FEDERAL UNIVERSITY
OF SANTA CATARINA

EEL5105 – Circuitos e Técnicas Digitais
Aula 3

Prof. Héctor Pettenghi

hector@eel.ufsc.br
http://hectorpettenghi.paginas.ufsc.br

3. Projeto de Circuitos Lógicos Combinacionais

Nesta aula: uso da álgebra booleana para o **projeto** de circuitos lógicos.

Ou seja: como é possível obter um circuito lógico a partir da especificação de um problema (real) e/ou de uma tabela verdade.

3.1. Formas Padrão/Canônicas

3.2. Projeto Usando as Formas Canônicas

3.3. Mapa de Karnaugh

2

Nesta aula veremos como a álgebra booleana é aplicada em projetos de circuitos lógicos, ou seja, como é possível obter um circuito a partir da especificação de um problema real e/ou uma tabela verdade.

Na seção 3.1 são abordadas as formas padrão, ou canônicas das funções booleanas.

Logo em seguida, na seção 3.2, é contemplado como um projeto prático é feito utilizando as formas canônicas.

Por último, na seção 3.3 veremos uma técnica para obter funções booleanas simplificadas, os mapas de Karnaugh.

3.1. Formas Padrão/Canônicas

3.2. Projeto Usando as Formas Canônicas

3.3. Mapa de Karnaugh

3

Começamos então com as formas padrão (soma de produtos e produto de somas) e canônicas, dentro de cada forma padrão.

As formas canônicas de uma equação booleana podem ser de dois tipos: soma de produtos ou produto de somas. Elas nada mais são do que formas em que todas as variáveis aparecem em todas as partes posteriormente comparadas em uma porta "or" ou uma porta "and". Vejamos...

3.1. Formas Padrão/Canônicas

- **3.1.1. Soma de Produtos**
  - Primeira forma canônica
  - Soma de **minitermos**
  - Todas as variáveis devem estar presentes em cada termo
  - Exemplos: $S(A,B,C) = ABC + A\bar{B}C + \bar{A}\bar{B}C$

$$S(A,B,C) = ABC + \bar{B}C$$

$$S(A,B,C,D) = ABCD + A\bar{B}C\bar{D} + \bar{A}\bar{B}CD + A\bar{B}\bar{C}D$$

4

A primeira forma canônica, ou soma de minitermos, é feita em forma de soma de produtos, ou seja, apresenta as variáveis comparadas a partir de uma porta "and" e as saídas dessas portas "and" unem-se como entradas de uma porta "or".

Observe que para ser uma forma canônica em todos os "produtos" deve haver sempre a presença de todas as variáveis.

3.1. Formas Padrão/Canônicas

- **3.1.1. Soma de Produtos**
  - Primeira forma canônica
  - Soma de **minitermos**
  - Todas as variáveis devem estar presentes em cada termo
  - Exemplos: $S(A,B,C) = \underbrace{ABC} + \underbrace{A\bar{B}C} + \underbrace{\bar{A}\bar{B}C}$

$$\cancel{S(A,B,C) = ABC + \bar{B}C}$$

$$S(A,B,C,D) = \underbrace{ABCD} + \underbrace{A\bar{B}C\bar{D}} + \underbrace{\bar{A}\bar{B}CD} + \underbrace{A\bar{B}\bar{C}D}_{\text{minitermo}}$$

5

Assim, o segundo exemplo não caracteriza uma forma canônica.

Cada um dos "produtos" da forma canônica é chamado de minitermo, o que origina o nome "soma de minitermos".

3.1. Formas Padrão/Canônicas

- **3.1.1. Soma de Produtos**
  - Representando uma função na forma de minitermos:

$$S = A + BC$$

6

Para obter a soma de minitermos de uma função booleana já simplificada, basta proceder da seguinte maneira...

3.1. Formas Padrão/Canônicas

- **3.1.1. Soma de Produtos**
  - Representando uma função na forma de minitermos:

$$
\begin{aligned}
S &= A + BC \\
&= A(B + \bar{B}) + (A + \bar{A})BC \\
&= AB + A\bar{B} + ABC + \bar{A}BC \\
&= AB(C + \bar{C}) + A\bar{B}(C + \bar{C}) + ABC + \bar{A}BC \\
&= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + ABC + \bar{A}BC \\
&= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC
\end{aligned}
$$

7

Utilizamos o teorema booleano 4-b) [X+X'=1], seguido do 7-a) [X*(Y+Z) = X*Y+X*Z], para inserir as variáveis faltantes em cada um dos "produtos"

Observe que, por exemplo, A*1=A.

3.1. Formas Padrão/Canônicas

- **3.1.2. Produto de Somas**
  - Segunda forma canônica
  - Produto de **maxitermos**
  - Todas as variáveis em cada termo
  - Exemplo: $S = (A+B+C) \cdot (A+B+\overline{C}) \cdot (\overline{A}+\overline{B}+C)$

8

A segunda forma canônica, ou produto de maxitermos, funciona exatamente do jeito contrário ao da soma de minitermos. Nela a função é representada em forma de produto de somas, ou seja, as variáveis são comparadas a partir de uma porta "or" e, em seguida, as saídas destas portas tornam-se as entradas de uma porta "and".

Veja o exemplo e note, mais uma vez, que para ser uma forma canônica em todas as "somas" deve haver a presença de todas as variáveis.

3.1. Formas Padrão/Canônicas

- **3.1.2. Produto de Somas**
  - Segunda forma canônica
  - Produto de **maxitermos**
  - Todas as variáveis em cada termo
  - Exemplo:

$$S = (A+B+C)\cdot(A+B+\overline{C})\cdot\underbrace{(\overline{A}+\overline{B}+C)}_{\text{maxitermo}}$$

9

Cada uma das "somas" da forma canônica chama-se maxitermo, o que origina o nome "produto de maxitermos".

