

**Experimento 1**

**Objetivo:**

Estabelecer a tabela verdade para o gate básico **AND**.

Todo circuito lógico executa uma expressão Booleana e, por mais complexo que seja, é formado pela interligação de portas lógicas básicas. Podemos obter a expressão Booleana que é executada por um circuito lógico qualquer.

Nesta experiência será examinado o bloco (ou Gate ) básico **E**.

Expressão Lógica

$S = A.B = AB$  onde o ponto significa AND. Pode-se omitir o ponto como na expressão AB que significa A e B.

A função AND pode ser definida como “saída **1** só se todas as entradas forem **1**” e o bloco **AND** funciona deste modo, isto é, a saída será + ( nível alto ) somente quando todas as entradas estiverem no nível + ( alto ).

A fig.1 mostra o símbolo e a tabela verdadeira para o bloco **AND**.

**Material e Componentes**

Fonte de alimentação – 5V d.c.

Resistor – 270R / 1/8W

Circuito – CI 7408

LED

Conectores

Protoboard

**Procedimento:**

1. Conecte no board, o CI 7408, fig.2, que consta de quatro blocos **AND**, cada um com duas entradas.

2. Ligue o pino 14 a +5V e o pino 7 ao “COMUM”, fig.3 .

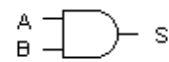
**Nota:**

Na série 74, todos os pinos de entrada sem conexão funcionam com nível **1**.

No caso da montagem prática as lâmpadas deverão ser conectadas ao circuito mediante resistores calculados conforme o Vcc. Lâmpada acesa vale **1**.

A fig.3 mostra como conectar no “board” um simples circuito para teste.

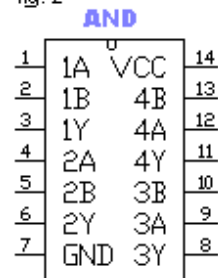
fig. 1 **7408**



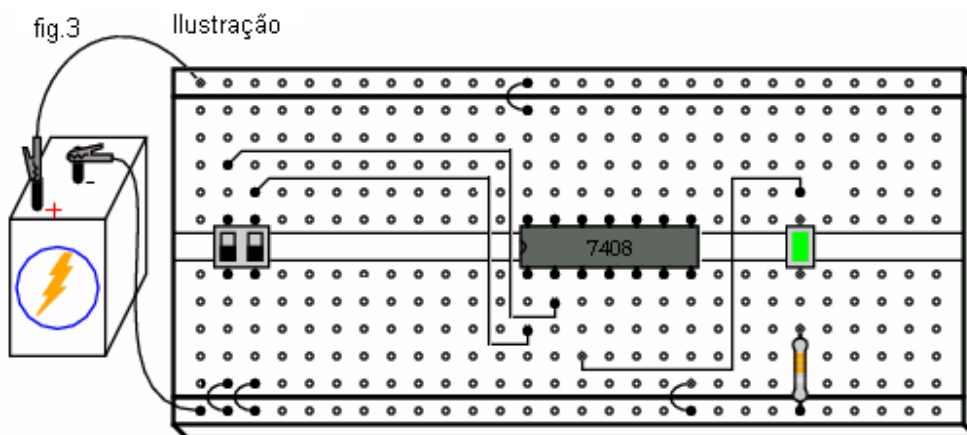
Bloco AND		
inputs		output
A	B	S
0	0	0
0	1	0
1	1	1
1	0	0

truth table

fig. 2



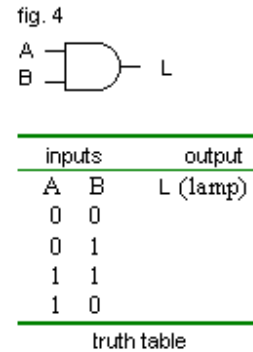
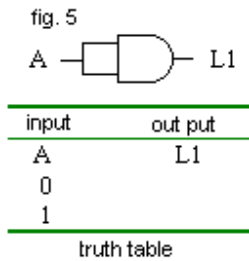
7408



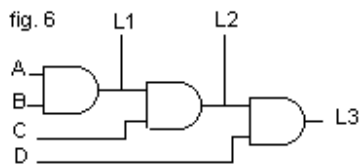
A figura mostra o board layout para o CI 7408

3. Monte o circuito da fig.4 e complete a tabela verdade.

4. Monte o circuito da fig.5 e complete a tabela verdade.

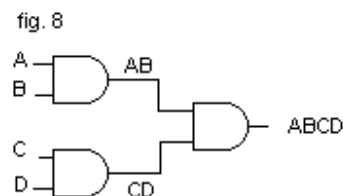
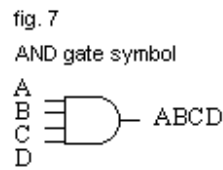


5. Monte o circuito da fig.6 e complete a tabela verdade.



inputs				output		
A	B	C	D	L1 = AB	L2 = ABC	L3 = ABCD
0	0	0	0			
0	0	0	1			
0	0	1	1			
0	0	1	0			
0	1	1	0			
0	1	1	1			
0	1	0	1			
0	1	0	0			
1	1	0	0			
1	1	0	1			
1	1	1	1			
1	1	1	0			
1	0	1	0			
1	0	1	1			
1	0	0	1			
1	0	0	0			

truth table



Usando o CI 7408 pode-se fazer circuitos **AND** com mais de três entradas, fig. 6 - 7. Por exemplo: para executar a função lógica  $L3 = ABCD$  fazemos  $L3 = (AB) C D$ .

