

FAMILIARIZAÇÃO COM PORTAS LÓGICAS E TEOREMA DE "DE MORGAN"

Ciro J. Egoavil¹, BARROS, E. C.², NASCIMENTO, L. A. F.², MOURA, A. F. L.²

¹Professor do Curso de Engenharia Elétrica - DAEE, Fundação Universidade Federal de Rondônia, Sala 206 - 2C, Porto Velho, Rondônia, Brasil (e-mail: ciro.egoavil@unir.br)

²Monitores da disciplina Eletrônica I - DAEE, Fundação Universidade Federal de Rondônia, Sala 206 - 2C, Porto Velho, Rondônia, Brasil (e-mail: eletronicamonitoria@gmail.com)

RESUMO

- Apresentar os conceitos, símbolos e tabelas verdade das principais portas lógicas;
- Mostrar o caráter universal das portas *NÃO-OU* e *NÃO-E*;
- Discutir os conceitos de *fan-in*, *fan-out* e o atraso nas saídas das portas lógicas;
- Familiarizar o discente com o teorema de *De Morgan*.

I. INTRODUÇÃO TEÓRICA

A implementação de circuitos lógicos é possível através da utilização de componentes eletrônicos discretos, entretanto na maioria das aplicações utiliza-se os conhecidos circuitos integrados. O desenvolvimento da eletrônica digital representou um avanço significativo para a tecnologia tornando possível, por exemplo, a construção de chips que contém pelo menos 4 bilhões de transistores em um invólucro de algumas polegadas [1].

A vantagem de sistemas digitais em relações àqueles com tecnologia analógica (para determinadas aplicações) se dá pelo fato de que em sistemas digitais é perfeitamente possível que se represente quantidades a partir de apenas dois símbolos, aos quais dá-se o nome de *níveis lógicos*, tais níveis são representados pelos símbolos 0 e 1. O símbolo 0 geralmente representa um estado baixo de tensão, enquanto que o símbolo 1 representa um nível alto de tensão em um sistema digital.

Com isto, se torna possível realizar operações lógicas com base em dois possíveis valores para determinadas variáveis, estes dois estados podem ser: *Ligado ou desligado*, *verdadeiro ou falso*, *sim ou não*, entre outros.

A base do sistema de numeração que descreve os sistemas digitais é chamada de *binária*, devido ao fato de só haver duas possibilidades de valores para um sistema binário. Cada símbolo que compõe um número binário é chamado de *Bit*, abaixo vê-se um número representado na base *binária* que possui 8 *bits*:

$$N = 10100001 \quad (1)$$

É possível realizar conversões entre as bases numéricas utilizando um conjunto de operações destinadas a este fim. As conversões mais recorrentes acontecem entre as bases *binária* e *decimal*, visto o fato de que as duas são as mais utilizadas. O conjunto de operações algébricas que fundamentam as operações envolvendo circuitos lógicos é a *Álgebra de Boole*.

A. ÁLGEBRA DE BOOLE

A utilização da *Álgebra de Boole* faz com que a análise de circuitos digitais se torne mais simples e, por este motivo, se torna uma ferramenta indispensável para analisar estes sistemas. Há uma série de operações *Booleanas* que são válidas. Abaixo estão listadas as principais propriedades no que diz respeito ao modo de se operar variáveis em *Álgebra Booleana*:

- $A + B = B + A$ (Comutativa)
- $A.B = B.A$ (Comutativa)
- $A + (B + C) = (A + B) + C = A + B + C$ (Associativa)
- $A.(B.C) = (A.B).C = A.B.C$ (Associativa)
- $A.(B + C) = A.B + A.C$ (Distributiva)
- $(A + B)(C + D) = A.C + A.D + B.C + B.D$ (Distributiva)
- $A + AB = A$ (Absorção)
- $A + \bar{A}.B = A + B$ (Absorção)

