Capítulo 2

Funções lógicas

George Boole (1815-1864), matemático e filósofo britânico, criou um sistema matemático de análise lógica chamado álgebra de Boole ou álgebra booleana. Esse sistema permitiu elaborar expressões conhecidas como funções lógicas, que possibilitaram o desenvolvimento da eletrônica digital. Para iniciar o estudo, vamos analisar o circuito da figura 2.1.

Sejam as variáveis S1, S2 e L, tais que:

S1 = S2 = 0 → chaves abertas
S1 = S2 = 1 → chaves fechadas
L = 0 → lâmpada apagada
L = 1 → lâmpada acesa

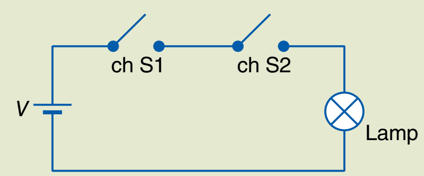


Figura 2.1
Circuito elétrico com duas chaves e uma lâmpada.

Assim, por exemplo:

- Se S1 = 1 (chave S1 fechada) e S2 = 1 (chave S2 fechada) → L = 1 (lâmpada acesa)
- Se S1 = 1 (chave S1 fechada) e S2 = 0 (chave S2 aberta) → L = 0 (lâmpada apagada)

A condição da lâmpada (acesa/apagada) é função (depende) da condição de cada uma das chaves (aberta/fechada) do circuito. Nessa função, não são consideradas quantidades (números), e sim os estados de variáveis, em que somente duas condições são possíveis: "0" ou "1". Essas variáveis, que podem assumir apenas dois estados (0/1, aberto/fechado, sim/não, verdadeiro/falso etc.), são chamadas **variáveis booleanas**, e os estados, **estados lógicos**, associados às variáveis. Quando estão atuando nessas condições, as variáveis booleanas são conhecidas como **funções booleanas**, que podem ser simples ou complexas. As funções booleanas simples são obtidas por meio de um conjunto de circuitos eletrônicos denominados **portas lógicas**. Associando portas lógicas, é possível implementar circuitos eletrônicos definidos por funções booleanas mais complexas.

As variáveis utilizadas nos circuitos são representadas pelas letras A, B, C, ..., N. Uma barra sobre uma variável booleana significa que seu valor sofrerá inversão.

Assim, se $A = 0$, $\overline{A} = 1$, e se $A = 1$, $\overline{A} = 0$, em que $\overline{A}$ lê-se: não A, A barra, A barrado ou complemento de A.

As funções booleanas apresentam resultados fornecidos pelas combinações possíveis devido a suas variáveis. Esses resultados são normalmente representados em forma de tabela.

Chamamos **tabela verdade** de uma função booleana a tabela que apresenta, geralmente de maneira ordenada, os valores da função y = f(A, B) para todas as combinações possíveis dos valores das variáveis.

Consideremos y uma função booleana das variáveis A e B, cuja tabela verdade é apresentada na tabela 2.1.

Tabela 2.1
Tabela verdade de y = f(A, B)

A	B	y
0	0	1
0	1	0
1	0	1
1	1	1

A tabela verdade é uma das maneiras de estabelecer a correspondência entre os valores da função e os das variáveis. A penúltima linha da tabela, por exemplo, informa que, nas condições A = 1 e B = 0, y = 1. Outra forma de estabelecer a correspondência é a expressão booleana da função, que será abordada mais adiante.

2.1 Portas lógicas

Portas lógicas são circuitos eletrônicos básicos que possuem uma ou mais entradas e uma única saída. Nas entradas e na saída, podemos associar estados "0" ou "1", ou seja, variáveis booleanas. Em eletrônica digital, quando utilizamos portas lógicas, atribuímos às entradas e às saídas valores de tensão. Nos circuitos exemplos de portas lógicas, associaremos ao 5 V o estado "1" e ao 0 V, o estado "0".

A porta lógica mais simples é denominada inversora. Nela, a saída é igual ao complemento da entrada (figura 2.2).

símbolo

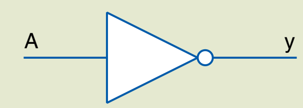


tabela verdade

A	y
0	1
1	0

expressão booleana

$y = \overline{A}$

A → entrada e y → saída

PORTA INVERSORA tem somente 1 entrada

Figura 2.2
Símbolo, tabela verdade e expressão booleana da porta inversora.

A porta OU (OR, em inglês) possui duas ou mais entradas. A saída sempre será igual a “1” quando uma das entradas for igual a “1” (figura 2.3). A saída será “0” somente se todas as entradas forem “0”.

Figura 2.3
Símbolo, tabela verdade e expressão booleana da porta OU.

símbolo

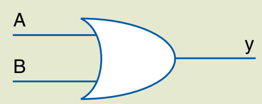


tabela verdade

A	B	y
0	0	0
0	1	1
1	0	1
1	1	1

expressão booleana

y = A + B (lê-se A OU B)

A e B → entradas

y → saída

A saída é “0” somente se todas as entradas forem zero

O símbolo “+” representa OU lógico e não significa uma soma aritmética, pois “0” e “1” não são números, mas estados lógicos das variáveis.

A porta NOU (NOR) corresponde à uma porta OU com a saída invertida (figura 2.4). A saída será “1” somente se todas as entradas forem “0”.

Figura 2.4
Símbolo, tabela verdade e expressão booleana da porta NOU.

símbolo

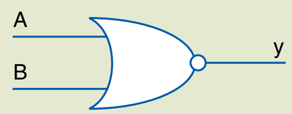


tabela verdade

A	B	y
0	0	1
0	1	0
1	0	0
1	1	0

expressão booleana

$y = \overline{A + B}$

A saída é “1” somente se todas as entradas forem zero

Observe que a “bolinha” no símbolo nega (complementa) a saída, equivalente à barra na expressão booleana, indicando que a porta NOU tem uma saída que corresponde ao complemento da saída da porta OU.

A porta E (AND) possui uma ou mais entradas e sua saída será “1” somente quando todas as entradas forem iguais a “1” (figura 2.5).

Figura 2.5
Símbolo, tabela verdade e expressão booleana da porta E.

símbolo

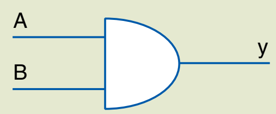


tabela verdade

A	B	y
0	0	0
0	1	0
1	0	0
1	1	1

expressão booleana

$y = AB$ ou $y = A \cdot B$

A saída é “1” somente se todas as entradas forem “1”

A porta NE (NAND) corresponde a uma porta E com a saída invertida (figura 2.6). A saída será “0” somente se todas as entradas forem “1”.