No caso do circuito **AND** de quatro entradas, fig.6, com três blocos **E** de duas entradas, entre as entradas e a saída **L3** existe a demora de três blocos **AND**. Isto algumas vezes pode ser inconveniente; um método usado para reduzir esta demora é o indicado no circuito da fig.8.

Note que a função lógica **AND** é definida por: "Saída **1** se todas as entradas forem **1**"

**Responda:**

Como obter "3-input **AND** gate" usando dois "2-input **AND** gates" ?

Referências:

António J.; Padilla G. Sistemas digitais. Portugal, Mc Graw-Hill, 1993  
 Palmer J.; Perlman D. Introduction to Digital Systems. USA, Mc Graw-Hill, 1993  
 Plant, Malcolm. Basic Electronics, London, SCDC Publications.

Prof. Franklin

**Experimento 2.**

**Objetivo:**

Estabelecer a tabela verdade para o gate **OR**

Introdução:

A função **OR** é definida como aquela que tem saída **1** quando houver uma ou mais entradas **1**.

Na fig.1 tem-se o símbolo **OR** e a tabela verdade.

Para um “multiple input **OR** gate”, um alto em qualquer das entradas dá um alto na saída.

**Material e Componentes**

Fonte de alimentação – 5V d.c.

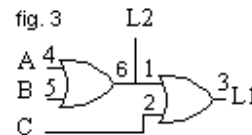
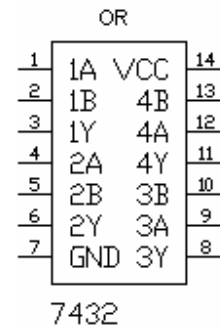
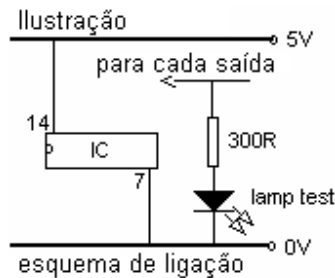
Resistor – 270R / 1/8W

Circuito – CI 7432

LED

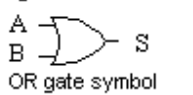
Conectores

Protoboard



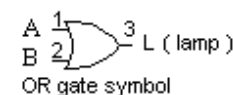
inputs			output	
A	B	C	L1	L2
0	0	0		
0	0	1		
0	1	1		
0	1	0		
1	1	0		
1	1	1		
1	0	1		
1	0	0		

fig.1



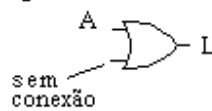
inputs		output
A	B	A + B = S
0	0	0
0	1	1
1	1	1
1	0	1

fig. 2a



inputs		output
A	B	A + B = L
0	0	
0	1	
1	1	
1	0	

fig. 2b



input	output
A	L
0	
1	

**Experiência:**

1. Conecte o CI 7432 no board.
2. Ligue o pino 14 a + 5V e o pino 7 ao comum.
3. Execute o circuito da fig.2a e “check” a tabela verdade.
4. Usando o circuito anterior, desconecte o pino 2 da chave B, fig.2b e complete a tabela.

Após completar a tabela observe que a saída L independe de A, o pino 2 que está desconectado (flutuando) está funcionando na entrada do bloco como se tivesse o nível lógico **1**. E isto pode ser considerado uma regra para os CIs da série 74XX. Vemos que neste caso o circuito não executa função nenhuma (saída sempre 1).

Na **série 74**, todos os pinos de entrada sem conexão funcionam como 1.

Blocos **OR** de várias entradas.

5. Monte no board o circuito da fig.3. Este circuito executa a função  $L1 = A + B + C$ .

Este circuito é também chamado de **OR** de dois níveis.

6. O circuito da fig. 4a executa a função

$L3 = A + B + C + D$ . “Check” o funcionamento.

Note que podemos construir o circuito de acordo com a equação equivalente  $L3 = (A+B) + (C + D)$ , fig.4b.

fig. 4a

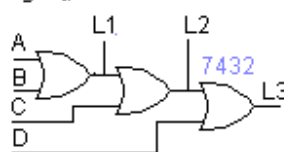
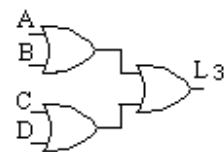


fig. 4b



inputs				output		
A	B	C	D	L1=A+B	L2=A+B+C	L3=A+B+C+D
0	0	0	0			
0	0	0	1			
0	0	1	1			
0	0	1	0			
0	1	1	0			
0	1	1	1			
0	1	0	1			
0	1	0	0			

**Experimento 3.**

**Objetivo:**

Estabelecer a tabela verdade para o gate **Not** (inversor)

Introdução:

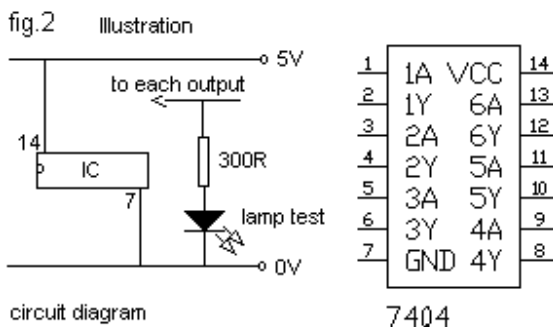
A implementação de circuitos lógicos, eventualmente requer a inversão de um sinal ou função, isto é, um “input of logic **1** gives an output of logic **0**”. Para este fim usa-se o Gate **Not** (Não). Também conhecido como inversor este bloco lógico é o mais simples de todos os gates, tendo somente uma entrada e uma saída. O símbolo e a tabela de funcionamento (função executada do circuito **Not**) estão mostrados na fig.1.

