PCS 3115 (PCS2215)

Sistemas Digitais I

Módulo 05 – Álgebra Booleana

Prof. Dr. Marcos A. Simplicio Jr.

versão: 3.0 (Jan/2016)

Conteúdo

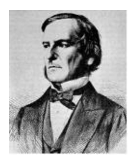
- Conceitos básicos
- Teoremas de 1 variável
- Teoremas de 2 variáveis
- Teoremas de n variáveis

- Referência: Cap. 4.1 do início até 4.1.4 (inclusive) do livro-texto.

Álgebra

- Definida por
  - Um conjunto de operações válidas
  - Um conjunto de valores que cada variável pode assumir
- Construídas usando símbolos
  - Letras: variáveis
  - Operadores lógicos
- Exemplo: álgebra com números reais
  - $(a+b) \cdot c/\sqrt{a}$

Álgebra Booleana e de Chaveamento



- 1854: George Boole cria um sistema algébrico baseado em um número finito de valores
  - Nascimento da Álgebra Booleana

- 1938: Shannon adapta a álgebra booleana para modelar o funcionamento de circuitos a relés
  - Chave aberta (sem corrente): 0

  - Chave fechada (com corrente): 1

- *Álgebra de chaveamento*: estrutura da Álgebra Booleana quando aplicada para descrever/analisar funções lógicas
  - Porém, a literatura costuma usar os termos como sinônimos (e nós também…)

Álgebra Booleana

- Dois valores possíveis: **{1, 0}**
  - Ou {Verdadeiro, Falso}, {V, F}, {on, off}, {alto, baixo}...
- Em outras palavras:
  - Se $a \neq 0 \rightarrow a = 1$
  - Se $a \neq 1 \rightarrow a = 0$

Álgebra Booleana

- Relação entre Álgebra Booleana e Eletrônica Digital
  - Obs.: notação positiva (V = 1, F =0)

Valor Lógico	Nível Lógico	Nível de Tensão TTL	Nível de Tensão CMOS (tipo AHC)
F	**0**	**0 – 0.8 V**	**0 – 1.5 V**
V	**1**	**2 – 5 V**	**3.5 – 5 V**

Tensões reconhecidas por *portas lógicas*
(obs.: CMOS AHC compatível com TTL)

Circuitos lógicos

- Dispositivos digitais que *implementam funções lógicas*.
  - Operam sobre um ou mais sinais lógicos de entrada
  - Produzem uma (e apenas uma) saída, dependente da função lógica implementada

Tabela-verdade

- Tabela com resultados para todas as possíveis combinações de entrada
  - Cada função tem sua própria tabela verdade

variáveis de entrada

todas as possíveis combinações na entrada

X	Y	Z	f(X,Y,Z)
0	0	0	f(0,0,0)
0	0	1	f(0,0,1)
0	1	0	f(0,1,0)
0	1	1	f(0,1,1)
1	0	0	f(1,0,0)
1	0	1	f(1,0,1)
1	1	0	f(1,1,0)
1	1	1	f(1,1,1)

função

valor de saída para uma dada combinação de entradas

Álgebra Booleana: operações básicas

- **Complemento** (NOT)
  - Também chamada de “negação” ou “inversão”
  - Operação unária (i.e., aplicada sobre uma variável por vez)
  - **Resultado**: valor oposto ao valor da entrada
    - Se X = 0, então X’ = 1
    - Se X = 1, então X’ = 0

- Símbolos:
  - $\overline{X}$, ~X, **X’**, ¬X, NOT(X)

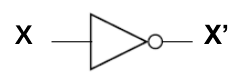
Símbolo usado na disciplina

- Tabela-verdade

X	X’
0	1
1	0

- Porta lógica

(representação gráfica)



Álgebra Booleana: operações básicas

- Operação E (AND)
  - Também chamada de “multiplicação lógica”
  - **Resultado**: 1 se e somente **todos** os termos forem 1

- Símbolos:

{ • , ∧ }

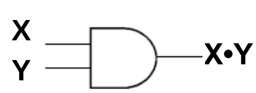
Símbolo usado na disciplina

- Tabela-verdade

X	Y	X•Y
0	0	0
0	1	0
1	0	0
1	1	1

- Porta lógica

(representação gráfica)



