

**Apostila  
para  
Eletrônica**

**ELETRÔNICA  
DIGITAL I**

# ÍNDICE

<b>1 - SISTEMA NUMÉRICO</b> .....	<b>3</b>
<b>1.1 - SISTEMA BINÁRIO</b> .....	<b>3</b>
Conversão Decimal para Binário.....	4
Conversão Binário para Decimal.....	5
<b>1.2 - SISTEMA OCTAL</b> .....	<b>6</b>
Conversão Decimal para Octal.....	6
Conversão Octal para Decimal.....	7
Conversão Octal em Binário.....	8
Conversão Binário em Octal.....	8
<b>1.3 - SISTEMA HEXADECIMAL</b> .....	<b>9</b>
Conversão Hexadecimal em Decimal.....	9
Conversão Decimal em Hexadecimal.....	11
Conversão Hexadecimal em Binário.....	12
Conversão Binário em Hexadecimal.....	13
<b>2 - ARITIMÉTICA BINÁRIA</b> .....	<b>14</b>
<b>2.1 - ADIÇÃO BINÁRIA</b> .....	<b>14</b>
<b>2.2 - SUBTRAÇÃO BINÁRIA</b> .....	<b>15</b>
<b>2.3 - COMPLEMENTO DE 1 (C – 1)</b> .....	<b>16</b>
<b>2.4 - MULTIPLICAÇÃO BINÁRIA</b> .....	<b>17</b>
<b>2.5 - DIVISÃO BINÁRIA</b> .....	<b>18</b>
<b>3 - PORTAS LÓGICAS</b> .....	<b>19</b>
<b>3.1 - CONCEITO DE VARIÁVEL BOOLEANA</b> .....	<b>20</b>
<b>3.2 - ELEMENTOS DE ÁLGEBRA BOOLEANA</b> .....	<b>20</b>
<b>3.3 - FUNÇÕES BOOLEANAS (PORTAS)</b> .....	<b>21</b>
3.3.1 - Função E (AND) - Porta E (AND).....	21
3.3.2 - Porta OU (OR).....	22
3.3.3 - Porta Inversora.....	23
3.3.4 - Porta NE (NAND).....	24
3.3.5 - Porta NOU (NOR).....	24
3.3.6 - Função OU Exclusivo - Porta OU Exclusivo (XOU).....	25
3.3.7 - Função NOU Exclusivo - Porta NOU Exclusivo (XNOU).....	27
<b>3.4 - OBTENDO A EXPRESSÃO BOOLEANA DO CIRCUITO</b> .....	<b>28</b>
<b>3.5 - OBTENDO A TABELA VERDADE DO CIRCUITO</b> .....	<b>29</b>
<b>4 - MAPA DE KARNAUGH</b> .....	<b>31</b>
<b>4.1 - MAPA COM 02 VARIÁVEIS</b> .....	<b>31</b>
<b>4.2 - MAPA COM 03 VARIÁVEIS</b> .....	<b>33</b>
<b>5 - FLIP–FLOP</b> .....	<b>35</b>
<b>5.1 - TIPO RS</b> .....	<b>35</b>
<b>5.2 - TIPO RS SÍNCRONO</b> .....	<b>36</b>

# 1 - SISTEMA NUMÉRICO

O sistema de numeração decimal é conhecido por todos nós. O sistema decimal tem uma característica de valor de posição, também é chamado de sistema de base 10, porque tem 10 símbolos diferentes - 0 à 9.

Os números binários (base 2) são extensamente usados em circuitos digitais.

Todos os sistemas de numeração (binário, octal, hexadecimal e decimal), podem ser usados para contagem, porque possuem a característica de valor de posição.

## 1.1 - SISTEMA BINÁRIO

O sistema de numeração binário usa apenas 2 símbolos - 0 e 1. Cada dígito binário é chamado de **Bit**, que é a menor unidade digital; é a presença ou ausência de corrente elétrica em um condutor. **Bit 0** = 0Vdc **Bit 1** = 5Vdc

A tabela abaixo mostra a comparação de uma contagem decimal com uma contagem binária:

Contagem Decimal →	Contagem binária				
	16	8	4	2	1
0					0
1					1
2				1	0
3				1	1
4			1	0	0
5			1	0	1
6			1	1	0
7			1	1	1
8		1	0	0	0
9		1	0	0	1
10		1	0	1	0
11		1	0	1	1
12		1	1	0	0
13		1	1	0	1

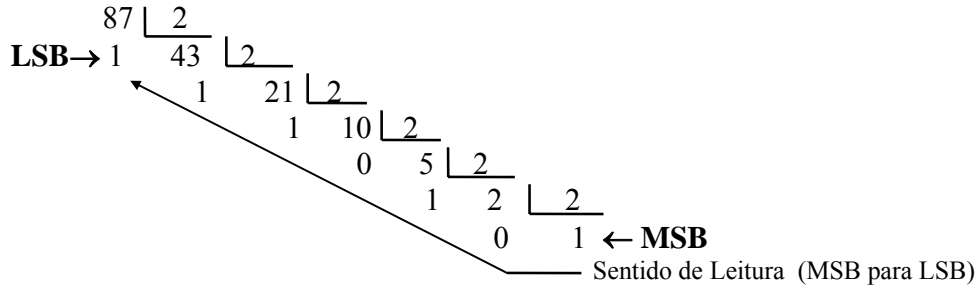
Na tabela acima podemos observar que o bit menos significativo (LSB = Least Significant Bit) está na coluna do 1.

Se o bit 1 aparecer na coluna do 1, um valor peso 1 é acrescentado na contagem binária. Se o bit 1 aparecer na coluna do 2, um valor peso 2 é acrescentado na contagem binária. Se o bit 1 aparecer na coluna do 4, um valor peso 4 é acrescentado na contagem binária. Se o bit 1 aparecer na coluna do 8, um valor peso 8 é acrescentado na contagem binária. Se o bit 1 aparecer na coluna do 16, um valor peso 16 é acrescentado na contagem binária, e assim por diante.

MSB (Most Significant Bit) = Bit mais significativo.

## Conversão Decimal para Binário

Para convertermos um número decimal para binário, basta dividi-lo por 2; o resultado dividir por 2; o novo resultado dividir por 2, e assim por diante até que o resultado seja igual à 1. Os restos serão 0 ou 1, que formarão o n° binário conforme exemplos abaixo.

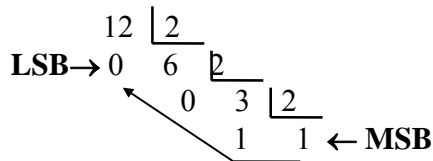


Portanto  $87_{10} = 1010111_2$

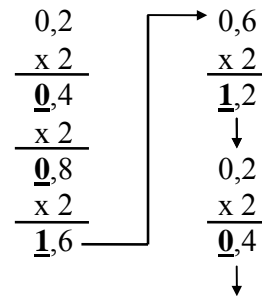
No exemplo abaixo convertemos o número decimal 12,2 para binário:

1° - Converter o n° inteiro dividindo por 2,

2° - Converter a fração multiplicando por 2. Do resultado o n° inteiro é extraído e colocado no resultado binário, continuando a multiplicação apenas com a fração.



Valor binário inteiro = 1100,



Valor binário fracionário = ,00110...

Portanto  $12,2_{10} = 1100,00110_2$

EXERCÍCIO 1 - Converter os números decimais abaixo para binário:

a)  $50 =$

b)  $27,32 =$

### Conversão Binário para Decimal

Para a conversão de um número Binário para Decimal, faz-se uso de uma tabela onde determinamos o valor (peso de posição) de cada **bit** (dígito binário).

