

TÉCNICO EM ELETROTÉCNICA

ELETRÔNICA DIGITAL

Funções e Portas Lógicas

SUMÁRIO

2. Funções Lógicas	2
2.1 Introdução	2
2.2 Funções Lógicas Básicas	3
2.2.1 Função Lógica NÃO (NOT)	3
2.2.2 Função Lógica E (AND)	3
2.2.3 Função Lógica OU (OR)	5
2.2.4 Função Lógica NÃO E (NAND)	7
2.2.5 Função Lógica NÃO OU (NOR)	7
2.3 Interligação entre Expressões, Circuitos e Tabela da Verdade	10
2.3.1 Expressões Booleanas Obtidas de Circuitos Lógicos	10
2.3.2 Circuitos Lógicos Obtidos de Expressões Booleanas	10
2.3.3 Tabelas da Verdade obtidas de Expressões Booleanas	12
2.3.4 Expressões Booleanas Obtidas de Tabelas da Verdade	12
2.3.5 Equivalência Entre Blocos Lógicos	13
Exercícios de Fixação	15
2.4 Circuitos Comerciais Básicos	19

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, esses 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, entre outros, apenas um pequeno grupo de circuitos lógicos básicos (que realizam funções lógicas), que são conhecidos como portas OU, E, NÃO e flip-flops.

Então, um circuito digital emprega um conjunto de funções lógicas, onde função é a relação existente entre as variável independente e a variável dependente (função) assim como aprendemos na matemática. 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 como o 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, ou seja, 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 em volts compatível com os circuitos digitais utilizados 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”)

Lógica Positiva: 0 V → 0 lógico +5 V → 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. Vamos conhecê-las...

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: A Simbologia representa um conjunto de circuitos eletrônicos que implementa a função lógica correspondente. A **Porta Lógica** Inversora é representada pelo seguinte símbolo:



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

$$Y = \bar{A} \quad (\text{esta equação representa a função lógica correspondente})$$

Entrada	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.

$$Y = f(A,B) = A.B = B.A \quad (\text{produto lógico})$$

Símbolo:

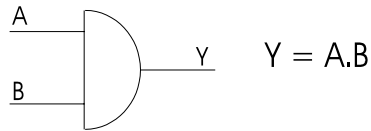
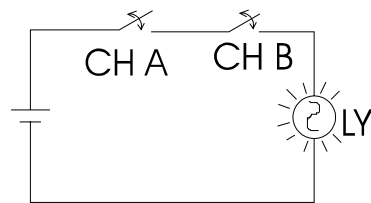


Tabela da Verdade:

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

Exemplo:



Convenção:

CH A aberta = 0

CH A fechada = 1

CH B aberta = 0

CH B fechada = 1

Lâmpada apagada = 0

Lâmpada acesa = 1

Se analisarmos todas as situações possíveis das chaves verifica-se que a lâmpada acende somente quando as chaves A e B estiverem fechadas (assume 1 somente quando todas as entrada forem 1).

2.2.2.1 Função lógica AND com mais de duas variáveis de entrada.

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

Comutatividade

Associatividade (propriedades aritméticas...)

Símbolo representativo:

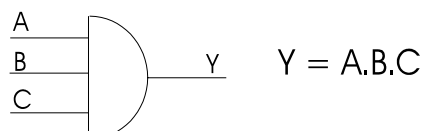
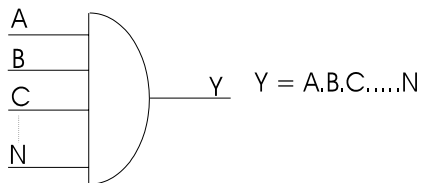


Tabela da Verdade(3 var. $\rightarrow 2^3$ combinações)

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:

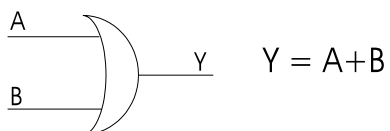


A tabela da verdade terá 2^N combinações na entrada e Y será 1 somente quando todas as entradas forem 1.