Figura 2.6
Símbolo, tabela verdade e expressão booleana da porta NE.

símbolo

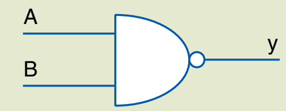


tabela verdade

A	B	y
0	0	1
0	1	1
1	0	1
1	1	0

expressão booleana

$y = \overline{AB}$ ou $y = \overline{A \cdot B}$

A saída é “0” somente se todas as entradas forem “1”

A porta OU EXCLUSIVO (XOR) possui uma ou mais entradas e fornecerá uma saída igual a “1” somente quando as entradas forem diferentes (figura 2.7).

Figura 2.7
Símbolo, tabela verdade e expressão booleana da porta OU EXCLUSIVO.

símbolo

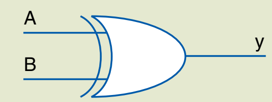


tabela verdade

A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

expressão booleana

$y = A \oplus B$

Saída 1 se as entradas forem diferentes

A porta NOU EXCLUSIVO (XNOR), também chamada de COINCIDÊNCIA, é equivalente a uma porta XOR com a saída invertida (figura 2.8). A saída será “1” se as entradas forem iguais.

Figura 2.8
Símbolo, tabela verdade e expressão booleana da porta NOU EXCLUSIVO.

símbolo

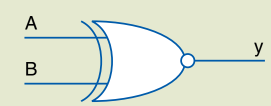


tabela verdade

A	B	y
0	0	1
0	1	0
1	0	0
1	1	1

expressão booleana

$y = A \odot B$

Saída 1 se as entradas forem iguais

2.2 Álgebra booleana

Vimos que na álgebra booleana o estudo de circuitos lógicos é baseado em apenas dois valores (0/1, aberto/fechado, sim/não, verdadeiro/falso etc.), que também podem ser representados por dois níveis distintos de tensão, chamados, por exemplo, nível alto (H – *high*) e nível baixo (L – *low*) ou simplesmente “0” (zero) e “1” (um). A análise das expressões também obedece a esse princípio e, portanto, é perfeitamente aplicável a nosso estudo.

Os símbolos H/L ou 0/1 podem ser empregados para representar situações do tipo:

- sim/não;
- verdadeiro/falso;
- ligado/desligado (*on/off*);
- aceso/apagado.

Obviamente, essas representações devem estar relacionadas a suas respectivas variáveis. Por exemplo, suponhamos que a uma chave do tipo liga/desliga seja atribuída a variável “K”. Com base nessa atribuição, podemos representar o estado da respectiva chave em um circuito como:

- K = 0 (zero) para a condição chave desligada (aberta);
- K = 1 (um) para a condição chave ligada (fechada).

Além disso, as funções booleanas são expressões que representam as relações entre as variáveis envolvidas em determinado processo por meio dos operadores lógicos “AND” (·) e “OR” (+).

Exemplo

Um sistema de alarme deverá soar quando os sensores A e C estiverem ativados ao mesmo tempo ou quando a chave B estiver ligada e pelo menos um dos sensores estiver ativado. Um modo de encontrar a solução para o problema é a tabela verdade. Para isso, constrói-se a tabela verdade com as variáveis de entrada envolvidas no problema proposto (no caso, A, B, C) e verificam-se, de acordo com a expressão, os níveis que a variável de saída (S) deverá possuir (tabela 2.2).

Tabela verdade

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

Toda função booleana de N variáveis pode ser escrita na forma canônica disjuntiva ou conjuntiva.

A forma canônica disjuntiva é obtida da tabela verdade de acordo com o seguinte procedimento:

a) Escreva um termo (operação lógica “E”) para cada linha em que a função é igual a “1”.

b) Junte os termos obtidos no item anterior com a operação “OU” (+).

Obs.: as variáveis serão barradas ou não conforme seu valor seja “0” ou “1” naquela linha.

Exemplo

Seja a tabela verdade a seguir

Tabela verdade

A	B	C	F	
0	0	0	1	→ $\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	0	
0	1	0	0	
0	1	1	1	→ $\overline{A}\,B\,C$
1	0	0	1	→ $A\,\overline{B}\,\overline{C}$
1	0	1	0	
1	1	0	0	
1	1	1	1	→ $A\,B\,C$

1ª linha: $\overline{A}\,\overline{B}\,\overline{C}$
4ª linha: $\overline{A}\,B\,C$
5ª linha: $A\,\overline{B}\,\overline{C}$
8ª linha: $A\,B\,C$

$$F = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,C + A\,\overline{B}\,\overline{C} + A\,B\,C$$

A forma canônica conjuntiva é obtida da tabela verdade de acordo com o seguinte procedimento:

a) Escreva um termo (operação lógica “OU”) para cada linha em que a função tem valor “0”.

b) Junte os termos obtidos no item anterior com a operação “E” (·).

Obs.: as variáveis serão barradas se naquela linha seu valor for “1” e não barrada se seu valor for “0”.

Exemplo

Na tabela verdade do exemplo anterior, verifica-se que a função é igual a “0” na segunda, terceira, sexta e sétima linhas.

Tabela verdade

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

2ª linha: $A + B + \overline{C}$
3ª linha: $A + \overline{B} + C$
6ª linha: $\overline{A} + B + \overline{C}$
7ª linha: $\overline{A} + \overline{B} + C$

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

2.2.1 Propriedades e teoremas da álgebra booleana

Os teoremas e propriedades da álgebra booleana permitem a simplificação de circuitos lógicos, objetivo final de todo projeto de circuitos digitais. As propriedades mais importantes são apresentadas a seguir.

Propriedade da intersecção

Está relacionada com as portas E. Os casos possíveis são:

$A \cdot 1 = A$
$A \cdot 0 = 0$

Obs.: essa propriedade é aplicável a um maior número de variáveis de entrada.

Exemplos

$A \cdot B \cdot 1 = A \cdot B$
$A \cdot B \cdot 0 = 0$

Propriedade da união

Está relacionada com as portas OU e divide-se em dois casos:

$B + (1) = 1$
$B + (0) = B$

Essa propriedade também é válida para portas OU com mais de duas entradas.

Exemplos

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

Propriedade da tautologia

Palavra de origem grega usada em lógica para descrever uma proposição que é verdadeira quaisquer que sejam os valores de suas variáveis.