128	64	32	16	8	4	2	1,	0,5	0,25	0,125	0,0625	0,0312
				1	0	0	0,					
						1	1,					
					1	0	1,	1	1			

EX: -  $1000_2 = 0 \times 1 + 0 \times 2 + 0 \times 4 + 1 \times 8$   
 $0 + 0 + 0 + 8 = 8_{10}$

-  $11_2 = 1 \times 1 + 1 \times 2$   
 $1 + 2 = 3_{10}$

-  $101,11_2 = 1 \times 1 + 0 \times 2 + 1 \times 4 + 1 \times 0,5 + 1 \times 0,25$   
 $1 + 0 + 4 + 0,5 + 0,25 = 5,75_{10}$

EXERCÍCIO 2 - Converter os números binários abaixo para decimal:

a)  $11101_2 = \underline{\hspace{2cm}}_{10}$

b)  $101010,1011_2 = \underline{\hspace{2cm}}_{10}$

c)  $10001101111_2 = \underline{\hspace{2cm}}_{10}$

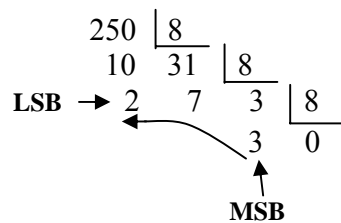
## 1.2 - SISTEMA OCTAL

O sistema de numeração Octal é constituído por apenas 8 símbolos: 0 a 7. Cada dígito octal é chamado de **Bit Octal**. Esse sistema numérico é pouco usado.

### Conversão Decimal para Octal

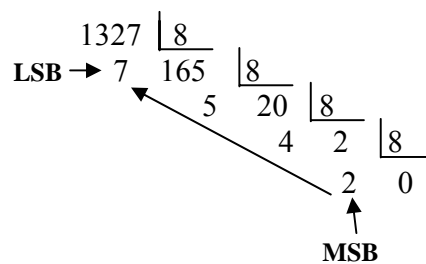
Para converter um número decimal em número octal, divide-se esse número inteiro por 8 até obter 0 no cociente.

Ex.1 :  $250_{10}$  p/ octal



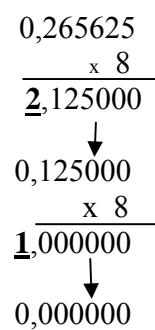
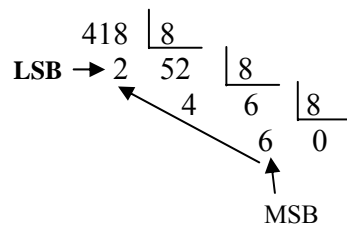
Portanto:  $250_{10} = 372_8$

Ex.2 :  $1327_{10}$  p/ octal



Portanto:  $1327_{10} = 2457_8$

Ex.3 :  $418,265625_{10}$  p/ octal



Portanto:  $418,265625_{10} = 642,21_8$

EXERCÍCIO 3 - Converte os números decimais abaixo para o sistema numérico octal:

a)  $499_{10} = \underline{\hspace{2cm}}_8$

b)  $1342,25_{10} = \underline{\hspace{2cm}}_8$

c)  $1,40625_{10} = \underline{\hspace{2cm}}_8$

### Conversão Octal para Decimal

A conversão de números octais para o sistema numérico Decimal, faz-se uso de uma tabela semelhante à conversão de binário p/ decimal:

...	<b>32768</b>	<b>4096</b>	<b>512</b>	<b>64</b>	<b>8</b>	<b>1,</b>	<b>0,125</b>	<b>0,015625</b>	<b>0,001953125</b>	...
			1	5	6	2,				
				1	2	3,				
						1,	3	2		

EX 1:  $1562_8 = 2 \times 1 + 6 \times 8 + 5 \times 64 + 1 \times 512$   
 $2 + 48 + 320 + 512 = 882_{10}$

EX 2:  $123_8 = 3 \times 1 + 2 \times 8 + 1 \times 64$   
 $3 + 16 + 64 = 83_{10}$

EX 3:  $1,32_8 = 1 \times 1 + 3 \times 0,125 + 2 \times 0,015625$   
 $1 + 0,375 + 0,03125 = 1,40625_{10}$

EXERCÍCIO 4 - Converter em Decimal os números Octais abaixo:

a)  $2457_8 = \underline{\hspace{2cm}}_{10}$

b)  $642,21_8 = \underline{\hspace{2cm}}_{10}$

### Conversão Octal em Binário

Para convertermos diretamente um número Octal em Binário, é importante sabermos que cada dígito (Bit) octal corresponde à 3 dígitos (Bits) binários, conforme exemplo abaixo

$$730_8 = \underline{\underline{111011000}}_2$$

7	3	0	8
↓	↓	↓	↓
111	011	000	2

EXERCÍCIO 5 - Converter em Binário os números Octais abaixo:

a)  $1235_8 = \underline{\hspace{2cm}}_2$

b)  $601,27_8 = \underline{\hspace{2cm}}_2$

### Conversão Binário em Octal

Sabendo que cada 3 dígitos (Bits) Binários corresponde à 1 dígito (Bit) Octal, separamos o número binário 3 em 3 Bits, partindo da virgula para a direita e para esquerda, completando com zeros, quando necessário para completar os 3 dígitos.

$$\text{Ex: } 1011100110_2 = \underline{\underline{1346}}_8$$

001	011	100	110	2
↓	↓	↓	↓	↓
1	3	4	6	8

EXERCÍCIO 6 - Converter em Octal os números Binários abaixo:

a)  $110110111101,101_2 = \underline{\hspace{2cm}}_8$

b)  $1010100010110,1_2 = \underline{\hspace{2cm}}_8$



### 1.3 - SISTEMA HEXADECIMAL

O sistema numérico Hexadecimal tem raiz (base) 16. Ele é constituído pelos símbolos (dígitos ou Bit Hexad.) de **0 a 9**, **A**, **B**, **C**, **D**, **E** e **F** conforme mostra a coluna hexadecimal da tabela abaixo.

A letra **A** significa uma contagem de 10 ; **B** significa uma contagem de 11 ; **C** uma contagem de 12; **D** de 13 ; **E** de 14 e **F** uma contagem de 15.

A vantagem do sistema hexadecimal é sua utilização em programação de baixo nível pela facilidade em converter diretamente em um número binário de 4 Bit's.

Na tabela abaixo é mostrado a equivalência do sistema Decimal , Binário e Hexadecimal.

DECIMAL	HEXADECIMAL	BINÁRIO
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

#### Conversão Hexadecimal em Decimal

Para a conversão de hexadecimal em decimal, se usa o sistema de tabela como mostra os exemplos abaixo:

...	1048576	65536	4096	256	16	1,	0,0625	0,00390625	0,00024414	...
				2	B	6,				
				A	3	F,	C			

$$2B6_{16} = \underline{694}_{10}$$

$$\begin{array}{r} 2 \times 256 + B(11) \times 16 + 6 \times 1 \\ 512 + 176 + 6 = 694_{10} \end{array}$$

$$A3F,C_{16} = \underline{2623,75}_{10}$$

$$\begin{array}{r} A(10) \times 256 + 3 \times 16 + F(15) \times 1 + C(12) \times 0,0625 \\ 2560 + 48 + 15 + 0,75 = 2623,75_{10} \end{array}$$

EXERCÍCIO 7 - Converter em Decimal os números Hexadecimais abaixo:

a)  $A2_{16} = \underline{\hspace{2cm}}_{10}$

b)  $B2C,D_{16} = \underline{\hspace{2cm}}_{10}$

c)  $3C2,F_{16} = \underline{\hspace{2cm}}_{10}$

## Conversão Decimal em Hexadecimal

Para converter um número Decimal em um número hexadecimal, se divide esse número decimal por 16 onde os restos formam o número hexad. conforme exemplo:

$$\text{EX 1: } 45 = \underline{\quad 2D \quad}_{16}$$

$$\begin{array}{r} 45 \quad | \quad 16 \\ \hline \text{LSB} \rightarrow 13 \quad 2 \\ \leftarrow \quad \uparrow \\ \text{MSB} \quad 0 \end{array} \quad | \quad \begin{array}{r} 16 \\ \hline 0 \end{array}$$

$$13_{10} = D_{16}$$

$$\text{Portanto: } 45_{10} = 2D_{16}$$

$$\text{EX 2: } 250,25 = \underline{\quad FA,4 \quad}_{16}$$

$$\begin{array}{r} 250 \quad | \quad 16 \\ \hline \text{LSB} \rightarrow 10 \quad 15 \\ \leftarrow \quad \uparrow \\ \text{MSB} \quad 0 \end{array} \quad | \quad \begin{array}{r} 16 \\ \hline 0 \end{array}$$