2.2.3 Função Lógica OU (OR)

A função lógica OR de duas variáveis realiza a seguinte operação de dependência:

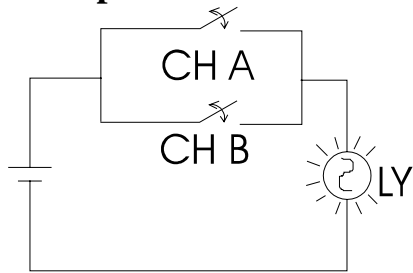
$$Y = f(A,B) = A+B \text{ (soma lógica)}$$

Símbolo:**Tabela da Verdade:**

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

1 1 1

Exemplo:



Utiliza-se as mesmas convenções adotadas para a porta AND. Ao analisar-se todas as situações que as chaves podem assumir verifica-se que a lâmpada acende quando CH A OU CH B OU ambas estiverem ligadas (a saída assume 0 somente quando todas as entradas forem 0).

2.2.3.1 Função lógica OR de mais de duas variáveis de entrada

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

Comutatividade
 Associatividade

Símbolo:

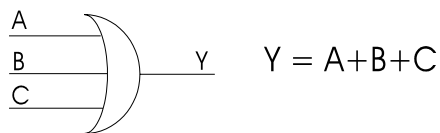
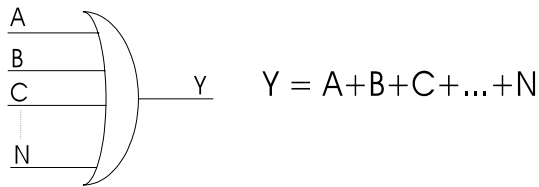


Tabela da Verdade:

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:



2.2.4 Função Lógica NÃO E (NAND)

Como o próprio nome diz esta função é uma combinação das funções AND e INVERSOR, onde é realizada a função E invertida.

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

Tabela da Verdade:

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

Símbolo:



2.2.5 Função Lógica NÃO OU (NOR)

Como o próprio nome diz esta função é uma combinação das funções OR e INVERSOR, onde é realizada a função OU invertida.

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








Tabela da Verdade:

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

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.

Tabela resumo das Portas (blocos) lógicas básicas:

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

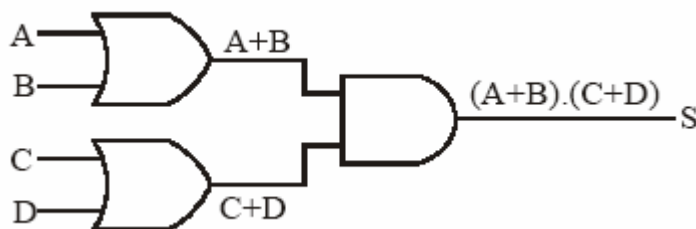
2.3 Interligação entre Expressões, Circuitos e Tabela da Verdade

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

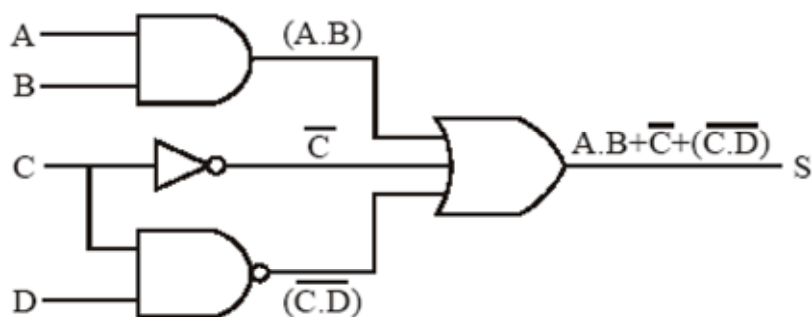
Todo o circuito lógico executa uma função booleana e, por mais complexo que seja, é formado pela interligação das portas lógicas básicas. Assim, pode-se obter a expressão booleana que é executada por um circuito lógico qualquer.

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



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 figura a seguir 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 NÃO E abaixo, e na entrada de algumas portas, como será visto mais adiante (-----o).

