

Capítulo II

Álgebra Booleana e Minimização Lógica

1 Introdução

Vimos no Capítulo I que a unidade básica construtiva de um sistema digital é a **Porta Lógica** e que **Funções Lógicas** com diversas variáveis de entrada podem ser obtidas mediante a interligação de portas lógicas básicas. Aliás, a própria porta lógica básica (NAND, NOR, XOR, etc...) executa uma função lógica elementar.

Vimos também no final do Capítulo I que para facilitar o tratamento analítico das diversas funções lógicas possíveis de serem implementadas através da interligação entre portas, utiliza-se a representação da função lógica através de Equações Booleanas, conforme mostra a Tabela I a seguir:

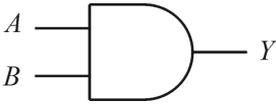
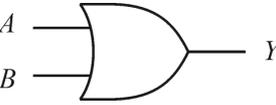
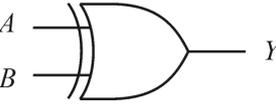
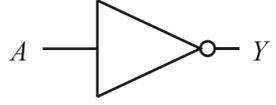
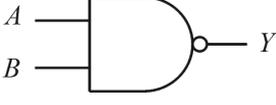
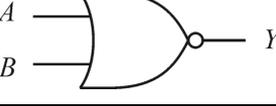
Função Lógica Básica	Símbolo Gráfico da Porta	Equação Booleana
AND		$Y = A \cdot B$
OR		$Y = A + B$
XOR		$Y = A \oplus B$
NOT		$Y = \bar{A}$
NAND		$Y = \overline{A \cdot B}$
NOR		$Y = \overline{A + B}$
XNOR		$Y = \overline{A \oplus B}$

Tabela 1: Equações Booleanas básicas correspondentes às Funções Lógicas Básicas.

Este capítulo descreve o método algébrico para análise e projeto de circuitos digitais que utilizam portas lógicas. As operações algébricas elementares do método algébrico Booleano consiste nas Equações Booleanas mostradas na Tabela I.

Veremos que:

- Não importando o número de variáveis de entrada, a quantidade e os tipos de portas lógicas interligadas necessárias para que se obtenha uma função lógica desejada na saída Y ,

- Não importando o número de variáveis de entrada da tabela verdade que descreve uma função lógica $Y = f(A, B, \dots)$

⇒ **Sempre** poderemos escrever uma equação algébrica Booleana que poderá ser simplificada e/ou otimizada através do uso dos **Teoremas e Postulados Booleanos**.

2 Teoremas e Postulados Booleanos

A Álgebra Booleana possui as mesmas propriedades da Álgebra Linear ordinária, se considerarmos:

- a operação lógica básica A AND B como a multiplicação $A \cdot B$ (ou AB)
- a operação A OR B como a soma $A + B$

Propriedade Comutativa:	$AB = BA$ $A + B = B + A$
Propriedade Associativa:	$A(BC) = (AB)C$ $A + (B + C) = (A + B) + C$
Propriedade Distributiva:	$A(B + C) = AB + AC$

Tabela 2: Propriedades da Álgebra Booleana.

P1	$A = 1$ se $A \neq 0$	P6	$0 + 0 = 0$
P2	$A = 0$ se $A \neq 1$	P7	$1 \cdot 0 = 0$
P3	$0 \cdot 0 = 0$	P8	$0 + 1 = 1$
P4	$1 + 1 = 1$	P9	$\bar{0} = 1$
P5	$1 \cdot 1 = 1$	P10	$\bar{1} = 0$

Tabela 3: Postulados da Álgebra Booleana.

T1	$A + 0 = A$	T8	$\overline{(\bar{A})} = A$
T2	$A \cdot 1 = A$	T9	$A + \bar{A} = 1$
T3	$A + 1 = 1$	T10	$A \cdot \bar{A} = 0$
T4	$A \cdot 0 = 0$	T11	$\overline{A + B + C + \dots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \dots$ (Teorema I de Morgan)
T5	$A + A = A$	T12	$\overline{A \cdot B \cdot C \dots} = \bar{A} + \bar{B} + \bar{C} + \dots$ (Teorema II de Morgan)
T6	$A \cdot A = A$	T13	$A(A + B) = A + AB = A(1 + B) = A$
T7	$\overline{(\bar{A})} = \bar{A}$	T14	$A + AB = A(A + B) = A$

Tabela 4: Teoremas da Álgebra Booleana.

Exemplo 1:

Determinar a expressão (equação) Booleana que representa a Tabela Verdade abaixo. Simplifique e otimize a expressão utilizando os resultados das Tabelas 2, 3 e 4. Desenhe a interligação de portas básicas que implementa esta Tabela Verdade.

<i>A</i>	<i>B</i>	<i>C</i>	<i>Y</i>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Tabela 5: Tabela verdade de uma função lógica hipotética de 3 variáveis.

