FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

T566 –SISTEMAS DIGITAIS AVANÇADOS

Aula 4 - Simplificação de Expressões Lógicas MinTermos & Maxtermos

Prof. Danilo Reis

Histórico Álgebra de Boole

- Desenvolvida pelo matemático e filósofo britânico George Boole para estudo da lógica (1850- Leis do Pensamento).
- 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 diferente de 0 ⇒ $x = 1$

Se x diferente de 1 ⇒ $x = 0$

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 Z1(A) = f(A) = ... (expressão usando var. A) Z2(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 resultadoscorrespondentes: a Tabela-Verdade

Álgebra de Boole: Tabela Verdade

Variáveis | Função Lógica

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

Lista das combinações possíveis dos estados das variáveis de entrada

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: precedência

- Precedência das Operações
  1. parêntesis
  2. “Negação”
  3. “E”
  4. “Ou”, “Ou-exclusivo”
- O uso de parêntesis altera a precedência “normal” dos operadores, como na álgebra comum.

Postulados básicos

Postulado 1:

Uma variável booleana x tem dois valores possíveis 0 e 1. Esses valores são exclusivos:

se x =0 então x não pode ser 1

se x =1 então x não pode ser 0

Postulado 2:

A operação NOT é definida como:

0 = 1 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

☑ Leis e Teoremas da Álgebra Booleana:

- *Operações com 0 and 1:*

1. $X + 0 = X$
2. $X + 1 = 1$

1D. $X \cdot 1 = X$
2D. $X \cdot 0 = 0$

- *Lei da Indepotência:*

3. $X + X = X$

3D. $X \cdot X = X$

- *Lei da Involução:*

4. $(\overline{\overline{X}}) = X$

- *Lei de Complementação:*

5. $X + \overline{X} = 1$

5D. $X \cdot \overline{X} = 0$

- *Lei comutativa:*

6. $X + Y = Y + X$

6D. $X \cdot Y = Y \cdot X$

- *Lei Distributiva:*

7. $A \,.\, (B + C) = A \,.\, B + A \,.\, C$
8. $A + B \,.\, C = (A + B) \,.\, (A + C)$

Propriedades básicas

☑ *Leis Associativas:*

7. $(X + Y) + Z = X + (Y + Z) = X + Y + Z$

7D. $(X \cdot Y) \cdot Z = X\ (Y \cdot Z) = X \cdot Y \cdot Z$

8. $X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$

8D. $X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$

9. $X \cdot Y + X \cdot \overline{Y} = X$

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

10. $X + X \cdot Y = X$

10D. $X \cdot (X + Y) = X$

11. $(X + \overline{Y}) \cdot Y = X \cdot Y$

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

Propriedades básicas

$$x+(\overline{x}.y) = x+y$$

- Partindo de $x+(\overline{x}.y)$ teremos
- Usando a lei distributiva $x+(\overline{x}.y) = (x+\overline{x}).(x+y)$
- Usando a lei de complementação $(x+\overline{x}).(x+y) = 1.(x+y)$
- Propriedade especial de **1**, $1.(x+y) = x+y$

Lei DeMorgan

"*The negation of a conjunction is the disjunction of the negations.*" and
"*The negation of a disjunction is the conjunction of the negations.*"

1. $\overline{X \cdot Y} = \overline{X} + \overline{Y}$
2. $\overline{X + Y} = \overline{X} \cdot \overline{Y}$

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.

$A + 0 = A$	$A . 1 = A$
$A + 1 = 1$	$A . 0 = 0$
$A + A = A$	$A . A = A$
$A + A = 1$	$A . A = 0$

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:

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

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

Funções Lógicas: Formas Padrão

Funções lógicas podem ser padronizadas a duas “formas padrão”:

- *Forma padrão de soma de produtos expressão é uma soma (OR) de produtos (AND) de variáveis e variáveis complementadas;*
- *Forma padrão de produto de somas expressão é um produto (AND) de somas (OR) de variáveis e variáveis complementadas.*

Forma Padrão: soma de produtos

As funções abaixo estão em sua forma canônica SDP:

F = A.B.C + A'.B.C + A.B'.C + A.B.C'

G = A'.B'.C' + A'.B.C' + A.B'.C'

A função abaixo não está em sua forma canônica

F = A.B + A'.C + B.C'

Forma Padrão: produto de somas

A função abaixo estão em sua forma canônica PDS:

$F(x,y) = (x'+Y).(x+y')$

A função abaixo não está em sua forma canônica PDS

$F(x,y) = x . (x+y')$

Estratégia similar a SDP para formatar uma função qualquer e obter a sua forma canônica.

Forma Padrão: produto de somas

A função abaixo estão em sua forma canônica PDS:

F(x,y) = (x'+Y).(x+y')

A função abaixo não está em sua forma canônica PDS

F(x,y) = x . (x+y')

Estratégia similar a SDP para formatar uma função qualquer e obter a sua forma canônica.