**Material**

- Fonte de alimentação – 5V d.c.
- Resistor – 270R / 1/8W
- Circuito – CI 7404
- LED
- Conectores
- Protoboard

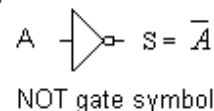
**Procedimento:**

1. Conecte um CI 7404 que consta de seis inversores, no “board”.
2. Ligue o pino 14 do CI a + 5V e o pino 7 ao comum, fig.2.



3. Usando os terminais 1 e 2, faça as ligações indicadas no circuito da fig.3.
4. Ligue a chave **A** para **0**, anote o resultado de **L** na tabela. A seguir ligue **A** para **1** e anote o resultado de **L** na tabela verdade.
5. Faça a montagem do circuito fig.4 e anote o resultado na tabela verdade.
6. Faça a montagem do circuito fig.5 e anote o resultado na tabela verdade.

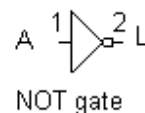
fig. 1



inputs	output
A	$S = \bar{A}$
0	1
1	0

truth table

fig. 3



inputs	output
A	L (lamp)
0	
1	

truth table

fig. 4



inputs	output
A	L1
0	
1	

truth table

7. Verifique na tabela do circuito da fig.5 que a saída de cada bloco **NÃO** é o inverso da entrada.

NOTA: Uma série de blocos **NÃO** tem várias aplicações práticas, por exemplo, como circuito de demora, outra aplicação é o buffer; a cada especificação de saída, de um dado bloco lógico indica quantas entradas de blocos lógicos de mesma família podem ser conectados à sua saída, isto é denominado de FANOUT, por exemplo, para Cis do tipo TTL o FANOUT típico é 10. Se a configuração do circuito exigir mais de 10 blocos, podem-se usar inversores como **buffer** para aumentar o FANOUT, conforme mostra a fig.6.

Resumo:

O bloco **NÃO** (inversor) é usado para inverter o sinal de entrada. Demoras (delays) com inversão de sinal podem ser obtidas usando-se um número ímpar de inversores em série. A demora total será a soma das demoras dos blocos. Os inversores podem ser usados também como buffers.

fig. 5

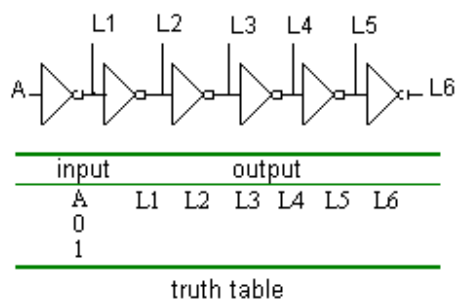
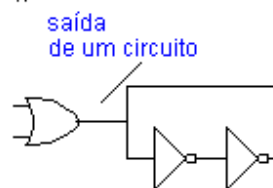


fig. 6



Referências:

- António J.; Padilla G. Sistemas digitais. Portugal, Mc Graw-Hill, 1993
- Palmer J.; Perlman D. Introduction to Digital Systems. USA, Mc Graw-Hill, 1993
- Plant, Macolm. Basic Eletronics, London, SCDC Publications.

Prof. Franklin

**Experimento 4.**  
 Gate **NAND** (NÃO-E)

**Objetivo:**

Estabelecer a tabela verdade para o gate **NAND**.  
 Estabelecer a tabela verdade para o gate **XOR**.

**Introdução**

A função NÃO-E pode ser definida como saída **0** somente se ambas entradas são **1**. Simbolicamente o bloco NÃO-E pode ser representado pelo símbolo de um bloco **E** seguido de um pequeno círculo que indica uma inversão. A função NÃO-E também pode ser formada por um bloco **E** seguido de um **NÃO**, fig.2.

O Bloco NÃO-E é um circuito universal. Com o bloco NÃO-E, pode-se executar qualquer circuito. Mostraremos a execução das funções básicas usando este bloco.

A fig.1 mostra o símbolo e a tabela verdade do bloco **NAND**. A expressão Booleana é:  
 $S = \overline{AB}$

fig. 1



input		output
A	B	$S = \overline{A \cdot B}$
0	0	1
0	1	1
1	1	0
1	0	1

truth table

fig. 2

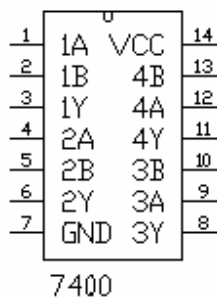
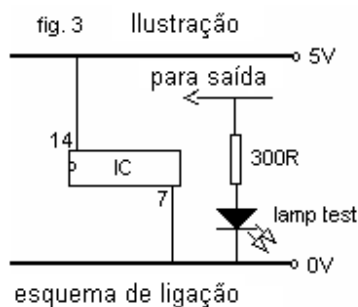
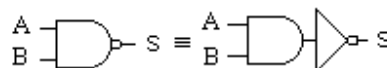
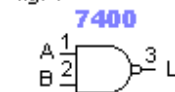


fig. 4



input		output
A	B	L (lamp)
0	0	
0	1	
1	1	
1	0	

truth table

**Material**

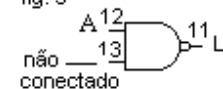
- Fonte de alimentação – 5V d.c.
- Resistor – 270R / 1/8W
- Circuito – CI 7400
- LED
- Conectores
- Protoboard

**Procedimento:**

1. Conecte o CI7400 ao “board”, como ilustra a fig.3.
2. Monte no “board” o circuito da fig.4 e complete a tabela verdade.
3. Execute o circuito da fig.5 e preencha a tabela de funcionamento.