Solução:

$$Y = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C}$$

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

Mas a função lógica XOR com duas variáveis *A* e *C* tem a seguinte Tabela Verdade/Expressão Booleana:

<i>A</i>	<i>C</i>	$Y = A \oplus C = \overline{A}C + A\overline{C}$
0	0	0
0	1	1
1	0	1
1	1	0

Logo,

$$Y = \overline{A}\overline{B}C + B(A \oplus C)$$

Utilizando o **T11** da Tabela 4 obtemos a seguinte Expressão Booleana simplificada:

$$Y = (\overline{A+B})C + B(A \oplus C)$$

Que resulta no seguinte circuito lógico:

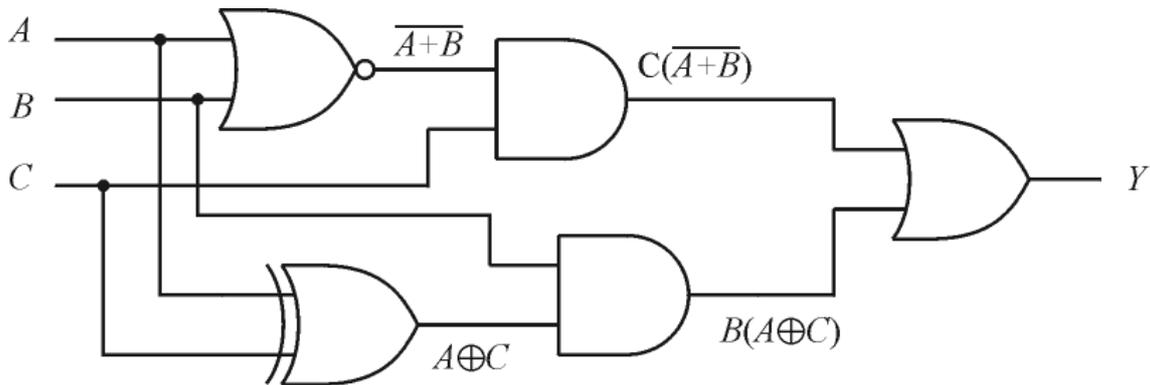


Figura 1: Interligação de portas básicas que implementa a Tabela Verdade da Tabela 5.

3 Mapas de Karnaugh

⇒ Um Mapa de Karnaugh (Mapa K) é a representação das linhas de uma Tabela Verdade em forma de quadrículos adjacentes.

⇒ Dois quadrículos adjacentes verticalmente ou horizontalmente em um mapa K correspondem à duas linhas da Tabela Verdade tal que **apenas uma variável** tenha seu valor lógico alterado de um quadrículo para o outro. Isto permite que a Propriedade Distributiva da Tabela 2 em conjunto com o teorema **T9** da Tabela 4 leve à eliminação de uma variável.

⇒ A simplificação lógica obtida com um Mapa K segue os seguintes princípios:

- (I) Seleciona-se uma combinação de quadrículos tal que inclua todos os quadrículos pelo menos uma vez, sendo o número de quadrículos selecionados uma potência inteira de 2. Ou seja, um quadrículo pode aparecer em mais de uma combinação.
- (II) As combinações devem ser selecionadas objetivando incluir o maior número de quadrículos por combinação, utilizando para tanto o menor número possível de combinações.

Exemplo 2:

Simplifique a Expressão Booleana resultante da Tabela Verdade abaixo.

<i>A</i>	<i>B</i>	<i>C</i>	<i>Y</i>
0	0	0	1
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	1

Solução:

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0	1	1	1	0
	1	0	1	1	1

Figura 2

$$Y = \overline{A}\overline{B} + C + AB$$

Exemplo 3:

Simplifique a Expressão Booleana resultante da Tabela Verdade abaixo.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
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	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Solução:

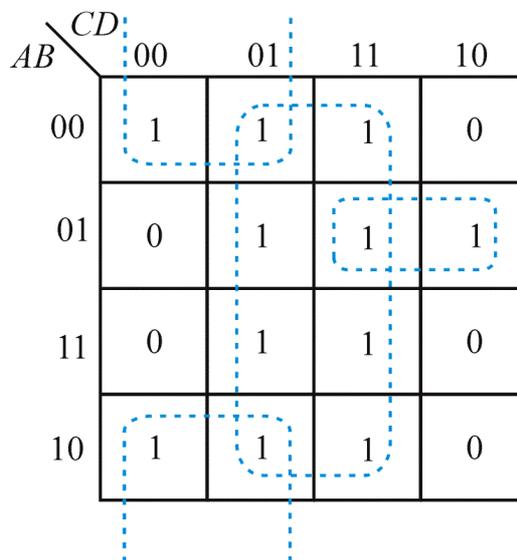


Figura 3

$$Y = \overline{A}BC + D + \overline{B}\overline{C}$$

Exemplo 4:

