Nestes, quatro dos cinco elétrons de valência do átomo dopante formam enlaces covalentes com os quatro átomos vizinhos. O elétron restante não fica ligado a nenhum átomo e é livre para mover-se através do cristal, convertendo-se em um portador potencial de corrente. Portanto, um semicondutor tipo N é um doador de elétrons. O principal elemento utilizado como doador em cristais de silício é o fósforo. Para cristais de germânio utilizam-se o antimônio e o arsênio.



Material tipo N Material tipo P

Basta-nos uma pesquisa sobre a variação das propriedades de um elemento químico em função de sua posição na tabela periódica, para justificarmos dentre outras coisas sua eletropositividade e eletronegatividade.

Eletropositividade de um átomo é uma medida de sua tendência em ceder elétrons. Um elétron é tanto mais eletropositivo quanto maior for a tendência de seu átomo em ceder elétrons e se transformar nos respectivos cátions.

Eletronegatividade de um átomo é uma medida de sua tendência em atrair elétrons. Um elemento é tanto mais eletronegativo quanto maior for a tendência de seu átomo em ceder elétrons e se transformar nos respectivos ânions.

Silício		Germânio	
Doadores	Aceitadores	Doadores	Aceitadores
Fósforo - P	Alumínio - Al	Antimônio - Sb	Índio - In
	Boro - B	Arsênio - As	Gálio - Ga

Os substratos dopados com impurezas trivalentes são denominados **semicondutores tipo P**. Nestes, os três elétrons de valência do átomo dopante formam enlaces covalentes com três átomos vizinhos de silício. O elétron faltante origina uma lacuna, o qual se comporta como uma carga positiva livre, capaz de atrair um elétron externo. Portanto, um semicondutor do tipo P é um **aceitador** de elétrons. Os principais elementos utilizados como impurezas aceitadoras em cristais de silício são o alumínio e o boro. Para cristais de germânio utilizam-se o índio e o gálio.

2.3.1. União PN

Os semicondutores tipos N e P por si mesmos, não tem maior utilidade prática, exceto quando estão fortemente dopados, e neste caso empregam-se como resistores que dependem da temperatura. Um dispositivo muito mais útil é a **união PN**, obtida ao dopar um cristal de silício ou germânio puro (substrato) com impurezas pentavalentes e trivalentes de modo que uma metade seja do tipo n e a outra metade seja do tipo p.

Todos os dispositivos semicondutores, incluindo os diodos e transistores estão baseados na combinação de duas ou mais camadas alternadas de materiais dos tipos N e P, ou seja, possuem uma ou mais uniões PN. O **diodo**, por exemplo, é uma união PN.

Em uma união PN há inicialmente um excesso de elétrons livres no lado N e de lacunas no lado P. Portanto, alguns elétrons do lado N serão atraídos por algumas lacunas do lado P, e vice-versa (veja figura 2.8).

Se for aplicada uma tensão externa **polarizando diretamente**, o pólo (+) da bateria repele as lacunas de P e o pólo (-) os elétrons de N, permitindo que atravessem a união. Se aplicarmos uma tensão externa **polarizando inversamente**, o pólo positivo (+) da bateria atrai os elétrons livres do material tipo N e o pólo negativo (-) as lacunas do material tipo P. Como resultado, a zona de esgotamento (divisa entre os dois materiais) se alarga e a união PN apresenta uma resistência muito elevada à passagem de corrente, comportando-se como um isolante.

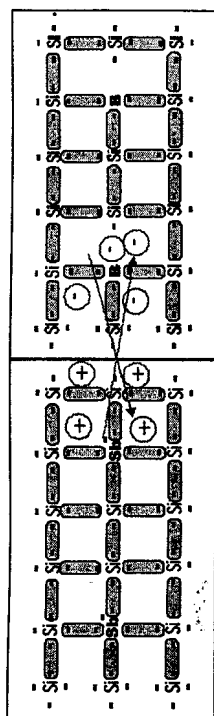


Figura 2.6

2.3.2. Diodos

Os diodos são componentes projetados para permitir a passagem de corrente elétrica em um sentido e bloqueá-la no sentido oposto. Fisicamente são formados por duas camadas de material semicondutor dopado, ou seja, tratados com impurezas especiais chamadas **material P e material N** (veja a figura 2.3), e possuem externamente dois terminais de conexão, chamados **anodo (positivo)** e **catodo (negativo)**. A posição do catodo é normalmente indicada através de uma faixa colorida impressa em uma ponta. São, portanto, componentes polarizados.