**Nota:** - Observe que a função executada, circuito da fig.5, é a função **NOT** (um dos pinos flutuando funciona com nível **1**)

fig. 5

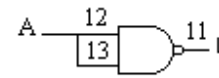


input	output
A	L
0	
1	

truth table

Então na equação com o pino  $11 = \overline{12.13}$  que corresponde no circuito anterior a  $L = \overline{A.1} = \overline{A}$ . O mesmo resultado poderia ser obtido se conectássemos o pino 13 a um ponto de nível 1, ou pino 13 a entrada A, fig.6.

fig. 6

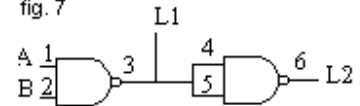


Na Álgebra de Boole consta o teorema  $\overline{A.A} = \overline{A}$ . Com este teorema observamos que a função executada pelo circuito anterior é a função NÃO, isto é:

o pino  $11 = \overline{12.13}$  ;  $L = \overline{A.A} = \overline{A}$

4. Monte o circuito da fig.7 e preencha a tabela.

fig. 7



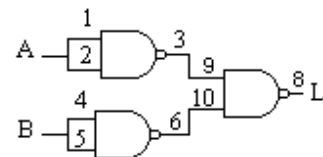
input		output	
A	B	L1	L2
0	0		
0	1		
1	1		
1	0		

truth table

Observe que  $L1 = \overline{AB}$  e que  $L2$  é o sinal de  $L1$  invertido, isto é,  $L2$  executa a função NÃO com  $L1$ , ou seja,  $L2 = \overline{L1} = \overline{\overline{AB}}$ . A Álgebra de Boole mostra que uma expressão submetida a duas negações resulta na expressão original (isto é, duas barras sobre a expressão não alteram a expressão), assim sendo  $L2 = \overline{\overline{AB}} = AB$ . Vemos então que  $L2$  é saída de um circuito E, portanto o circuito da fig.7 executa a função E.

5. Monte o circuito da fig.8 e complete a tabela.

fig. 8



input		output
A	B	L
0	0	
0	1	
1	1	
1	0	

truth table

Note que a função executada é a função OU, isto é,  $L = A + B$ . Pode-se explicar isto mediante o Teorema de DE MORGAN:

a)  $\overline{X + Y} = \overline{X} . \overline{Y}$

b)  $\overline{X.Y} = \overline{X} + \overline{Y}$

Note que no circuito anterior o pino  $3 = \overline{A}$  e o pino  $6 = \overline{B}$

$$L = \overline{3.6} = \overline{\overline{A}\overline{B}}$$

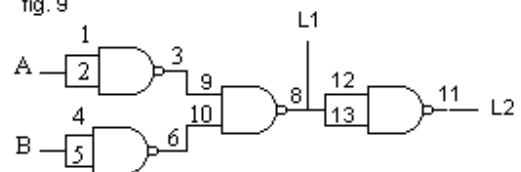
pelo teorema de De Morgan,

$L = \overline{\overline{A} + \overline{B}} = A + B$  mostrando que o circuito anterior executa a função OU.

6. Monte o circuito da fig.9 e preencha a tabela.

Observe que  $L1$  corresponde à saída de um circuito OU, isto é,  $L1 = A + B$  e  $L2$  é a negação de  $L1$ , isto é  $L2 = \overline{L1} = \overline{A+B}$ , então  $L2$  é a saída de um circuito NÃO-OU. O circuito da fig.9 executa a função NÃO-OU.

fig. 9



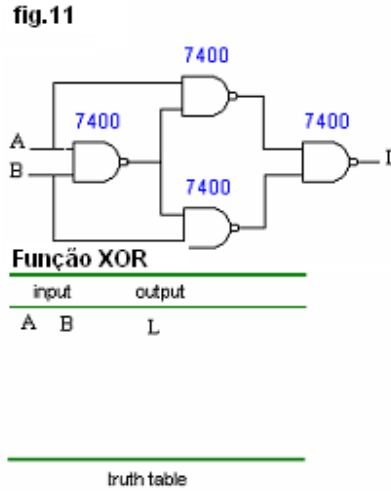
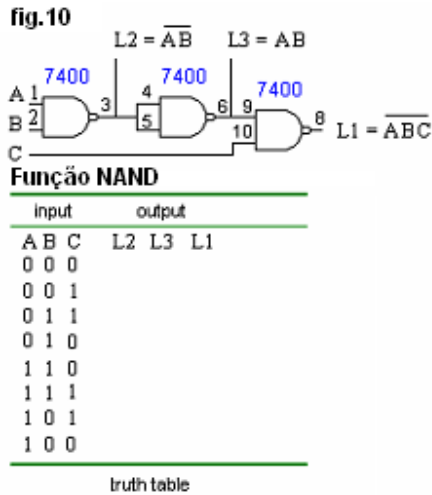
input		output	
A	B	L1	L2
0	0		
0	1		
1	1		
1	0		

truth table

7. Com o CI 7400 execute o circuito da fig.10,  $L1 = \overline{A.BC}$  e complete a tabela.

8. Conecte o CI 7400 ao board. Monte o circuito da fig.11. Verifique que a função executada é a função XOR. Escreva a expressão simplificada do circuito.

