Lógica Digital

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

Lógica Digital

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

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

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

Lógica Digital

Em geral, usa - se uma faixa de tensão para representar o valor falso ou verdadeiro de uma variável lógica.
Lógica Positiva -a tensão mais positiva representa o valor V (1) e a mais negativa o valor F(0).
Lógica Negativa - o valor V é representado pela tensão mais negativa (1) e F pela tensão mais positiva (0).
Lógica Mista - no mesmo sistema, usam-se as lógicas positiva e negativa.

Circuitos Digitais

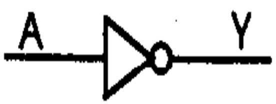
O passo seguinte na evolução dos sistemas digitais foi a implementação dos sistemas lógicos (funções lógicas Booleanas), utilizando-se dispositivos eletrônicos (circuitos digitais), obtendo-se assim, rapidez na solução dos problemas (descritos pela álgebra de Boole). Nos circuitos digitais tem-se somente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

Funções Lógicas

2.2.1 Função Lógica NÃO (NOT)

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

Símbolo:



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

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

$Y = \overline{A}$

Entrada → ← Saída

A	Y
0	1
1	0

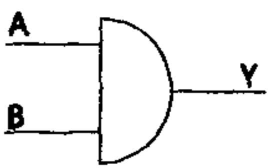
Funções Lógicas

2.2.2 Função Lógica E (AND)

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

$Y = f(A,B) = A.B = B.A$ (produto lógico)

Símbolo:

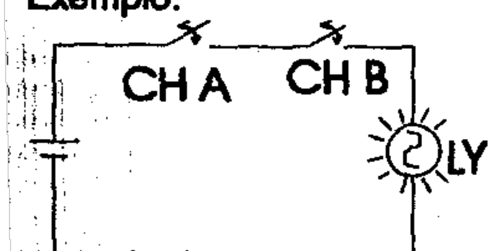


$Y = A.B$

Tabela da Verdade

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

Exemplo:



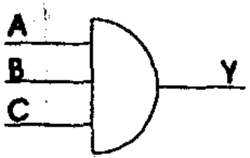
<u>Convenção:</u>
CH A aberta = 0
CH A fechada = 1
CH B aberta = 0
CH B fechada = 0
Lâmpada apagada = 0
Lâmpada acesa = 1

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).

Funções Lógicas

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

$$Y = \underbrace{A.B.C = B.A.C = C.A.B}_{\text{Comutatividade}} = \underbrace{(A.B).C = A.(B.C)}_{\text{Associatividade}}$$



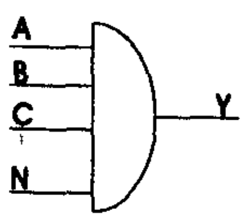
Y=A.B.C

Tabela da Verdade
(3 var. → 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:

Y=A.B.C...N



Y = A.B.C.....N

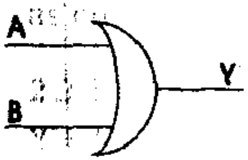
A tabela da verdade terá 2^N combinações na entrada e Y será 1 somente quando todas as entradas forem 1.

Funções Lógicas

2.2.3 Função Lógica OU (OR)

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

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

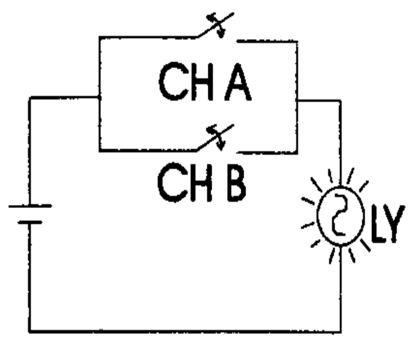


$Y=A+B$

Tabela da Verdade

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

Exemplo:



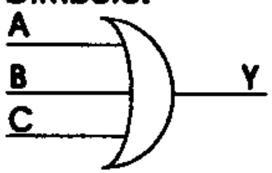
Utiliza-se as mesmas convenções adotadas para a porta AND. Ao analisar-se todas as situações que as chaves podem assumir verifica-se que a lâmpada acende quando CH A **OU** CH B **OU** ambas estiverem ligadas (a saída assume 0 somente quando todas as entradas forem 0).

Funções Lógicas

2.2.3.1 Função lógica OR de mais de duas variáveis de entrada

$$Y = \underbrace{A+B+C = C+B+A = B+C+A}_{\text{Comutatividade}} = \underbrace{A+(B+C) = (A+B)+C}_{\text{Associatividade}}$$

Símbolo:

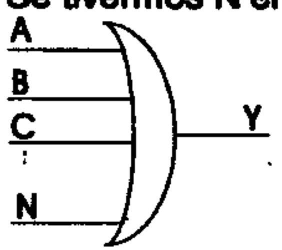


Y=A+B+C

Tabela da Verdade

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

Se tivermos N entradas, teremos:



Y=A+B+C+...+N

Funções Lógicas

2.2.4 Função Lógica NÃO E (NAND)

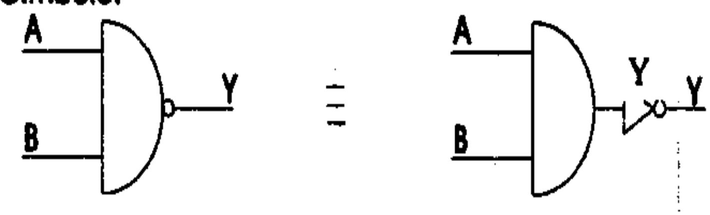
Como o próprio nome diz esta função é uma combinação das funções AND e INVERSOR, onde é realizada a função E invertida.