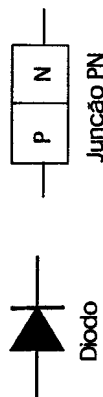


Figura 2.7

2.3.3. Transistores

Os transistores, em geral, são componentes primariamente projetados para serem utilizados como amplificadores, ou seja, para controlar correntes grandes a partir de correntes ou voltagens pequenas. Esta operação é denominada **amplificação**. Também são utilizados como interruptores eletrônicos para permitir ou bloquear a passagem de corrente sem ações mecânicas.

Os transistores inventados em 1948, são os dispositivos eletrônicos mais importantes da atualidade e os que realmente iniciaram a revolução eletrônica. Podem vir como peças soltas ou discretas, ou estar incorporados em **circuitos integrados**, os quais podem chegar a conter várias centenas de milhares de transistores em um espaço muito pequeno.

Segundo Paixão e Honda (1999) funcionalmente, esse dispositivo faz a mesma coisa que a válvula e também conta com três partes: uma chamada base que controla o fluxo de elétrons (como a grade para válvulas), uma chamada emissor que fornece os elétrons (como o filamento), e uma terceira denominada coletor que capta os elétrons (como a placa).

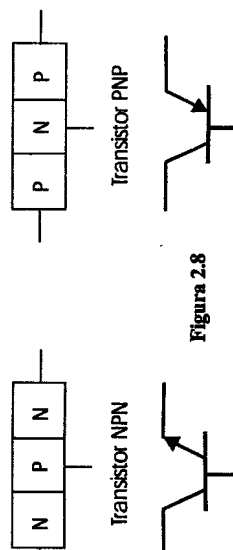


Figura 2.8

3. CIRCUITOS ELETRÔNICOS

O que é um circuito?

De um modo geral, um circuito é a combinação de componentes conectados de forma a proporcionar uma ou mais trajetórias fechadas para a circulação da corrente e permitir o aproveitamento dos elétrons em movimento para produzir um trabalho útil. A seguir estão listadas algumas definições importantes na análise de um circuito:

Nó: é todo ponto do circuito comum a três ou mais condutores.

Ramo: é todo trecho do circuito compreendido entre dois nós consecutivos.

Malha: é todo conjunto de ramos formando um circuito fechado.

Um circuito pode vir a ser muito complexo. No entanto, independentemente de sua complexidade, todos precisam de pelo menos uma fonte de energia, um par de condutores e uma carga.

Além disso a maioria dos circuitos também requer dispositivos de controle para regular o fluxo de elétrons até a carga e dispositivos para bloqueá-lo automaticamente quando se produz uma condição anormal de funcionamento.

4. PORTAS LÓGICAS – CIRCUITOS ELETRÔNICOS DIGITAIS

Os bits- dígitos binários- são representações de potenciais elétricos que assumem apenas duas situações possíveis em uma faixa de tempo.

As portas lógicas, são circuitos eletrônicos bem definidos que estabelecem a saída de um circuito dependente das situações possíveis nas entradas do mesmo, onde teremos apenas dois estados possíveis: o estado 0 (zero) e o estado 1 (um).

O estado zero (0) representará, por exemplo: portão fechado, aparelho desligado, ausência de tensão, chave aberta, não, etc.; o estado um (1) representará então: portão aberto, aparelho ligado, presença de tensão, chave fechada, sim, etc.

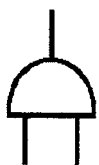
Note, então, que se representarmos por zero (0) uma situação, representaremos por um (1) a situação contrária.

O conceito aplicado em cada porta lógica pode ser estendido para quantas variáveis na entrada o circuito possuir.

4.1 PORTA E ou AND

A porta E é um circuito que executa a função E. Ou seja, teremos a saída (s) no "estado um" se, e somente se todas as entradas - uma e outra - forem nível alto; e teremos a saída "igual a zero" nos demais casos.