3.1. Formas Padrão/Canônicas

- **3.1.2. Produto de Somas**
  - Representando uma função na forma de maxitermos:

$$S = A + BC$$

Para representar uma função booleana já simplificada em produto de maxitermos basta proceder da seguinte forma...

3.1. Formas Padrão/Canônicas

- **3.1.2. Produto de Somas**
  - Representando uma função na forma de maxitermos:

Teorema 7(b)

$$
\begin{aligned}
S = A + BC &= (A+B)(A+C) \\
&= (A+B+C\overline{C})(A+C+B\overline{B}) \\
&= (A+B+C)(A+B+\overline{C})(A+C+B)(A+C+\overline{B}) \\
&= (A+B+C)(A+B+\overline{C})(A+C+B)(A+C+\overline{B}) \\
&= (A+B+C)(A+B+\overline{C})(A+\overline{B}+C)
\end{aligned}
$$

11

Neste caso, primeiro precisamos utilizar o teorema 7-b) [X+(Y*Z) = (X+Y)*(X+Z)]. Em seguida adicionamos as variáveis faltantes em cada uma das "somas" utilizando o teorema 4-a) [X*X' = 0] seguido pelo 7-b) [X+(Y*Z) = (X+Y)*(X+Z)] (em que X = (A+B)).

Observe que (A+B+0) = (A+B).

3.1. Formas Padrão/Canônicas

3.2. Projeto Usando as Formas Canônicas

3.3. Mapa de Karnaugh

Agora veremos como utilizamos a primeira e a segunda formas canônicas para resolver produzir determinada tabela verdade.

3.2. Projeto Usando as Formas Canônicas

- Minitermos:

A	B	C								ABC	
0	0	0									
0	0	1									
0	1	0									
0	1	1									
1	0	0									
1	0	1									
1	1	0									
1	1	1									

13

Utilizando a primeira forma canônica, temos que cada minitermo representa uma determinada combinação de números binários das entreadas do circuito.

3.2. Projeto Usando as Formas Canônicas

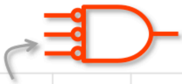
- Minitermos:

A	B	C								ABC	
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	

14

Por exemplo: caso a saída deva ser '1' quando ABC = "111", utilizamos o minitermo ABC. Isso porque 1*1*1 = 1.

3.2. Projeto Usando as Formas Canônicas

- Minitermos: 

A	B	C	A'B'C'							ABC	
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	

15

A mesma lógica segue para as demais situações.

3.2. Projeto Usando as Formas Canônicas

- Minitermos:

A	B	C	A’B’C’	A’B’C						ABC	
0	0	0	1							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	0							0	
1	1	1	0							1	

16

Quando a saída deve ser '1' em ABC="000", utilizamos A'B'C'. Isso porque 0'*0'*0' = 1*1*1 = 1.

3.2. Projeto Usando as Formas Canônicas

- Minitermos:

A	B	C	A’B’C’	A’B’C	A’BC’					ABC	
0	0	0	1	0						0	
0	0	1	0	1						0	
0	1	0	0	0						0	
0	1	1	0	0						0	
1	0	0	0	0						0	
1	0	1	0	0						0	
1	1	0	0	0						0	
1	1	1	0	0						1	

17

Quando a saída deve ser '1' em ABC="001", utilizamos A'B'C.

3.2. Projeto Usando as Formas Canônicas

- Minitermos:

A	B	C	A’B’C’	A’B’C	A’BC’				ABC	
0	0	0	1	0	0				0	
0	0	1	0	1	0				0	
0	1	0	0	0	1				0	
0	1	1	0	0	0				0	
1	0	0	0	0	0				0	
1	0	1	0	0	0				0	
1	1	0	0	0	0				0	
1	1	1	0	0	0				1	

Quando a saída deve ser '1' em ABC="010", utilizamos A'BC'.

3.2. Projeto Usando as Formas Canônicas

- Minitermos:

A	B	C	A'B'C'	A'B'C	A'BC'	A'BC	AB'C'	AB'C	ABC'	ABC	
0	0	0	1	0	0	0	0	0	0	0	m_0
0	0	1	0	1	0	0	0	0	0	0	m_1
0	1	0	0	0	1	0	0	0	0	0	m_2
0	1	1	0	0	0	1	0	0	0	0	m_3
1	0	0	0	0	0	0	1	0	0	0	m_4
1	0	1	0	0	0	0	0	1	0	0	m_5
1	1	0	0	0	0	0	0	0	1	0	m_6
1	1	1	0	0	0	0	0	0	0	1	m_7

Cada minitermo resulta em 1 para uma única combinação de valores das variáveis de entrada!

19

E, pela mesma lógica, obtemos todos os possíveis minitermos.

Observe que cada minitermo resulta em '1' para uma única combinação de valores das variáveis de entrada.

3.2. Projeto Usando as Formas Canônicas

- **Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de minitermos?**

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

20

Mas, considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de minitermos?

3.2. Projeto Usando as Formas Canônicas

- **Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de minitermos?**

A	B	C	S	A'B'C'	A'B'C	A'BC'	A'BC	AB'C'	AB'C	ABC'	ABC	
0	0	0	1	1	0	0	0	0	0	0	0	m_0
0	0	1	0	0	1	0	0	0	0	0	0	m_1
0	1	0	1	0	0	1	0	0	0	0	0	m_2
0	1	1	1	0	0	0	1	0	0	0	0	m_3
1	0	0	0	0	0	0	0	1	0	0	0	m_4
1	0	1	0	0	0	0	0	0	1	0	0	m_5
1	1	0	0	0	0	0	0	0	0	1	0	m_6
1	1	1	1	0	0	0	0	0	0	0	1	m_7

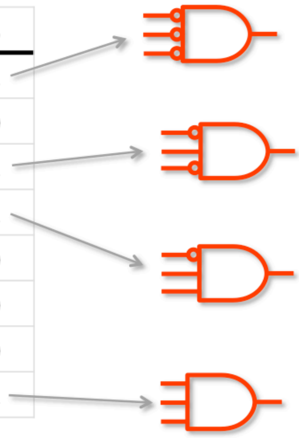
21

Relembrando da tabela de minitermos, vemos que devemos utilizar os minitermos m0, m2, m3 e m7.