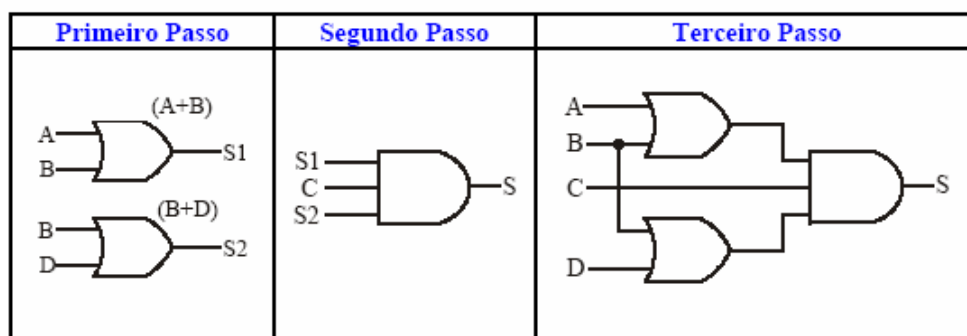


2.3.2 Circuitos Lógicos Obtidos de Expressões Booleanas

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



Exercícios.

Esboce os circuitos obtidos a partir das seguintes expressões:

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

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:

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

Para exemplificar este processo, utiliza-se a expressão:

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

A expressão contém 4 variáveis: A, B, C e D, logo, existem $2^4=16$ 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
A	B	C	D	$A.\bar{B}.C$	$A.\bar{D}$	$\bar{A}.B.D$	Final
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	1	1
0	1	1	0	0	0	0	0
0	1	1	1	0	0	1	1
1	0	0	0	0	1	0	1
1	0	0	1	0	0	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	0	0	1
1	1	0	0	0	1	0	1
1	1	0	1	0	0	0	0
1	1	1	0	0	1	0	1
1	1	1	1	0	0	0	0

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 conseqüentemente, 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 (a)
1	0	0	0
1	0	1	1 (b)
1	1	0	1 (c)
1	1	1	1 (d)

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

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


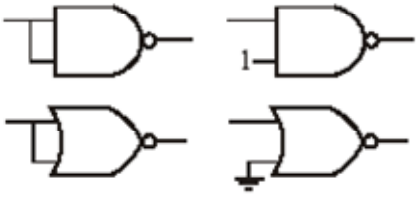








Para se obter a expressão basta realizar a soma booleana de cada termo acima:

$$S = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$$

Nota-se que o método permite obter, de qualquer tabela, uma expressão padrão formada sempre pela soma de produtos. Utilizando a álgebra de Boole e também mapas de Karnaugh é possível realizar a simplificação de expressões de funções lógicas, possibilitando a obtenção de circuitos reduzidos e portanto mais baratos. Estas técnicas não fazem parte de nosso objetivo, mas é interessante conhecê-las através de uma bibliografia adicional.

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 conseqüente minimização do custo do sistema.

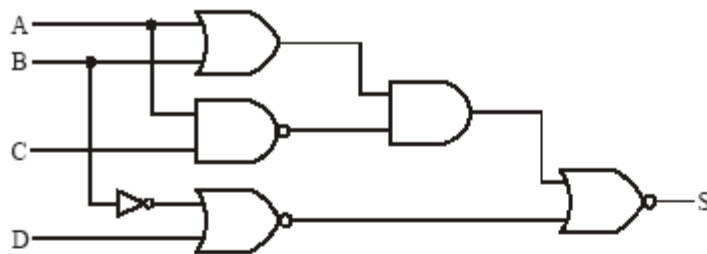
BLOCO LÓGICO	BLOCO EQUIVALENTE
	
 $S = \overline{A+B}$	 $S = \overline{A \cdot B}$
 $S = A+B$	 $S = \overline{\overline{A} \cdot \overline{B}}$
 $S = \overline{A \cdot B}$	 $S = \overline{A+B}$
 $S = A \cdot B$	 $S = \overline{\overline{A+B}}$

