Circuitos Digitais
Portas Lógicas e Álgebra de Boole

Aula 3

Prof. Abel Guilhermino

Álgebra de Boole (ou Boleana)

- Desenvolvida pelo matemático britânico George Boole para estudo da lógica.
- Definida sobre um conjunto de dois elementos:

(falso, verdadeiro) (0, 1) (baixo, alto)

- Seus elementos, a princípio, não tem significado numérico.
- Postulados: se x é uma variável boleana então:
  - Se $x \neq 0 \Rightarrow x = 1$
  - Se $x \neq 1 \Rightarrow x = 0$

Álgebra de Boole: funções

- Uma variável boleana só pode assumir apenas um dos valores possíveis (0 e 1)
- Uma ou mais variáveis e operadores podem ser combinados formando uma função lógica
  - $Z_1(A) = f(A) = ...$ (expressão usando var. A)
  - $Z_2(A,B) = f(A,B) = ...$ (expr. usando var. A e B)
- Resultados de uma função lógica podem ser expressos numa tabela relacionando todas as combinações possíveis dos valores que suas variáveis podem assumir e seus resultados correspondentes: a Tabela-Verdade.

Álgebra de Boole: Tabela Verdade

Variáveis — Função Lógica

Lista das combinações possíveis dos estados das variáveis de entrada

A	B	Z=f(A,B)
0	0	0
0	1	1
1	0	1
1	1	1

Resultados da função lógica para cada combinação dos estados de entrada

- Tabela-Verdade relaciona os resultados (saída) de uma função lógica para todas as combinações possíveis de suas variáveis (entrada).
- Na Tabela-Verdade acima a função lógica Z possui duas variáveis A e B, sendo Z = f(A, B) = A + B

Álgebra de Boole: operações

- São definidas algumas operações elementares na álgebra boleana:
  - Operação "Não" (NOT)
  - Operação "E" (AND)
  - Operação "Ou" (OR)
  - NAND
  - NOR
  - Operação "Ou-Exclusivo" (Exclusive-Or ou XOR)
  - XNOR

Álgebra de Boole

- Porta Lógica NOT
  - É a porta Inversora
  - Operador: Barra, Apóstrofo

$\overline{A}$, A'

  - Símbolo

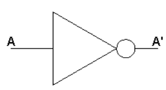


Tabela da Verdade

A	F = A'
0	1
1	0

Álgebra de Boole

- Porta Lógica OR
  - Necessita de duas ou mais entradas
  - Operador: +

F = A + B

- Símbolo

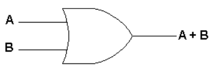


Tabela da Verdade

A	B	F = (A+B)
0	0	0
0	1	1
1	0	1
1	1	1

Álgebra de Boole

- Porta Lógica AND
  - Necessita de duas ou mais entradas
  - Operador: .

F = A . B

- Símbolo

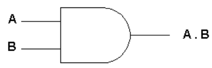


Tabela da Verdade

A	B	F = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1

Álgebra de Boole

- Porta Lógica NAND
  - Equivalente a uma porta AND seguido de uma NOT
  - Operador:

F = (A . B)'

- Símbolo

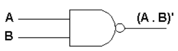


Tabela da Verdade

A	B	F = (A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

Álgebra de Boole

- Porta Lógica NOR
  - Equivalente a uma porta OR seguido de uma NOT
  - Operador:

F = (A + B)'

- Símbolo

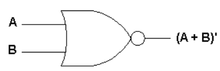


Tabela da Verdade

A	B	F = (A+B)'
0	0	1
0	1	0
1	0	0
1	1	0

Álgebra de Boole

- Porta Lógica XOR
  - É o OU Exclusivo
  - Operador:

F = (A ⊕ B)

- Símbolo

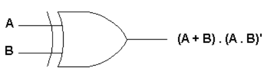


Tabela da Verdade

A	B	F = (A⊕B)
0	0	0
0	1	1
1	0	1
1	1	0

Álgebra de Boole

- Porta Lógica XNOR
  - É o complemento da Função XOR
  - Operador:

F = (A ⊕ B)'

- Símbolo

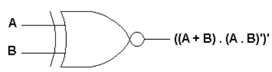
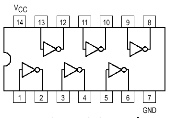