Com circuitos NAND pode-se executar qualquer função lógica.

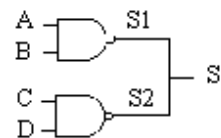


Nota:

Lógica por Fios ou “Wired Logic”:

O gate AND por fios pode ser formado ligando-se a saída de dois ou mais blocos TTL a um só ponto, conforme indica a fig.12 e a correspondente tabela. Para circuitos TTL não é uma boa prática, devendo ser evitada (exceto quando se usam blocos TTL especiais, chamados de coletor aberto), pois em geral provoca um consumo excessivo. A conexão das duas saídas a um só ponto funciona como um bloco E, isto é, se algumas das saídas S1 ou S2 estiver no nível (baixo), a saída S estará 0 (baixa). Em outras palavras, S só será 1 (alto) se S1 for 1 (alto), e S2 for 1 (alto), conforme mostra a tabela; então  $S = S1.S2$ , mas  $S1 = \overline{AB}$  e  $S2 = \overline{CD}$  então  $S = \overline{AB} . \overline{CD}$  Com este tipo de circuito há economia de um bloco E.

fig. 12



Referências:

- António J. Padilla G. Sistemas digitais. Portugal, Mc Graw-Hill, 1993
- Palmer J.; Perlman D. Introduction to Digital Systems. USA, Mc Graw-Hill, 1993
- Plant, Macolm. Basic Eletronics, London, SCDC Publications

Prof. Franklin



**Objetivo:** Verificar alguns teoremas da Álgebra de Boole.

Introdução:

Todos os circuitos digitais funcionam ao conectar nas suas entradas sinais digitais. As saídas são também sinais elétricos do mesmo tipo. Estes sinais são formados exclusivamente por dois níveis de tensão que correspondem aos dois estados estáveis ( corte e saturação ) dos elementos eletrônicos básicos que constituem as portas lógicas e outros blocos integrados.

Qualquer informação que se deseja tratar, processar ou armazenar usando sistemas digitais deverá ser codificado num tipo de linguagem adequada. Na prática, qualquer número, letra, símbolo, instrução ou operação é convertido num conjunto de sinais elétricos digitais.

A álgebra de Boole é fundamental para se trabalhar com circuitos lógicos. Os teoremas usuais da álgebra de Boole são:

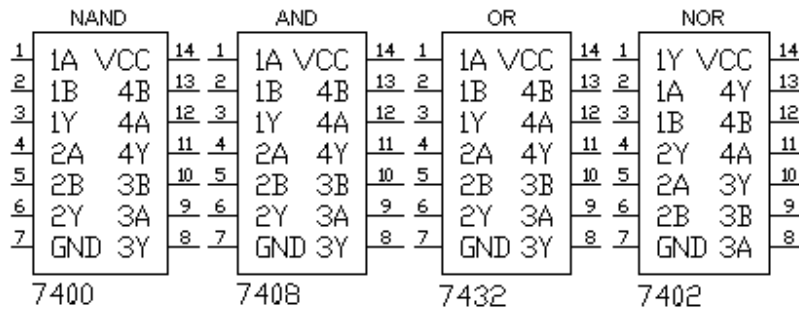
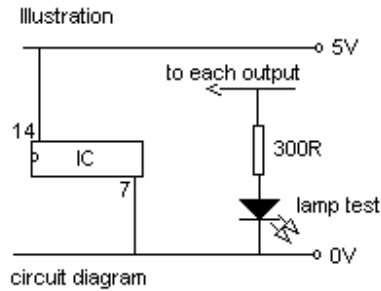
- |   |  |
|---|--|
| 1. $0 \cdot X = 0$  | 1. $0 + X = X$   |
| 2. $1 \cdot X = X$  | 2. $1 + X = 1$   |
| 3. $X \cdot X = X$  | 3. $X + X = X$   |
| 4. $X \cdot \bar{X} = 0$  | 4. $X + \bar{X} = 1$   |
| 5. $XY = YX$  | 5. $X + Y = Y + X$   |
| 6. $\overline{XYZ} = (\overline{XY})Z = X(\overline{YZ})$         | 6. $\overline{X + Y + Z} = \overline{(X + Y)} + Z = \bar{X} + (Y + Z)$ |
| 7. $\overline{XY \dots Z} = \bar{X} + \bar{Y} + \dots + \bar{Z}$  | 7. $\overline{X + Y \dots + Z} = \bar{X} \cdot \bar{Y} \dots \bar{Z}$  |
| 8. $\overline{XY + XZ} = X(\overline{Y + Z})$                     | 8. $(X + Y)(X + Z) = X + YZ$   |
| 9. $\overline{XY + X\bar{Y}} = X$                                 | 9. $(X + Y)(X + \bar{Y}) = X$  |
| 10. $X + \overline{XY} = X + Y$                                   | 10. $X(\overline{X + Y}) = \bar{X}$                                    |
| 11. $X + \bar{X}Y = X + Y$  | 11. $X(\bar{X} + Y) = XY$  |
| 12. $ZX + Z\bar{X}Y = ZX + ZY$                                    | 12. $(Z + X)(Z + \bar{X} + Y) = (Z + X)(Z + Y)$                        |
| 13. $\overline{XY + \bar{X}Z + YZ} = \bar{X} + \bar{Y} + \bar{Z}$ | 13. $(X + Y)(\bar{X} + Z)(Y + Z) = (X + Y)(\bar{X} + Z)$               |
| 14. $\overline{XY + \bar{X}Z} = (X + Z)(\bar{X} + Y)$             | 14. $(X + Y)(\bar{X} + Z) = XZ + \bar{X}Y$                             |

Os circuitos seguintes examinam alguns teoremas da Álgebra de Boole.