Porta AND (E)



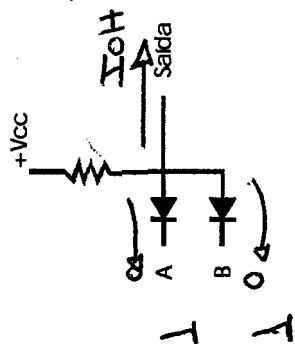
Símbolo lógico

(Mostre o fluxo da corrente)

Figura 4.1

Tabela verdade

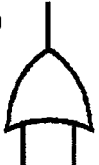
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1



4.2 PORTA OU ou OR

A porta OU é um circuito que executa a função OU. Ou seja, teremos a saída (s) nos "estado um" quando uma ou mais variáveis de entrada forem iguais a um (1), e teremos a saída no estado zero (0) no caso restante -quando todas as entradas forem iguais a zero.

Porta OR (OU)



Símbolo lógico

Tabela verdade

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

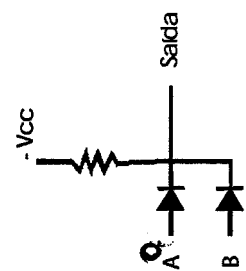


Figura 4.2

4.3 PORTA NÃO ou NOT (Inversora)

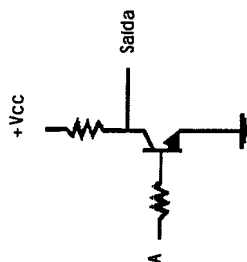
Porta NOT (Inversora)



Tabela verdade

A	S
0	1
1	0

Figura 4.3



A porta NÃO é um circuito que executa a função NÃO. É aquela que inverte o estado da variável (s), ou seja, se a variável estiver em zero (0) vai para um (1), se a variável estiver em um (1) vai para o (0).

4.4 PORTA NÃO E, NE ou NAND

Como o próprio nome "NÃO E" diz: essa porta é uma composição da porta E com a porta NÃO, ou seja, teremos a porta E invertida. Observe que a tabela verdade da NAND é exatamente o inverso da tabela verdade da porta AND.

Porta NAND (NÃO-E)

Símbolo lógico

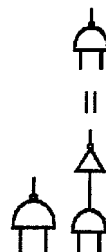
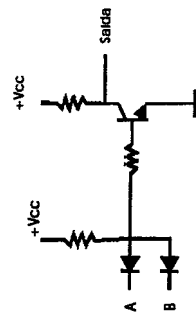


Tabela verdade

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

Figura 4.4



4.5 PORTA NÃO OU, NOU ou NOR

Como o próprio nome "NÃO OU" diz: essa porta é uma composição da porta OU com a porta NÃO, ou seja, teremos a porta OU invertida. Observe que a tabela verdade da NOR é exatamente o inverso da tabela verdade da porta OR.

Porta NOR (NÃO-OU)

Símbolo lógico

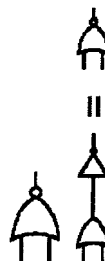
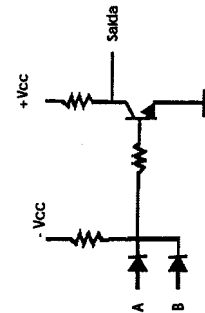


Tabela verdade

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

Figura 4.5



4.6 PORTA EXCLUSIVE-OR(OU-EXCLUSIVAS)

Uma porta OR reconhece palavras (entradas) com um ou mais 1s. A porta EXCLUSIVE-OR de duas entradas é diferente; ela reconhece apenas palavras que tem um número ímpar de 1s. Na figura 4.13 temos duas entradas baixas; assim, a saída final é baixa. Se A for baixa e B for alta, a porta AND superior terá uma saída alta; portanto, a saída final será alta. Analogamente, um A alto e um B baixo resultam numa saída final que é alta. Se ambas as entradas forem altas, ambas as portas AND terão saídas baixas e a saída final será baixa.

A saída será alta quando A ou B for alta, mas não ambas; por isso é que o circuito é conhecido como uma porta EXCLUSIVE-OR. Em outras palavras, a saída de uma porta XOR de duas entradas será um 1 somente quando as entradas forem diferentes.

A figura 4.14 é o símbolo padrão de uma porta XOR de 2 entradas. Sempre que vemos este símbolo, lembramos do funcionamento ou ação: as entradas devem ser diferentes para se obter uma saída alta.