Álgebra Booleana: operações básicas

- Operação OU (OR)
  - Também chamada de “adição lógica”
  - **Resultado**: 1 se **<u>qualquer um</u>** dos termos for 1

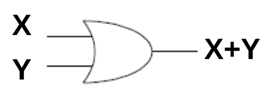
- Símbolos:
  { **+** , v }

  Símbolo usado na disciplina

- Tabela-verdade

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

- Porta lógica
  (representação gráfica)



Operadores: ordem de precedência

- Do nível de parêntesis mais *interno* para o nível mais *externo*
  1. Complemento de variável individual (NOT)
  2. Operação E
  3. Operação OU

Sequência de avaliação

NOT > E > OU
' > • > +

Ex.: $\overbrace{\overset{1}{X'} + \underbrace{Y \bullet Z \bullet \overset{1}{W'}}_{2}}^{3}$

Teoremas de 1 variável

- (T1) X+0=X (T1') X•1=X → identidade
- (T2) X+1=1 (T2') X•0=0 → elemento nulo
- (T3) X+X=X (T3')X•X=X → idempotência
- (T4) (X')'=X → involução
- (T5) X+X'=1 (T5') X•X'=0 → complemento

- ➔ Podem ser provados por indução perfeita: listando todas as variáveis
  - Ou seja: basta verificar as tabelas-verdade

Teoremas de 2 ou 3 variáveis

(T6) X+Y=Y+X	(T6') X•Y=Y•X	→ comutativa
(T7) (X+Y)+Z=X+(Y+Z)	(T7') (X•Y)•Z=X•(Y•Z)	→ associativa
(T8) X•(Y+Z)=X•Y+X•Z	(T8') X+Y•Z=(X+Y)•(X+Z)	→ distributiva
(T9) X+X•Y=X	(T9') X•(X+Y)=X	→ cobertura
(T10) X•Y+X•Y'=X	(T10') (X+Y)•(X+Y')=X	→ combinação
(T11) X•Y+X'•Z+Y•Z = X•Y+X'•Z		→ consenso
(T11') (X+Y)•(X'+Z)•(Y+Z) = (X+Y)•(X'+Z)		

Teoremas de n variáveis

(TC) $X+X+...+X = X$ (TC') $X \cdot X \cdot ... \cdot X = X$ → idempotência generalizada

(T13) $(X_1 \cdot X_2 \cdot ... \cdot X_n)' = X_1' + X_2' + ... + X_n'$
(T13) $(X_1 + X_2 + ... + X_n)' = X_1' \cdot X_2' \cdot ... \cdot X_n'$ → De Morgan

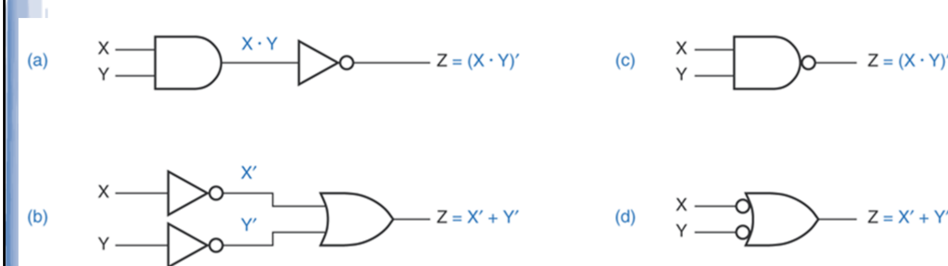
(T14) $[F(X_1, X_2, ..., X_n, +, \cdot)]' = F(X_1', X_2', ..., X_n', \cdot, +)$ → De Morgan generalizado

(T15) $F(X_1, X_2, ..., X_n) = X_1 \cdot F(1, X_2, ..., X_n) + X_1' \cdot F(0, X_2, ..., X_n)$
(T15') $F(X_1, X_2, ..., X_n) = [X_1 + F(0, X_2, ..., X_n)] \cdot [X_1' + F(1, X_2, ..., X_n)]$