## Experimento

Procedimento: Conecte os circuitos integrados seguintes no board: 1x 7400 1x7402 1x 7408 1x7432.

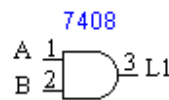
Faça as alimentações destes circuitos, isto é, faça suas conexões ao  $V_{cc} = +5V$  e ao GND.



1. Monte no board os circuitos da fig. 1.

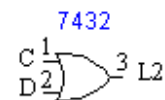
Depois de completadas as tabelas, obtém-se no caso do gate AND,  $L1 = 0$  (off). Verificando o teorema  $0.X = 0$  no caso do circuito  $A = 0$  e  $B = X$ . Note que 0 vezes qualquer valor é 0. No caso do gate OR obtém-se  $L2 = 1$  (on) verificando o teorema  $1 + X = 1$ , isto é, "1 mais qualquer valor é 1", onde no circuito, C foi fixado em 1 e D é o X do Teorema.

fig. 1



Inputs		outputs
A	B	L1(lamp)
0	0	
0	1	

truth table



Inputs		outputs
C	D	L2(lamp)
1	0	
1	1	

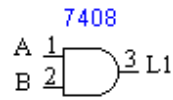
truth table

2. Monte no board o circuito da fig. 2

Neste caso o valor de L1 é igual a B como se pode verificar na tabela, o que confere o teorema:  $1.X = X$ , visto que  $X = B$  e  $A = 1$ .

1.B = B  
 1.0 = 0    B = 0    Pelos postulados temos mostrado que a saída é igual a B.  
 1.1 = 1    B = 1

fig. 2

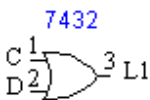


Inputs		output
A	B	L1(lamp)
1	0	
1	1	

truth table

3. Monte o circuito da fig. 3 e complete a tabela. Verifique o teorema  $0 + X = X$ .

fig. 3

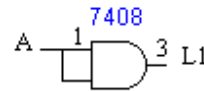


Inputs		output
A	B	L1(lamp)
0	1	
0	0	

truth table

4. Monte o circuito da fig. 4 e complete a tabela. Verifique o teorema  $XX = X$ .

fig. 4

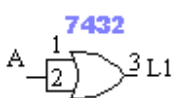


Input	output
A	L1
1	
0	

truth table

5. Monte o circuito da fig. 5 e complete a tabela. Verifique o teorema  $X + X = X$ .

fig. 5

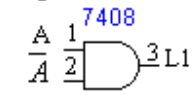


Input	output
A	L1
1	
0	

truth table

6. Monte o circuito da fig. 6 e complete a tabela. Verifique o teorema  $X\bar{X} = 0$ .

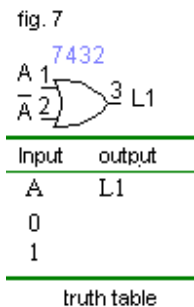
fig. 6



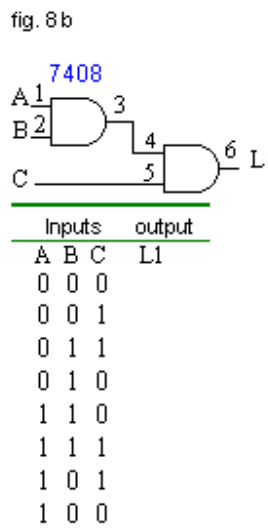
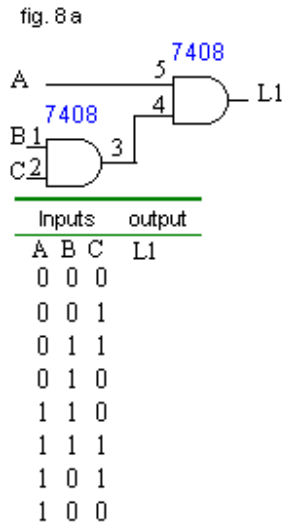
Input	output
A	L1
0	
1	