$$\text{OBS: } 10_{10} = A_{16}$$

$$15_{10} = F_{16}$$

$$\begin{array}{r} 0,25 \\ \times 16 \\ \hline 4,00 \\ \downarrow \end{array}$$

$$0,00$$

$$\text{Portanto: } 250,25_{10} = FA,4_{16}$$

EXERCÍCIO 8 - Converter em Hexadecimal os números Decimais abaixo:

a)  $57_{10} = \underline{\hspace{2cm}}_{16}$

b)  $132_{10} = \underline{\hspace{2cm}}_{16}$

c)  $540,20_{10} = \underline{\hspace{2cm}}_{16}$

### Conversão Hexadecimal em Binário

Para convertermos diretamente um número Hexad. em Binário, é importante sabermos que cada dígito (Bit) Hexad. corresponde à 4 dígitos (Bits) binários, conforme exemplo abaixo:

$$\text{EX 1: } 3\text{B}9_{16} = \underline{\underline{001110111001}}_2$$

3	B	9 <sub>16</sub>
↓	↓	↓
0011	1011	1001 <sub>2</sub>

$$\text{EX 2: } 47,\text{FE}_{16} = \underline{\underline{01000111,11111110}}_2$$

4	7	,	F	E <sub>16</sub>
↓	↓	,	↓	↓
0100	0111	,	1111	1110 <sub>2</sub>

EXERCÍCIO 9 - Converter em Binário os números Hexadecimais abaixo:

a)  $5\text{F}_{16} = \underline{\hspace{2cm}}_2$

b)  $\text{C}2\text{B},\text{B}_{16} = \underline{\hspace{2cm}}_2$

c)  $\text{DE},\text{A}3_{16} = \underline{\hspace{2cm}}_2$

## Conversão Binário em Hexadecimal

Sabendo que cada 4 dígitos (Bits) Binários corresponde à 1 dígito (Bit) Hexad., separamos o número binário 4 em 4 Bits, partindo da virgula para a direita e para esquerda, completando com zeros, quando necessário para completar os 4 dígitos.

$$\text{Ex 1: } 101010000101_2 = \underline{\quad A85 \quad}_{16}$$

1010	1000	0101 <sub>2</sub>
↓	↓	↓
10 (A)	8	5 <sub>16</sub>

$$\text{Ex 2: } 10010,011011_2 = \underline{\quad 12,6C \quad}_{16}$$

0001	0010 ,	0110	1100 <sub>2</sub>
↓	↓	↓	↓
1	2 ,	6	12 (C) <sub>16</sub>

EXERCÍCIO 10 - Converter em Hexadecimal os números Binários abaixo:

a)  $1010111_2 = \underline{\hspace{2cm}}_{16}$

b)  $11111_2 = \underline{\hspace{2cm}}_{16}$

c)  $10001,1101_2 = \underline{\hspace{2cm}}_{16}$

**OBS:** Para as conversões abaixo não é possível fazer diretamente, é necessário passar p/ outro sistema numérico:

- Hexad. em Octal
- Octal em Hexad.

## 2 - ARITIMÉTICA BINÁRIA

A aritimética binária é a execução das operações matemáticas com números binários. Podemos executar vários tipos de operações, onde iremos estudar as principais.

### 2.1 - ADIÇÃO BINÁRIA

Como se sabe, no sistema decimal, quando se quer somar 9 com 1, o resultado é sempre 0 e vai 1, ou seja, é igual a 10. No sistema binário, ocorre o mesmo quando se soma 1 com 1. O resultado é 0 e vai 1, ou seja 10. As regras para a adição binária são as seguintes:

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	0 e vai 1 ( este resultado é escrito 10, mas diz-se “ um zero”.

Exemplos:

a)  $100 + 10 =$

binário	decimal
100	4
+ 010	+ 2
<hr/>	
110	6

b)  $1010 + 111 =$

binário	decimal
1010	10
+ 0111	+ 7
<hr/>	
10001	17

EXERCÍCIO 11 – Efetue as adições binárias abaixo:

a)  $101 + 11 =$

b)  $1011011 + 1011010 =$

c)  $11,01 + 101,11 =$

d)  $11 + 1010 + 11 + 110 =$

e)  $111,11 + 11,01 + 1011,10 + 101,11 =$

## 2.2 - SUBTRAÇÃO BINÁRIA

No sistema decimal, quando se quer subtrair 20 menos 9, empresta-se 1 para o 0 do minuendo e o resultado é 1, portanto esse 1 que foi emprestado devolve para o subtraendo na frente do 9 para que possa ser subtraído de 2 onde o resultado será 1. Então a diferença ficaria 1 e 1, ou seja, é igual a 11.

$$\begin{array}{r} 20 \\ - 9 \\ \hline 11 \end{array}$$

minuendo  
subtraendo  
diferença

No sistema binário, ocorre o mesmo quando subtrai 1 de 0. O resultado é 1 e empresta 1. As regras para a subtração binária são as seguintes:

$$\begin{array}{r} 0 - 0 = 0 \\ 0 - 1 = 1 \text{ e empresta } 1 \\ 1 - 0 = 1 \\ 1 - 1 = 0 \end{array}$$

Exemplos:

a)  $11 - 1 =$

binário	decimal
11	3
- 01	- 1
<hr/>	
10	2

b)  $10 - 1 =$

binário	decimal
10	2
- 01	- 1
<hr/>	
01	1

EXERCÍCIO 12 – Efetue as subtrações abaixo:

a)  $1001 - 101 =$

b)  $10000 - 11 =$

c)  $110,01 - 100,10 =$

d)  $110000 - 11101 =$

### 2.3 - COMPLEMENTO DE 1 (C – 1)

Complemento de 1 é a outra maneira de se calcular a subtração binária, onde se faz a soma dos números binários, invertendo-se o subtraendo (trocando onde for 0 por 1 e onde for 1 por 0).

Quando o complemento dar 1, soma-se 1 e o resultado será positivo, conforme exemplo a. Quando o complemento dar 0, acha-se novamente o complemento (invertendo-se o número binário) e o resultado será negativo, conforme exemplo b.

Exemplos:

a)  $1001 - 101 =$

binário	decimal
1001	9
- 0101	- 5
<hr/>	
	4
1001	
+ 1010	(trocar 0/1)
<hr/>	
<u>1</u> 0011	
0011	
+ 1	soma o complemento <u>1</u>
<hr/>	
0100	resultado

b)  $101 - 110 =$

binário	decimal
101	5
- 110	- 6
<hr/>	
	- 1
101	
+ 001	(trocar 0/1)
<hr/>	
<u>0</u> 110	
1. 001	inverte - resultado

EXERCÍCIO 13 – Efetue as subtrações abaixo:

a)  $101110 - 110110 =$

b)  $100111 - 110111 =$

c)  $110 - 11011 =$

d)  $11010110 - 11010010 =$



## 2.4 - MULTIPLICAÇÃO BINÁRIA

A multiplicação binária é exatamente igual a multiplicação decimal. No exemplo abaixo os bits de sinais não foram usados. Observe que o primeiro multiplicando (10011) é deslocado para a esquerda com relação ao segundo multiplicando (1101). Depois disso, os resultados parciais são somados para obter o produto final.

Exemplos:

a)  $10011 \times 1101 =$

binário	decimal
10011	19
x 01101	x 13
<hr/>	
10011	57
00000	<u>19</u>
10011	247
10011	
<hr/>	
11110111	

EXERCÍCIO 14 – Efetue as multiplicações abaixo:

a)  $1101 \times 1010 =$

b)  $10011010110 \times 1111 =$

c)  $10101 \times 110 =$

d)  $1001 \times 1011 =$

## 2.5 - DIVISÃO BINÁRIA

A divisão binária é igual à divisão decimal. A divisão de números com sinal é feita de maneira idêntica à multiplicação, onde os números negativos são transformados em positivos através do complemento.