→ Expansão de Shannon

Álgebra Booleana: operações derivadas

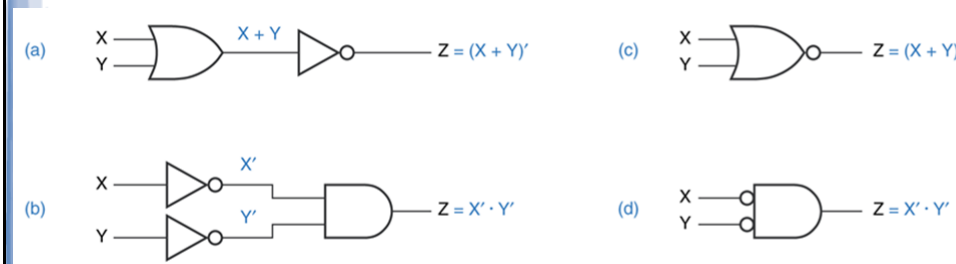
- **Circuitos equivalentes**, de acordo com Teorema de De Morgan: **NAND** ou **NÃO-E**
  - Combinação de inversão e operação E



From *Digital Design: Principles and Practices*, Fourth Edition, John F. Wakerly, ISBN 0-13-186389-4.

Álgebra Booleana: operações derivadas

- **Circuitos equivalentes**, de acordo com Teorema de De Morgan: **NOR** ou **NÃO-OU**
  - Combinação de inversão e operação OU



From *Digital Design: Principles and Practices*, Fourth Edition, John F. Wakerly, ISBN 0-13-186389-4.

Álgebra Booleana: operações derivadas

- Operação OU-Exclusivo (XOR)
  - X ⊕ Y = (X'•Y) + (X•Y')
  - Resultados (equivalentes)
    - 1 se **apenas uma** das entradas for 1, 0 caso contrário
    - 1 se **ambas as entradas são diferentes**, 0 caso contrário

- Símbolo:
  { ⊕ }

- Tabela-verdade

X	Y	X⊕Y
0	0	0
0	1	1
1	0	1
1	1	0

- Porta lógica
  (representação gráfica)

Exercícios

1. Simplificar a expressão $(x+y+z)\bullet(x\bullet y'+y\bullet z+x\bullet z')$
2. Calcular (desenvolver) $f = (x\bullet y+y'\bullet z+x\bullet z')'$
3. Comprovar que $[(x + x'\bullet y) = (x+y)]$ por Diagrama de Venn.

Exercícios: Respostas

1. Simplificar a expressão $(x+y+z)\bullet(x\bullet y'+y\bullet z+x\bullet z')$

$(x+y+z)\bullet(x\bullet y+y\bullet z+x\bullet z') =$

$= x\bullet x\bullet y'+x\bullet y\bullet z+x\bullet x\bullet z'+x\bullet y\bullet y'+y\bullet y\bullet z+x\bullet y\bullet z'+x\bullet y'\bullet z+y\bullet z\bullet z+x\bullet z\bullet z'=$

$= x\bullet y' + x\bullet y\bullet z + x\bullet z' + x\bullet 0 + \underline{y\bullet z} + x\bullet y\bullet z' + x\bullet y'\bullet z + \underline{y\bullet z} + x\bullet 0 =$

$= x\bullet y' + x\bullet y\bullet z + x\bullet z' + y\bullet z + x\bullet y\bullet z' + x\bullet y'\bullet z =$

$= x\bullet y'\bullet(1+z) + y\bullet z\bullet(x+1) + x\bullet z'\bullet(1+y) =$

$= x\bullet y' + y\bullet z + x\bullet z' =$

$= x\bullet(y'+z') + y\bullet z =$

$= x\bullet(y\bullet z)' + y\bullet z = x\bullet a'+a =$

$= [(x\bullet a')'\bullet a']' = [(x'+a)\bullet a']'$ (De Morgan)

$= [x'\bullet a' + a\bullet a']' = [x'\bullet a' + 0]' = [x'\bullet a']' = x + a$