truth table

7. Monte o circuito da fig. 7 e complete a tabela.  
 Verifique o teorema  $X + \bar{X} = 1$ .

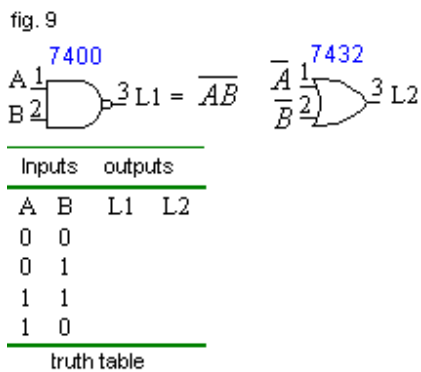


8. Monte os circuitos da fig. 8 e complete a tabela.

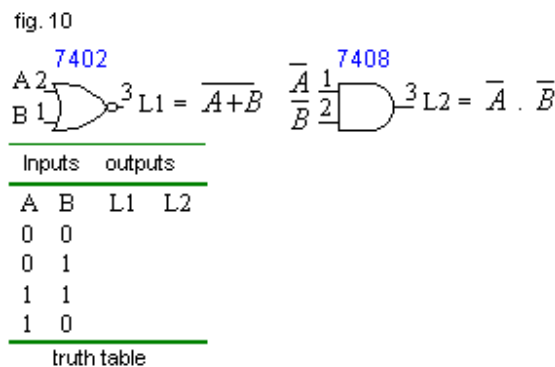


No primeiro caso, circuito da fig. 8 a, tem-se o circuito  $A(BC)$ , e no segundo caso, circuito da fig. 8 b, tem-se o circuito  $(AB)C$ . Comparando-se as duas tabelas, observa-se que  $(AB)C = A(BC)$ . De modo análogo, poderíamos mostrar, usando blocos OU, que  $(A + B) + C = A + (B + C)$ .

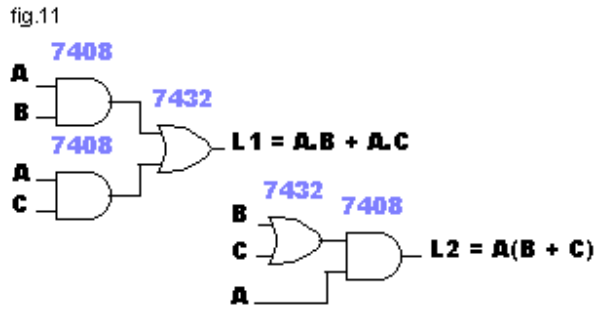
9. Monte os circuitos da fig. 9 e complete as tabelas. Nos circuitos:  $L1 = \overline{AB}$  e  $L2 = \overline{A + B}$ .  
 Verifique o Teorema de De Morgan  $\overline{AB} = \overline{A} + \overline{B}$ .



10. Monte os circuitos da fig. 10 e complete a tabela.  
 Verifique o Teorema de De Morgan  $\overline{A + B} = \overline{A} \cdot \overline{B}$ .



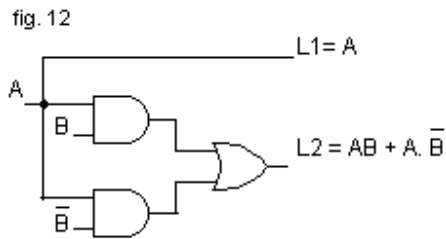
11. Monte os circuitos da fig. 11 e complete a tabela.  
 Verifique o teorema  $AB + AC = A(B + C)$ .



Inputs			output	
A	B	C	L1	L2
0	0	0		
0	0	1		
0	1	1		
0	1	0		
1	1	0		
1	1	1		
1	0	1		
1	0	0		

truth table

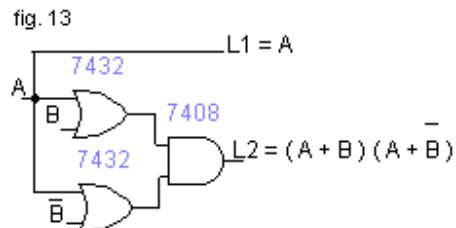
12. Monte o circuito da fig. 12 e complete a tabela.  
 Verifique o teorema  $AB + A.\bar{B} = A$



Inputs		outputs	
A	B	L1	L2
0	0		
0	1		
1	1		
1	0		

truth table

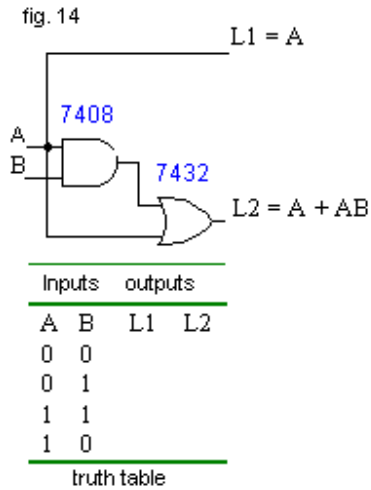
13. Monte o circuito da fig. 13 e complete a tabela.  
 Verifique o teorema  $(A + B)(A + \bar{B}) = A$ .



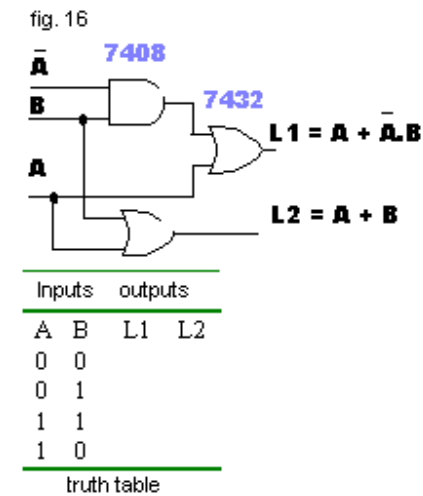
Inputs		outputs	
A	B	L1	L2
0	0		
0	1		
1	1		
1	0		

truth table

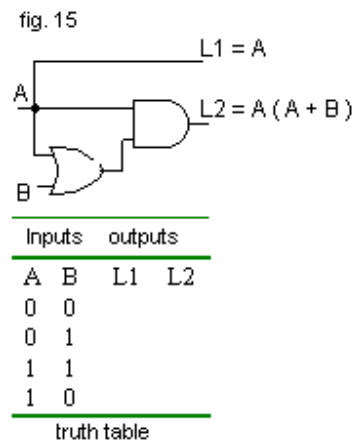
14. Monte o circuito da fig. 14 e complete a tabela.  
Verifique o Teorema  $A + AB = A$ .