Exemplos:

a)  $110100 \times 100 =$

binário		decimal
110100	$\underline{100}$	52 $\underline{4}$
0101	1101	12 13
00100		0
000		

EXERCÍCIO 15 – Efetue as divisões abaixo:

a)  $101110,11 : 1010 =$

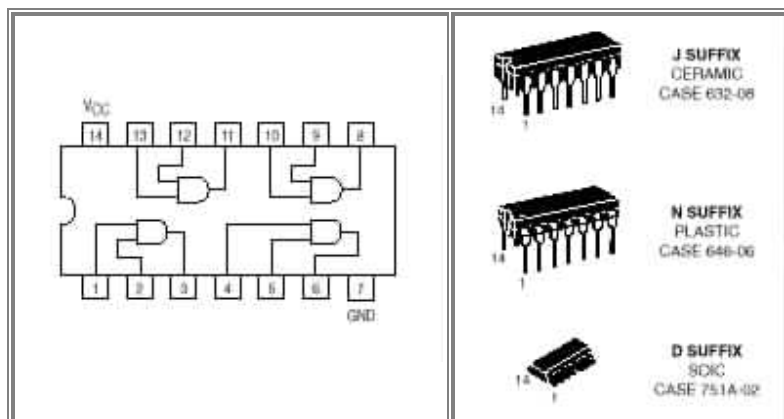
b)  $110,101 : 101 =$

c)  $101110,11 : 11 =$

d)  $1101 : 10,1 =$

### 3 - PORTAS LÓGICAS

Quando se deseja construir um circuito lógico (ou digital) relativamente simples, usa-se uma placa de circuito impresso com soquetes sobre os quais insere-se um circuito integrado (CI) digital. A maioria dos CI's já são padronizados, e os mais comuns pertencem à série denominada 7400. Os mais simples utilizam a tecnologia de Integração em Pequena Escala (SSI - Small Scale Integration). Como um exemplo, veja o CI 7408, que contém quatro portas AND, cuja relação de suas entradas e saídas com os pinos de seu encapsulamento pode ser vista no diagrama.



Além de seu diagrama, são mostradas também as três formas de encapsulamento desse chip, sendo que duas delas exibem tipos diferentes de material e a outra é diferente na forma com a qual deve ser fixada na placa de circuito impresso. Como indica seu diagrama, esse CI possui 14 pinos (olhando-o de cima, a numeração de tais pinos é importante, pois o pino 1 sempre começa do mesmo lado; isso é suficiente para você efetuar as ligações elétricas necessárias para a montagem de seu circuito lógico), sendo que o pino 7 é o terra (Ground = 0 V; V indica volts) e o pino 14 é a tensão da fonte de energia ( $V_{CC} = 5\text{ V}$ ). A maioria dos CI's SSI, que serão utilizados em nosso curso, possui 14 ou 16 pinos.

### 3.1 - CONCEITO DE VARIÁVEL BOOLEANA

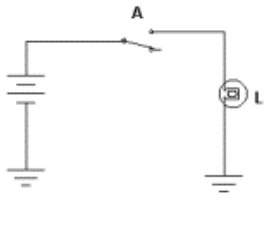
Chamamos de variável Booleana a uma variável que pode assumir só duas condições (dois valores).

Um exemplo de variável Booleana é uma chave, que só pode estar aberta ou fechada, não existe outra condição. Outro exemplo é uma lâmpada, que só pode estar acesa ou apagada.

Em eletrônica digital costumamos associar a uma variável Booleana os símbolos "0" e "1" aos estados que a variável pode assumir. Desta forma lâmpada acesa poderia ser "1" e conseqüentemente apagada "0", mas poderia ser o contrario depende da convenção adotada. Uma variável Booleana pode ser dependente de outras variáveis Booleanas. Por exemplo, em resposta à condição de uma chave (variável A) a qual pode estar aberta ou fechada podemos ter a condição de uma lâmpada (variável L) acesa ou apagada.

Na Fig. 01 podemos convencionar que chave aberta  $A=0$ , a chave fechada, portanto será  $A=1$  da mesma forma teremos para lâmpada apagada  $L=0$  e acesa  $L=1$ . Para caracterizar o comportamento lógico estabelecemos o que chamamos de tabela verdade do circuito.

Expressão Booleana:  $L=A$

	<b>A</b>	<b>L</b>	<b>A</b>	<b>L</b>
	ABERTA	APAGADA	0	0
	FECHADA	ACESA	1	1
( a )	( b )			
Fig01: ( a ) Circuito com chaves ( b ) Tabela Verdade				

### 3.2 - ELEMENTOS DE ÁLGEBRA BOOLEANA

Em álgebra Booleana assim como na álgebra comum, as letras são usadas para representar as variáveis. Na Álgebra Booleana usamos letras maiúsculas para representar uma variável Booleana. Uma variável Booleana só pode ter duas condições às quais associaremos os símbolos "0" ou "1". O símbolo = tem usualmente o significado de "é equivalente", isto é, se o lado direito da equação é 0, então o lado esquerdo também será 0. Desta forma a declaração:

$Y=A$  significa que Y é 1 se A é 1, Y é 0 se A for 0.

Na figura 01 usamos uma chave, que representa a variável A se relacionando com a variável L (lâmpada) pela expressão:

$L=A$  (observe isso na tabela verdade)

O símbolo Booleano com uma barra acima da variável significa a negação ou o complemento da variável.

Desta forma  $\overline{A}$  é lido como A barrado. Se  $A=1$  então  $\overline{A}$  será igual a 0 e vice-versa.

### 3.3 - FUNÇÕES BOOLEANAS (PORTAS)

Uma função Booleana relaciona duas ou mais variáveis Booleanas entre si através de uma expressão chamada de **Expressão Booleana**.

#### 3.3.1 - Função E (AND) - Porta E (AND)

Antigamente os circuitos lógicos eram feitos (implementados) com relés, hoje usamos portas lógicas em CI (Circuito Integrado) para realizar uma determinada lógica (determinada função). A seguir mostraremos as principais funções lógicas e as portas lógicas que realizam a lógica da função.

As duas chaves, A e B estão ligadas em série para ligar a lâmpada L.

A lógica existente é: “A lâmpada acenderá se chave A e a chave B estiverem fechadas na Fig02a”.

Dizemos que esta é uma lógica E (AND em inglês). A porta lógica correspondente é chamada porta E (AND) e cujo símbolo está representado na Fig03a.

A Expressão Booleana é:  $L = A \cdot B$  (lê-se A e B, mas por analogia com a operação multiplicação dizemos também A vezes B).

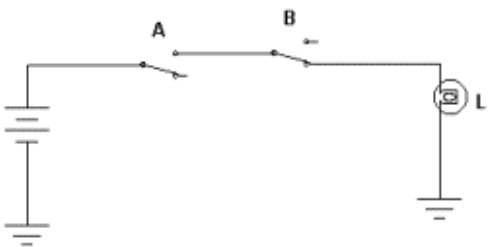
	<b>B</b>	<b>A</b>	<b>L</b>
	0 (ABERTA)	0 (ABERTA)	0 (APAGADA)
	0 (ABERTA)	1 (FECHADA)	0 (APAGADA)
	1 (FECHADA)	0 (ABERTA)	0 (APAGADA)
	1 (FECHADA)	1 (FECHADA)	1 (ACESA)
( a )	( b )		

Fig02: ( a ) circuito com chaves para lógica E ( b ) Tabela Verdade

A seguir os símbolos da porta E (AND) e a sua Tabela Verdade.