$= x + y\bullet z$

Exercícios: Respostas

2. Calcular (desenvolver) $f = (x \bullet y + y' \bullet z + x \bullet z')'$

$f = (x \bullet y + y' \bullet z + x \bullet z')'$

$= (x \bullet (y+z') + y' \bullet z)'$ T8

$= [x \bullet (y+z') + (y+z')']'$ T13

$= (x \bullet a + a')'$

$= (x \bullet a)' \bullet a$

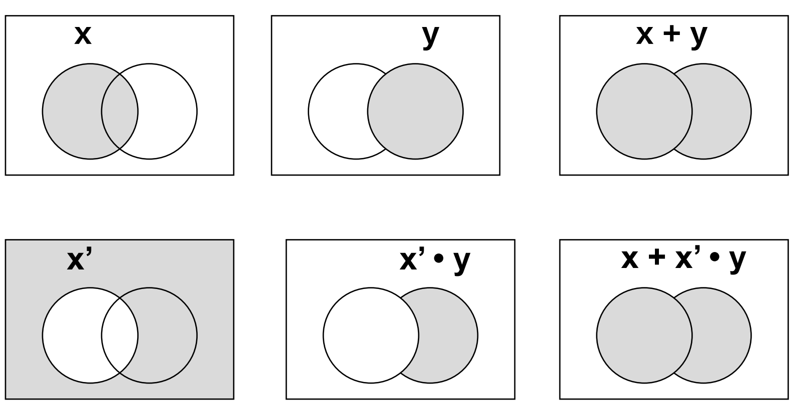
$= (x' + a') \bullet a$

$= x' \bullet a + a \bullet a'$ T5

$= x' \bullet a = x' \bullet (y + z')$

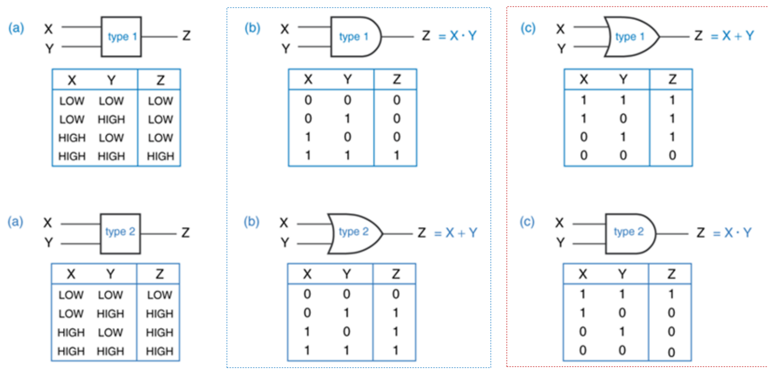
Exercícios: Respostas

3. Comprovar que $[(x + x' \bullet y) = (x+y)]$ por Diagrama de Venn.



Princípio da Dualidade

- Qualquer teorema em álgebra de chaveamento continua verdadeiro se trocarmos 0 por 1, e • por +
  - ➔ Pode-se projetar circuitos com lógica positiva (“alto = 1”) ou negativa (“alto = 0”), usando o mesmo conjunto de portas lógicas



(a) type 1

X	Y	Z
LOW	LOW	LOW
LOW	HIGH	LOW
HIGH	LOW	LOW
HIGH	HIGH	HIGH

(b) type 1, $Z = X \cdot Y$

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

(c) type 1, $Z = X + Y$

X	Y	Z
1	1	1
1	0	1
0	1	1
0	0	0

(a) type 2

X	Y	Z
LOW	LOW	LOW
LOW	HIGH	HIGH
HIGH	LOW	HIGH
HIGH	HIGH	HIGH

(b) type 2, $Z = X + Y$

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

(c) type 2, $Z = X \cdot Y$

X	Y	Z
1	1	1
1	0	0
0	1	0
0	0	0

Circuito desejado **Lógica positiva** **Lógica negativa**

Tabela Verdade e Expressões Algébricas

- Tabela verdade: representação um tanto básica...
  - Todas as entradas/saídas ➔ pode ficar grande
- Mas qualquer informação na Tabela Verdade pode ser representada algebricamente!
  - Para isso, precisamos de algumas definições...

variáveis de entrada → | função ←

todas as possíveis combinações na entrada

X	Y	Z	f(X,Y,Z)
0	0	0	f(0,0,0)
0	0	1	f(0,0,1)
0	1	0	f(0,1,0)
0	1	1	f(0,1,1)
1	0	0	f(1,0,0)
1	0	1	f(1,0,1)
1	1	0	f(1,1,0)
1	1	1	f(1,1,1)

valor de saída para uma dada combinação de entradas

Definições (1/3)

- **Literal**: variável de chaveamento ou seu complemento.
- **Termo produto**: único literal, ou produto lógico de 2 ou mais literais.
  - Ex.: z’, w•x•y, x•y’•z, w’•y’•z
- **Expressão da soma de produtos**: soma lógica de termos produto.
  - Ex.: z’ + w•x•y + x•y’•z + w’•y’•z