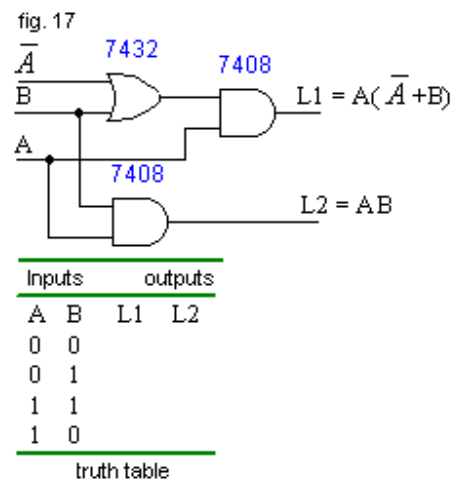
16. Monte o circuito da fig. 16 e complete a tabela.  
Verifique o teorema  $A + \bar{A}B = A + B$ .



15. Monte o circuito da fig. 15 e complete a tabela.  
Verifique o Teorema  $A(A+B) = A$ .



17. Monte o circuito da fig. 17 e complete a tabela.  
Verifique o teorema  $A(\bar{A} + B) = AB$ .

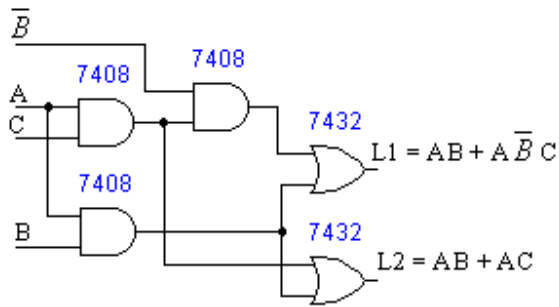


18. Monte o circuito da fig. 18 .

Verifique o teorema

$$AB + A\bar{B}C = AB + AC.$$

fig. 18



Inputs	outputs	
A B C	L1	L2
0 0 0		
0 0 1		
0 1 1		
0 1 0		
1 1 0		
1 1 1		
1 0 1		
1 0 0		

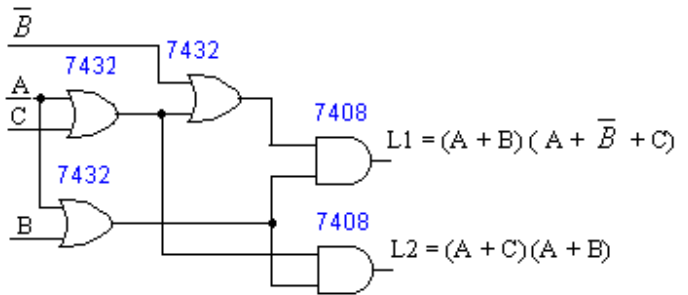
truth table

19. Monte o circuito da fig. 19.

Verifique o teorema

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

fig. 19



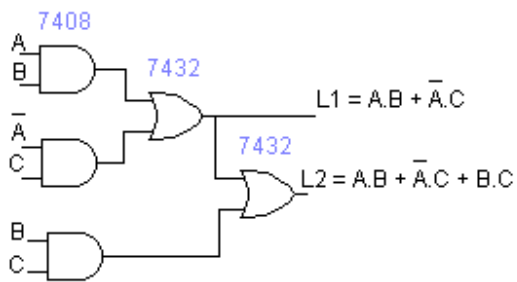
Inputs	outputs	
A B C	L1	L2
0 0 0		
0 0 1		
0 1 1		
0 1 0		
1 1 0		
1 1 1		
1 0 1		
1 0 0		

truth table

20. Monte no board o circuito da fig. 20  
 Verifique o teorema

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

fig. 20



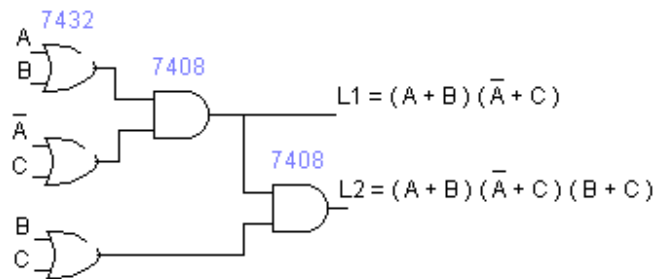
Inputs			outputs	
A	B	C	L1	L2
0	0	0		
0	0	1		
0	1	1		
0	1	0		
1	1	0		
1	1	1		
1	0	1		
1	0	0		

truth table

21. Monte no board o circuito da fig. 21.  
 Verifique o teorema

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

fig. 21



Inputs			outputs	
A	B	C	L1	L2
0	0	0		
0	0	1		
0	1	1		
0	1	0		
1	1	0		
1	1	1		
1	0	1		
1	0	0		

truth table

Referências:

- António J.; Padilla G. Sistemas digitais. Portugal, Mc Graw-Hill, 1993
- Palmer J.; Perlman D. Introduction to Digital Systems. USA, Mc Graw-Hill, 1993
- Plant, Macolm. Basic Electronics, London, SCDC Publications, 1990
- Harowitz P.; Hill W. The Art of Electronics, USA, Cambridge University Press, 1989.

Prof Franklin