3.2. Projeto Usando as Formas Canônicas

- Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de minitermos?

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



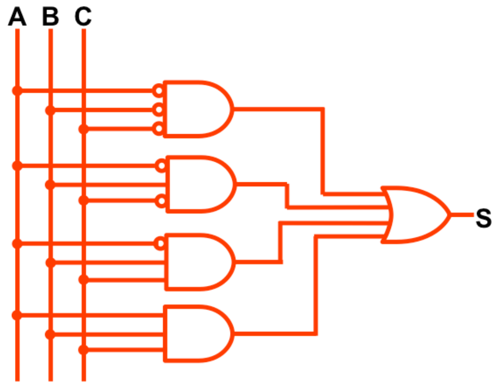
22

Assim, as portas lógicas do circuito serão estas.

3.2. Projeto Usando as Formas Canônicas

- **Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de minitermos?**

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



E estas portas lógica conectam-se da seguinte maneira.

Observe a junção das portas "and" em uma "or".

3.2. Projeto Usando as Formas Canônicas

- **Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de minitermos?**

A	B	C	S	
0	0	0	1	m_0
0	0	1	0	m_1
0	1	0	1	m_2
0	1	1	1	m_3
1	0	0	0	m_4
1	0	1	0	m_5
1	1	0	0	m_6
1	1	1	1	m_7

Fazendo uma **soma de minitermos**:

$$S = m_0 + m_2 + m_3 + m_7$$

$$= A'B'C' + A'BC' + A'BC + ABC$$

simplificando

$$S = A'C' + BC$$

24

Isso, representado em forma de função, fica da seguinte forma.

3.2. Projeto Usando as Formas Canônicas

- Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de maxitermos?

A	B	C	S	A+B+C	A+B+C'	A+B'+C	A+B'+C'	A'+B+C	A'+B+C'	A'+B'+C	A'+B'+C'	
0	0	0	1	0	1	1	1	1	1	1	1	M_0
0	0	1	0	1	0	1	1	1	1	1	1	M_1
0	1	0	1	1	1	0	1	1	1	1	1	M_2
0	1	1	1	1	1	1	0	1	1	1	1	M_3
1	0	0	0	1	1	1	1	0	1	1	1	M_4
1	0	1	0	1	1	1	1	1	0	1	1	M_5
1	1	0	0	1	1	1	1	1	1	0	1	M_6
1	1	1	1	1	1	1	1	1	1	1	0	M_7

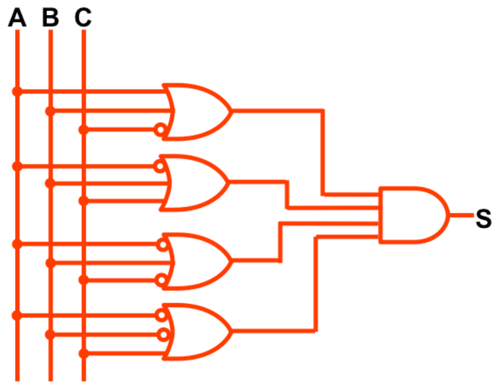
Para fazer o mesmo a partir da tabela de maxitermos procedemos de modo semelhante. Veja que cada maxitermo resulta em '0', e não em '1' (como na tabela de minitermos), para um aúnica combinação de valores das variáveis de entrada.

Assim, utilizaremos para esta tabela verdade os maxitermos M1, M4, M5 e M6.

3.2. Projeto Usando as Formas Canônicas

- **Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de maxitermos?**

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

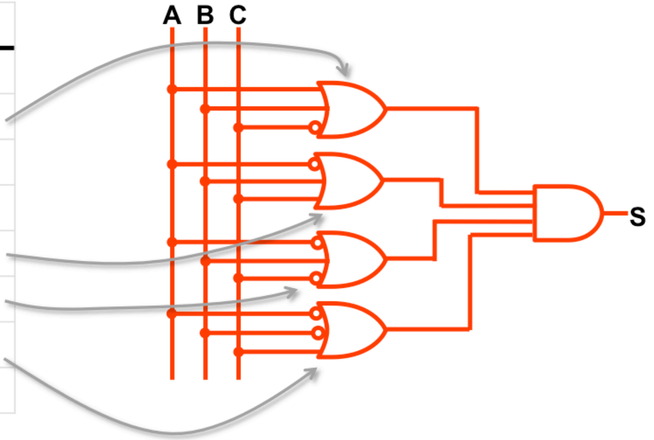


O circuito lógico correspondente é este.

3.2. Projeto Usando as Formas Canônicas

- **Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de maxitermos?**

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



27

Perceba as porta "or", obtidas a partir dos maxitermos, com as saídas conectadas em uma porta "and".

3.2. Projeto Usando as Formas Canônicas

- **Considerando um problema qualquer com três variáveis de entrada, como podemos obter a função lógica (e, portanto, o circuito lógico correspondente) a partir do conhecimento da tabela de maxitermos?**

A	B	C	S	
0	0	0	1	M_0
0	0	1	0 ←	M_1
0	1	0	1	M_2
0	1	1	1	M_3
1	0	0	0 ←	M_4
1	0	1	0 ←	M_5
1	1	0	0 ←	M_6
1	1	1	1	M_7

Fazendo um **produto de maxitermos**:

$$S = M_1 \,.\, M_4 \,.\, M_5 \,.\, M_6$$

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

simplificando

$$S = A'\,C' + BC$$

28

A função lógica fica da assim.

Observe que a função lógica simplificada obtida a partir da soma de minitermos é a mesma obtida a partir do produto de maxitermos, visto que elas vieram da mesma tabela verdade.

3.2. Projeto Usando as Formas Canônicas

- **Exemplo**: Obter as funções de saída para a seguinte tabela verdade usando soma de minitermos.