É válida para portas E e portas OU e pode ser verificada nos seguintes casos:

$A \cdot A = A$
$A + A = A$

Essa propriedade é válida para um maior número de variáveis.

Exemplo

$A \cdot B + A \cdot B + C = A \cdot B + C$

Propriedade dos complementos

Se aplicarmos um sinal lógico e seu complemento a uma porta lógica, simultaneamente a saída será "0" ou "1", dependendo do tipo de porta.

Exemplos

$A \cdot \overline{A} = 0$
$A + \overline{A} = 1$

Propriedade da dupla negação

Essa propriedade afirma que o complemento do complemento de uma variável é igual a ela própria. Em forma de expressão matemática, temos, como exemplo:

$\overline{\overline{A}} = A$

Propriedade comutativa

Essa propriedade é semelhante à da álgebra convencional e pode ocorrer nos seguintes casos:

$A \cdot B = B \cdot A$
$A + B = B + A$

Propriedade associativa

É outra propriedade semelhante à da álgebra convencional. Os casos possíveis são:

$(A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C$
$A + (B + C) = (A + B) + C = A + B + C$

Propriedade distributiva

Também é semelhante à da álgebra convencional.

Exemplos

$A \cdot (B + C) = A \cdot B + A \cdot C$
$A + B \cdot C = (A + B) \cdot (A + C)$

Propriedade da absorção

Os casos mais elementares são:

$A + A \cdot B = A$
$A + A \cdot B = A + B$
$(A + B) \cdot B = A \cdot B$

Em decorrência dessas identidades, podemos encontrar outras um pouco mais complexas:

$A \cdot B + A \cdot B = A$
$(A + B) \cdot (A + B) = A$
$A \cdot (A + B) = A$
$A \cdot (A + B) = AB$
$A \cdot B + A \cdot C = (A + C) \cdot (A + B)$

Dualidade

Seja F uma função booleana. Define-se a **função dual** de F como aquela obtida quando mudamos os operadores **+** por · e · por **+** e os valores "0" por "1" e "1" por "0".

Postulados da dualidade:

1a) $X = 0$ se $x \neq 1$ | 1b) $X = 1$ se $X \neq 0$
2a) $X = 1$ se $x = 0$ | 2b) $X = 0$ se $X = 1$
3a) $0 \cdot 0 = 0$ | 3b) $1 + 1 = 1$
4a) $1 \cdot 1 = 1$ | 4b) $0 + 0 = 0$
5a) $1 \cdot 0 = 0 \cdot 1 = 0$ | 5b) $0 + 1 = 1 + 0 = 1$

1º teorema de De Morgan

"O complemento do produto é igual à soma dos complementos"

$\overline{A} \cdot \overline{B} = \overline{A} + \overline{B}$

Podemos comprovar esse teorema pela tabela verdade a seguir:

Tabela verdade para uma porta NAND

A	B	$\overline{A}$	$\overline{B}$	A•B	$\overline{A \bullet B}$	$\overline{A}+\overline{B}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

2º teorema de De Morgan

"O complemento da soma é igual ao produto dos complementos"

$\overline{A + B} = \overline{A} \cdot \overline{B}$

Esse teorema também pode ser comprovado pela tabela verdade.

Como consequência dos teoremas de De Morgan as funções lógicas já conhecidas podem ser reescritas por um bloco equivalente, permitindo, assim, redesenhar os circuitos lógicos caso seja conveniente.

As equivalências básicas são:

a)Portas NAND (figura 2.9).

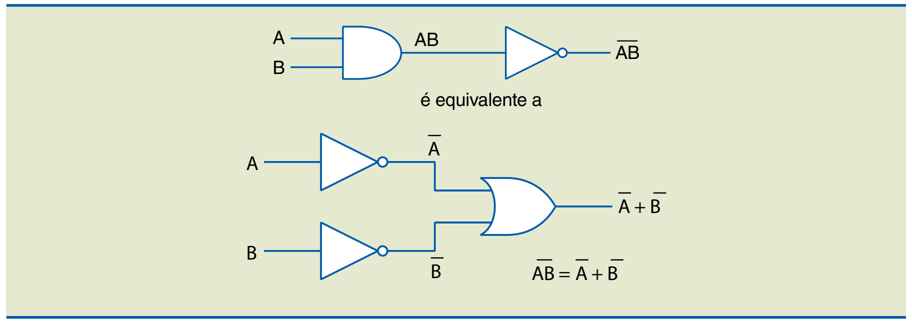


Figura 2.9
Equivalência entre as portas NAND.

Ou seja (figura 2.10):

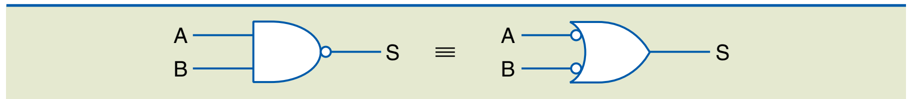


Figura 2.10
Representações simplificadas das portas NAND.

b)Portas NOR (figura 2.11).

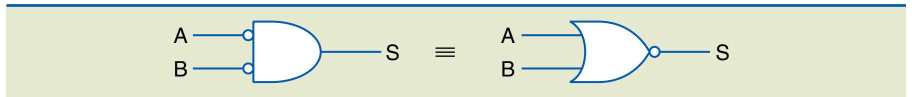


Figura 2.11
Representações simplificadas das portas NOR.

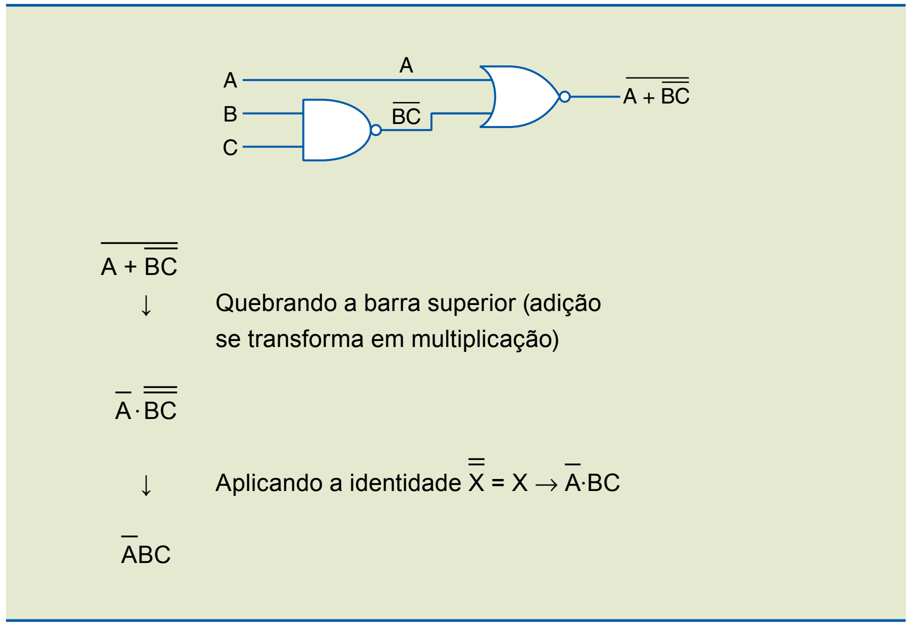
Exemplo

Consideremos a seguinte expressão lógica:

$\overline{(A + (\overline{B \cdot C}))}$

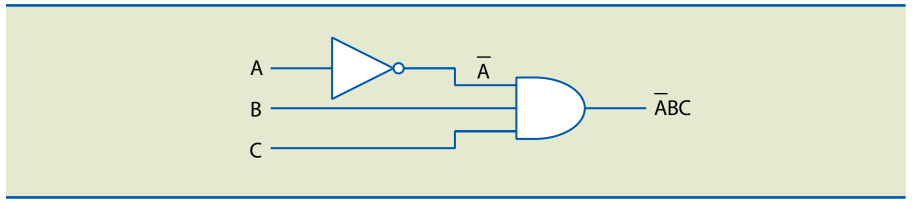
O circuito lógico correspondente implementado com portas lógicas E, OU e INVERSORAS terá o aspecto ilustrado na figura 2.12.

Figura 2.12
Representação do circuito lógico com portas lógicas E, OU e INVERSORAS.



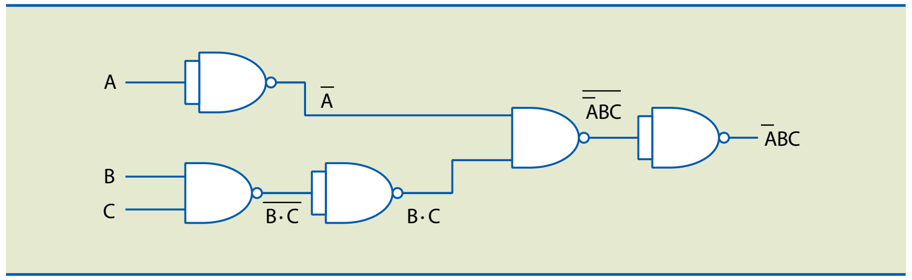
Pela aplicação das identidades do circuito da figura 2.12, o circuito lógico reduz-se conforme apresenta a figura 2.13.

Figura 2.13
Representação simplificada do circuito lógico com portas lógicas E, OU e INVERSORAS.



Reaplicando os teoremas de De Morgan para substituir os blocos lógicos da figura 2.13 pelos equivalentes, obtemos a figura 2.14.

Figura 2.14
Representação simplificada do circuito lógico com portas lógicas E, OU e INVERSORAS com substituição dos blocos lógicos da figura 2.13 por seus equivalentes.



2.3 Descrição de funções lógicas

Os circuitos lógicos podem ser representados por funções booleanas, ou seja, admite-se que todos os circuitos lógicos estabelecem as relações entre entradas e saída obedecendo à função booleana que os representa. Quando necessário, é possível obter a função booleana por meio da tabela verdade do circuito. Além disso, o circuito lógico pode ser descrito pela conexão de portas lógicas básicas, independentemente de sua complexidade. A seguir, são descritas as relações entre as formas de representação de um circuito lógico.

2.3.1 Circuito lógico

Consideremos o circuito lógico da figura 2.15. Vamos obter a função lógica $S = f(A, B, C, D)$, da saída do circuito.

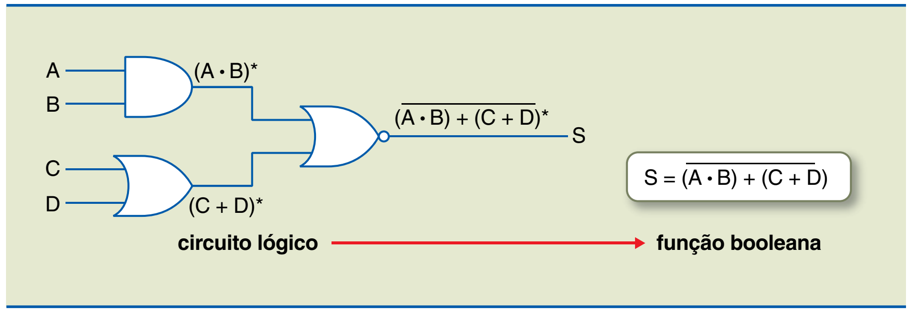


Figura 2.15
Representação da função y = A, B, C, D.

Analisando esse circuito, podemos notar que colocamos na saída de cada porta lógica a expressão booleana correspondente (*), que será a entrada de outra porta lógica, e assim repetimos o procedimento sucessivamente até a expressão booleana da saída.

Vamos analisar outra situação, considerando a função booleana $y = A \cdot B + C \cdot (B + D)$. Como se trata de uma expressão algébrica (álgebra booleana), devemos respeitar na implementação do circuito a ordem das operações, associando a multiplicação à operação “E” e a soma à operação “OU”. As operações entre parênteses devem ser feitas agrupadas (figura 2.16).

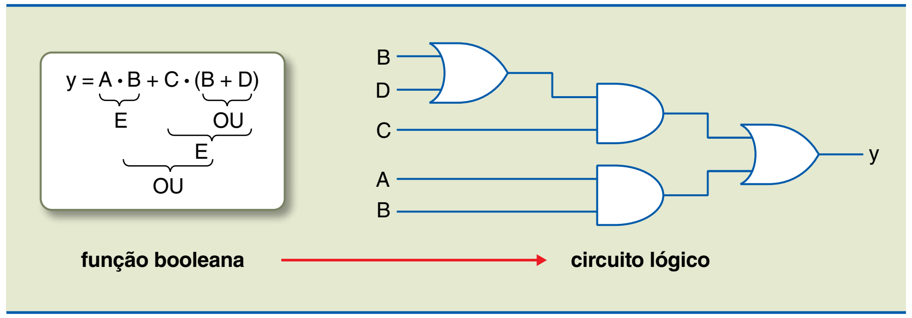


Figura 2.16
Representação da função $y = A \cdot B + C \cdot (B+D)$.