B. TEOREMAS DE DE MORGAN

Além dos teoremas acima, tem-se os conhecidos: *Teoremas de De Morgan*, chamados assim em homenagem ao

matemático que os formulou. Os teoremas de *De Morgan* em conjunto com as propriedades listadas acima compõem as ferramentas necessárias para analisar qualquer circuito lógico com 3 ou mais variáveis de entrada (Os mapas de Karnaugh serão vistos posteriormente) visto que é possível estender os teoremas vistos para casos mais gerais onde há mais variáveis envolvidas. As expressões dos teoremas de *De Morgan* para duas variáveis podem ser vistas nas equações abaixo:

- Teorema 1:

$$(\overline{A + B}) = \overline{A} \cdot \overline{B}$$

- Teorema 2:

$$\overline{A} \cdot \overline{B} = \overline{(A + B)}$$

C. REPRESENTAÇÃO DE PORTAS LÓGICAS E IMPLEMENTAÇÃO UTILIZANDO CIRCUITOS TTL

Os circuitos integrados TTL *transistor-transistor-logic*, são os mais utilizados para a implementação de portas lógicas. Com estes, é possível obter circuitos *flip-flops*, registradores de dados, portas lógicas *E*, *OU* e *NOT*, entre várias outras aplicações. Abaixo estão listadas as principais operações lógicas a serem estudadas, a partir destas é possível obter toda uma gama de dispositivos que realizam operações lógicas combinando-as, é possível que em determinadas aplicações as operações lógicas sejam aplicadas sem q estas estejam combinadas com nenhuma outra.

- PORTA LÓGICA *E* ou *AND*:

A operação lógica *E* caracteriza-se por ser aquela onde o valor na saída apresenta nível lógico 1 se, e somente se, todas as variáveis de entrada possuem valor lógico 1. A tabela (1) é chamada de Tabela verdade da operação lógica "*E*":

Tabela 1. Tabela verdade da operação lógica *E*

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

A representação da porta lógica *E* é mostrada na figura (1):

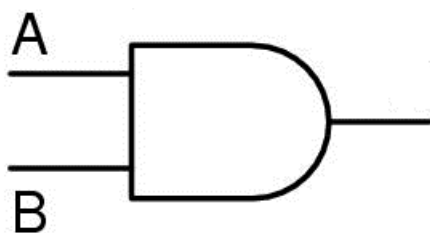


Figura 1. Representação de uma porta lógica *E* [1]

Um Circuito integrado TTL que realiza a função *E* é o *SN7408N*, sua configuração de pinagem está apresentada na figura (2):

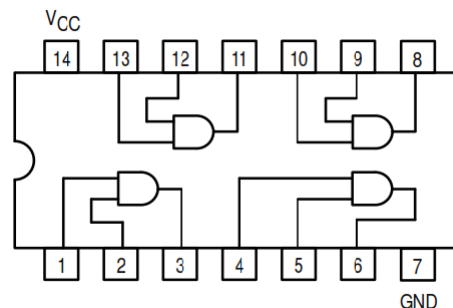


Figura 2. Configuração de pinos do circuito integrado *SN7408N* [4]

Num único invólucro de um circuito integrado *SN7408N* estão disponíveis pelo menos 4 portas *AND* que podem ser utilizadas independentemente entre si. Na figura (3) é possível verificar o invólucro mais comum deste circuito integrado.



Figura 3. Circuito integrado *SN7408N* [4]

- PORTA LÓGICA *OU* ou *OR*

Esta operação lógica é caracterizada pelo fato de que a saída lógica apresenta valor 1 se *pelo menos uma* das entradas tem valor lógico 1, esta característica pode ser vista na tabela (2) que é a tabela verdade desta operação lógica: A porta *OU* é útil quando se deseja obter um sinal

Tabela 2. Tabela verdade da operação lógica "*OU*"

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

para uma variável de saída todas as vezes em que pelo menos uma das variáveis de entrada altere o seu estado. Quando se monitora processos químicos, por exemplo, pode ser necessário medir variáveis como pressão e temperatura e com portas *OU* se torna possível receber um sinal na saída toda vez que pelo menos uma destas variáveis mudar seu estado. A representação de uma porta *OU* pode ser vista na figura (4):