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

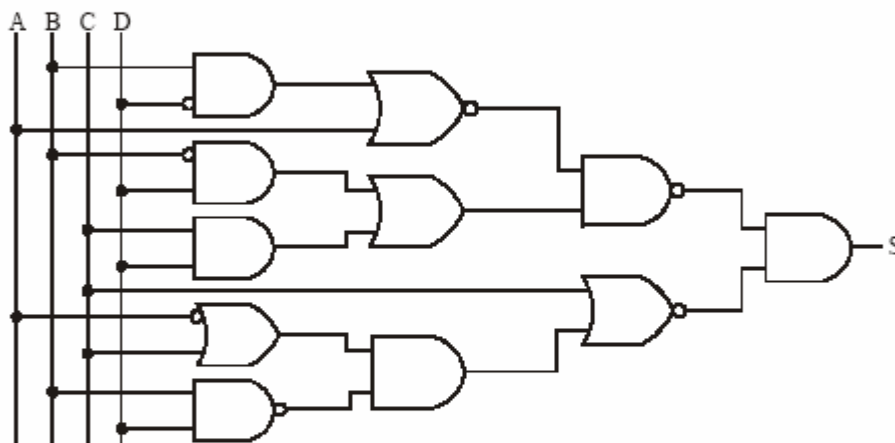
Exercícios de Fixação

Determine as expressões das funções lógicas dos circuitos abaixo:

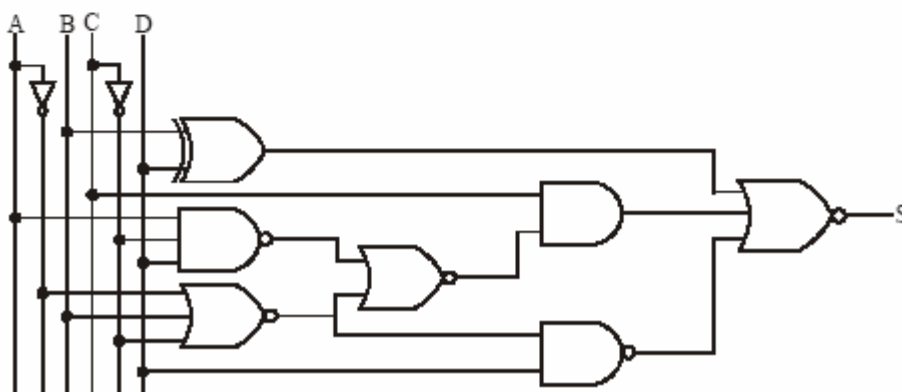
a) Circuito 1



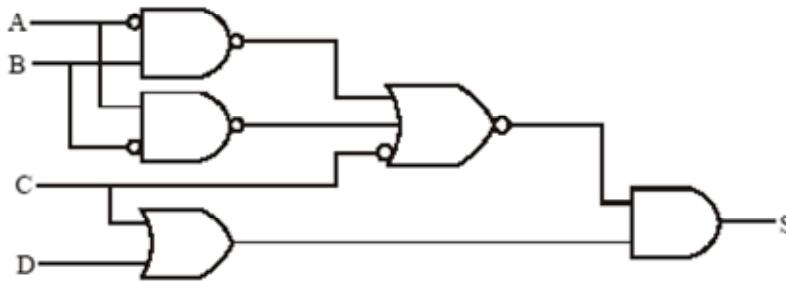
b) Circuito 2



c) Circuito 3



d) Circuito 4



2) Desenhe o circuito que executa as seguintes expressões:

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

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

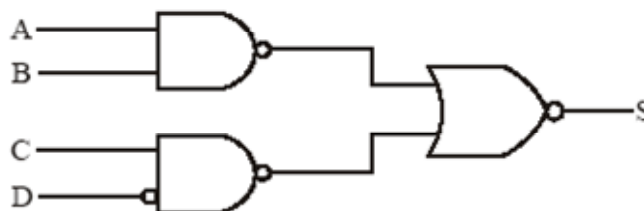
c) $S = (A \oplus B) \cdot \overline{[A \cdot \overline{B} + (\overline{B + D}) + C \cdot \overline{D} + (\overline{B \cdot C})]} + \overline{A} \cdot B \cdot \overline{C}$

3) Levante a tabela verdade das seguintes expressões:

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

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

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



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

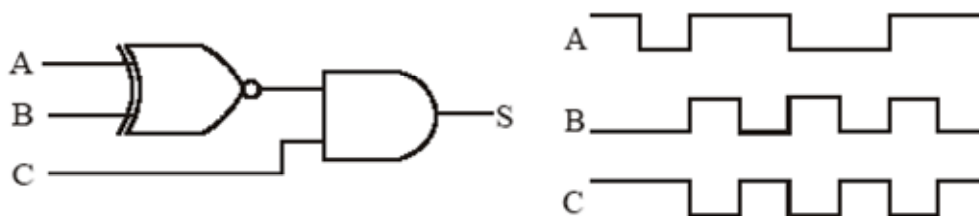
a) Tabela 1

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

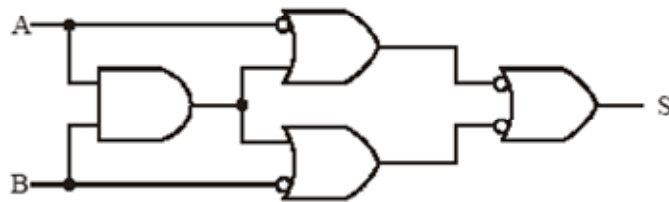
b) Tabela 2

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	0
1	1	1	1	1

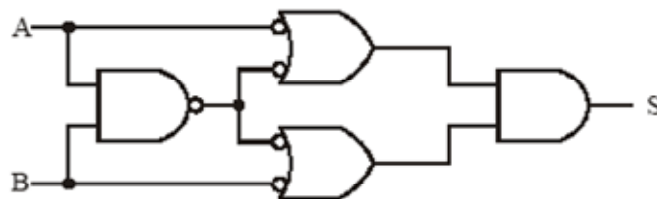
6) Desenhe o sinal de saída do circuito abaixo:



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



8) Mostre que o circuito é um circuito Coincidência.



9) Prove que:

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

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

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

11) Esquematize o circuito Coincidência usando apenas porta NOU.

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

13) Esquematize o circuito Coincidência, utilizando apenas 4 portas NOU.

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

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

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

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

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

2.4 Circuitos Comerciais Básicos

Sabe-se que todos os circuitos digitais, por mais complexos que sejam, são obtidos através de portas lógicas. As portas lógicas, por sua vez, não são encontradas comercialmente de uma forma discreta (como os resistores) e sim encapsuladas em Circuitos Integrado – CI'S, que serão melhor explorados nas aulas de Instrumentação.

TODO circuito integrado possui um conjunto de contatos externos, denominados “pinos” (leads ou ainda, terminais), cada qual com sua função específica. São numerados a partir do número “1” no sentido anti-horário. O pino “1” é identificado olhando-se o CI pela parte superior, conforme mostra a Figura 1. TODO circuito integrado possui um manual no qual a função de cada um de seus pinos está descrita. Os CI's que implementam funções lógicas podem possuir uma ou mais portas, geralmente todas de uma mesma função.