2.3.2 Tabela verdade 2

Vamos obter a tabela verdade da função booleana $y = A \cdot B \cdot C + AC + BC$. Para isso, adotamos o seguinte procedimento:

1) Montamos a coluna completa de todas as combinações possíveis das variáveis (número de linhas = $2^n + 1$, n = número de variáveis).
2) Montamos as colunas auxiliares em quantidade igual ao número de "parcelas" da função booleana.
3) Montamos a última coluna para y.

Os cálculos referentes às colunas das "parcelas" da função booleana, em geral, são feitos mentalmente ou em rascunho.

Tabela verdade de $y = A \cdot B \cdot C + AC + BC$

A	B	C	A•B•C	AC	BC	y
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	1	0	1
1	0	1	0	0	0	0
1	1	0	0	1	0	1
1	1	1	0	0	1	1

É possível obter a expressão booleana por meio da tabela verdade. Para isso, vamos considerar a tabela verdade a seguir:.

Tabela verdade

	A	B	C	y
1	0	0	0	0
2	0	0	1	0
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	0
8	1	1	1	1

Para montarmos a função booleana a partir dos valores da tabela verdade, adotamos o seguinte procedimento:

1) Consideramos somente as linhas da tabela em que $y = 1$.
2) Fazemos "E" das variáveis que têm valor "1" com os complementos das que têm valor "0", por exemplo:

linha 3 $\rightarrow$ A=0, B=1 e C=0 $\rightarrow \overline{A} \cdot B \cdot \overline{C}$
linha 5 $\rightarrow$ A=1, B=0 e C=0 $\rightarrow A \cdot \overline{B} \cdot \overline{C}$
linha 6 $\rightarrow$ A=1, B=0 e C=1 $\rightarrow A \cdot \overline{B} \cdot C$
linha 8 $\rightarrow$ A=1, B=1 e C=1 $\rightarrow A \cdot B \cdot C$

3) Fazemos "OU" dos valores obtidos

$$y = A \cdot B \cdot C + A \cdot B \cdot C + A \cdot B \cdot C + A \cdot B \cdot C$$

Obs.: a numeração das linhas registradas à esquerda não é necessária; serve somente como referência para a explicação.

2.3.3 Simplificação de funções lógicas

O mapa (ou diagrama) de Karnaugh é uma forma ordenada utilizada para minimizar uma expressão lógica, que geralmente produz um circuito com configuração mínima. É construído com base na tabela verdade e pode ser facilmente aplicado em funções envolvendo duas a seis variáveis. No caso de sete ou mais variáveis, o método torna-se complicado e devemos usar técnicas mais elaboradas.

Representa-se o mapa de Karnaugh por uma tabela em forma de linhas e colunas. Essa tabela, de acordo com o número de variáveis, é dividida em células obedecendo à proporção 2^n, em que n é o número de variáveis de entrada envolvidas.

Mapa para uma variável de entrada (figura 2.17)

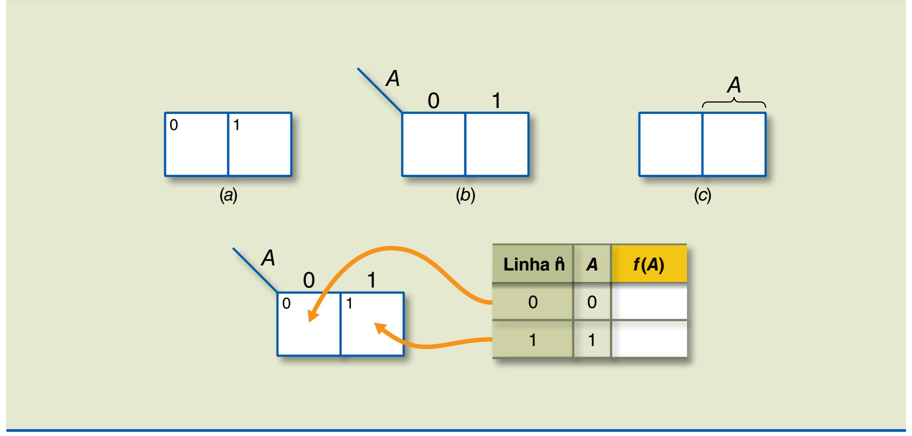


Figura 2.17
Mapa para uma variável de entrada.

Mapa para duas variáveis de entrada (figura 2.18)

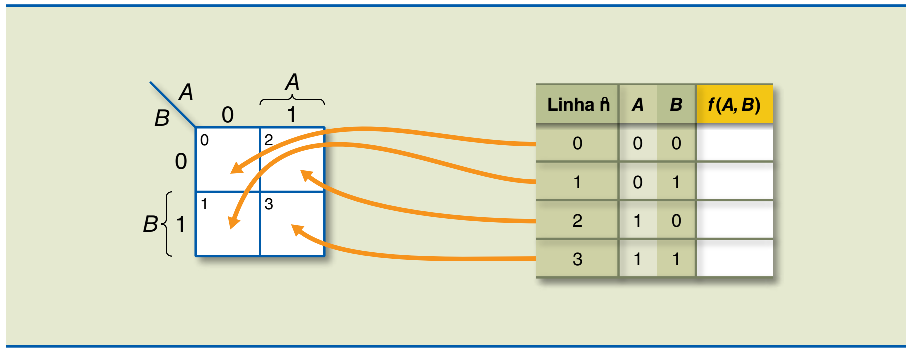


Figura 2.18
Mapa para duas variáveis de entrada.

A figura 2.19 apresenta a tabela verdade e o mapa de Karnaugh correspondente para duas variáveis.

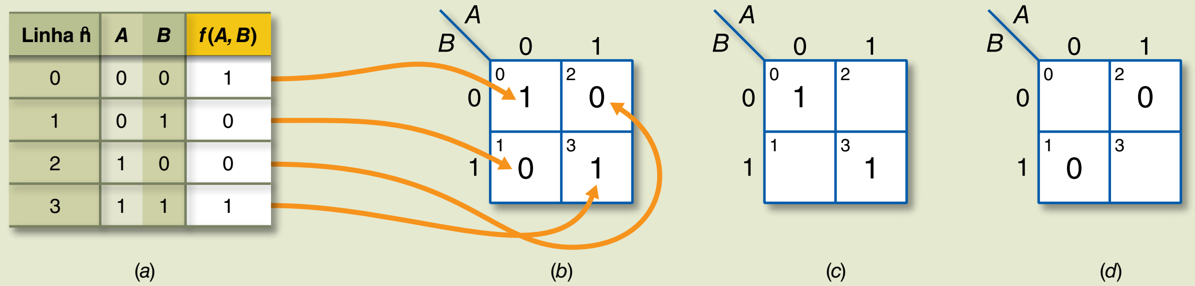


Figura 2.19
Representação do mapa para duas variáveis de entrada.