Simplifique a Expressão Booleana resultante da Tabela Verdade abaixo.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
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	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Solução:

	<i>CD</i>	00	01	11	10
<i>AB</i>	00	0	0	0	0
	01	1	1	0	0
	11	1	1	0	0
	10	0	0	0	0

Figura 4

$$Y = B\bar{C}$$

Exemplo 5:

Simplifique a Expressão Booleana resultante da Tabela Verdade abaixo.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
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	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Solução:

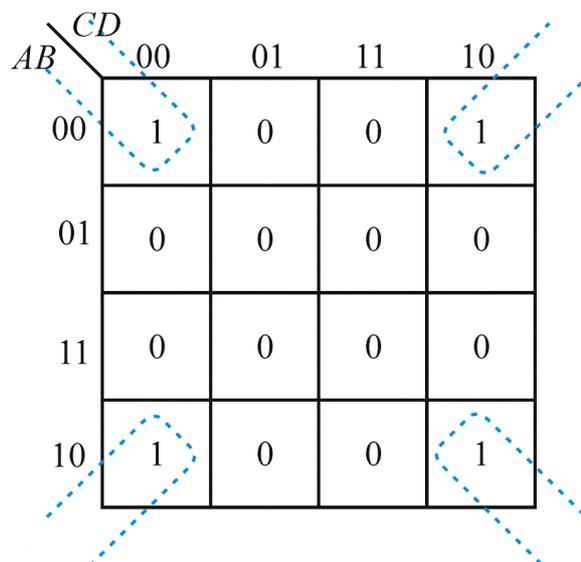


Figura 5

$$Y = \overline{B\overline{D}}$$

Exemplo 6:

Simplifique a Expressão Booleana resultante da Tabela Verdade abaixo.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
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	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	0
1	1	1	1	0

Solução:

		<i>CD</i>			
	<i>AB</i>	00	01	11	10
00		0	0	0	0
01		1	1	1	1
11		0	0	0	0
10		0	0	0	0

Figura 6

$$Y = \overline{AB}$$

Exemplo 7:

Simplifique a Expressão Booleana resultante da Tabela Verdade abaixo.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Solução:

		<i>CD</i>			
		00	01	11	10
<i>AB</i>	00	0	1	1	0
	01	0	0	0	0
	11	0	0	0	0
	10	0	1	1	0

Figura 7

$$Y = \overline{BD}$$

Exemplo 8:

Simplifique a Expressão Booleana resultante da Tabela Verdade abaixo.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Solução:

		<i>CD</i>			
		00	01	11	10
<i>AB</i>	00	1	1	0	0
	01	1	1	0	0
	11	1	1	0	0
	10	1	1	0	0

Figura 8

$$Y = \bar{C}$$

Exemplo 9:

Simplifique a Expressão Booleana resultante da Tabela Verdade abaixo.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Solução:

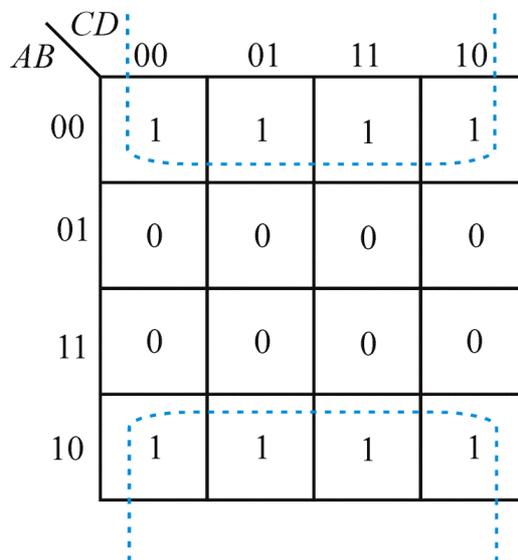


Figura 9

$$Y = \overline{B}$$

3.1 Método de uso dos Mapas de Karnaugh

- Para efeito de sistematizar o uso de um Mapa K na minimização lógica, sugere-se adotar o seguinte procedimento:
 - (I) Assinalar inicialmente apenas os quadrículos que **não podem ser combinados com nenhum outro**.
 - (II) Identificar os quadrículos que podem ser combinados com um **único** outro quadrículo **somente de uma maneira**. Assinalar estas combinações de **dois quadrículos** por combinação. Quadrículos que podem ser combinados em **grupos de dois de mais de uma maneira** são deixados temporariamente de lado.
 - (III) Identificar quadrículos que podem ser combinados com **três** outros quadrículos **somente de uma maneira**. Assinalar estas combinações de **quatro quadrículos** por combinação. Quadrículos que podem ser combinados em **grupos de quatro de mais de uma maneira** são deixados temporariamente de lado.
 - (IV) Identificar quadrículos que podem ser combinados com **sete** outros quadrículos **somente de uma maneira**. Assinalar estas combinações de **oito quadrículos** por combinação. Quadrículos que podem ser combinados em **grupos de oito de mais de uma maneira** são deixados temporariamente de lado.
 - (V) Repetir o processo para grupos de **16,32**, etc...
 - (VI) Se, uma vez encerrado o processo acima, ainda restarem quadrículos não incluídos em agrupamentos, estes quadrículos podem ser combinados uns com os outros ou com quadrículos já incluídos em outros agrupamentos (**se houver adjacência e o agrupamento resultante contiver uma potência inteira de 2**).
 - (VII) É importante lembrar que **o objetivo é obter o menor número de agrupamentos possível, cada agrupamento contendo o maior número possível de quadrículas que resulte em uma potência inteira de 2**.

⇒ As Figuras 10, 11, 12 e 13 mostram a aplicação do método em um exemplo específico:

		<i>CD</i>			
<i>AB</i>		00	01	11	10
	00	1	0	1	0
	01	1	1	1	1
	11	1	0	1	1
	10	0	1	0	0

Figura 10: Mapa de Karnaugh para a função lógica descrita por $Y = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + ABC\overline{D} + ABCD + ABC\overline{D} + \overline{A}BC\overline{D}$
 Etapa (I) do método para sistematização do uso de mapas K.

		<i>CD</i>			
<i>AB</i>		00	01	11	10
	00	1	0	1	0
	01	1	1	1	1
	11	1	0	1	1
	10	0	1	0	0

Figura 11: Mapa de Karnaugh para a função lógica descrita por $Y = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + ABC\overline{D} + ABCD + ABC\overline{D} + \overline{A}BC\overline{D}$
 Etapa (II) do método para sistematização do uso de mapas K.

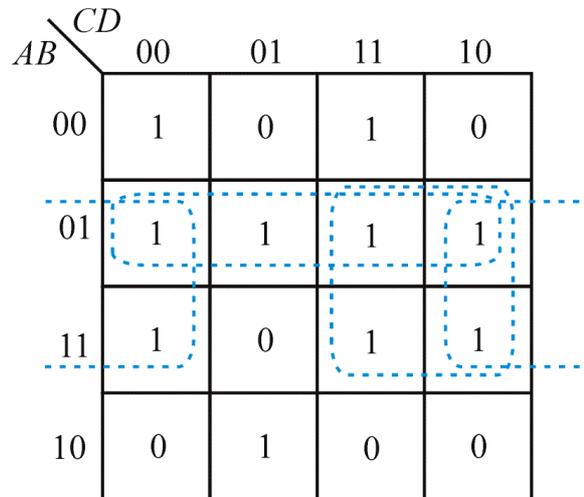


Figura 12: Mapa de Karnaugh para a função lógica descrita por $Y = \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}\overline{D}$ Etapa (III) do método para sistematização do uso de mapas K.

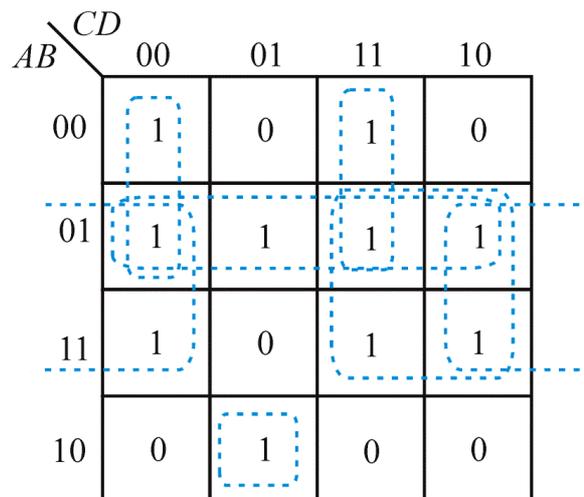


Figura 13: Mapa de Karnaugh para a função lógica descrita por $Y = \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}B\overline{C}\overline{D}$ Mapa K completo. A função lógica minimizada resulta em $Y = \overline{A}BCD + \overline{A}CD + \overline{A}CD + \overline{A}B + \overline{B}D + \overline{B}C$.