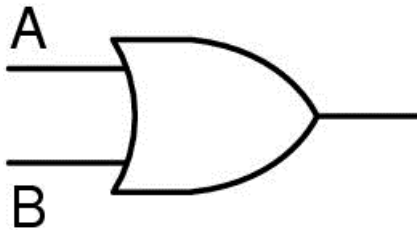


Figura 4. Representação da porta lógica OU [2]

Um circuito integrado *TTL* que implementa a operação lógica OU é o *SN74LS32N*. Sua configuração de pinos pode ser vista na figura (5), onde se observa as semelhanças presente nos invólucros deste circuito integrado com o que já foi apresentado anteriormente.

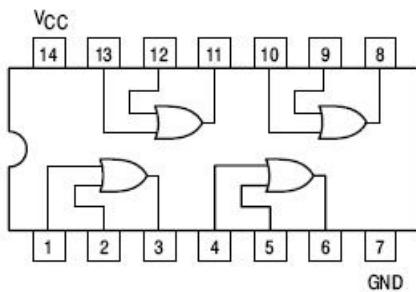


Figura 5. Configuração de pinos do circuito integrado *SN74LS32N* [5]

É importante que as semelhanças entre os invólucros não permitam que o projetista de um circuito selecione portas lógicas erradas para seu projeto, visto que apesar de haver semelhança na construção, as operações que elas realizam são completamente diferentes:

Na figura (6) é possível observar um circuito integrado *HD74LS32P*:

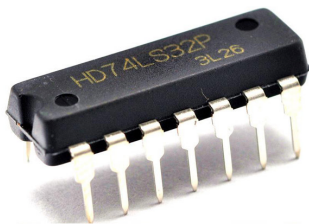


Figura 6. Circuito integrado *HD74LS32P* [5]

• PORTA LÓGICA NOT OU INVERSORA

A primeira característica a ser observada sobre a operação lógica *NOT* é que, diferente das anteriores, esta apresenta uma única variável de entrada. A operação *NOT* consiste em fornecer na saída de uma porta lógica a variável complementar àquela na entrada, para isto, usa-se o símbolo conhecido como complementar, que pode ser uma barra acima da variável. Como o conjunto universo dos números binários só possui dois possíveis valores, a tabela verdade para esta operação fica :

Tabela 3. Tabela verdade da operação "NOT" ou "INVERSORA"

A	\bar{A}
0	1
1	0

A representação da porta lógica *NOT* pode ser vista na figura (7):

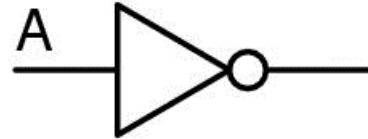


Figura 7. Representação da porta lógica NOT [2]

Os circuitos integrados desenvolvidos para realizar esta operação lógica contam, normalmente, com pelo menos 6 portas lógicas independentes num único invólucro de 14 pinos, a possibilidade de se implementar mais portas em um único invólucro se dá pelo fato de que há apenas um terminal de entrada para cada porta lógica.

Na figura (8) pode-se ver a configuração de pinos de um circuito integrado 7404:

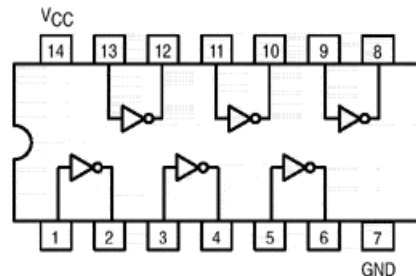


Figura 8. Configuração dos terminais do circuito integrado 7404 [4]

O circuito integrado visto na figura (9) é um *SN74HC04N*, e é utilizado quando se deseja realizar a operação lógica "NOT":



Figura 9. Circuito integrado *SN74HC04N* [5]