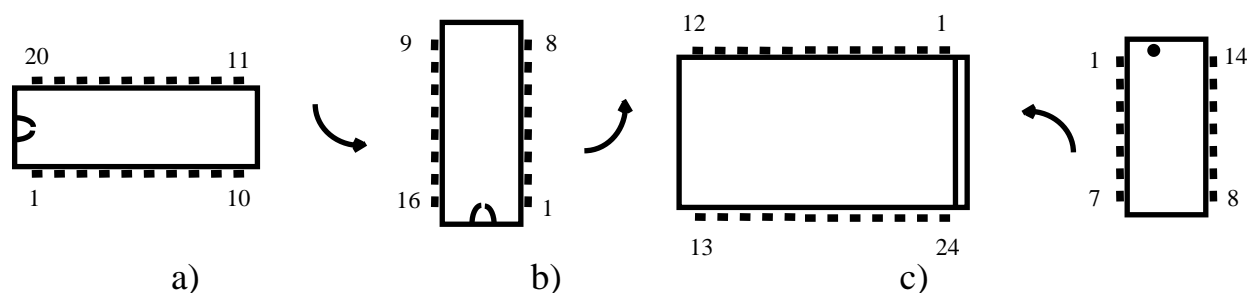
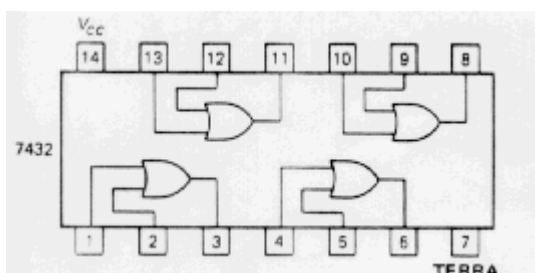
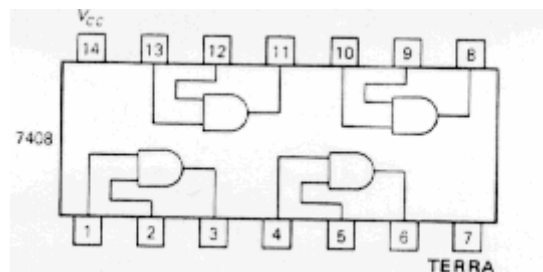
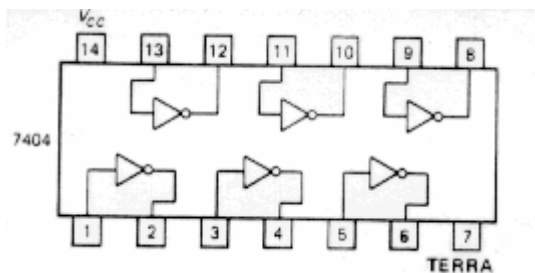
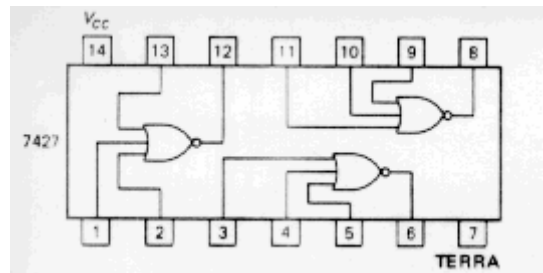
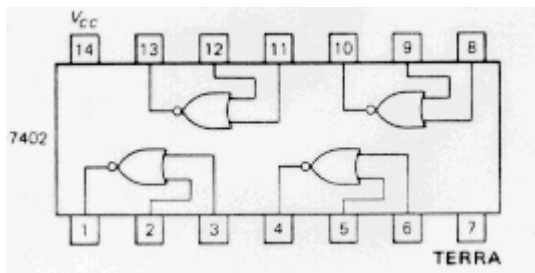
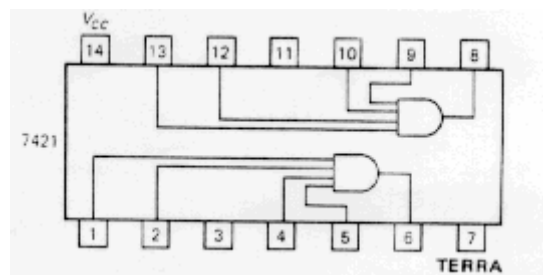
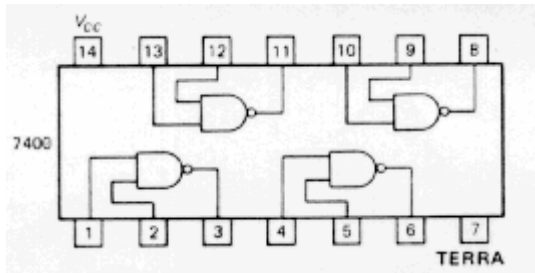


Figura 1 - Vista superior, em diferentes posições, da pinagem de um CI e suas diferentes formas de indicação. a) CI de 20 pinos com pino “1” identificado por “chanfro”; b) CI de 16 pinos orientado em outra direção; c) CI de 24 pinos com “traço” de identificação do pino “1” (repare que a contagem dos pinos continua sendo realizada no sentido anti-horário); d) CI de 14 pinos com pino “1” identificado por um “ponto”;