Mintermos e Maxtermos

Os conceitos de Mintermos e Maxtermos são utilizados para reescrever-se uma função lógica em uma forma padronizada no sentido de obter-se uma simplificação da mesma. Esta padronização serve como base, por exemplo, na utilização de Arranjos e PLAs.

Mintermos e Maxtermos

- Na soma padrão de produtos, cada termo correspondente a um produto é denominado mintermo.
- Analogamente, no produto padrão de somas, cada termo correspondente a uma soma é denominado de maxtermo.
- Embora as formas padrões não sejam as formas mais simplificadas (e por vezes mais complexas que as formas originais) se prestam a sistematização da simplificação.

Mintermos e Maxtermos

Cada mintermo ou maxtermo se associa a uma possibilidade de entrada de uma função lógica. Por exemplo

Y=f(A,B)=(A.B)’

Mintermo	Maxtermo	A	B	Y
A'.B'	A+B	0	0	1
A'.B	A+B'	0	1	1
A.B'	A'+B	1	0	1
A.B	A'+B'	1	1	0

Mintermos e Maxtermos

Numerando as entradas da tabela verdade é possível se identificar os mintermos e maxtermos genericamente:

mintermos: 0 equivale variável complementada
1 equivale variável

maxtermos: 0 equivale variável
1 equivale variável complementada

Assim a entrada 0, que equivale a A=0 e B=0:

mintermo: A'.B'

maxtermo: A+B

Mintermos e Maxtermos na Tabela Verdade

Linha nº	A	B	C	f(A,B,C)	Mintermos	Maxtermos
0	0	0	0	1	$m_0 = A'B'C'$	$M_0 = A + B + C$
1	0	0	1	0	$m_1 = A'B'C$	$M1 = A + B + C'$
2	0	1	0	1	$m_2 = A'BC'$	$M_2 = A + B' + C$
3	0	1	1	1	$m_3 = A'BC$	$M_3 = A + B' + C'$
4	1	0	0	0	$m_4 = AB'C'$	$M_4 = A' + B + C$
5	1	0	1	0	$m_5 = AB'C$	$M_5 = A' + B + C'$
6	1	1	0	1	$m_6 = ABC'$	$M_6 = A' + B' + C$
7	1	1	1	1	$m_7 = ABC$	$M_7 = A' + B' + C'$

Formas Canônicas

Mintermos forma canônica geral para uma função de n váriaveis

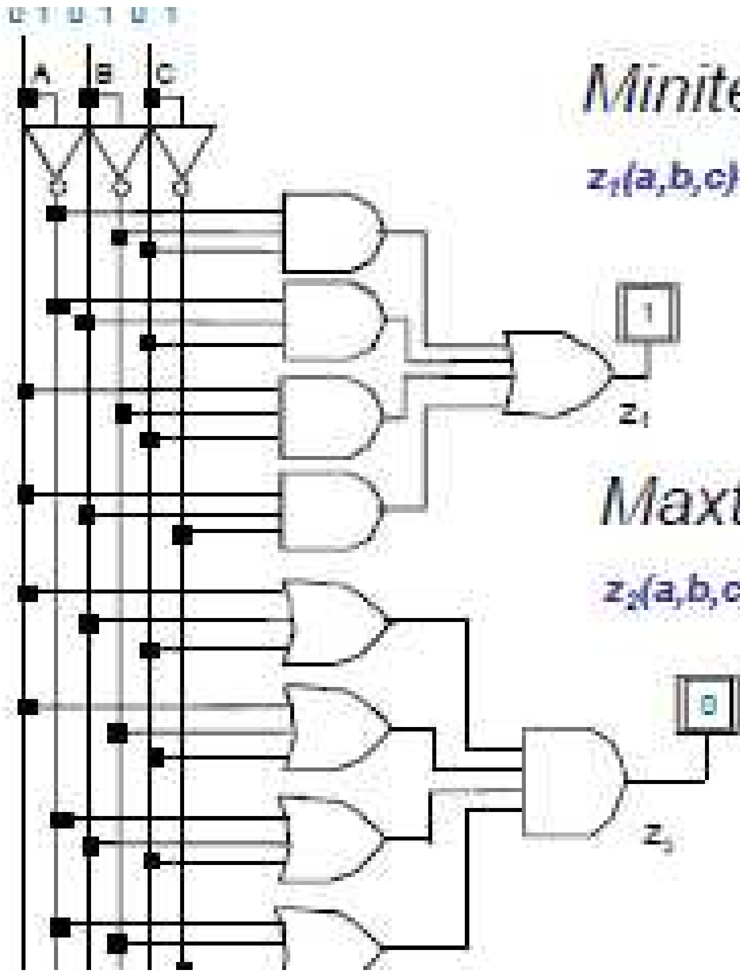
$$F(A, B..., X) = \sum_{i=0}^{2^n} m_i$$