Mapa para três variáveis de entrada (figura 2.20)

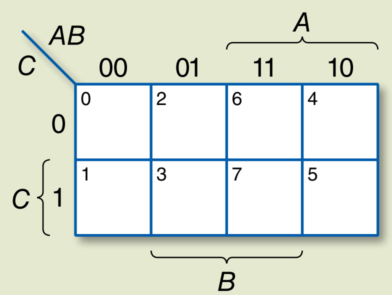


Figura 2.20
Mapa para três variáveis de entrada.

A figura 2.21 apresenta um exemplo de como deve ser representado o mapa para três variáveis, a partir da tabela verdade correspondente.

Figura 2.21
Mapa para três variáveis de entrada.

A	B	C	X	
0	0	0	1	→ $\bar{A}\bar{B}\bar{C}$
0	0	1	1	→ $\bar{A}\bar{B}C$
0	1	0	1	→ $\bar{A}B\bar{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	→ $AB\bar{C}$
1	1	1	0	

$$X = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C}$$

(b)

	$\bar{C}$	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$	1	0
AB	1	0
$A\bar{B}$	0	0

Mapa para quatro variáveis de entrada (figura 2.22)

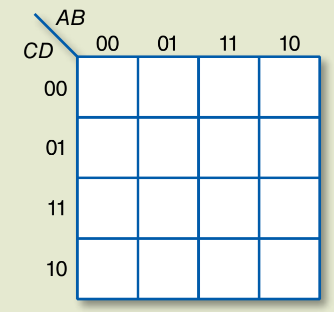


Figura 2.22
Mapa para quatro variáveis de entrada.

A figura 2.23 apresenta um exemplo de como deve ser representado o mapa para quatro variáveis, a partir da tabela verdade correspondente.

Figura 2.23
Mapa para quatro variáveis de entrada.

A	B	C	D	X	
0	0	0	0	0	→ $\bar{A}\bar{B}\bar{C}D$
0	0	0	1	1	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	→ $\bar{A}B\bar{C}D$
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	0	
1	1	0	1	1	→ $AB\bar{C}D$
1	1	1	0	0	
1	1	1	1	1	→ $ABCD$

$$X = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + ABCD$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

Mapa para cinco variáveis de entrada (figura 2.24)

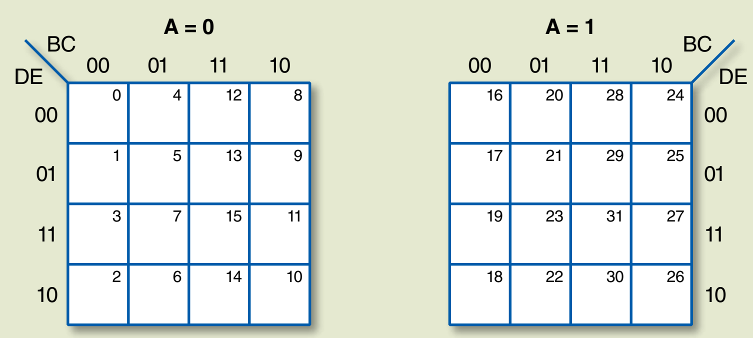


Figura 2.24
Mapa para cinco variáveis de entrada.

Na figura 2.25, podemos observar a representação do mapa para seis variáveis de entrada.

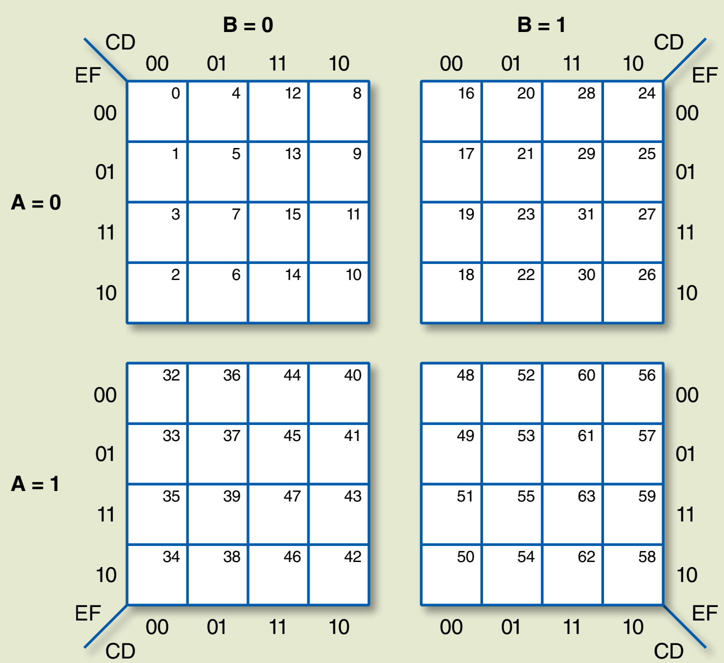


Figura 2.25
Mapa para seis variáveis de entrada.

A seguir, vamos analisar o processo de minimização utilizando os diagramas de Karnaugh e, depois, ver alguns exemplos.

Minimização de funções utilizando o mapa de Karnaugh

Para realizarmos a minimização de funções lógicas utilizando o método do mapa de Karnaugh, devemos obedecer às seguintes regras:

1) Identificar as células nas quais os níveis de saída são iguais a “1”.
2) Formar enlaces ou agrupamentos de células logicamente adjacentes cujos níveis de saída são iguais a “1”.

Obs.: duas células são adjacentes se apenas uma das variáveis de entrada correspondentes troca de valor; portanto, as células localizadas nos vértices do mapa também são adjacentes entre si.

3) Os agrupamentos formados devem conter o maior número possível de células logicamente adjacentes, mas esse número tem sempre de ser uma potência de 2, ou seja, agrupamentos que tenham 1, 2, 4, 8, 16, 32, ... elementos.