A	B	C	S_1	S_2	S_3
0	0	0	0	1	0
0	0	1	1	1	0
0	1	0	0	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	1	1

29

Como exercício, tente fazer este exemplo. Lembre-se que cada saída (S1, S2 E S3) corresponde a uma função lógica, assim teremos três funções lógicas.

PROBLEMAS

Problema 3.1 Obter função para a seguinte tabela verdade

A	B	C	S_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

a) Por inspeção da tabela de verdade, expresse s_1 na forma soma de minitermos.

b) Simplifique a expressão obtida em forma de soma de minitermos usando o teorema da adjacência, de modo a obter o número mínimo de termos de soma de produto.

Solução:

Tabela de verdade

ABC	S_1	
000	1	→ m_0
001	1	→ m_1
010	0	
011	0	
100	1	→ m_4
101	1	→ m_5
110	0	
111	1	→ m_7

a) $S_1(A,B,C) = \sum m(0,1,4,5,7) = m_0 + m_1 + m_4 + m_5 + m_7 =$

$$= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + ABC.$$

b) $S_1(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + ABC =$

$$= \bar{A}\,\bar{B}\,\underbrace{(\bar{C}+C)}_{=1} + A\bar{B}\,\underbrace{(\bar{C}+C)}_{=1} + AC\,\underbrace{(\bar{B}+B)}_{=1} =$$

$$= \bar{A}\bar{B} + A\bar{B} + AC = B\,\underbrace{(\bar{A}+A)}_{=1} + AC = B + AC$$

30

No apartado a) identificamos os minitermos na tabela de verdade e colocamos a função S1 expressada como o somatorio dos minitermos.

No apartado b) aplicamos as reduções algébricas aprendidas na aula anterior para reduzir a expressão de minitermos.

3.1. Formas Padrão/Canônicas

3.2. Projeto Usando as Formas Canônicas

3.3. Mapa de Karnaugh

31

Agora veremos sobre o Mapa de Karnaugh.

3.3. Mapa de Karnaugh

- Apresenta as mesmas informações da **tabela verdade**
- Cada linha da **tabela verdade** corresponde a um quadrado do **Mapa de Karnaugh**
- Numeração ordenada conforme o código Gray

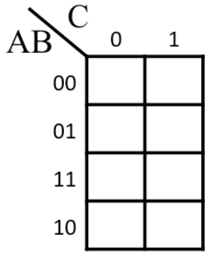
32

Um Mapa de Karnaugh apresenta as mesmas informações da tabela verdade, sendo que cada linhas da tabela verdade corresponde a um quadrado do Mapa de Karnaugh.

3.3. Mapa de Karnaugh

- Apresenta as mesmas informações da **tabela verdade**
- Cada linha da **tabela verdade** corresponde a um quadrado do **Mapa de Karnaugh**
- Numeração ordenada conforme o código Gray
- Exemplo:

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



33

Monta-se um Mapa de Karnaugh da seguinte forma.

Veja que a numeração lateral é feita em código Gray, e não em binário, como na tabela verdade.

3.3. Mapa de Karnaugh

- Apresenta as mesmas informações da **tabela verdade**
- Cada linha da **tabela verdade** corresponde a um quadrado do **Mapa de Karnaugh**
- Numeração ordenada conforme o código Gray
- Exemplo:

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

AB \ C	0	1
00	1	0
01	1	1
11	0	1
10	0	0

34

Em seguida passamos as saídas diretamente da tabela verdade para o mapa, como no slide.

3.3. Mapa de Karnaugh

- Exemplo de projeto:

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

S =

AB \ C	0	1
00	1	0
01	1	1
11	0	1
10	0	0

Agora vamos tirar a função lógica simplificada a partir do Mapa de Karnaugh.

3.3. Mapa de Karnaugh

- Exemplo de projeto:

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$
\begin{aligned}
S &= \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC \\
&= \overline{A}\,\overline{C}(B + \overline{B}) + (\overline{A} + A)BC \\
&= \overline{A}\,\overline{C} + BC
\end{aligned}
$$

AB \ C	0	1
00	1	0
01	1	1
11	0	1
10	0	0

36

Considerando que, a partir da técnica da soma de minitermos, a expressão e sua simplificação são estas...

3.3. Mapa de Karnaugh

- Exemplo de projeto:

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$
\begin{aligned}
S &= \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC \\
&= \overline{A}\,\overline{C}\boxed{(B + \overline{B})} + \boxed{(\overline{A} + A)}BC \\
&= \overline{A}\,\overline{C} + BC
\end{aligned}
$$

AB \ C	0	1
00	1	0
01	1	1
11	0	1
10	0	0

37

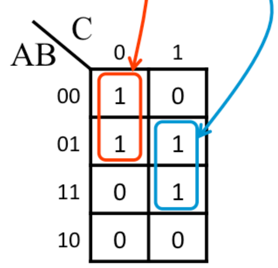
Vemos que o seguinte se expressa entre as etapas de simplificação: no primeiro produto, B pode ser tanto '1' como '0' para produzir a saída desejada, enquanto, no segundo, A pode ser tanto '1' como '0'.

3.3. Mapa de Karnaugh

- Exemplo de projeto:

$$S = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + ABC$$
$$= \bar{A}\bar{C}\boxed{(B + \bar{B})} + \boxed{(\bar{A} + A)}BC$$
$$= \bar{A}\bar{C} + BC$$

As simplificações que podem ser feitas usando álgebra booleana aparecem como valores iguais em quadros vizinhos em uma Mapa de Karnaugh!



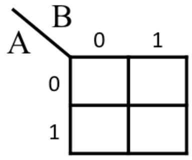
38

Assim, quando olhamos para o mapa, conseguimos perceber que as simplificações que podem ser feitas usando álgebra booleana aparecem como valores iguais em quadro vizinhos em um Mapa de Karnaugh.