• PORTA LÓGICA NAND

Na operação *NAND*, a letra "N" significa *NOT* e esta porta nada mais é do que uma porta *AND* com um inversor

acoplado. Por isso, sua saída é o oposto da *AND*. Seu símbolo é o mesmo do *AND*, mas com um *o* em sua saída, para dizer que o valor da sua saída é invertido. É possível construir uma porta *NAND* conectando uma porta *AND* a um inversor. A tabela (4) representa a tabela verdade da operação lógica *NAND*, é possível verificar que esta operação é a negação da operação lógica *AND*:

Tabela 4. Tabela verdade da operação lógica "NAND"

A	B	A.B
0	0	1
0	1	1
1	0	1
1	1	0

A implementação de uma porta lógica *NAND* pode ser feita através da combinação entre uma porta lógica *AND* e uma porta lógica *NOT*, entretanto existem circuitos integrados que realizam a operação *NAND* disponíveis em mercado.

Na figura (10) é possível verificar a representação de uma porta lógica "NAND":

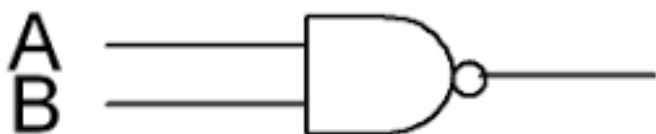


Figura 10. Configuração dos terminais do circuito integrado 7404 [5]

Um circuito integrado que realiza a função "NAND" é o *SN74LS00P*, a configuração dos terminais deste circuito pode ser vista na figura (11):

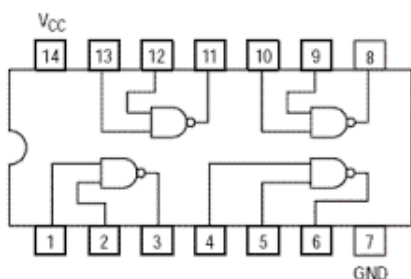


Figura 11. Configuração dos terminais do circuito integrado 7404 [5]

• PORTA LÓGICA NOR

Esta operação é análoga a operação *NAND*, porém a operação *NOR* é a negação da operação *OU*, logo sua tabela verdade fica como na tabela (5):

Tabela 5. Tabela verdade da operação lógica NOR

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

O diagrama esquemático de um circuito integrado 7402, usado para implementar uma porta lógica *NAND*, pode ser visto na figura (12):

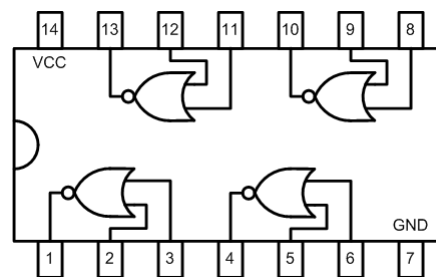


Figura 12. Configuração dos terminais do circuito integrado "7402"[5]

• PORTAS LÓGICAS "OU EXCLUSIVO"

A porta lógica *OU-EXCLUSIVO* de duas entradas compara dois bits e a saída será 1 se, e somente se, eles forem diferentes. Uma porta *OU-EXCLUSIVO* de várias entradas terá a saída igual a q se tiver um número ímpar de níveis lógicos 1 na entrada.

A configuração dos terminais de um circuito integrado que realiza a função lógica *OU-EXCLUSIVO*, pode ser vista na figura (13):

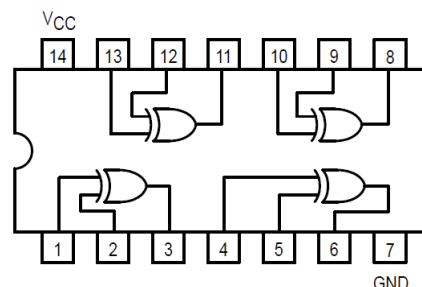


Figura 13. Configuração dos terminais do circuito integrado "7486"[5]

A porta lógica *NÃO-OU-EXCLUSIVO* compara dois bits binários e a saída será 1 se, e somente se, eles forem iguais. No caso de várias entradas a saída só será 1 se houver um número par de níveis lógicos 1 na entrada. Ou seja, esta operação lógica aplica uma inversão a operação *OU-EXCLUSIVO*.