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

Tabela da Verdade

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

Símbolo:



Funções Lógicas

2.2.5 Função Lógica NÃO OU (NOR)

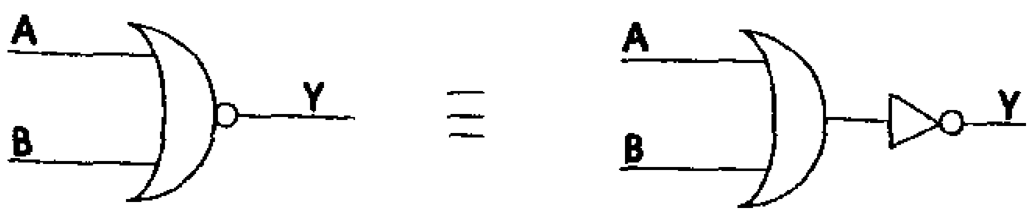
Como o próprio nome diz esta função é uma combinação das funções OR e INVERSOR, onde é realizada a função OU invertida.

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

Tabela da Verdade

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

Símbolo:



BLOCOS LÓGICOS BÁSICOS

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

Funções Lógicas

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.

Funções Lógicas

2.3 Interligação entre Expressões, Circuitos e Tabela da Verdade

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

2.3.1 Expressões Booleanas Obtidas de Circuitos Lógicos

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

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

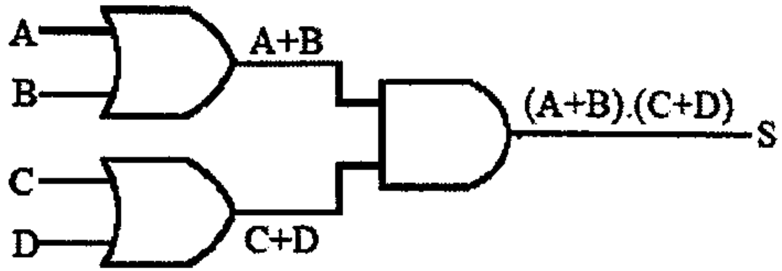


Figura 2.17 – Circuito lógico.

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

Funções Lógicas

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

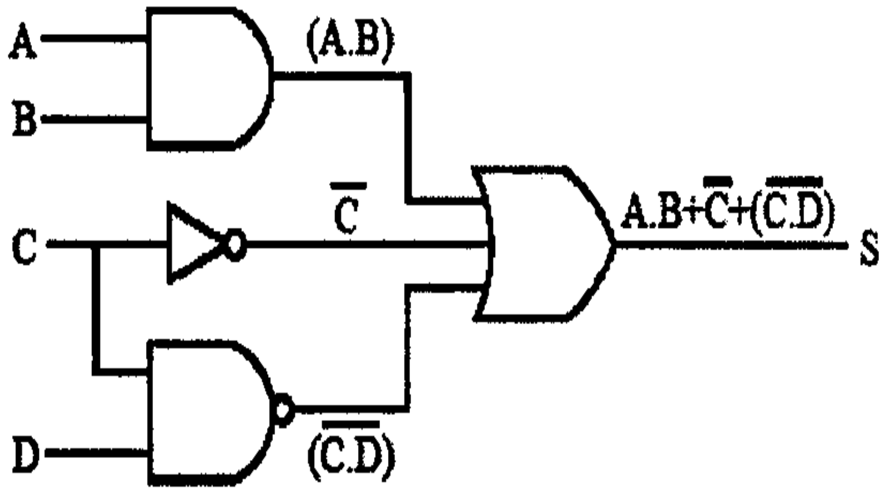


Figura 2.18 – Circuito lógico.

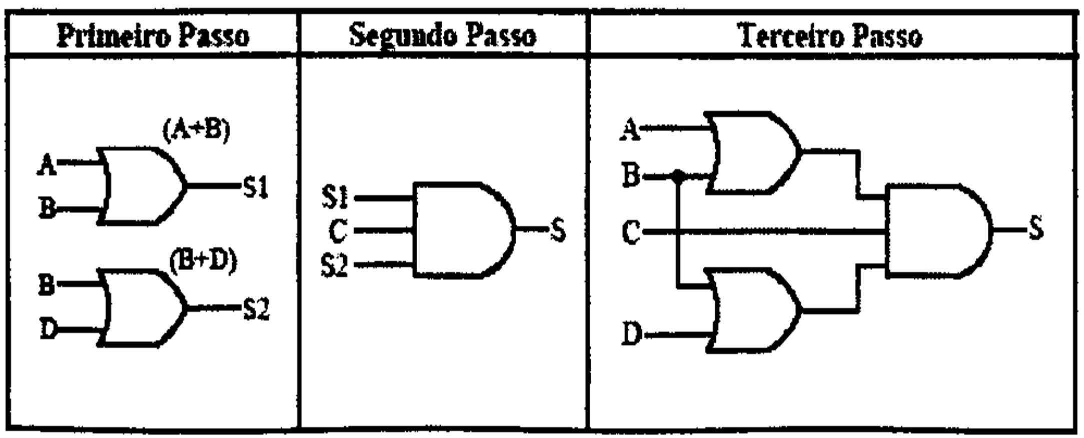
Funções Lógicas

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.

Funções Lógicas

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.



Funções Lógicas

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

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