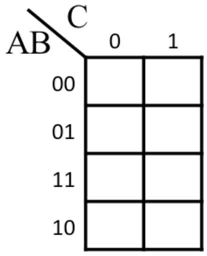
Assim, se os agrupamentos forem feitos da forma mais eficiente, o Mapa de Karnaugh nos fornece a função lógica mais simples. Isso, com mapas de Karnaugh pequenos (até 4 ou 5 variáveis), pode tornar o processo de simplificação de uma função mais eficiente.

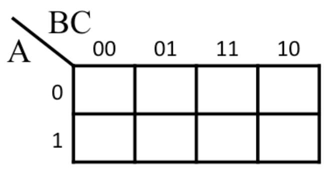
3.3. Mapa de Karnaugh

- 2 variáveis:



- 3 variáveis



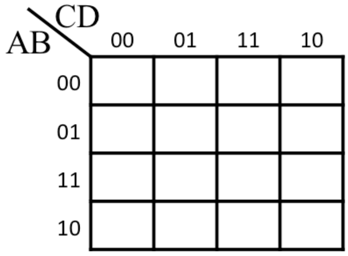


39

No exemplo anterior, tínhamos um Mapa de Karnaugh para três variáveis. Aqui vemos como ele ficaria para duas variáveis e uma outra forma para o de três variáveis.

3.3. Mapa de Karnaugh

- 4 variáveis:



- 5, 6, 7, etc...

40

Aqui como ele ficaria para quatro variáveis.

Lembrando que ele pode ser feito para cinco, seis, sete ou quantas mais variáveis forem necessárias, mas também perde sua eficiência com esse crescimetnto.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - De onde surgem as simplificações
  - 2, 4, 8, etc.

41

Mas como o agrupamento de quadros deve ser feito?

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:

AB \ C	0	1
00	0	0
01	1	0
11	1	0
10	0	0

42

Ele deve ser feito com o máximo número de 1's possível, desde que ele seja correspondente a uma potência de 2. Assim, pode ser de 2 quadros, como no exemplo deste slide.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:

AB \ C	0	1
00	0	0
01	1	0
11	1	0
10	0	0

$$S = B\overline{C}$$

43

Da forma como o agrupamento foi feito ele produziu a função S = B*C'. Veja que só foram preservadas as variáveis que, no agrupamento, possuem sempre o mesmo valor no código Gray. Como B é sempre '1', então permanece na função como B. Como C é sempre '0', então permanece na função como C'.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:

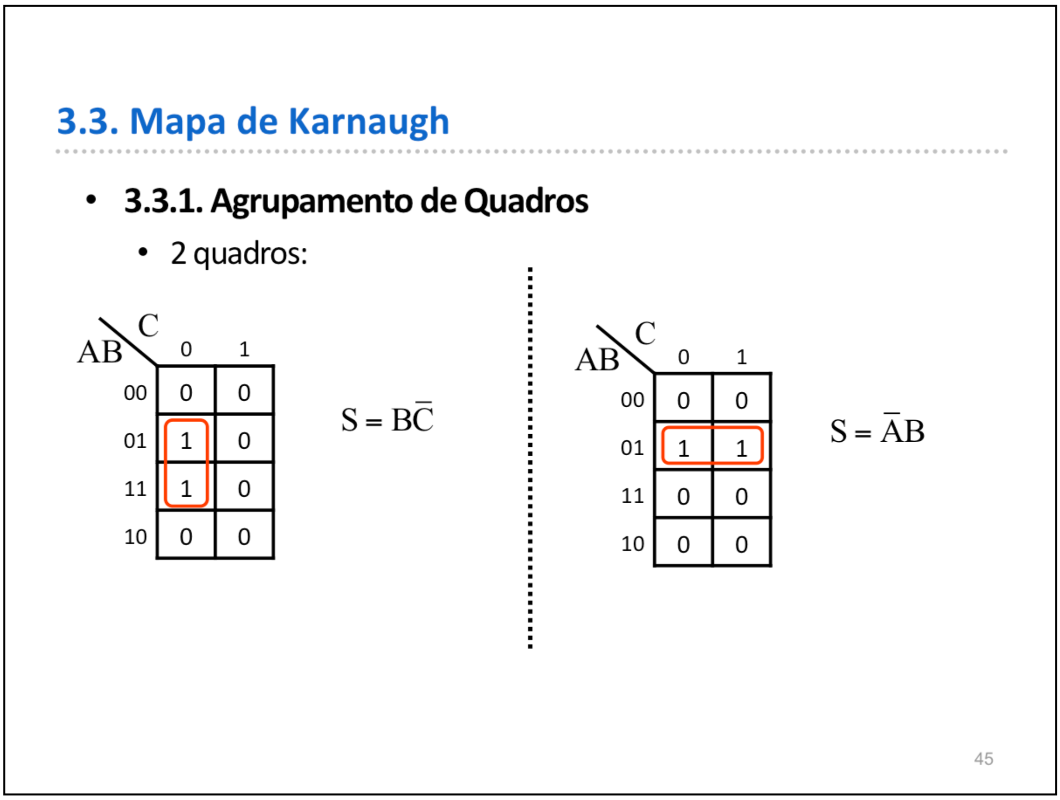
AB \ C	0	1
00	0	0
01	1	0
11	1	0
10	0	0

$S = B\overline{C}$

AB \ C	0	1
00	0	0
01	1	1
11	0	0
10	0	0

44

Outra possibilidade de agrupamento de 2 quadros é a deste slide.



Ela produz a saída S=A'*B.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:

AB \ C	0	1
00	1	0
01	0	0
11	0	0
10	1	0

46

Caso tivéssemos uma situação assim...

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:

AB \ C	0	1
00	1	0
01	0	0
11	0	0
10	1	0

$$S = \overline{B}\,\overline{C}$$

47

Poderíamos fazer o seguinte. Lembre-se que B e C são sempre '0'.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Aqui podemos fazer dois agrupamentos...

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$$S = \overline{A}\,\overline{B}C + A\overline{B}\,\overline{D}$$

49

Estes são eles, produzindo S = A'*B'*C+A*B'*D'. Veja que cada agrupamento produz um "produto".