Uma aplicação da XOR é o comparador de bits que veremos mais adiante no item 5 e 5.2

Porta XOR (OU-EXCLUSIVO)

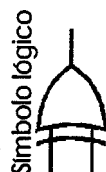


Tabela verdade

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

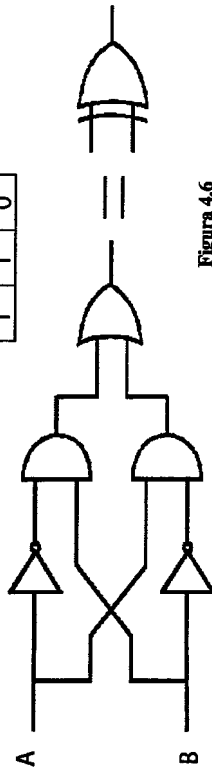


Figura 4.6

5. APLICAÇÕES DE PORTAS LÓGICAS

No início da era Eletrônica todos os problemas eram resolvidos por sistemas analógicos, também conhecidos por sistemas lineares.

Com o avanço da tecnologia, esses mesmos problemas começaram a ser solucionados através da eletrônica digital. Esse ramo da eletrônica emprega nas suas máquinas, tais como: computadores, processadores de dados, sistemas de controle e comunicação digital, apenas um pequeno grupo de circuitos lógicos básicos, que são conhecidos como portas lógicas.

Atualmente há vários tipos de circuitos lógicos bastante comuns. Os usuários de circuitos lógicos digitais podem utilizar unidades CI oferecidas por vários fabricantes. Como não há um circuito específico aceito universalmente, parece razoável considerar alguns dos mais utilizados para entender a operação e suas vantagens e desvantagens relativas.

A lógica transistor-transistor (transistor-transistor logic - TTL) é uma forma de se construir porta lógica que, apesar de poder ter componentes separados, é própria para ser fabricada na forma integrada.

5.1 Circuito Oscilador ou Circuito Gerador de Clock

O circuito oscilador ou gerador de clock é um circuito muito utilizado nas placas de diversos equipamentos, inclusive na placa-mãe do computador - executando operações de extrema importância para o correto funcionamento sincronizado dos demais circuitos da placa e também do processador.

No entanto, para cada tipo de dispositivo, os circuitos osciladores são escolhidos em sua categoria de CI's levando-se em consideração alguns fatores como: 1) custo, 2) dissipação de potência, 3) velocidade de operação 4) imunidade a ruído.

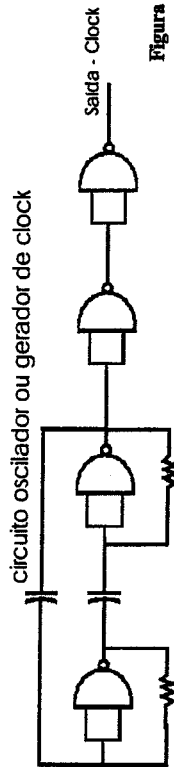


Figura 5.1

Analisando o circuito da figura 5.15, vemos que há 4 portas NOR de duas entradas, 2 capacitores e 2 resistores. Este circuito pode ser construído utilizando-se, por exemplo, o CI SN7400. A figura 5.16 mostra o esquema do SN7400. Na figura 5.17 temos uma ilustração do circuito oscilador construído a partir do SN7400. No CI temos da esquerda para a direita, uma entrada da 1ª porta ligada ao capacitor, ao resistor e à saída da 2ª porta; e a outra entrada da 1ª porta na terra (GND). Depois temos uma entrada da 2ª porta ligada à saída da 1ª porta; e a outra entrada da 2ª porta na terra (GND). Adiante temos as duas entradas da 3ª porta ligadas à saída da 2ª porta e ao capacitor. Prosseguindo temos as entradas da 4ª porta ligadas à saída da 3ª porta e ao resistor. Por final, temos a saída do circuito na 4ª porta para gerar o clock.