Tabela da Verdade

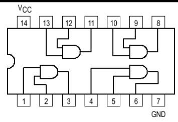
A	B	F = (A⊕B)'
0	0	1
0	1	0
1	0	0
1	1	1

"NOT" CI 7404



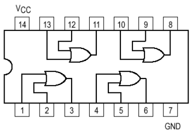
- Desligue a Alimentação
- A montagem de todos os circuitos intregrados no protoboard deverá obedecer:
- Alim.(V_{CC}) : **+5.0 V pino 14 (fio vermelho)**.
- Terra (**GND**) : **0.0 V pino 7 (fio preto)**.
- Conecte um fio de protoboard longo, em série uma resistência de 1 kΩ e um LED. Através de outro fio, conecte o LED à terra.
- Utilize esta *ponteira lógica* para analisar alguns sinais na entrada e na saída do integrado.
- Esta ponteira apresenta lógica positiva (saída alta=> led aceso).
- Ajustar o gerador para uma frequência de 10 KHz e utilizar a ***saída TTL***. Esta é a saída adequada para funcionar operando junto com integrados de lógica TTL, ela já fornece o sinal no intervalo esperado de tensão para alimentá-los.
- Caso seja necessário, utilize diodos para evitar tensão negativa na entrada do integrado.
- Ligue o gerador de onda quadrada em alguma das entradas inversoras.
- Observar no osciloscópio a saída invertida. Meça o tempo de atraso da saída em relação à entrada.

"AND" CI 7408



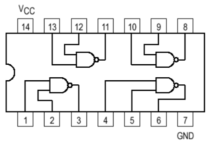
- Desligue a Alimentação e troque o CI
- ***Tabela verdade***
- Utilizando a ponteira lógica, obtenha a tabela-verdade.
- A obtenção de dará através da conexão da alimentação (**1**) e do terra (**0**) às entradas da porta **AND**.
- Utilize lógica inversa para a ponteira (conecte o led à alimentação) e obtenha a nova tabela lógica.
- ***A porta "AND" como controlador de transmissão.***
- Conecte o gerador de onda com pulso quadrado de 100 ms à entrada ***A*** de uma porta AND.
- Com um fio de protoboard, contacte a entrada ***B*** a **1** ou a **0** (GND ou V_{cc}).
- Observe os valores na saída, em função dos sinais de entrada.

"OR" CI 7432



- ***DESLIGUE A ALIMENTAÇÃO* e *substitua o integrado.***
- De forma similar à montagem anterior, observe maneira similar à montagem anterior,observe o funcionamento destas portas.
- Obtenha as tabelas verdade com lógica positiva e negativa em função das entradas.

"NAND" CI 7400



- ***DESLIGUE A ALIMENTAÇÃO* e *substitua o integrado.***
- A partir de um circuito integrado 7400 (quatro portas "NAND") construa um operador XOR e obtenha sua tabela verdade (lógica direta), utilizando a ponteira lógica.
- Esquematize as ligações necessárias, utilizando as leis de Morgan e as identidades booleanas.
- Antes de realizar esta montagem, prove que é possível realizá-la esta montagem. Para isto, utilize as leis de Morgan e as identidades booleanas já apresentadas.

Circuitos Digitais
Álgebra de Boole

Prof. Abel Guilhermino

Álgebra de Boole: precedência

- Precedência das Operações
  - (0) parêntesis
  - (1) "Negação"
  - (2) "E"
  - (3) "Ou", "Ou-exclusivo"
- O uso de parêntesis altera a precedência "normal" dos operadores, como na álgebra comum.

Postulados básicos da álgebra de chaveamento

- Postulado 1:
  - Uma variável booleana x tem dois valores possíveis 0 e 1. Esses valores são exclusivos:
    - **se x =0 então x $\neq$ 1**
    - **se x =1 então x $\neq$ 0**
- Postulado 2:
  - A operação NOT é definida como:
    - **$\bar{0} = 1$** **$\bar{1} = 0$**
- Postulado 3
  - As operações AND e OR são definidas como:

0 . 0 = 0 **0 + 0 = 0**
0 . 1 = 0 **0 + 1 = 1**
1 . 0 = 0 **1 + 0 = 1**
1 . 1 = 1 **1 + 1 = 1**

  - A partir destes postulados podem ser construídos os teoremas que permitem manipular e simplificar expressões lógicas.

Propriedades básicas da álgebra de chaveamento

☑ **Leis e Teoremas da Álgebra Booleana:**

- *Operações com 0 and 1:*
  - **1. X + 0 = X** **1D. X • 1 = X**
  - **2. X + 1 = 1** **2D. X • 0 = 0**
- *Lei da Indepotência:*
  - **3. X + X = X** **3D. X • X = X**
- *Lei da Involução:*
  - **4. $\overline{(\overline{X})}$ = X**
- *Lei de Complementação:*
  - **5. X + $\overline{X}$ = 1** **5D. X • $\overline{X}$ = 0**
- *Lei comutativa:*
  - **6. X + Y = Y + X** **6D. X • Y = Y • X**
- *Lei Distributiva:*
  - **7. A . (B + C) = A . B + A . C**
  - **8. A + B . C = (A + B) . (A + C)**