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:
    - Supondo uma escolha infeliz de agrupamentos:

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$$S = \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C}\,\overline{D} + \overline{B}C\overline{D}$$

Outra alternativa seria fazer estes agrupamentos. Entretanto, veja que eles não foram feitos com o máximo número de 1's possível, então não produzem a função mais simples.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 2 quadros:
    - Supondo uma escolha infeliz de agrupamentos:

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$$
\begin{aligned}
S &= \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C}\,\overline{D} + \overline{B}C\overline{D} \\
&= \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C}\,\overline{D} + (A + \overline{A})\overline{B}C\overline{D} \\
&= \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}C\overline{D} \\
&= \overline{A}\,\overline{B}C(1 + \overline{D}) + A\overline{B}\,\overline{D}(\overline{C} + C) \\
&= \overline{A}\,\overline{B}C + A\overline{B}\,\overline{D}
\end{aligned}
$$

51

Para chegarmos na equação mais simples ainda teríamos que aplicar os teoremas booleanos, como exposto neste slide.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 4 quadros:

AB \ C	0	1
00	0	1
01	0	1
11	0	1
10	0	1

52

Os agrupamentos podem ser também de 4 quadros (2^2 = 4).

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 4 quadros:

AB \ C	0	1
00	0	1
01	0	1
11	0	1
10	0	1

S = C

53

Neste agrupamento feito, a única variável que permanece inalterada é C, sempre '1'.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 4 quadros:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	0	0

Poderíamos ter também uma situação assim.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 4 quadros:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	0	0

$S = AB$

55

Em tal agrupamento, A e B têm sempre os mesmos valores.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 4 quadros:

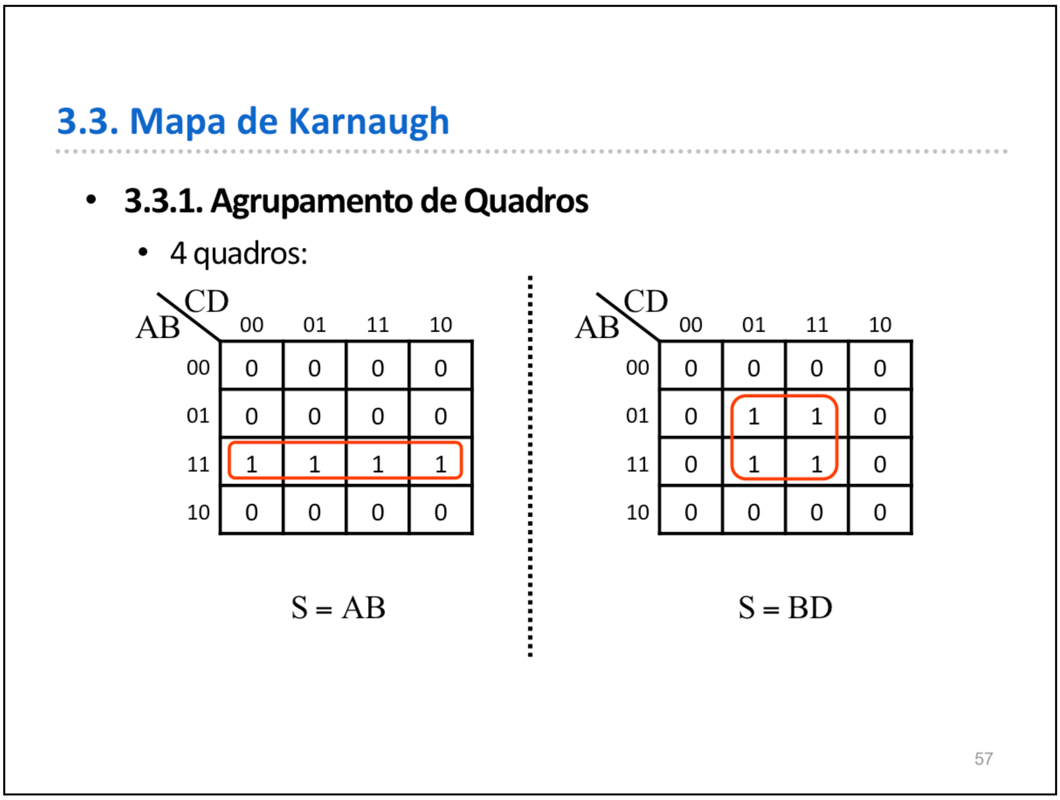
AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	0	0

S = AB

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

56

Outra situação é esta.



Que pode produzir o seguinte agrupamento, onde B e D permanecem sempre os mesmos.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 4 quadros:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	0	0	1
10	1	0	0	1

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Nestestes dois casos podemos utilizar um agrupamento entre as bordas.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 4 quadros:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	0	0	1
10	1	0	0	1

$$S = A\overline{D}$$

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

59

Assim.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 4 quadros:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	0	0	1
10	1	0	0	1

$$S = A\overline{D}$$

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$$S = \overline{B}\,\overline{D}$$

60

E assim. Veja que isto só é possível porque o Mapa de Karnaugh é feito com código Gray.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 8 quadros:

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

AB \ CD	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	1	1	0	0
10	1	1	0	0

61

O agrupamento com 8 quadros (2^3 = 8) segue os mesmos princípios.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 8 quadros:

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

$S = B$

AB \ CD	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	1	1	0	0
10	1	1	0	0

$S = \overline{C}$

62

Assim, os exemplos deste slide produzem estas funções.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 8 quadros:

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

AB \ CD	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

E neste caso, como pode ser o agrupamento para que tenha o maior número possível de 1's?