Ligação das entradas e saídas do circuito oscilador

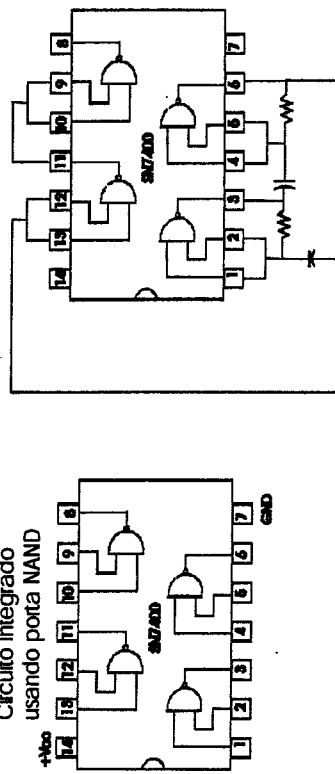


Figura 5.2

Figura 5.3

5.2 Comparador de bits e o Comparador de palavras

Este circuito foi projetado para reconhecer apenas entradas de palavras em que os bits são iguais, e desconsiderar as entradas quando os bits forem diferentes.

Em razão da inversão no lado de saída, a tabela verdade de uma porta XNOR é o complemento de uma tabela-verdade XOR. Conforme mostrado na tabela verdade, a saída é alta quando as entradas são as mesmas. Por esta razão a porta XNOR de duas entradas é idealmente adequada à comparação de bits, reconhecendo quando dois bits de entradas são idênticos. (O Exemplo tal nos diz mais alguma coisa a respeito da comparação de bits.) Por causa da inversão no lado de saída, estas portas XNOR realizam a função complementar das portas XOR. Em vez de reconhecer palavras de paridade ímpar, as portas XNOR reconhecem palavras de paridade par.

O que o circuito faz? O circuito é um comparador de palavras; ele reconhece duas palavras (entradas) idênticas. A porta XNOR mais a esquerda compara A5 e B5, se elas forem as mesmas, Y5 será 1. A segunda porta compara A4 e B4; se elas forem as mesmas, Y4 será 1. Alternadamente, as portas restantes comparam os bits que são deixados, produzindo um saída 1 para bits iguais e uma saída 0 para bits desiguais.

Se as palavras A e B forem idênticas, todas as portas OR terão saídas altas e a porta AND terá IGUAL alto. Se as palavras A e B diferirem uma ou mais posições de bit, a porta AND terá um IGUAL baixo.

Símbolo lógico

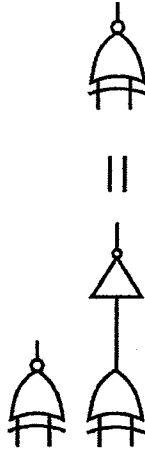


Tabela verdade

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

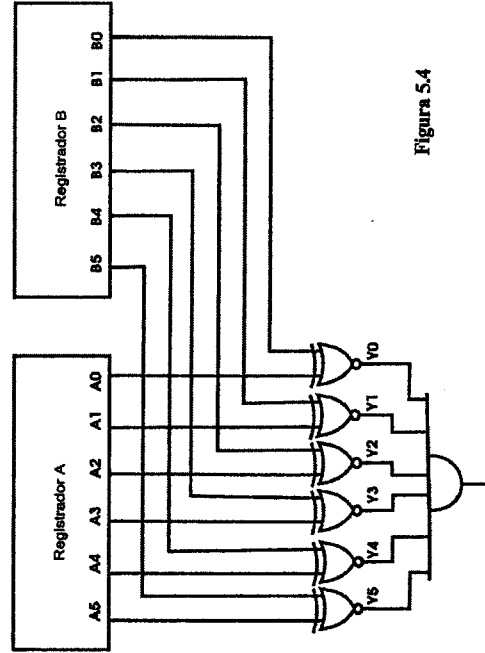


Figura 5.4

5.3 Conversor decimal-binário

O que esse circuito faz? A figura 5.5 é um *codificador*, de decimal em binário, um circuito que converte decimal em binário. Por exemplo, quando o botão 3 é pressionado as portas OR designadas Y1, Y0 tem as entradas altas; portanto a palavra de saída é Y3, Y2, Y1, Y0 = 0011. Se o botão 5 for comutado, as portas OR designadas Y2 e Y0 terão entradas altas e a palavra de saída tornar-se-á Y3, Y2, Y1, Y0 = 0101. Quando a chave 9 for comprimida, Y3, Y2, Y1, Y0 = 1001.

A intenção com a utilização de circuitos digitais é construir sistemas de automação, cálculo ou controle que tragam benefícios ao ser humano que utiliza o sistema numérico decimal no dia-a-dia e não o sistema binário usado pelos circuitos digitais.