Maxtermos forma canônica geral para uma função de n váriaveis

$$F(A, B, ..., X) = \prod_{i=0}^{2^n} M_i$$

Implementação em PALs ou PLAs

a	b	c	z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



Minitermos

$z_1(a,b,c)= m_1+m_3+m_5+ m_6$

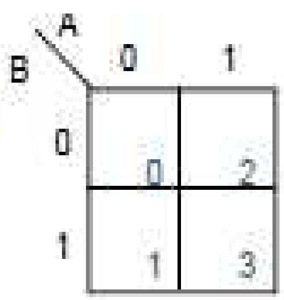
Maxtermos

$z_2(a,b,c)=M_0.M_2.M_4.M_7$

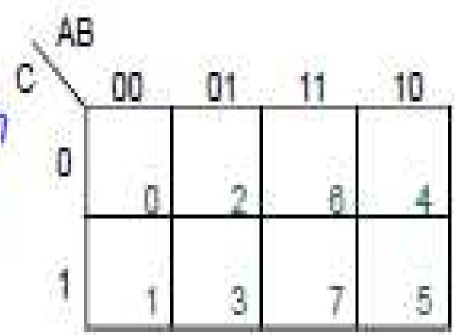
Mapa de Karnaugh

Mapa de Karnaugh é um método alternativo para se representar uma tabela verdade. Ele propõe uma solução gráfica da informação contida na tabela verdade.

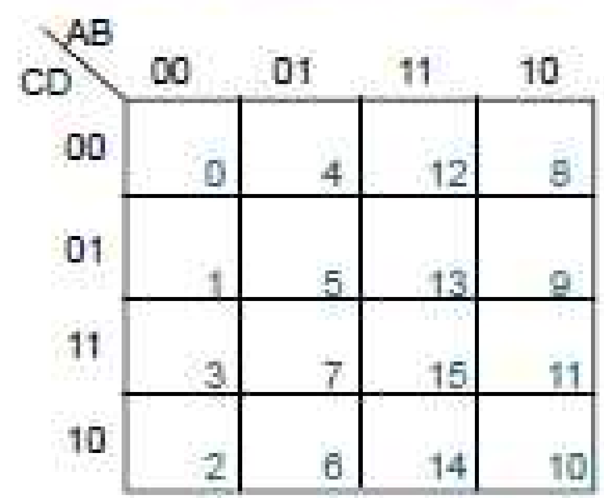
☑ *Mapa de Karnaugh para duas variáveis*



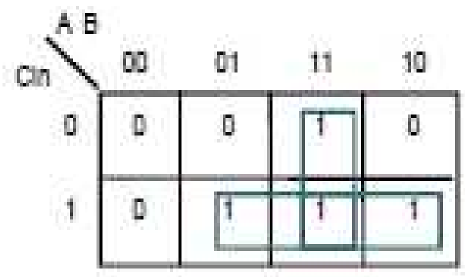
☑ *Mapa de Karnaugh para três variáveis*



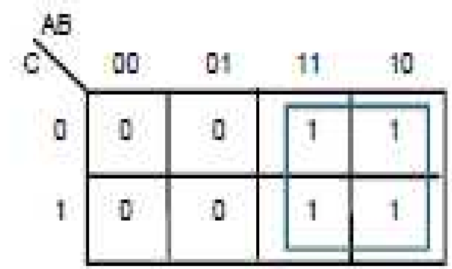
☑ *Mapa de Karnaugh para quatro variáveis*



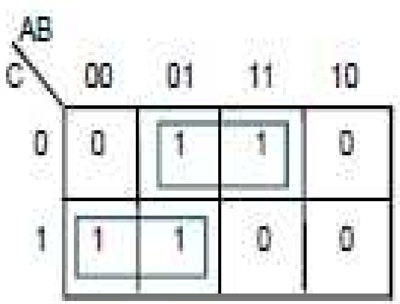
Simplificação com Karnaugh



f(Cin,A,B) = A B + B Cin + A Cin

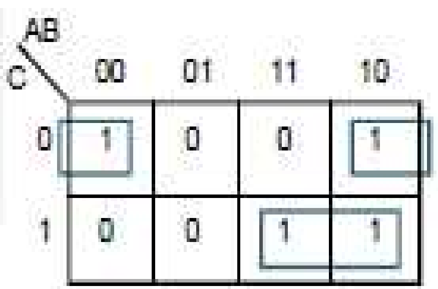


F(A,B,C) = A



F(A,B,C) = Σm(1,2,3,6)

F' = B C' + A' C



F(A,B,C) = Σm(0,4,5,7)

F = B' C' + A C

Referências

- http://pt.wikipedia.org/wiki/George_Boole
- http://www.cin.ufpe.br/~agsf/Sistemas_Digitais.htm