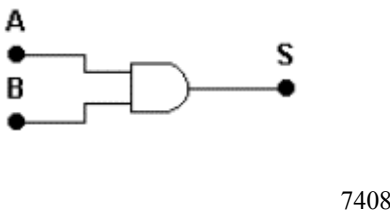
	<b>B</b>	<b>A</b>	<b>S</b>
	0	0	0
	0	1	0
	1	0	0
	1	1	1
( a )	( b )		

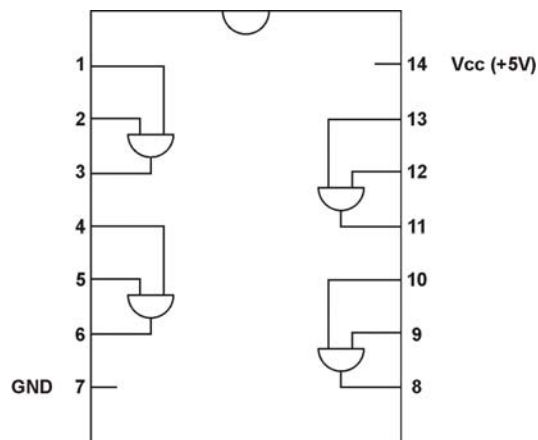
Fig03: ( a ) Porta E, símbolo ( b ) Tabela Verdade

**7408** – CI com 4 portas de 2 entradas.

**7411** – CI com 3 portas de 3 entradas.

**7421** – CI com 2 portas de 4 entradas

Abaixo está a figura do CI 7408 com a relação das entradas, saídas e os pinos do encapsulamento:



### 3.3.2 - Porta OU (OR )

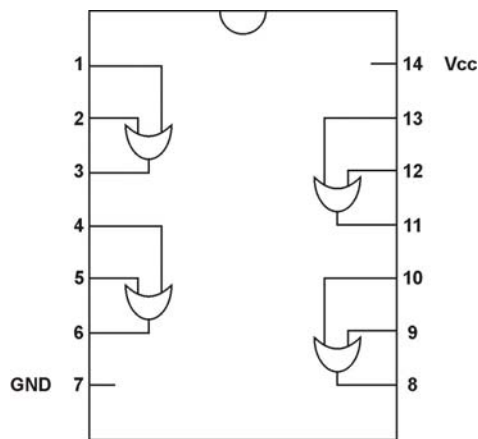
A função OU (OR) tem a seguinte lógica se pensarmos em termos de chaves (não esqueça chave aberta, “0“, fechada, “1“): “A lâmpada acenderá se a Chave A OU a chave B estiverem fechadas na Fig04a“.

Expressão Booleana:  $L = A+B$  (Lemos A ou B, mas por analogia com a operação soma dizemos A mais B).

A figura 04b mostra o símbolo e a figura 04c a tabela verdade.

		<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>S</th> </tr> </thead> <tbody> <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> </tbody> </table>	B	A	S	0	0	0	0	1	1	1	0	1	1	1	1
B	A	S															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
( a )	( b )	( c )															
Fig04: Função OU ( a ) circuito com chaves ( b ) Porta OU símbolo ( c ) Tabela verdade																	

Abaixo está a figura do CI 7432 com a relação das entradas, saídas e os pinos do encapsulamento:



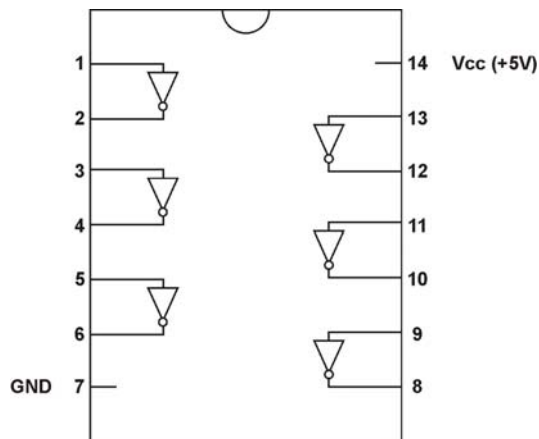
### 3.3.3 - Porta Inversora

A função Não (NOT) ou função Inversora dá uma saída que é o complemento (inverso) da entrada.

Expressão Booleana:  $L = \bar{A}$

		<table border="1"> <thead> <tr> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	S	0	1	1	0	<table border="1"> <thead> <tr> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	S	0	1	1	0
A	S														
0	1														
1	0														
A	S														
0	1														
1	0														
( a )	( b )	( c )													
Fig05: Função inversora - Circuito com chave ( a ) - Porta inversora ( b ) - Tabela verdade ( c )															

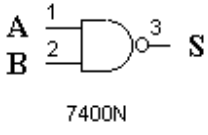
Abaixo está a figura do CI 7404 com a relação das entradas, saídas e os pinos do encapsulamento:



### 3.3.4 - Porta NE (NAND)

A lógica desta função corresponde à negação da função E (AND). A Fig06 dá o símbolo da porta lógica e a sua tabela verdade.

Expressão Booleana:  $L = \overline{AB}$

 <p>(a)</p>	<b>B</b>	<b>A</b>	<b>S</b>
	0	0	1
	0	1	1
	1	0	1
	1	1	0
(a)		(b)	
Fig06: (a) Porta NE - Símbolo (b) Tabela verdade			

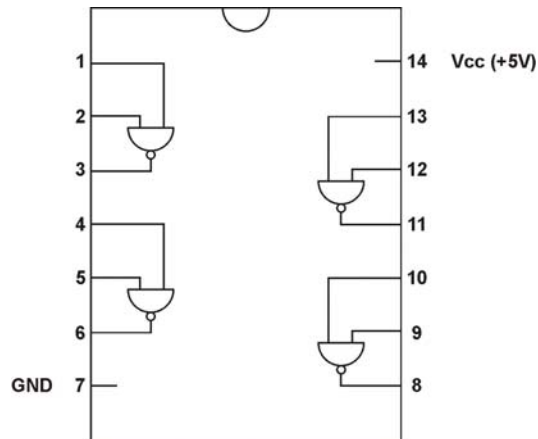
**7400** – CI com 4 portas de 2 entradas.

**7410** – CI com 3 portas de 3 entradas.

**7420** – CI com 2 portas de 4 entradas

**7430** – CI com 1 portas de 8 entradas.

Abaixo está a figura do CI 7400 com a relação das entradas, saídas e os pinos do encapsulamento:

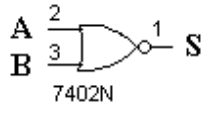




### 3.3.5 - Porta NOU (NOR)

A lógica desta função corresponde à negação da função OU ( NOR ). A Fig07 dá o símbolo da porta lógica e a sua tabela verdade.

Expressão Booleana:  $L = \overline{A+B}$

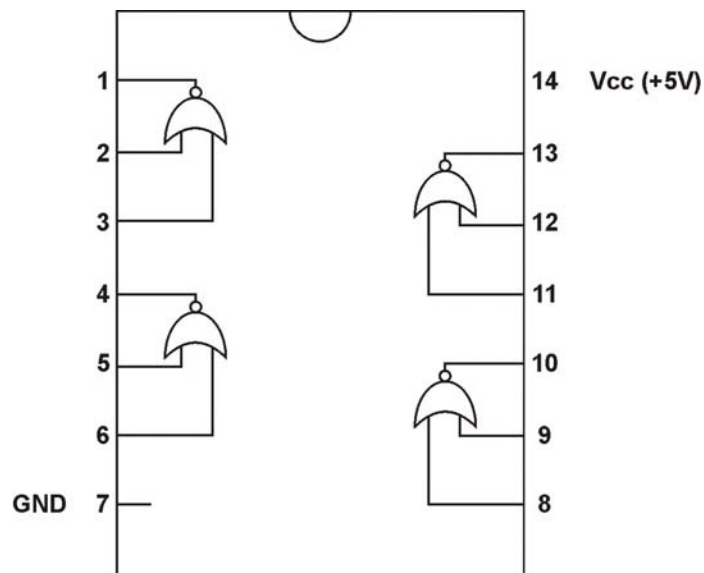
	<b>B</b>	<b>A</b>	<b>S</b>
	0	0	1
	0	1	0
	1	0	0
	1	1	0
( a )	( b )		
Fig07: Porta NOU - ( a ) Símbolo ( b ) Tabela verdade			

**7402** – CI com 4 portas de 2 entradas.

**7427** – CI com 3 portas de 3 entradas.

**7425** – CI com 2 portas de 4 entradas

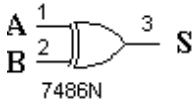
Abaixo está a figura do CI 7402 com a relação das entradas, saídas e os pinos do encapsulamento:



### 3.3.6 - Função OU Exclusivo - Porta OU Exclusivo (XOU)

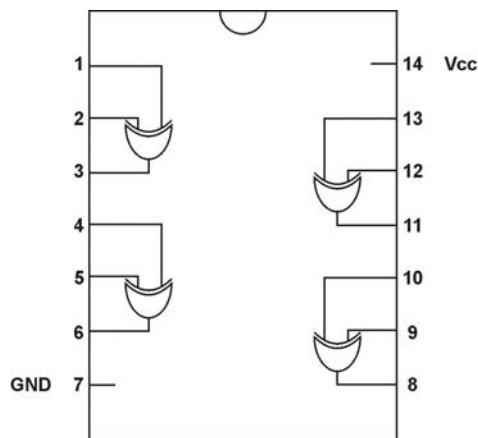
Essa função e a porta lógica correspondente formam juntamente com a coincidência outros circuitos básicos de sistemas digitais.