Como já foi explicado no tópico sobre portas OR, uma ou mais entradas altas produzem uma saída alta.

Os botões da figura 5.5 são botões como os de um teclado de um computador. Os bits ou as saídas, fora das portas OR, formam uma palavra de 4 bits, designada por Y3, Y2, Y1, Y0. Observe que toda vez que um botão é pressionado é fechado contato com a tensão +5v.

A palavra de saída designada por Y3, Y2, Y1, Y0 sempre é igual ao equivalente em binário do botão que está sendo pressionado.

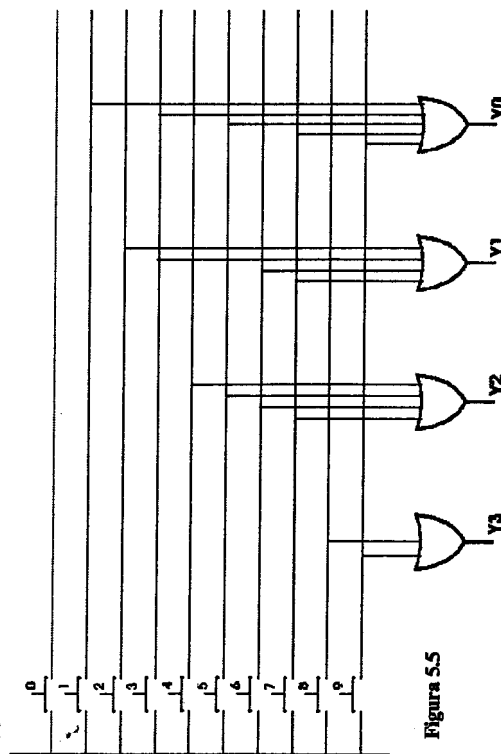


Figura 5.5

5.4 Conversor binário-decimal

O que faz o circuito da figura 5.6? Este é um *decodificador* binário para decimal, um circuito que converte de binário para decimal. Por exemplo, quando o conteúdo do registrador for 0011, a porta AND Y3 terá todas as entradas altas; portanto Y3 é alta. Além disso, o conteúdo 0011 do registrador significa que todas as outras portas AND tem pelo menos uma entrada baixa. Em consequência, todas as outras portas AND tem saídas baixas. (Analisar o circuito para convencer-se.)

A intenção com a utilização de circuitos digitais é construir sistemas de automação, cálculo ou controle que tragam benefícios ao ser humano que utiliza o sistema numérico decimal no dia-a-dia e não o sistema binário usado pelos circuitos digitais.

Um registrador é um grupo de elementos de memória que funcionam juntos como uma unidade. Eles também podem ser construídos com portas lógicas.

Se o conteúdo do registrador mudar para 0100, somente a porta AND Y4 terá todas as entradas altas; portanto, somente Y4 será alta. Se o conteúdo do registrador mudar para 0111, somente Y7 será a única saída alta.

Este circuito é também chamado decodificador de 4 linhas para 10 linhas porque há 4 linhas de entrada e 10 linhas de saída. Um outro nome para ele é decodificador de 1 de 10 porque somente 1 de 10 linhas de saída tem uma alta tensão.

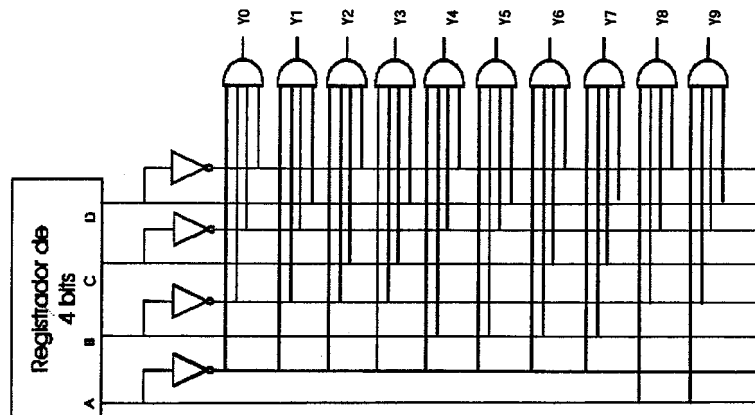


Figura 5.6

6. PROJETOS FUTUROS

Um computador é um amontoado inútil de Hardware até que alguém o programe. Isto significa o carregamento de *instruções* passo-a-passo na memória antes do início de um processamento de dados.