3.2 Mapas de Karnaugh para a 5 Variáveis

● Suponhamos que queiramos minimizar a função lógica $Y1 = f(A, B, C, D, E)$ definida por:

$$Y1 = \overline{A}\overline{B}\overline{C}\overline{D}\overline{E} + \overline{A}\overline{B}\overline{C}\overline{D}E + \overline{A}\overline{B}\overline{C}D\overline{E} + \overline{A}\overline{B}\overline{C}DE + \overline{A}\overline{B}C\overline{D}\overline{E} + \overline{A}\overline{B}C\overline{D}E + \overline{A}\overline{B}CD\overline{E} + \overline{A}\overline{B}CDE + \overline{A}B\overline{C}\overline{D}\overline{E} + \overline{A}B\overline{C}\overline{D}E + \overline{A}B\overline{C}D\overline{E} + \overline{A}B\overline{C}DE + \overline{A}BC\overline{D}\overline{E} + \overline{A}BC\overline{D}E + \overline{A}BCD\overline{E} + \overline{A}BCDE + A\overline{B}\overline{C}\overline{D}\overline{E} + A\overline{B}\overline{C}\overline{D}E + A\overline{B}\overline{C}D\overline{E} + A\overline{B}\overline{C}DE + AB\overline{C}\overline{D}\overline{E} + AB\overline{C}\overline{D}E + ABC\overline{D}\overline{E} + ABCDE$$

⇒ O Mapa K para $Y1 = f(A, B, C, D, E)$ é:

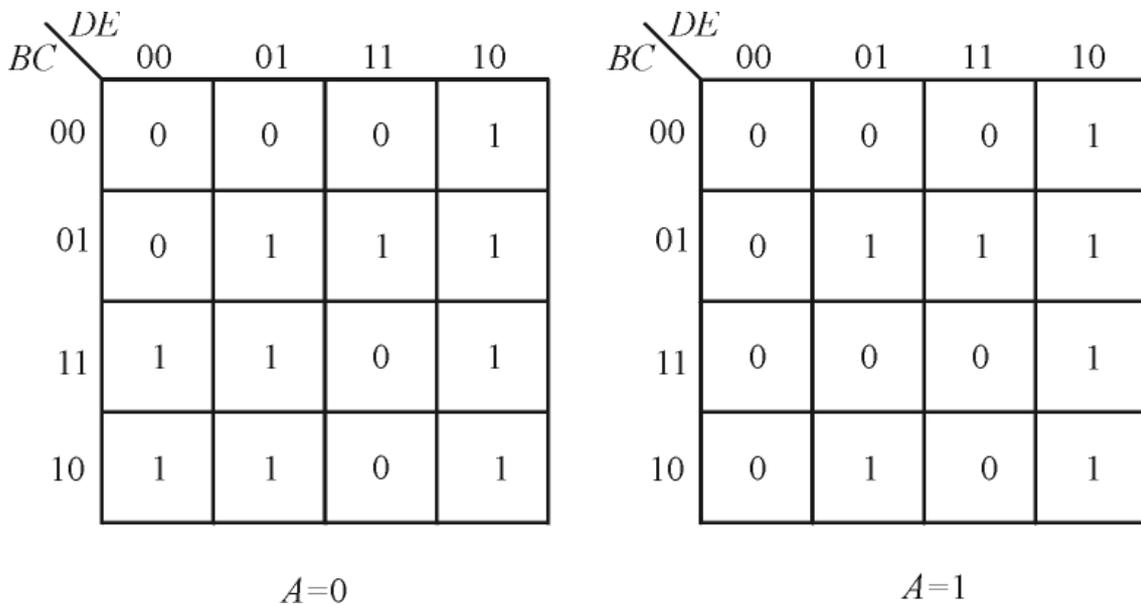


Figura 14: Mapa de Karnaugh para a função lógica $Y1 = f(A, B, C, D, E)$.