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - 8 quadros:

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$S = \overline{D}$

AB \ CD	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

$S = \overline{B}$

64

Pelas bordas, produzindo estas funções.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - Processo completo
    - Exemplo 1:

AB \ CD	00	01	11	10
00	0	0	0	1
01	0	1	1	0
11	0	1	1	0
10	0	0	1	0

65

Os agrupamentos também podem ter números de 1's diferentes dentro um Mapa de Karnaugh só.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - Processo completo
    - Exemplo 1:

AB \ CD	00	01	11	10
00	0	0	0	1
01	0	1	1	0
11	0	1	1	0
10	0	0	1	0

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

66

Neste caso, a função mais simples (no slide) é obtida com um agrupamento de 4 quadros, outro de 2 e mais um de 1.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - Processo completo
    - Exemplo 2:

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	1	1	1
11	0	0	0	1
10	1	1	0	1

67

Tente agora obter a função deste Mapa de Karnaugh.

3.3. Mapa de Karnaugh

- **3.3.1. Agrupamento de Quadros**
  - Processo completo
    - Exemplo 2:

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	1	1	1
11	0	0	0	1
10	1	1	0	1

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

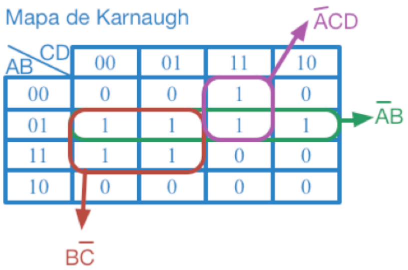
68

Esta é a função mais simples.

PROBLEMAS

Problema 3.2. Obtenha os agrupamentos associados ao seguinte mapa de Karnaugh:

AB \ CD	00	01	11	10
00	0	0	1	0
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0



$$Função(A,B,C,D) = B\bar{C} + \bar{A}B + \bar{A}CD$$

69

Fazemos os agrupamentos máximos que cobrem todos um uns da tabela.

3.3. Mapa de Karnaugh

- **3.3.2. Condições de Irrelevância**

Agora veremos as como se comportam as condições de irrelevância na tabela verdade e no Mapa de Karnaugh.

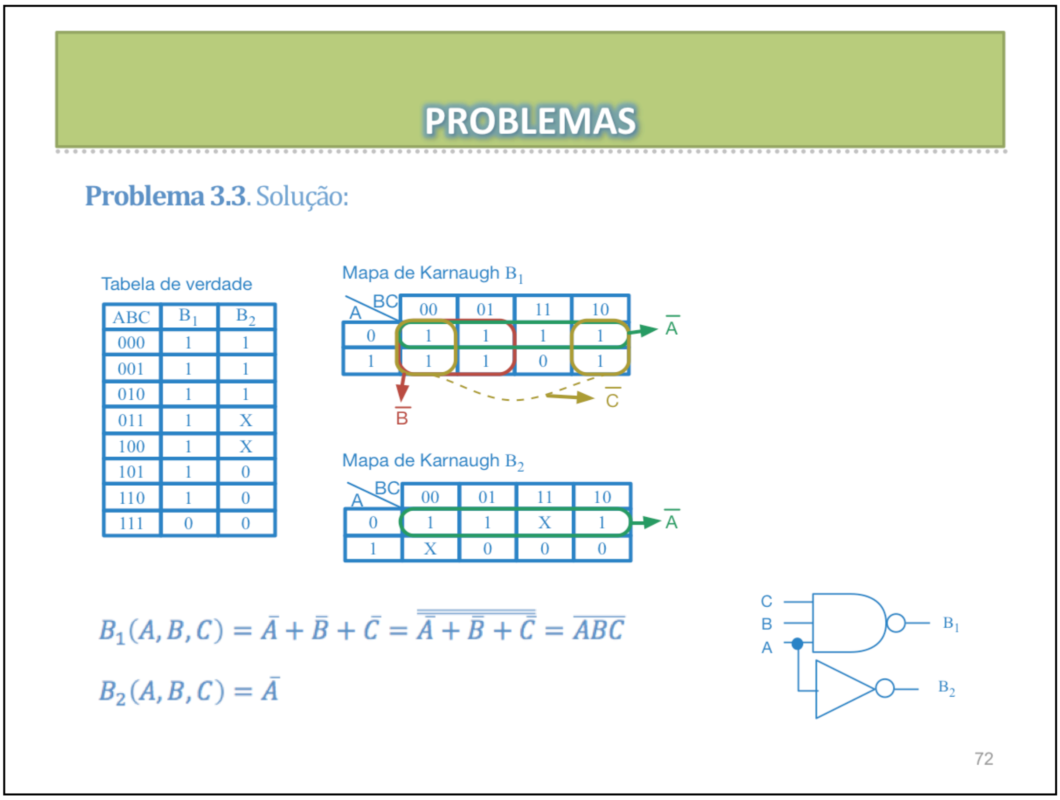
PROBLEMAS

Problema 3.3. Um reservatório de água é controlado a partir de um sensor digital que indica, em uma saída de 3 bits, o nível atual da água no reservatório (0 ou 000_2 indica que o reservatório está vazio e 7 ou 111_2 indica que o reservatório está em sua capacidade máxima). Projete um circuito digital que faça o controle de duas bombas (bomba 1 e bomba 2) que enchem esse reservatório seguindo as seguintes regras:

a) Caso o nível esteja abaixo de 3, as duas bombas devem estar ligadas.
b) Caso o nível esteja acima de 4, apenas a bomba 1 deve estar ligada.
c) Para nível 3 e 4 a bomba 1 está ligada.
d) Para evitar um transbordamento, nenhuma das bombas deve estar ligada caso o nível seja igual a 7.

71

Considere o seguinte problema.



Montamos a tabela de verdade tendo em consideração que para a bomba 2 não existem condições indicadas no enunciado para nível 3 e 4 (condições de irrelevancia ou “don’t care).

A seguir fazemos os agrupamentos máximos que cobrem todos os uns nos mapas de Karnaugh para B1 e B2.

O seguinte passo é obter a expressão reduzida. Notar que para B1 aplicamos dois complementos e o teorema de Morgan para expressa-lo em usando apenas uma porta NAND de três entradas. Está redução adicional é opcional e não obrigatoria para a resolução do problema.

Por último projetamos o circuito.