Expressão Booleana:  $Y = A \oplus B$

	<b>B</b>	<b>A</b>	<b>S</b>
	0	0	0
	0	1	1
	1	0	1
	1	1	0
( a )	( b )		
Fig08: Porta XOU - ( a ) Símbolo ( b ) Tabela verdade			

**OBS.:** Somente possuirá 1 na saída da porta XOU quando possuir um número ímpar de 1s nas entradas.

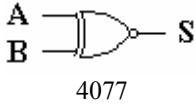
Abaixo está a figura do CI 7486 com a relação das entradas, saídas e os pinos do encapsulamento:



### 3.3.7 - Função NOU Exclusivo - Porta NOU Exclusivo (XNOU)

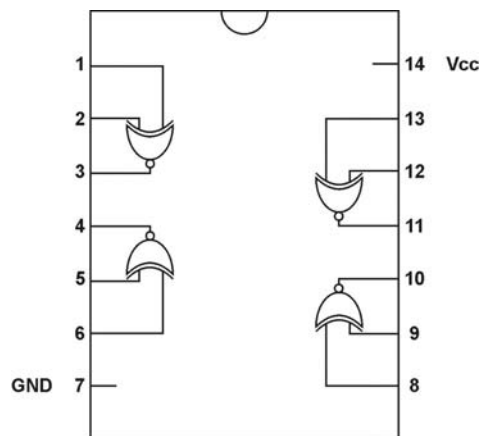
Observar que a relação entre a função OU Exclusivo e a NOU Exclusivo é:

Expressão Booleana:  $Y = A \oplus B$

 <p>( a )</p>	<b>B</b>	<b>A</b>	<b>S</b>
	0	0	1
	0	1	0
	1	0	0
	1	1	1
( b )			
Fig08: Porta XNOU - ( a ) Símbolo ( b ) Tabela verdade			

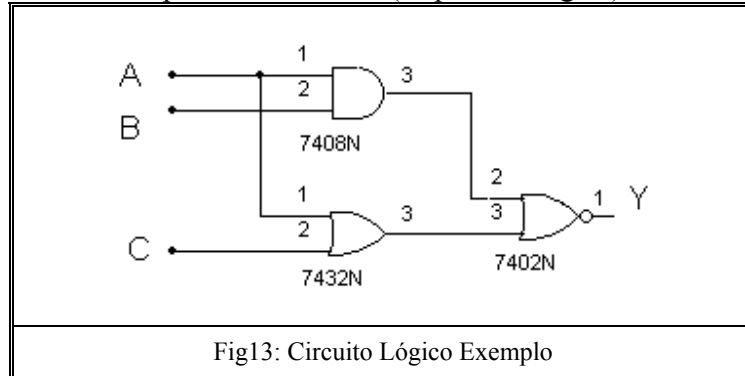
**OBS.:** Somente possuirá 1 na saída da porta XNOU quando possuir um número par de 1s ou números 0s nas entradas.

Abaixo está a figura do CI 4077 com a relação das entradas, saídas e os pinos do encapsulamento:



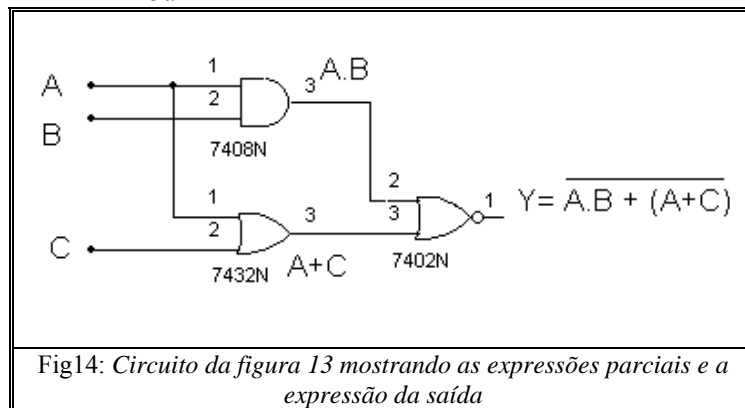
### 3.4 - OBTENDO A EXPRESSÃO BOOLEANA DO CIRCUITO

Podemos ter um problema inverso, isto é, o circuito pode ser especificado e precisamos obter a Tabela Verdade e a expressão Booleana (Expressão lógica).



Para obter a expressão da saída em função das entradas (expressão lógica ou Booleana), a partir das entradas devemos escrever a expressão da saída de cada porta lógica básica encontrada até chegarmos na saída. No exemplo, na saída da porta E de duas entradas temos  $A.B = X$ . Na saída da porta OU temos  $A+C = Z$ .  $X$  e  $Z$  são as entradas da porta NOR, cuja saída é a saída do circuito.

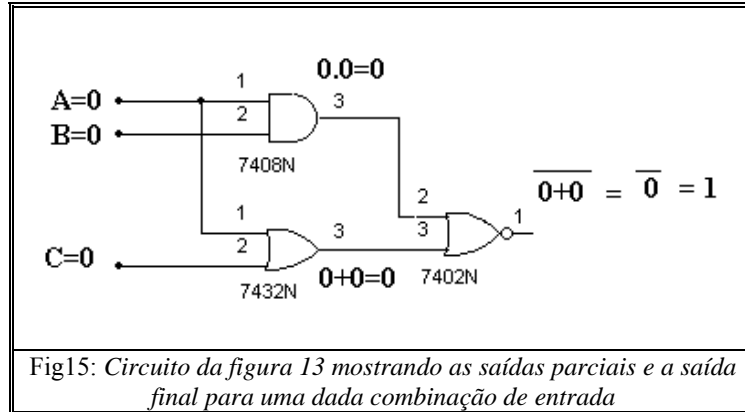
Portanto:  $Y = \overline{Z + X}$  ou  $Y = \overline{A+C + A.B}$



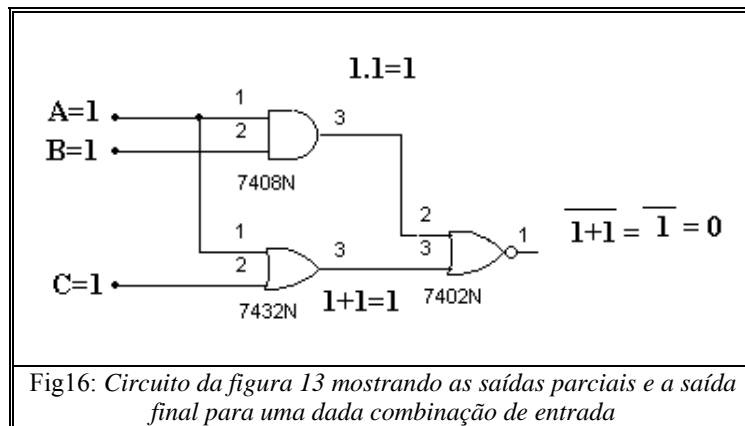
### 3.5 - OBTENDO A TABELA VERDADE DO CIRCUITO

Para obter a Tabela Verdade, existe dois caminhos, ou achar a expressão booleana do circuito e calcular linha a linha ou analisar no próprio circuito.