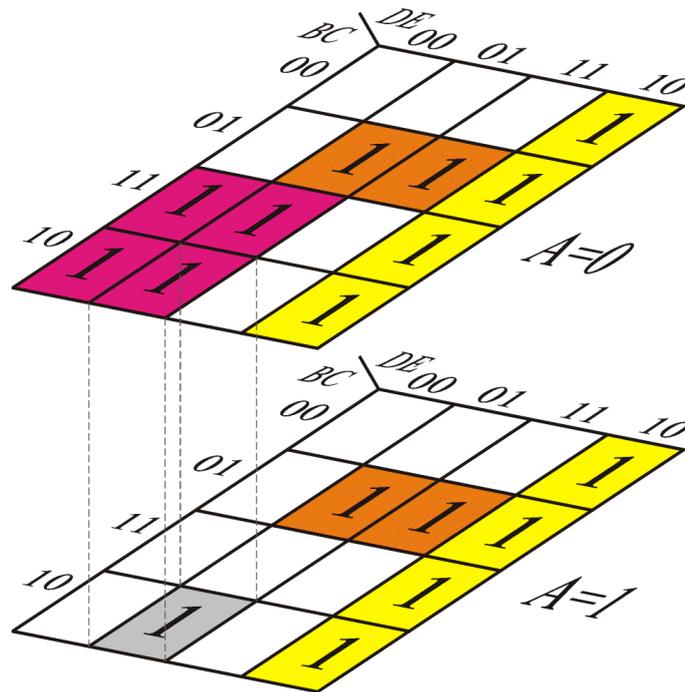


Figura 15: Adjacências entre quadrículos no Mapa de Karnaugh para a função lógica $Y1 = f(A, B, C, D, E)$ dada, caracterizadas sob um ponto de vista tridimensional. O termo resultante para o agrupamento amarelo é $D\bar{E}$, para o agrupamento laranja é $\bar{B}CE$, para o agrupamento magenta é $\bar{A}B\bar{D}$, e para o agrupamento cinza (superposto ao magenta para $A = 0$) é $\bar{B}C\bar{D}E$. Portanto, a função lógica minimizada resulta em $Y1 = D\bar{E} + \bar{B}CE + \bar{A}B\bar{D} + \bar{B}C\bar{D}E$.

● Suponhamos agora que queiramos minimizar a função lógica $Y2 = f(A, B, C, D, E)$ definida por:

$$Y2 = \bar{A}\bar{B}\bar{C}\bar{D}\bar{E} + \bar{A}\bar{B}\bar{C}\bar{D}E + \bar{A}\bar{B}\bar{C}D\bar{E} + \bar{A}\bar{B}\bar{C}DE + \bar{A}\bar{B}C\bar{D}\bar{E} + \bar{A}\bar{B}C\bar{D}E + \bar{A}\bar{B}CD\bar{E} + \bar{A}\bar{B}CDE + \bar{A}B\bar{C}\bar{D}\bar{E} + \bar{A}B\bar{C}\bar{D}E + \bar{A}B\bar{C}D\bar{E} + \bar{A}B\bar{C}DE + \bar{A}BC\bar{D}\bar{E} + \bar{A}BC\bar{D}E + \bar{A}BCD\bar{E} + \bar{A}BCDE$$

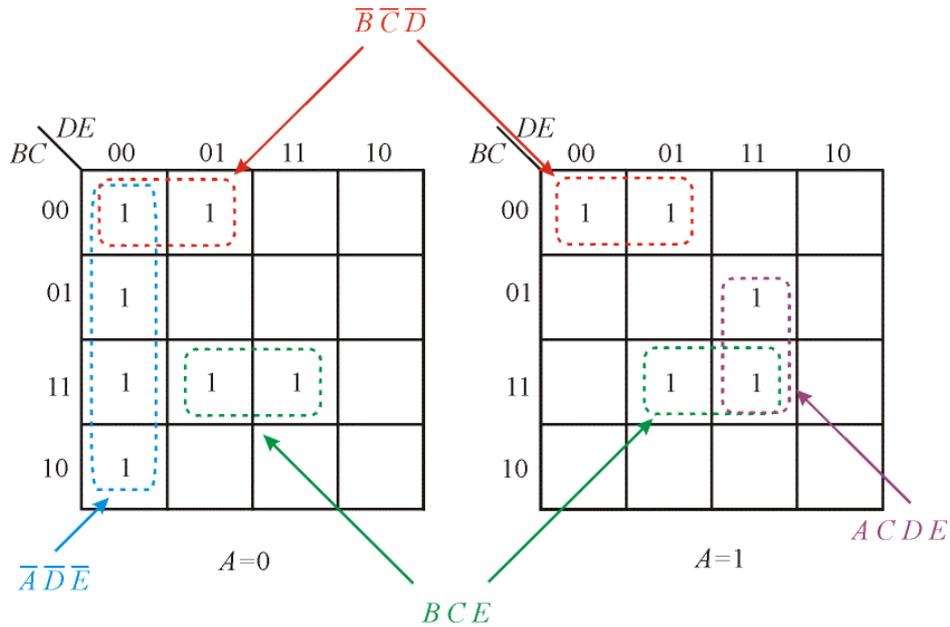


Figura 16: Adjacências entre quadrículos no Mapa de Karnaugh para a função lógica $Y2 = f(A, B, C, D, E)$ dada, caracterizadas sob um ponto de vista bidimensional. A função lógica minimizada resulta em $Y2 = ACDE + \overline{ADE} + BCE + \overline{BCD}$.