Propriedades básicas da álgebra de chaveamento

☑ *Leis Associativas:*

7. (X + Y) + Z = X + (Y + Z) = X + Y + Z **7D. (X • Y) • Z = X (Y • Z) = X • Y• Z**

8. X • (Y+ Z) = (X • Y) + (X •Z) **8D. X + (Y • Z) = (X + Y) • (X + Z)**

9. X • Y + X • $\overline{Y}$ = X **9D. (X + Y) • (X + $\overline{Y}$) = X**

10. X + X • Y = X **10D. X • (X + Y) = X**

11. (X + $\overline{Y}$) • Y = X•Y **11D. (X • $\overline{Y}$) + Y = X + Y**

☑ ***Lei de DeMorgan:***

1. $\overline{(X + Y + Z + ...)} = \overline{X} \cdot \overline{Y} \cdot \overline{Z} \cdot ...$ **1D. $\overline{(X \cdot Y \cdot Z \cdot ...)} = \overline{X} + \overline{Y} + \overline{Z} + ...$**
2. {F(X1,X2,...,Xn,0,1,+,•)}' = {F($\overline{X1}$,$\overline{X2}$,...,$\overline{Xn}$,1,0,•,+)}

Propriedades básicas da algebra de chaveamento

- Exemplo:
  - Provar que x+$\bar{x}$ = 1
  - Usando a tabela verdade:

x	x+$\bar{x}$	1
0	1	1
1	1	1

  - Provar x+(x.y) =x (lei da absorção)
  - Usando a tabela verdade

x	y	x+(x.y)	x
0	0	0	0
0	1	0	0
1	0	1	1
1	1	1	1

Propriedades da álgebra de chaveamento

- Provar que x+($\bar{x}$.y) = x+y
  - Partindo de x+($\bar{x}$.y) teremos
  - Usando a lei distributiva x+($\bar{x}$.y) = (x+$\bar{x}$).(x+y)
  - Usando a lei de complementação (x+$\bar{x}$).(x+y) = 1.(x+y)
  - Propriedade especial de **1**, 1.(x+y) = x+y
- Exemplo:
  - Simplificar a seguinte função lógica:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$f(x_1, x_2, x_3) = \bar{x}_1.\bar{x}_2.\bar{x}_3 + \bar{x}_1.\bar{x}_2.x_3 + x_1.\bar{x}_2.x_3 =$$
$$= \bar{x}_1.\bar{x}_2.\bar{x}_3 + \bar{x}_1.\bar{x}_2.x_3 + \bar{x}_1.\bar{x}_2.x_3 + x_1.\bar{x}_2.x_3$$
$$= \bar{x}_1.\bar{x}_2(\bar{x}_3 + x_3) + \bar{x}_2.x_3(\bar{x}_1 + x_1)$$
$$= \bar{x}_1.\bar{x}_2 + \bar{x}_2.x_3$$
$$= \boxed{\bar{x}_2.(\bar{x}_1 + x_3)}$$

Indepotência

Distributiva e Complementação

Distributiva

Álgebra de Boole: dualidade

- Existe um princípio especial na álgebra boleana denominado "princípio da dualidade":
  - Para uma equação boleana qualquer, se trocarmos as operações E (**.**) e operações OU (+) entre si assim como valores 0s e 1s entre si, obteremos uma equação igualmente válida.
  - A + 0 = A A . 1 = A
  - A + 1 = 1 A . 0 = 0
  - A + A = A A . A = A
  - A + $\bar{A}$ = 1 A . $\bar{A}$ = 0

Equivalência e Suficiência de Operações

- Equivalência das operações
  - Qualquer função lógica pode ser expressa em termos das operações AND, OR e NOT.

    $A \oplus E = \bar{A}E + A\bar{E}$
- Suficiência das operações
  - Apenas as operações AND e NOT ou OR e NOT são suficientes para expressar qualquer operação:

    $\bar{A}E + A\bar{E}$

  (aplicando De Morgan) $\Rightarrow \overline{\overline{\bar{A}E} \,.\, \overline{A\bar{E}}}$

Simplificação

- Os teoremas, propriedade e identidades da álgebra boleana podem ser aplicados para simplificarmos funções lógicas e, com isso, reduzirmos o número necessário de operações.
- Simplificando:
  - $w = x.y + \overline{y.x}.z$ temos $w = xy + z$
  - $w = x\,(\bar{x} + y)$ temos $w = xy$
  - $w = \bar{x}\,(x + y) + \bar{z} + z.y$ temos $w = y + \bar{z}$
  - $w = (w + y + \bar{x})(w + y + x)(\bar{y} + z)(w + z)$

Exercícios

(1) Prove se as seguintes simplificações são possíveis:

- $w = x.y + \overline{y.x}.z$ temos $w = xy + z$
- $w = x\,(\bar{x} + y)$ temos $w = xy$
- $w = \bar{x}\,(x + y) + \bar{z} + z.y$ temos $w = y + \bar{z}$

(2) Simplifique:

- $w = (w + y + \bar{x})(w + y + x)(\bar{y} + z)(w + z)$
- $w = \overline{v + v\bar{x} + yz}$