Agrupamentos possíveis	
HEXA	16 quadros
OITAVA	8 quadros
QUADRA	4 quadros
PAR	2 quadros
TERMO ISOLADO	1 quadro

Nota: sempre que um grupo é formado, a variável que muda de estado é a eliminada. Por exemplo: se o grupo engloba parte da região A e parte da região $\overline{A}$, a variável A é eliminada.

4) Cada agrupamento assim formado corresponde a uma função lógica “E” envolvendo as variáveis de entrada entre uma célula e outra que mantêm o nível lógico.
5) A expressão lógica final corresponde a uma função “OU” envolvendo os agrupamentos anteriormente mencionados.

Exemplos de minimização

Exemplos para três variáveis de entrada

1. $Z = f(A, B, C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B + AB\overline{C} + AC$ (figura 2.26)

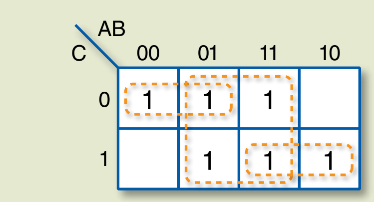
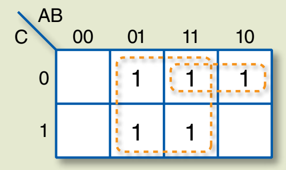


Figura 2.26
Simplificação das três variáveis de entrada para o exemplo 1.

2. $Z = f(A, B, C) = \overline{A}B + B\overline{C} + BC + A\overline{B}\,\overline{C}$ (figura 2.27)

A expressão lógica minimizada é $B + A\overline{C}$.

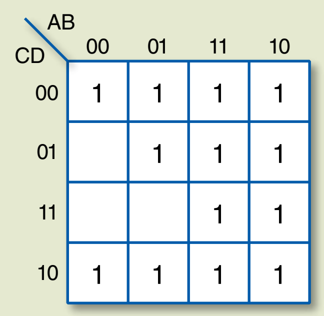
Figura 2.27
Simplificação das três variáveis de entrada para o exemplo 2.



Exemplos para quatro variáveis de entrada

1. Dado o diagrama de Karnaugh da figura 2.28, obtenha a expressão lógica minimizada.

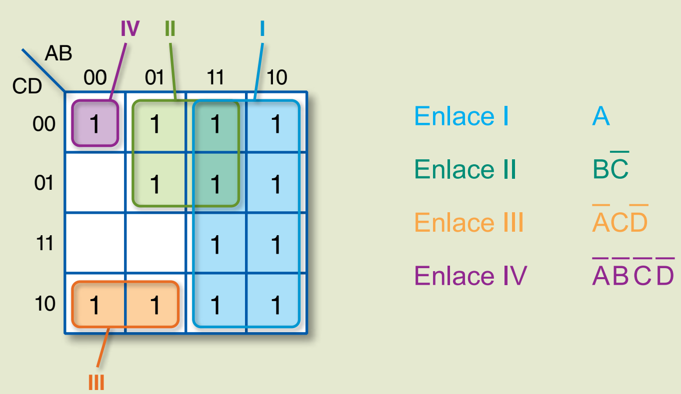
Figura 2.28
Simplificação das quatro variáveis de entrada para o exemplo 1.



Solução:

Para ilustrar o processo, primeiramente não de forma ideal, suponhamos que tivéssemos selecionado os agrupamentos apresentados na figura 2.29.

Figura 2.29
Representação dos quatro enlaces.

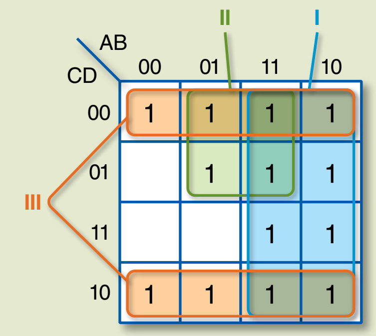


De acordo com os enlaces anteriores, a expressão obtida seria:

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

Mas será essa a expressão mínima? Se selecionarmos adequadamente os enlaces de acordo com as regras expostas anteriormente, obteremos a figura 2.30.

Figura 2.30
Nova representação com os três enlaces.



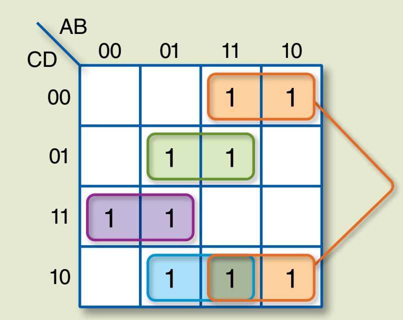
Considerando esses novos enlaces, obteremos a seguinte expressão mínima:

$f = \overline{D} + B\overline{C} + A$

2. Minimize a expressão lógica dada a seguir (figura 2.31).

$f = \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}C\overline{D} + AB\overline{C}\,\overline{D} +$
$+ AB\overline{C}D + ABC\overline{D}$

Figura 2.31
Representação com os quatro enlaces.



Solução:

Expressão lógica minimizada:

$F(A,B,C,D) = B\overline{C}D + A\overline{D} + \overline{A}CD + BC\overline{D}$

Exemplo para cinco variáveis de entrada

Considere as figuras 2.32 e 2.33.

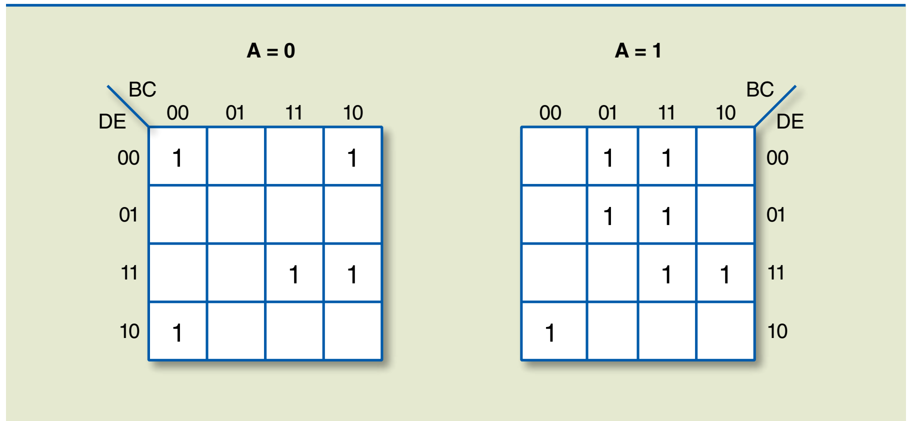


Figura 2.32
Mapa para cinco variáveis de entrada.