As instruções são operações de rotinas programadas que o computador pode executar.

Com um conjunto de instruções básicas como as da tabela 6.1, por exemplo, poderíamos programar o computador para executar operações de adição e subtração como mostra a tabela 6.2.

Mnemônicos	Operação	Representação
LDA	Carregue os dados da RAM no acumulador.	0000
ADD	Soma os dados da RAM com o acumulador.	0001
SUB	Subtrai os dados da RAM do acumulador.	0010
OUT	Carregue os dados do acumulador no registrador de saída	0011
HTL	Pare o processamento.	0000

Tabela 6.1

Endereço	Conteúdo
0H	LDA 9H
1H	ADD AH
2H	ADD BH
3H	SUB CH
4H	OUT
5H	HTL
6H	FFH
7H	FFH
8H	FFH
9H	01H
AH	02H
BH	03H
CH	04H
DH	FFH
EH	FFH
FH	FFH

Tabela 6.2

Analisemos o conteúdo da tabela 6.2. O intervalo de endereços entre 0H e 5H estão armazenando instruções relacionadas na tabela 6.1. Nos demais endereços, encontramos dados armazenados.

O programa é executado a partir do endereço mais baixo (0H). A primeira instrução carrega o conteúdo do endereço 9H da RAM no acumulador (um registrador de memória intermediária que fica dentro do processador), e assim o conteúdo do acumulador torna-se $A = 01H$. A segunda instrução soma o conteúdo do endereço AH da RAM ao conteúdo do acumulador: $A = 01H + 02H = 03H$. Similarmente, a terceira instrução soma o conteúdo do endereço BH da RAM ao acumulador: $A = 03H + 03H = 06H$. A instrução SUB subtrai o conteúdo do endereço CH da RAM para produzir $A = 06H - 04H = 02H$. A instrução OUT carrega o conteúdo do acumulador para a porta de saída; portanto, o indicador visual em binário mostra 0000 0010, o equivalente, em decimal, ao número 2. A instrução HTL interrompe o processamento dos dados. É claro que podemos usar na saída um conversor binário-decimal conforme apresentado na seção 5.4.

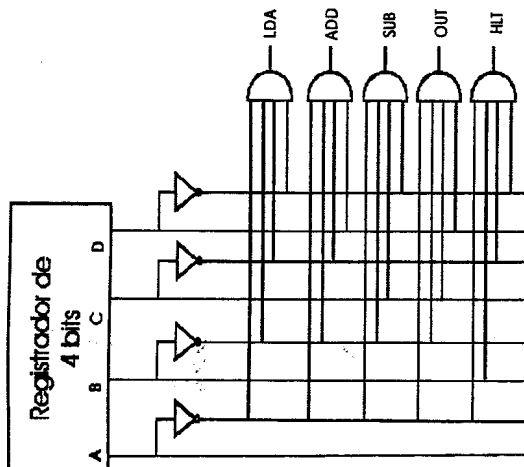


Figura 6.1

Em breve, num outro projeto, estaremos tratando sobre a programação básica dos computadores, relacionado às portas lógicas, os sistemas de numeração, linguagem de programação, etc.

7. CONCLUSÃO

Através da utilização conveniente das portas lógicas, podemos implementar todas as expressões geradas pela Álgebra de Boole, que como vimos na introdução, é a base de construção dos computadores, processadores de dados, sistemas de controle e de comunicação digital.

8. AGRADECIMENTOS

Meus agradecimentos ao coordenador Eduardo Santos, ao professor Anderson Pereira Patrício e a todos que usaram de paciência e compreensão para comigo no auxílio desta obra.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- Paixão, R. R. & Honda, R. (1999) "Processadores Intel", Ed. Érica.
- Boylestad, R. & Nashelsky, L. (1984) "Dispositivos Eletrônicos e Teoria de Circuitos", Ed. PHB.
- Capuano, F.G. & Ideta, I.V. (1985) "Elementos de eletrônica digital", 6ª edição. São Paulo: Ed. Érica 1984.
- Malvino, P. A. (1985) "Microcomputadores e Microprocessadores" Ed. McGraw Hill.
- Carvalho, G.C. (1980) "Química Moderna" 1º Vol. São Paulo, Livraria Nobel.