3.3 Mapas de Karnaugh par a 6 Variáveis

● Suponhamos que queiramos minimizar a função lógica $Y3 = f(A, B, C, D, E, F)$ definida por:

$$Y3 = \overline{ABCDEF} + \overline{ABCDEF}$$

⇒ O Mapa K para $Y3 = f(A, B, C, D, E, F)$ é:

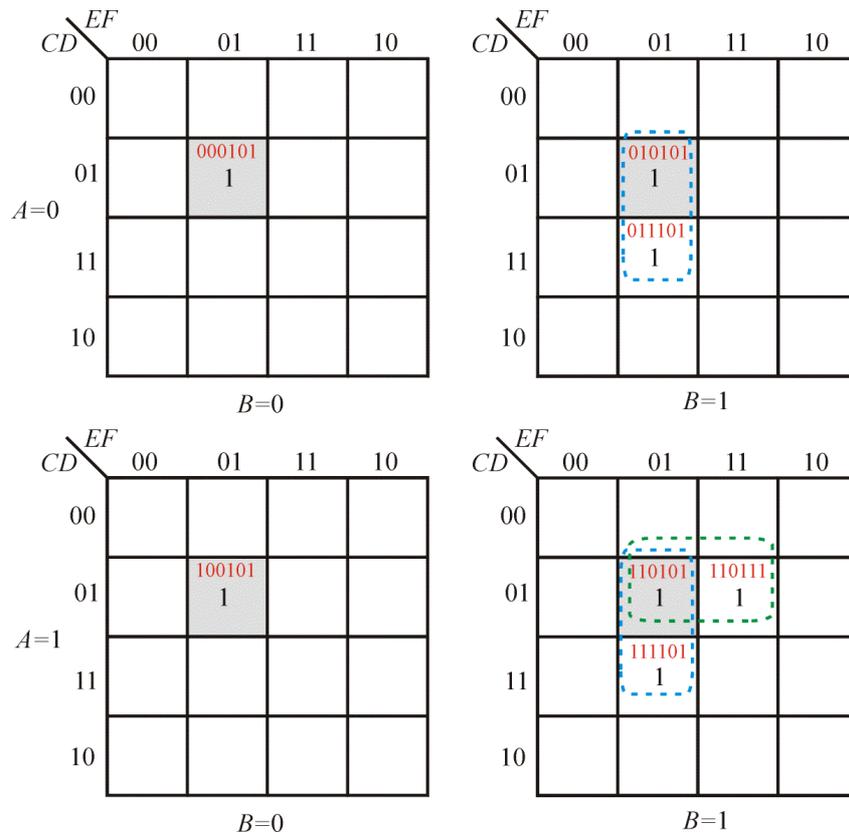


Figura 17: Adjacências entre quadrículos no Mapa de Karnaugh para a função lógica $Y3 = f(A, B, C, D, E, F)$ dada, caracterizadas sob um ponto de vista bidimensional. A função lógica minimizada resulta em $Y3 = \overline{C}\overline{D}\overline{E}F + B\overline{D}\overline{E}F + ABC\overline{D}F$.

4 Funções Incompletamente Especificadas (*don't care condition*)

● Vamos supor que um determinado processo industrial a ser controlado por um circuito lógico tenha uma variável Y representada por:

A	B	C	D	Y
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	X
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	0
1	1	0	1	0
1	1	1	0	X
1	1	1	1	X

Tabela 6: Tabela verdade de $Y = f(A, B, C, D)$.

⇒ O valor “X” atribuído à saída Y em determinadas linhas da Tabela Verdade significa que, para os específicos valores lógicos das variáveis A, B, C e D nestas linhas, o valor lógico da saída Y é **irrelevante para o processo controlado (*don't care*)**.

⇒ O mapa K resultante é

		CD			
		00	01	11	10
AB	00				
	01	1	1	X	1
	11			X	X
	10	1	1	X	X

Figura 17: Mapa K para a Tabela 6.

⇒ Mas, uma vez que os quadrículos contendo X representam situações irrelevantes ao processo industrial, podemos atribuir a cada X um valor lógico conveniente no contexto de minimização lógica de forma a nos permitir agrupar o maior número possível de quadrículos gerando o menor número possível de agrupamentos:

		<i>CD</i>			
<i>AB</i>		00	01	11	10
00					
01		1	1	1	1
11					
10		1	1	1	1

Figura 18: Mapa K para a Tabela 6 com os valores lógicos dos Xs atribuídos objetivando a minimização da função lógica resultante. A função lógica minimizada resulta em $Y = \overline{A}B + A\overline{B} = A \oplus B$.

5 Distância de Hamming

- As linhas de uma Tabela Verdade formam **Palavras Binárias** formadas por tantos bits (bit: *binary unit*) quantos sejam o número de variáveis da função lógica descrita pela tabela. Por exemplo, as linhas nas 4 primeiras colunas da Tabela 6 formam palavras binárias de 4 bits.
- Em muitas situações práticas de controle digital de processos industriais tais palavras binárias constituem **Instruções de Comando** que devem ser enviadas por longas distâncias através de um **Canal de Transmissão** (cabo coaxial, fibra ótica, etc...) antes de chegarem ao destino onde a instrução desencadeará uma ação específica no processo controlado. A Tabela 7 mostra um possível exemplo com 4 instruções de comando cada uma delas definida por uma palavra binária de 5 bits:

Palavra-Código binária (Instrução de Comando) enviada remotamente	Ação efetuada no Ponto de Destino
$[0 \ 0 \ 0 \ 0 \ 0]$	Abre comporta da represa
$[0 \ 1 \ 0 \ 1 \ 1]$	Fecha comporta da represa
$[1 \ 0 \ 1 \ 0 \ 1]$	Liga motor da bomba de dreno
$[1 \ 1 \ 1 \ 1 \ 0]$	Desliga motor da bomba de dreno

Tabela 7: Exemplo de processo remotamente controlado. O conjunto de instruções de comando é formado por 4 instruções cada uma delas definida por uma palavra binária de 5 bits.