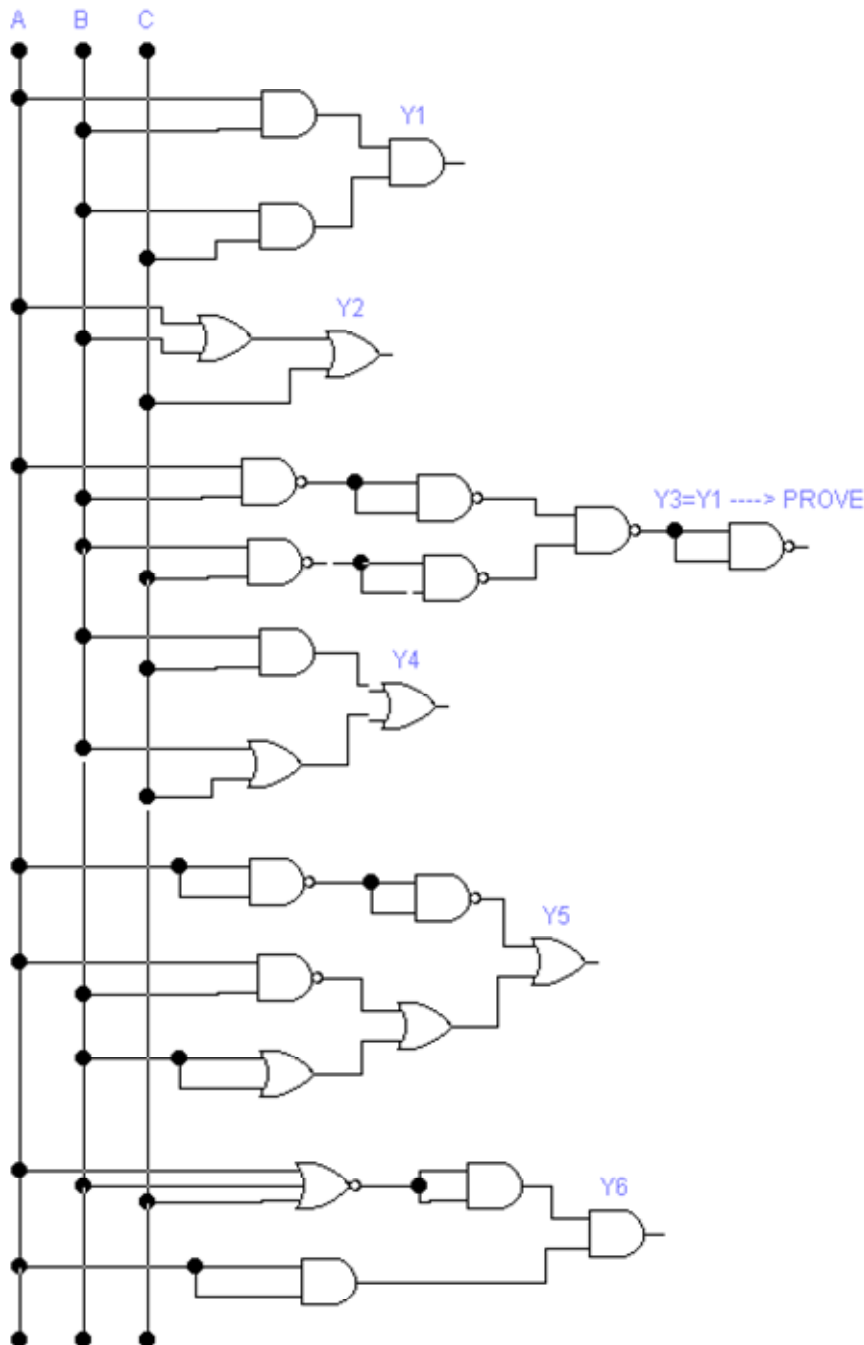
Deve-se tomar todo o cuidado possível no manuseio de circuitos integrados, pois os mesmos podem vir a ser facilmente danificados através das DESCARGAS ELETROSTÁTICAS ACUMULADAS quando tocamos seus terminais. Portanto, jamais deve-se tocar os pinos de um CI, ou as pistas de uma placa de circuito impresso sem a proteção adequada para o desvio destas descargas.

Abaixo vocês podem observar alguns CI's comercialmente disponíveis. Identifique qual a função desempenhada por cada CI.