Veremos primeiramente pelo circuito. Deveremos obter a saída (Y) para todas as combinações de entrada. Por exemplo se  $A=B=C=0$  qual será o valor da saída? Veja figura a seguir para compreender isso:



O que acontece se  $A=B=C=1$  ? Veja figura a seguir para compreender isso:



EXERCÍCIO 16 – Complete a tabela verdade abaixo em função do circuito acima:

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

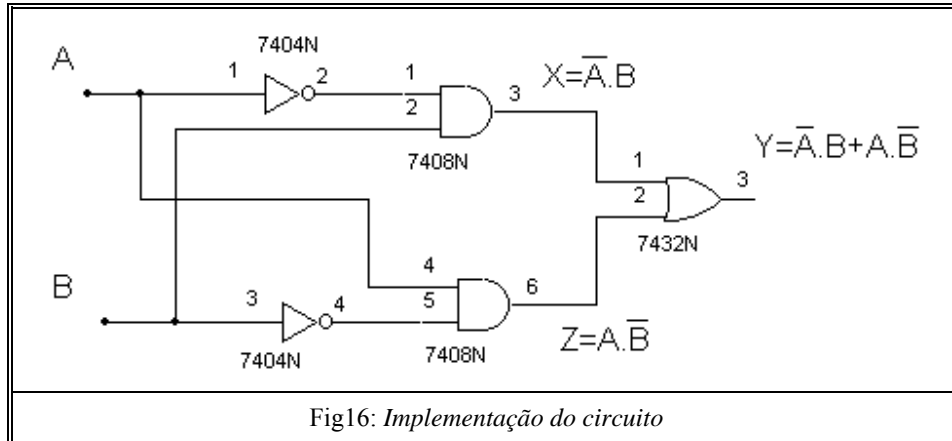
Veremos agora como obter o circuito através da expressão booleana, obtendo também a tabela verdade.

Primeiramente vamos obter o circuito.

Seja  $X = \bar{A}.B$  e  $Z = A.\bar{B}$  então  $Y = X + Z$

Precisamos de duas portas E, duas inversoras e uma OU, portanto 3 CI's diferentes.

Resultando circuito:



O processo para obter a Tabela Verdade já foi visto, então completaremos apenas com as respostas.

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

EXERCÍCIO 17 – Dada as funções abaixo, obter a tabela verdade e fazer o circuito usando portas lógicas.

a)  $Y = \bar{A}.\bar{B} + A.B$

b)  $Y = (\bar{A} + B) + A.\bar{B}.\bar{C}$

c)  $S = (A+B).C.(B+D)$

## 4 - MAPA DE KARNAUGH

É uma ferramenta poderosa usada para simplificar a expressão booleana de uma função. O mapa de Karnaugh auxilia na verificação de circuitos, se pode ser construído um outro circuito similar porém com número menor de portas e conseqüentemente de CI's. Consiste em colocar em um mapa os valores da saída da função para cada entrada.

### 4.1 - MAPA COM 02 VARIÁVEIS

Dada a expressão seguimos alguns passos para a simplificação desta expressão.

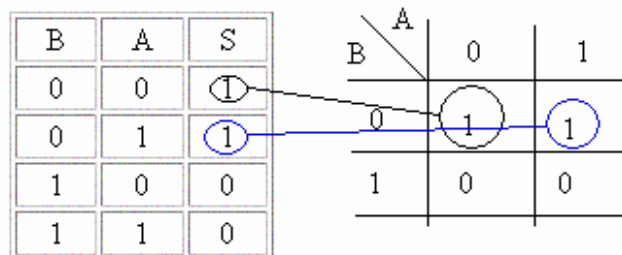
1º Monta-se uma tabela verdade sem se preocupar com a saída.

B	A	S (Y)
0	0	
0	1	
1	0	
1	1	

2º Substituir as variáveis da expressão com os valores da tabela verdade linha a linha. O resultado obtido coloca-se na coluna de saída da tabela verdade em sua respectiva linha.

B	A	S (Y)
0	0	1
0	1	1
1	0	0
1	1	0

3º Montar o mapa colocando o valor de saída obtido pelos cálculos, no ponto correspondente. Observe que para cada linha da tabela verdade corresponde a uma célula do mapa.



4º Uma vez montado o mapa temos que simplifica-lo. A simplificação consiste em agrupar células vizinhas. Podemos fazer laços em torno dos grupos adjacentes de somente um, dois quatro, oito, dezesseis, etc.

Os laços podem se sobrepor, onde devemos laçar o maior número de níveis alto (bits 1) possível, desde que estejam na vertical ou horizontal, jamais na diagonal. No exemplo acima as duas células com bit 1, são vizinhas e portanto podem ser enlaçadas, como visto na figura a seguir.

	A	
B \	0	1
0	1	1
1	0	0

Estes laços indicam quais linhas e colunas iremos utilizar. Neste caso iremos utilizar a 1ª linha e a 1ª e 2ª coluna.

Cada laço resultará em um produto das variáveis, neste produto aparecerão as variáveis que não mudam, no exemplo o complemento de B.

**a função não depende de A**

	A	
B \	0	1
0	1	1
1	0	0

$\bar{B}$  não muda

Logo a expressão da função é

$$S = \bar{B}$$

EXERCÍCIO 18 – Fornecido as tabelas verdades abaixo, simplificar utilizando o mapa de Karnaugh:

a)

B	A	S (Y)
0	0	0
0	1	0
1	0	0
1	1	1

b)

B	A	S (Y)
0	0	0
0	1	1
1	0	1
1	1	1



## 4.2 - MAPA COM 03 VARIÁVEIS

Para a simplificação de expressões booleanas com 03 variáveis usa-se o mesmo método de 02 variáveis.

1º Monta-se uma tabela verdade sem se preocupar com a saída.

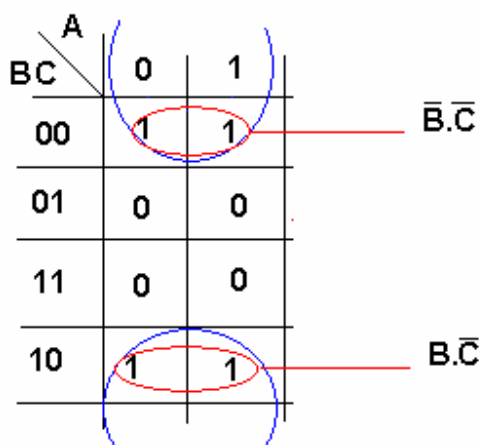
C	B	A	S (Y)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

2º Substituir as variáveis da expressão com os valores da tabela verdade linha a linha. O resultado obtido coloca-se na coluna de saída da tabela verdade em sua respectiva linha.

C	B	A	S (Y)
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

3º Montar o mapa colocando o valor de saída obtido pelos cálculos, no ponto correspondente. Observe que para cada linha da tabela verdade corresponde a uma célula do mapa.

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



Observe que as células de cima são vizinhas das duas de baixo pois ao passar de uma destas células para a de cima ou vice versa, só uma variável mudará.

Quando existem mais de dois grupos de células agrupadas, o resultados será uma operação de soma (OU).

$$S = \bar{B}.\bar{C} + B.\bar{C}$$

A expressão acima também pode ser representada desta forma:

$$S = B'.C' + B.C'$$

Aplicando a Algebra Booleana teremos:

$$S = B'.C' + B.C'$$

$$S = C'.(B' + B)$$

$$S = C'.1$$

$$S = C'$$

O mesmo resultado poderia ser obtido diretamente considerando um único laço envolvendo todos bits 1.

EXERCÍCIO 19 – Fornecido as tabelas verdades abaixo, simplificar utilizando o mapa de Karnaugh:

a)

C	B	A	S (Y)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

b)

C	B	A	S (Y)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

## 5 - FLIP-FLOP

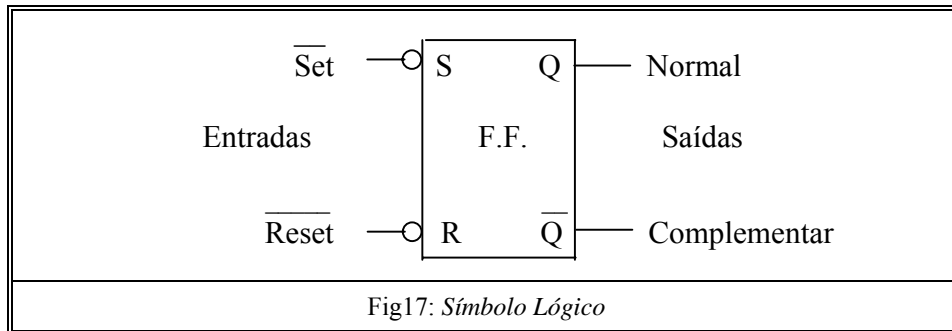
Nos circuitos combinacionais (portas lógicas), nota-se que p/ uma determinada combinação de valores atribuídos nas entradas, obtém-se um único sinal na saída.