- **Sempre** que palavras binárias são enviadas através de um Canal de Transmissão estas ficam sujeitas a algum tipo de **Interferência** (ruído aleatório, interferência de outras fontes de energia, interferência intersimbólica, etc...).

⇒ Portanto, devido à interferência sofrida no canal de transmissão, as Instruções de Comando de um processo remotamente controlado podem chegar ao ponto de destino com alguns de seus bits tendo seu valor lógico invertido. Isto constitui um **Erro de Transmissão** que deve ser corrigido.

- Uma técnica de correção de erros de transmissão é a denominada FEC (*Forward Error Correction*). Em palavras simples, quando uma palavra binária chega ao seu destino ela é comparada com uma tabela contendo todas as possíveis Instruções de Comando de um processo, denominada Tabela de Decodificação.

- A comparação é efetuada com base na **Distância de Hamming** entre a palavra binária recebida e aquelas contidas na Tabela de Decodificação. A **Distância de Hamming** entre duas palavras binárias é a contagem dos bits com valores lógicos complementares em posições correspondentes nas duas palavras.

● Por exemplo, suponhamos que para o processo remotamente controlado da Tabela 7, seja enviada em determinado instante a instrução $[0\ 1\ 0\ 1\ 1]$ (fecha comporta da represa) e que, por ação de interferência no canal de transmissão, seja recebido no ponto de destino a palavra binária errada $[0\ 1\ 0\ 0\ 1]$ (erro no segundo bit da direita para a esquerda).

⇒ O decodificador FEC no ponto de destino calcula as Distâncias de Hamming entre a palavra recebida e todas as possíveis instruções válidas (Tabela de Decodificação):

Palavra-Código binária (Instrução de Comando) enviada remotamente	Distância de Hamming da Palavra-Código recebida $[0\ 1\ 0\ 0\ 1]$:
$[0\ 0\ 0\ 0\ 0]$	2
$[0\ 1\ 0\ 1\ 1]$	1
$[1\ 0\ 1\ 0\ 1]$	3
$[1\ 1\ 1\ 1\ 0]$	4

Tabela 8: Distâncias de Hamming entre a palavra $[0\ 1\ 0\ 0\ 1]$ e todas as possíveis instruções válidas (Tabela de Decodificação).

⇒ A seguir, o decodificador FEC no ponto de destino faz a seguinte inferência: A instrução originalmente transmitida é aquela que resulta na **menor Distância de Hamming da palavra recebida sob erro** (instrução que é “mais parecida” com a palavra recebida).

⇒ Portanto, da Tabela 8, o decodificador FEC infere que a instrução originalmente transmitida foi $[0\ 1\ 0\ 1\ 1]$ (menor Distância de Hamming entre as 4 obtidas). Note que o decodificador efetuou uma inferência correta, porque a palavra originalmente transmitida é efetivamente a palavra inferida.

⇒ Observe que se tivesse sido recebida uma palavra binária com mais de um bit em erro, o decodificador FEC deste exemplo não teria capacidade de corrigir os erros múltiplos. Portanto, surge a questão: Qual é o fator que determina a capacidade de um decodificador FEC corrigir erros múltiplos ?

⇒ O fator que rege a Capacidade de Correção de Erro de um decodificador FEC é a **Distância de Hamming** entre as palavras binárias do conjunto de instruções.

⇒ Quanto **maior** for a **Mínima Distância de Hamming obtida entre todas as palavras binárias do conjunto de instruções**, **maior** será a capacidade de correção do decodificador FEC. Demonstra-se que o número de erros simultâneos t que um decodificador FEC é capaz de corrigir dado por

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \quad (1)$$

onde d_{\min} representa a Mínima Distância de Hamming obtida entre todas as palavras binárias do conjunto de instruções e $\lfloor \cdot \rfloor$ é o operador que resulta no menor inteiro mais próximo do argumento.

⇒ No exemplo da Tabela 7, $d_{\min} = 3$ resultando $t = 1$, o que significa que o decodificador FEC consegue corrigir no máximo um bit recebido em erro. Para aumentar a capacidade de correção teríamos que utilizar instruções representadas por palavras binárias com um número maior do que 5 bits, de modo a aumentar a Mínima Distância de Hamming entre elas.