Definições (2/3)

- **Termo soma**: único literal, ou soma lógica de 2 ou mais literais.
  - Ex.: z’, w+x+y, x+y’+z, w’+y’+z
- **Expressão do produto de somas**: produto lógico de termos soma.
  - Ex.: z’•(w+x+y)•(x+y’+z)•(w’+y’+z)
- **Termo Normal**: termo soma ou produto em que nenhuma variável aparece mais de uma vez.
  - Um termo não-normal pode ser simplificado com os teoremas T3 (X+X=X, X•X=X) e T5 (X+X'=1, X•X’=0)

Definições (3/3)

- **Mintermo**: termo produto normal com n literais.
  - Existem 2^n mintermos.
- **Maxtermo**: termo soma normal com n literais.
  - Existem 2^n maxtermos.
- Existe uma relação direta entre **Tabela Verdade e mintermos/maxtermos**!
  - **Mintermo**: termo produto cuja saída é 1.
  - **Maxtermo**: termo soma cuja saída é 0.

Tabela verdade e (min/max)termos

- Exemplo para 3 entradas
  - mintermos: produto (ANDs) deve dar 1
  - maxtermo: soma (ORs) deve dar 0

linha	X	Y	Z	f(X,Y,Z)	*exemplo*	mintermo	maxtermo
0	0	0	0	f(0,0,0)	*1*	X'•Y'•Z'	X+Y+Z
1	0	0	1	f(0,0,1)	*0*	X'•Y'•Z	X+Y+Z'
2	0	1	0	f(0,1,0)	*0*	X'•Y•Z'	X+Y'+Z
3	0	1	1	f(0,1,1)	*1*	X'•Y•Z	X+Y'+Z'
4	1	0	0	f(1,0,0)	*1*	X•Y'•Z'	X'+Y+Z
5	1	0	1	f(1,0,1)	*0*	X•Y'•Z	X'+Y+Z'
6	1	1	0	f(1,1,0)	*1*	X•Y•Z'	X'+Y'+Z
7	1	1	1	f(1,1,1)	*1*	X•Y•Z	X'+Y'+Z'

Tabela verdade: Soma Canônica

- Ou primeira forma canônica: **soma dos mintermos** correspondentes às **saídas 1**
  - No exemplo: X'•Y'•Z' + X'•Y•Z + X•Y'•Z' + X•Y•Z' + X•Y•Z

    $F = \sum_{X,Y,Z}(0,3,4,6,7)$

linha	X	Y	Z	f(X,Y,Z)	*exemplo*	mintermo
0	0	0	0	f(0,0,0)	***1***	**X'•Y'•Z'**
1	0	0	1	f(0,0,1)	*0*	X'•Y'•Z
2	0	1	0	f(0,1,0)	*0*	X'•Y•Z'
3	0	1	1	f(0,1,1)	***1***	**X'•Y•Z**
4	1	0	0	f(1,0,0)	***1***	**X•Y'•Z'**
5	1	0	1	f(1,0,1)	*0*	X•Y'•Z
6	1	1	0	f(1,1,0)	***1***	**X•Y•Z'**
7	1	1	1	f(1,1,1)	***1***	**X•Y•Z**

Tabela verdade: Produto Canônico

- Ou segunda forma canônica: **produto dos maxtermos** correspondentes às **saídas 0**
  - No exemplo: (X+Y+Z') • (X+Y'+Z) • (X'+Y+Z')

    $F = \prod_{X,Y,Z}(1,2,5)$

linha	X	Y	Z	f(X,Y,Z)	*exemplo*	maxtermo
0	0	0	0	f(0,0,0)	*1*	X+Y+Z
1	0	0	1	f(0,0,1)	***0***	**X+Y+Z'**
2	0	1	0	f(0,1,0)	***0***	**X+Y'+Z**
3	0	1	1	f(0,1,1)	*1*	X+Y'+Z'
4	1	0	0	f(1,0,0)	*1*	X'+Y+Z
5	1	0	1	f(1,0,1)	***0***	**X'+Y+Z'**
6	1	1	0	f(1,1,0)	*1*	X'+Y'+Z
7	1	1	1	f(1,1,1)	*1*	X'+Y'+Z'