Exercício:

Considerando os CIs comerciais vistos no item anterior, realize o projeto (o diagrama com os CIs utilizados) e obtenha a expressão e tabela verdade para o circuito digital abaixo:



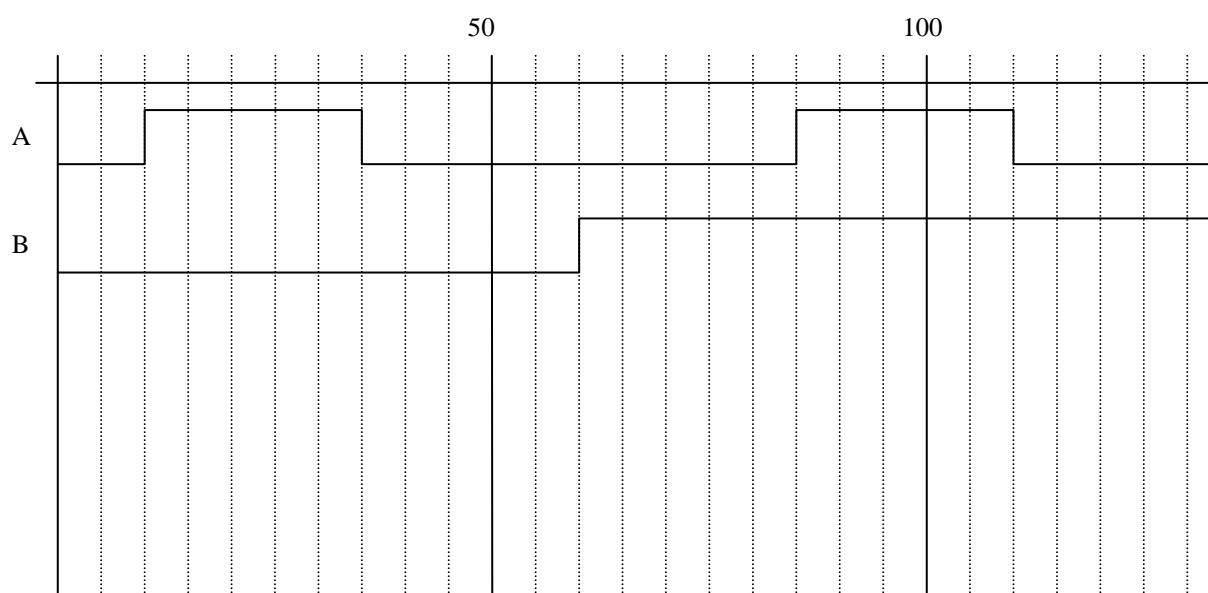
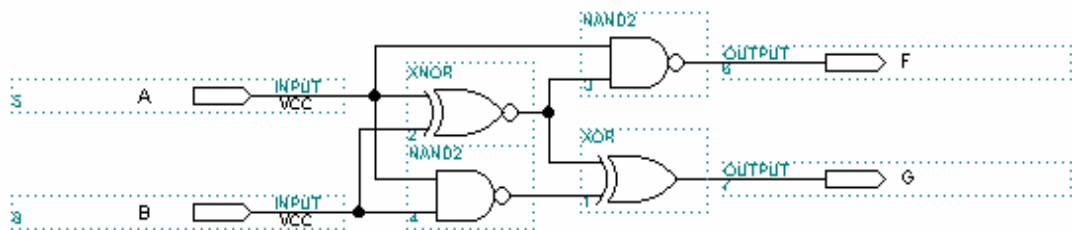
Exercícios Desafio:

1) A tabela verdade de um subtrator de 1 bit é indicada abaixo:

A	B	D	BL
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

A e B são os bits a serem subtraídos, D é a diferença entre eles e BL indica o *Borrow* (o “vem 1”, análogo ao carry do somador). Projete um circuito para executar a diferença entre as entradas A e B e descreva sua função lógica.

2) Considere o circuito abaixo. Desenhe a forma de onda das saídas do circuito para as entradas indicadas no diagrama de sinal abaixo. Considere um *delay* de 5 unidades de tempo para cada *gate(PORTA)* básico (observe o delay do XNOR – toda a porta possui algum...)



3) Considere o circuito abaixo. No tempo t_0 a chave é fechada e no tempo t_1 , aberta. Desenhe a forma de onda dos pontos internos B e C e da saída do circuito. Assuma que o *delay* de cada *gate* é de uma unidade de tempo.