Na tabela (6) é possível observar os resultados de saída, com base nos valores de entrada de duas variáveis, para as operações "OU-EXCLUSIVO e NÃO-OU-EXCLUSIVO":

Tabela 6. Tabela verdade da operação lógica "OU - EXCLUSIVO"

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

II. MATERIAIS UTILIZADOS

- Fonte de tensão
- 1 Circuito Integrado 7404
- 1 Circuito Integrado 7408
- 1 Circuito Integrado 7432
- 1 Circuito Integrado 7486
- 4 Resistores De 560Ω (verde-azul-marrom)
- 3 Resistores De 220Ω (vermelho-vermelho-marrom)
- 3 Led's De Cores Diferentes
- Protoboard

III. PARTE EXPERIMENTAL

A. IMPLEMENTAÇÃO E VERIFICAÇÃO DE UM CIRCUITO LÓGICO COM PORTAS E NOT

Implemente uma porta "OU" utilizando apenas portas "E" e "INVERSORA", montando na protoboard a o esquema da figura 14.

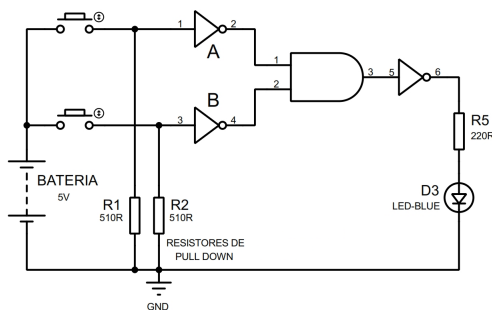


Figura 14. Circuito 1

Verifique a tensão no resistor de saída do circuito.

Os terminais onde se deve inserir VCC e GND em cada circuito integrado estão especificados nas figuras (2), (5), (8), (11), (12) e (13).

Verifique o funcionamento do circuito (1) e preencha a tabela (1):

Tabela (1)

Tabela verdade para o funcionamento do circuito 1

A	B	S

B. PORTA E COM PORTAS OU E INVERSORA

Utilizando o Teorema de *De Morgan*, agora, implemente a porta lógica "E" usando apenas portas lógicas "OU" e "INVERSORA". Implemente na protoboard o esquema do circuito 2:

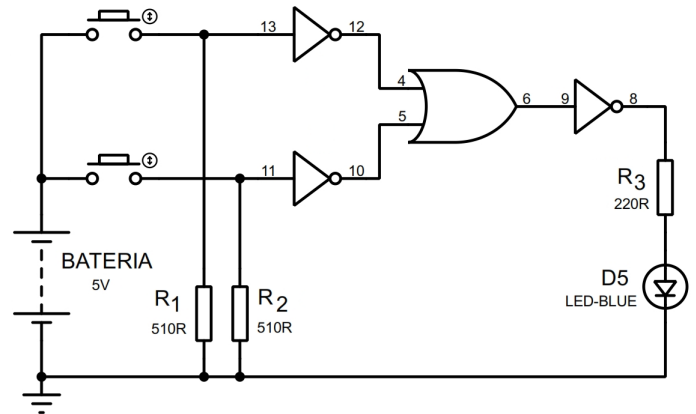


Figura 15. Circuito 2

Verifique o funcionamento do circuito da 2 e, a partir dos resultados encontrados, preencha a tabela (2):

Tabela (2)

Tabela verdade para o funcionamento do circuito 2

A	B	S

C. PORTAS NÃO-E DE 3 ENTRADAS

Implementar o circuito 3 na protoboard utilizando o CI 74LS08 e o CI 74LS04, verifique e identifique corretamente os pinos dos CIs, alimente os terminais com 5 V da fonte:

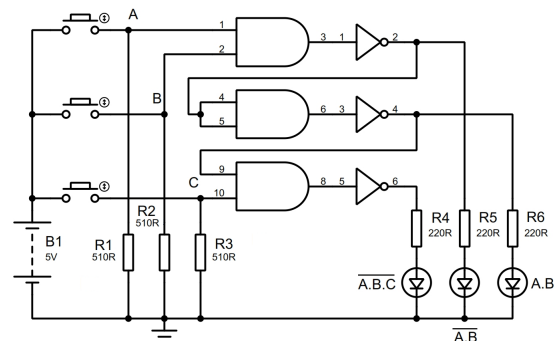


Figura 16. Circuito 3

Preencha a tabela (3) de acordo com o funcionamento do circuito da figura (18):

Tabela (3)

Tabela verdade para o funcionamento do circuito 3

Entradas			Saídas		
A	B	C	$\overline{A \cdot B}$	$A \cdot B$	$\overline{A \cdot B \cdot C}$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

D. PORTAS OU-EXCLUSIVO UTILIZANDO APENAS PORTAS NÃO-E:

Implementar o circuito 4 e verificar o seu funcionamento para todas as combinações das variáveis de entrada:

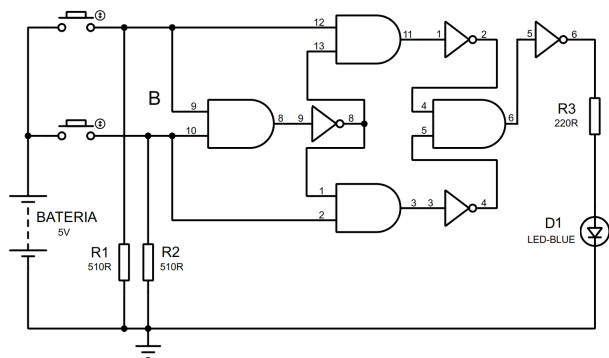


Figura 17. Circuito 4

Após verificado o funcionamento do circuito 4 preencher a tabela (4):

Tabela (4)

Tabela verdade para o funcionamento do circuito 4

Entradas		Saída
A	B	$\overline{A} \cdot B + A \cdot \overline{B}$
0	0	
0	1	
1	0	
1	1	

E. PORTAS OU-EXCLUSIVO

Implementar o circuito da 5 na protoboard utilizando o CI 74LS86.

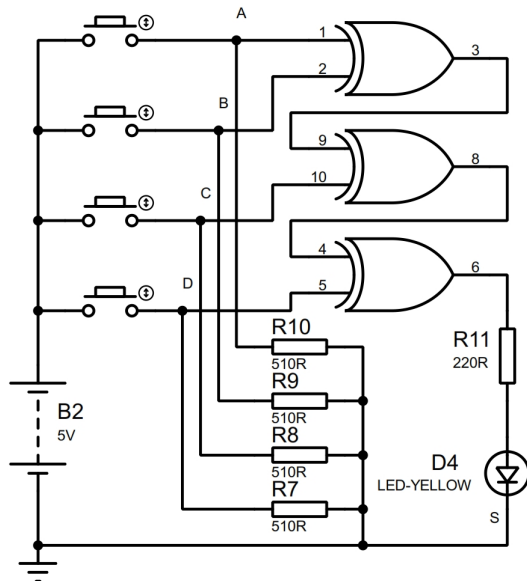


Figura 18. Circuito 5

Após verificado o funcionamento do circuito preencher a tabela (5):

Tabela (5)

Tabela verdade para o funcionamento do circuito da figura 20

A	B	C	D	$A \oplus B \oplus C \oplus D$
0	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	0	0	0	
1	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	

Referências

[1] LIMA, S. Evolução dos processadores e seu futuro; Universidade São Francisco, Itatiba 2012.
 [2] SEDRA, Adel S. SMITH, Kenneth C. "Microeletrônica", 5ª ed. São Paulo: Pearson Prentice Hall, 2007.
 [3] MALVINO, Albert P. "Eletrônica", Volume 1, 4ª ed. São Paulo: Makron Books, 1995.
 [4] TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L..Sistemas Digitais: Princípios e Aplicações. 10ª ed. São Paulo: Pearson, 2007. 830 p.

- [5] TEXAS INSTRUMENTS, Digital Logic: Pocket Data Book, Dallas, Texas, 2006.

...