Exercícios

- Escreva a tabela da verdade para:

  a) F = X•(Y+Z')

  b) F = X'•Y+X'•Y'•Z

  c) F = W'•X+W•(Y'+Z)

  d) F = (W•Z)'•(X'+Y')'

- Escreva soma e produto canônico para:

  a) F = $\sum_{X,Y}(1,2)$

  b) F = $\prod_{A,B}(0,1,2)$

  c) F = X'+Y•Z

Exercícios: Resposta

1) Escreva a tabela-verdade para

   a) F = X • (Y + Z')

➔ Identificar variáveis de entrada

- Variáveis: X, Y e Z

➔ Criar uma coluna à esquerda para cada variável de entrada

- Três colunas à esquerda
- Nota: 2^n linhas para *n* variáveis

X	Y	Z	
0	0	0	*0*
0	0	1	*1*
0	1	0	*2*
0	1	1	*3*
1	0	0	*4*
1	0	1	*5*
1	1	0	*6*
1	1	1	*7*

Exercícios: Resposta

➔ Criar colunas à direita conforme precedência:

f = X • (Y + Z') ➔ *Criar coluna para Z'*

X	Y	Z	Z'
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Exercícios: Resposta

➔ Criar colunas à direita conforme precedência:

f = X • (Y + Z') ➔ *Criar coluna para (Y+Z')*

X	Y	Z	Z'	Y+Z'
0	0	0	1	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	1

Exercícios: Resposta

➔ Criar colunas à direita conforme precedência:

f = X • (Y + Z') ➔ *Criar coluna para X•(Y+Z')*

X	Y	Z	Z'	Y+Z'	X•(Y+Z')
0	0	0	1	1	0
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	1	1

Exercícios: Resposta

b) F = X'•Y+X'Y'•Z

X	Y	Z	F
0	0	0	0
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

c) F = W'•X+W•(Y'+Z)

W	X	Y	Z	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

d) F = (W•Z)'•(X'+Y')'

W	X	Y	Z	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Exercícios: Resposta

- Escreva soma e produto canônico para:

a) $F = \sum_{X,Y}(1,2)$ ➔ $F = X'\bullet Y + X\bullet Y' = (X'+Y') \bullet (X+Y)$

01+10 **00 • 11**

b) $F = \prod_{A,B}(0,1,2)$ ➔ $F = A\bullet B \quad = (A+B)\bullet(A+B')\bullet(A'+B)$

11 **00 • 01 • 10**

c) $F = X'+Y\bullet Z$ ➔ (três variáveis):

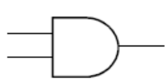
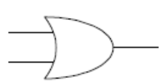
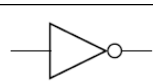
$X'Y'Z'+X'Y'Z+ X'YZ'+X'YZ+XYZ+X'YZ$

000 001 010 011 111

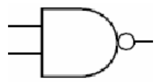
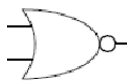
$(X'+Y+Z) \bullet (X'+Y+Z') \bullet (X'+Y'+Z)$

100 101 110

Resumo

Blocos Lógicos Básicos				
Porta lógica	**Símbolo Usual**	**Tabela verdade**	**Função Lógica**	**Expressão**
E AND		A B S 0 0 0 0 1 0 1 0 0 1 1 1	**Função E:** 1 se todas as entradas forem 1; 0 nos outros casos	$S = A\bullet B$
OU OR		A B S 0 0 0 0 1 1 1 0 1 1 1 1	**Função OU:** 0 se todas as entradas forem 0; 1 nos outros casos	$S = A+B$
NÃO NOT		A S 0 1 1 0	**Função NOT:** inverte o valor da entrada	$S = A'$

Resumo

Blocos Lógicos Básicos				
Porta lógica	**Símbolo Usual**	**Tabela verdade**	**Função Lógica**	**Expressão**
NÃO-E NAND		**A B S** **0 0 1** **0 1 1** **1 0 1** **1 1 0**	**Função NÃO-E:** Inverso da função E	S=(A•B)’
NÃO-OU NOR		**A B S** **0 0 1** **0 1 0** **1 0 0** **1 1 0**	**Função NOR:** Inverso da função OU	S=(A+B)’
OU-Exclusivo XOR		**A B S** **0 0 0** **0 1 1** **1 0 1** **1 1 0**	**Função XOR:** 1 quando as duas entradas forem distintas entre si	S=A⊕B