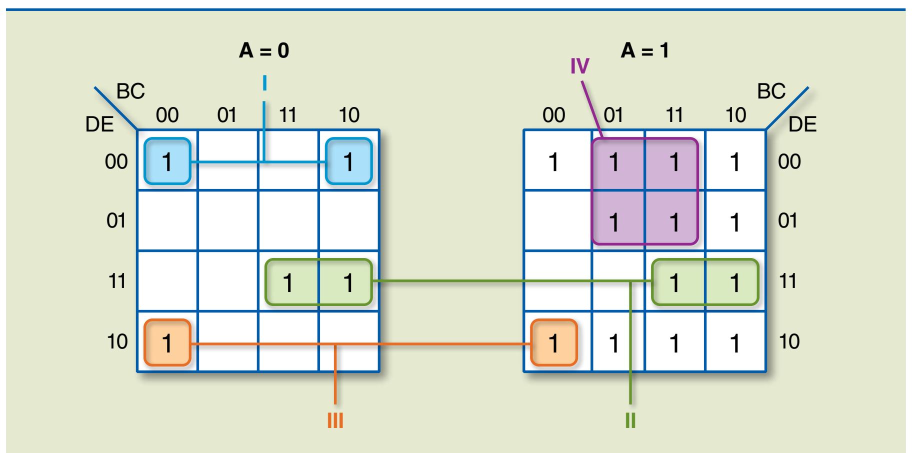


Figura 2.33
Representação dos quatro enlaces.

O resultado obtido é:

$f = \overline{A}\,\overline{C}\,\overline{D}\,\overline{E} + BDE + \overline{B}\,\overline{C}D\overline{E} + AC\overline{D}$

Exemplo para seis variáveis de entrada

Considere as figuras 2.34 e 2.35.

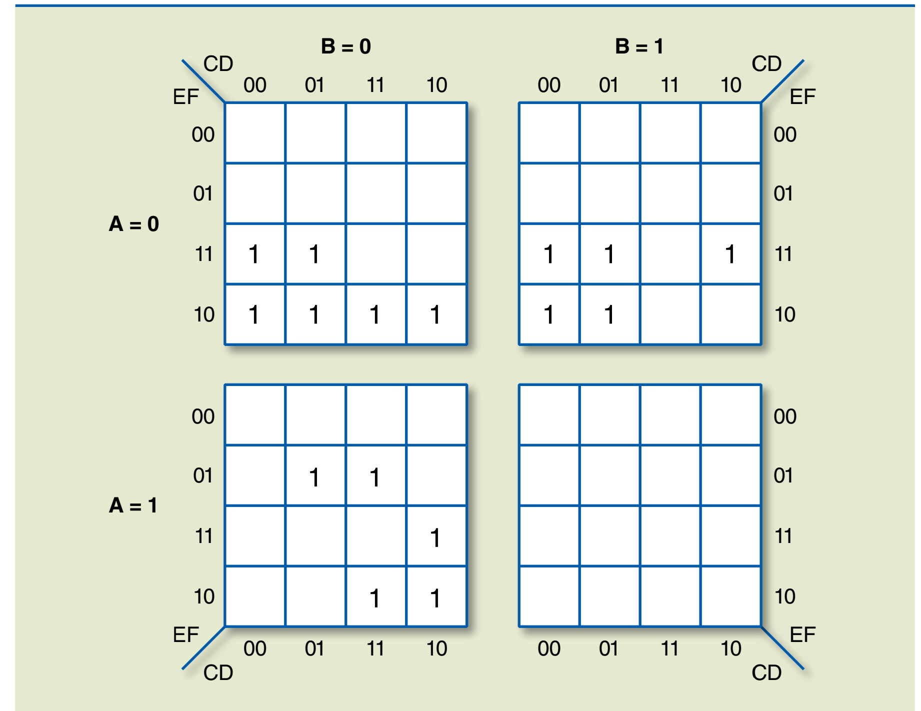


Figura 2.34
Mapa para seis variáveis de entrada.

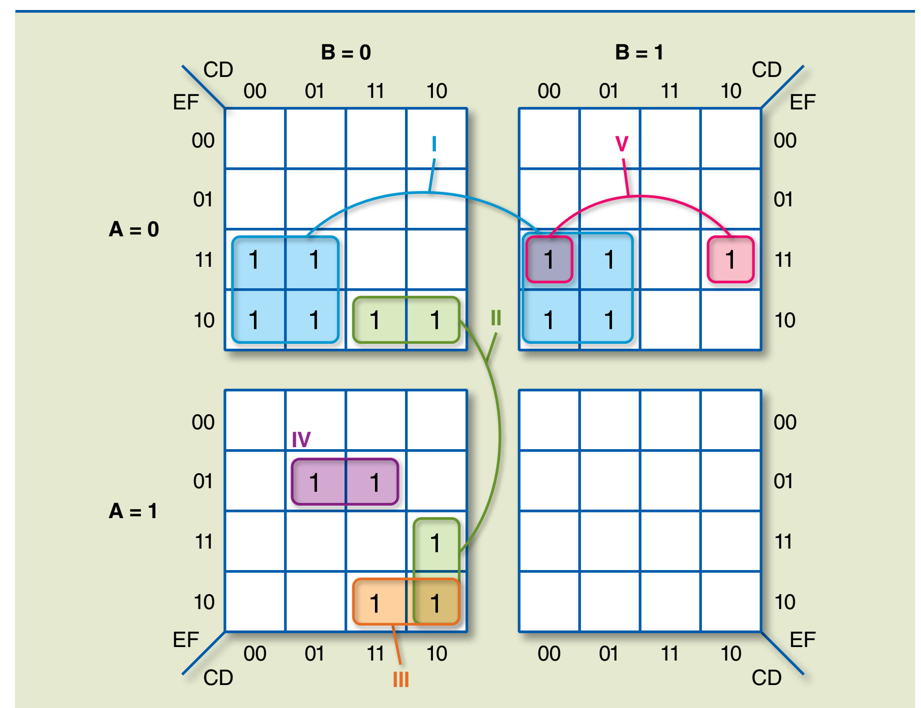


Figura 2.35
Representação dos cinco enlaces.

$F = \overline{A}\,\overline{C}E + \overline{B}CE\overline{F} + A\overline{B}C\overline{D}E + A\overline{B}D\overline{E}F + \overline{A}B\overline{D}EF$