3.3. Mapa de Karnaugh

- **3.3.2. Condições de Irrelevância (Bomba 2)**

A	B	C	$B2$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	X
1	0	0	X
1	0	1	0
1	1	0	0
1	1	1	0

73

A, B e C fazem parte da saída de 3 bits do sensor que indicam o nível da água. Elas serão as entradas do nosso circuito. B2 é saída que indica se a bomba 2 deve estar ativa (1) ou não (0). As condições listadas no enunciado produzem os seguintes valores para B2. Onde há X, há uma condição de irrelevância (ou "don't care", "não importa"), onde a saída pode assumir qualquer valor.

3.3. Mapa de Karnaugh

- **3.3.2. Condições de Irrelevância**

A	B	C	$B2$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	X
1	0	0	X
1	0	1	0
1	1	0	0
1	1	1	0

AB \ C	0	1
00	1	1
01	1	X
11	0	0
10	X	0

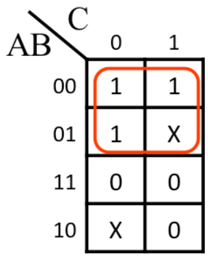
74

O Mapa de Karnaugh correspondente a esta tabela verdade é este.

3.3. Mapa de Karnaugh

- **3.3.2. Condições de Irrelevância**

A	B	C	*B2*
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	X
1	0	0	X
1	0	1	0
1	1	0	0
1	1	1	0



$B2 = \overline{A}$

75

Fazendo um agrupamento de 4 quadros, assumindo que um dos "don't care" é '1', obtemos a função lógica B2 = A'.

3.3. Mapa de Karnaugh

- **3.3.2. Condições de Irrelevância**
  - Exemplo:

M	F1	F2	F3	ABRIR
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	X
0	1	0	0	1
0	1	0	1	X
0	1	1	0	X
0	1	1	1	X

M	F1	F2	F3	ABRIR
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	X
1	1	0	0	0
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

76

Vamos a outro exemplo.

3.3. Mapa de Karnaugh

- **3.3.2. Condições de Irrelevância**
  - Exemplo:

$MF_1 \backslash F_2F_3$	00	01	11	10
00	0	1	X	1
01	1	X	X	X
11	0	X	X	X
10	0	0	X	0

77

A partir da tabela verdade do slide anterior obtemos este Mapa de Karnaugh.

3.3. Mapa de Karnaugh

- **3.3.2. Condições de Irrelevância**
  - Exemplo:

MF_1 \ F_2F_3	00	01	11	10
00	0	1	X	1
01	1	X	X	X
11	0	X	X	X
10	0	0	X	0

$$ABRIR = \bar{M}F_1 + \bar{M}F_2 + \bar{M}F_3$$

78

Podemos assumir que quatro dos oito "don't care" sejam '1', o que nos leva aos agrupamentos e à função lógica apresentada no slide.

3.3. Mapa de Karnaugh

- **3.3.3. Mapa de Karnaugh e Maxitermos**
  - Também é possível...

$M\,F_1 \backslash F_2\,F_3$	00	01	11	10
00	0	1	X	1
01	1	X	X	X
11	0	X	X	X
10	0	0	X	0

79

Ou ainda, alternativamente...

3.3. Mapa de Karnaugh

- **3.3.3. Mapa de Karnaugh e Maxitermos**
  - Também é possível...

$MF_1 \backslash F_2F_3$	00	01	11	10
00	0	1	X	1
01	1	X	X	X
11	0	X	X	X
10	0	0	X	0

Agrupar os 0's, em vez dos 1's, assumindo alguns "don't care" como 0 também.

3.3. Mapa de Karnaugh

- **3.3.3. Mapa de Karnaugh e Maxitermos**
  - Também é possível...

$MF_1 \backslash F_2F_3$	00	01	11	10
00	0	1	X	1
01	1	X	X	X
11	0	X	X	X
10	0	0	X	0

$$\text{ABRIR} = \overline{M} \cdot (F_1 + F_2 + F_3)$$
$$= \overline{M}F_1 + \overline{M}F_2 + \overline{M}F_3$$

81

Entretanto, neste caso, devemos representar a função lógica obtida diretamente a partir do Mapa de Karnaugh em forma de produto de somas. Neste caso, as variáveis que permanecem iguais a '1' são as que devemos negar.

Observe que, após a aplicação do Teorema de DeMorgan, a equação final é igual à obtida através da agrupação de 1's.

FEDERAL UNIVERSITY
OF SANTA CATARINA

EEL5105 – Circuitos e Técnicas Digitais
Aula 3

Prof. Héctor Pettenghi

hector@eel.ufsc.br
http://hectorpettenghi.paginas.ufsc.br

Exercícios

- Os exercícios da **1ª edição** do livro do **Vahid** indicados abaixo são os recomendados:
  - **2.53-2.54, 2.57-2.58, 6.2-6.8**
- **A versão digital da 1ª edição do livro do Vahid está disponível no site da BU**
  - http://www.bu.ufsc.br/framebases.html opção **Minha Biblioteca**
  - Se, após o login na **Minha Biblioteca**, o livro não aparecer na lista de livros, use o seguinte link: http://integrada.minhabiblioteca.com.br/books/9788577802371

83

Exercícios

- Os exercícios de diferentes edições do livro do Tocci indicados abaixo são os recomendados:

10ª Edição	11ª Edição
4.1 até o 4.19	4.4 até o 4.19
4.26 (muito recomendado)	4.26 (muito recomendado)
4.27 até o 4.30	4.27 até o 4.30

  - Dê preferência aos exercícios marcados com * pois eles têm respostas no final do livro;
  - No caso dos exercícios sem resposta, você pode testar o resultado em algum simulador.

- **Versões digitais da 10ª e 11ª edições do livro do Tocci estão disponíveis no site da BU**
    - http://www.bu.ufsc.br/framebases.html
    - Acessar a **Biblioteca Virtual 3.0** com seu login e senha da BU