Os Flip-Flops são característicos por possuírem uma realimentação da saída p/ a entrada. São chamados também de circuito de trava (Latch) ou multivibrador biestáveis.

Os biestáveis úteis podem ser montados através de portas lógicas ou adquiridos em forma de CI (Circuitos Integrado). Os biestáveis são interligados p/ formar circuitos lógicos seqüenciais p/ armazenamento, temporização, contagem e seqüencial.

### 5.1 - TIPO RS

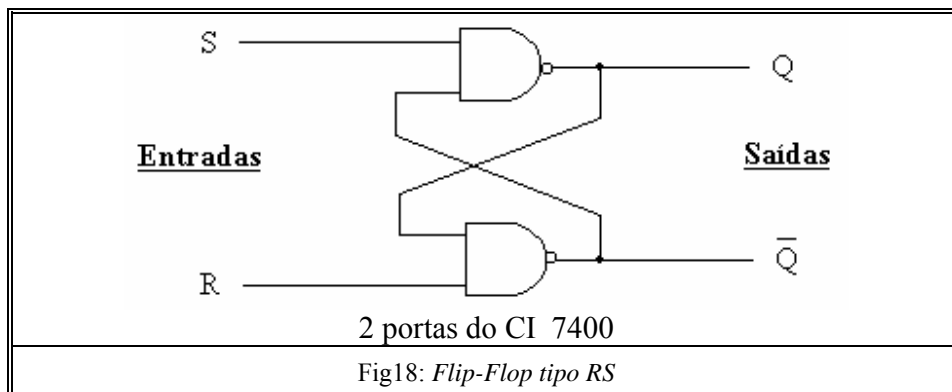
O circuito de trava (latch) do tipo RS é um dispositivo assíncrono, onde não opera simultaneamente com um dispositivo de temporização (clock).



O símbolo lógico (fig. 17) mostra duas entradas, designadas estabelecer (Set – S) e restabelecer (Reset – R). Sendo ativo em nível de tensão baixo (0 Vdc) indicado pelas inversoras nas entradas R e S.

A saída Q é considerada a saída Normal e a saída  $\bar{Q}$  (não Q) é o inverso da Q, ou seja, o seu complemento.

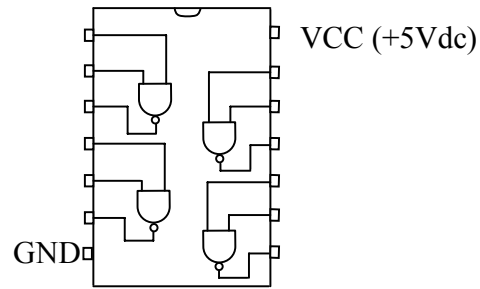
Um biestável RS está mostrado na fig. 18 a partir de duas portas NE. Uma tabela verdade define a operação do Flip-Flop (fig 19a) e a configuração interna do CI 7400 (fig 19b)



Modo de Operação	Entradas		Saídas	
	S	R	Q	$\bar{Q}$
Proibido	0	0	1	1
Set	0	1	1	0
Reset	1	0	0	1
Hold	1	1	ñ muda	

Tabela Verdade - FF RS

Fig 19a



CI 7400

Fig 19b

## 5.2 - TIPO RS SÍNCRONO

O biestável RS sincronizado acrescenta um aspecto síncrono, onde opera simultaneamente com um dispositivo de temporização (clock) como mostra a fig 20.

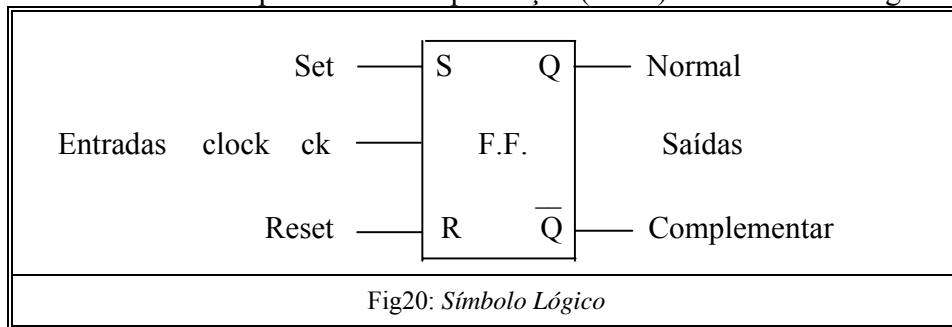


Fig20: Símbolo Lógico

Duas portas NE foram acrescentadas ao circuito de trava RS p/ formar o biestável RS sincronizado (fig 21a). Sua tabela verdade é mostrado na fig 21b

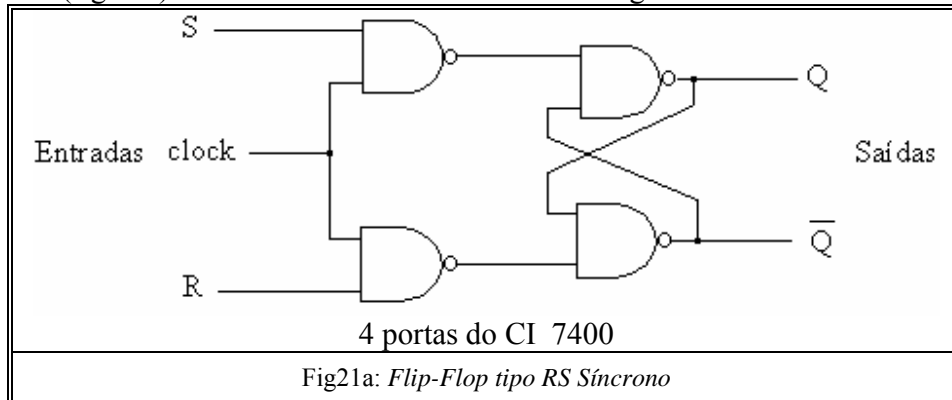


Fig21a: Flip-Flop tipo RS Síncrono

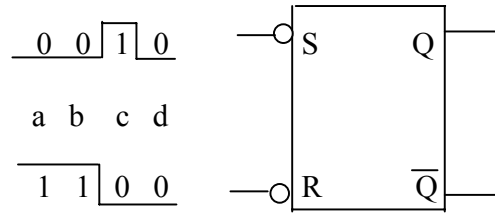
Modo de Operação	Entradas			Saídas	
	CK	S	R	Q	$\bar{Q}$
Hold		0	0	não muda	
Reset		0	1	0	1
Set		1	0	1	0
Proibido		1	1	1	1

Tabela Verdade - FF RS

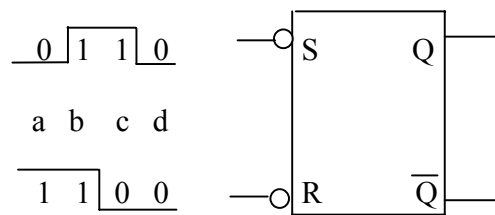
Fig 21b

EXERCÍCIO 20 – Fornecido os blocos abaixo, colocar os sinais e os bits das saídas normais e complemento para cada seqüência de pulsos nas entradas dos Flip-Flops.

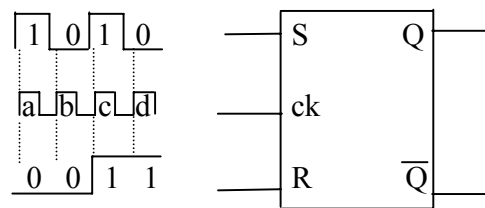
a)



b)



c)



d)

