AULA 5 – CIRCUITOS LÓGICOS COMBINACIONAIS

Nas aulas anteriores estudamos a operação de todas as portas lógicas básicas e usamos a álgebra Booleana para descrever e analisar circuitos que foram feitos a partir da combinação de portas lógicas. Esses circuitos podem ser classificados como circuitos lógicos combinacionais porque, em qualquer instante de tempo, o nível lógico da saída do circuito depende da combinação dos níveis lógicos presentes nas entradas.
Um circuito combinacional não possui a característica de memória. Portanto sua saída depende apenas dos valores atuais das entradas.

FORMA DE SOMA-DE-PRODUTOS:

Os métodos de simplificação e projetos de circuitos lógicos que estudaremos requerem que a expressão esteja na forma de soma-de-produtos. Alguns exemplos de expressões desse tipo são:

$ABC + \overline{A}B\overline{C}$ $\qquad AB + \overline{A}B\overline{C} + \overline{C}\,\overline{D} + D$ $\qquad \overline{A}B + C\overline{D} + EF + GK + H\overline{L}$

Cada uma destas expressões consiste em dois ou mais termos AND conectados por uma expressão OR. Vale observar que na forma de soma de produtos a barra não deve cobrir mais do que uma variável do mesmo termo AND.

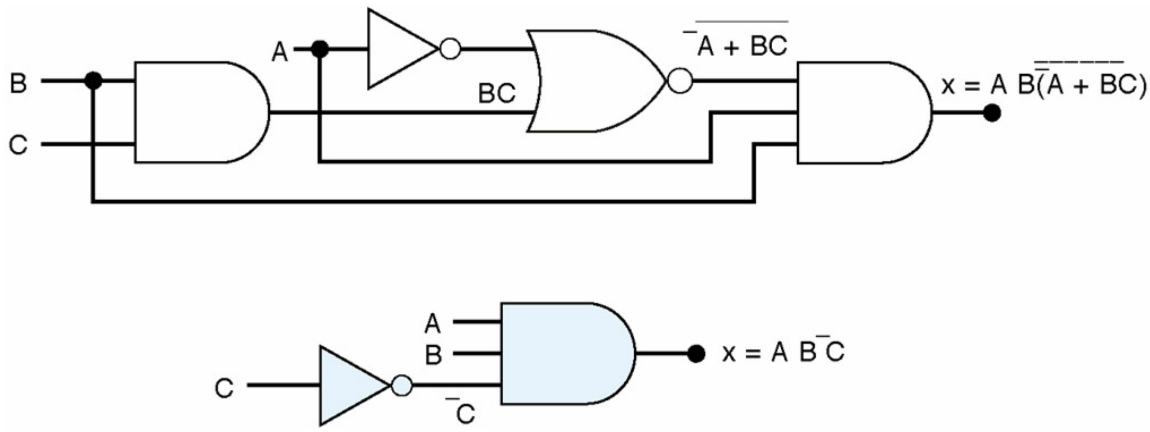
FORMA DE PRODUTO-DE-SOMAS:

Esta outra forma de expressões lógicas também é utilizada, consiste em dois ou mais termos OR conectados por operações AND, alguns exemplos de expressões deste tipo seguem abaixo:

$(A + \overline{B} + C)(A + C)$ $\qquad (A + \overline{B})(\overline{C} + D)F$

SIMPLIFICAÇÃO DE CIRCUITOS LÓGICOS:

Uma vez obtida a expressão de um circuito lógico podemos simplificá-la a uma forma que utilize menos portas, seja, portanto de menor custo e mais fácil de implementar.
Existem duas formas de simplificação, através da álgebra Booleana ou através de um método gráfico chamado Mapa de Karnaugh.
Abaixo podemos ver um exemplo de simplificação:



Ambos os circuitos realizam a mesma operação lógica, só que o circuito abaixo utiliza um número menor de portas.

SIMPLIFICAÇÃO ALGÉBRICA:

Podemos utilizar os teoremas da Álgebra de Boole para a simplificação de expressões, nem sempre é tão simples saber qual teorema utilizar ou mesmo se a expressão atingiu a sua forma mais simples, acaba sendo um processo de tentativa e erro em algumas situações, naturalmente com a experiência vai ficando menos difícil. Vejamos o exemplo abaixo:

$$z = ABC + A\overline{B}.(\overline{\overline{A}.\overline{C}})$$
$$z = ABC + A\overline{B}(\overline{\overline{A}} + \overline{\overline{C}})$$
$$z = ABC + A\overline{B}(A + C)$$
$$z = ABC + A\overline{B}A + A\overline{B}C$$
$$z = ABC + A\overline{B} + A\overline{B}C$$
$$z = AC(B + \overline{B}) + A\overline{B}$$
$$z = AC + A\overline{B}$$
$$z = A(C + \overline{B})$$

Comentários:
Na primeira ação aplicamos o teorema de Demorgan, depois eliminamos as inversões duplas, aplicamos a distributiva na terceira linha, na quarta temos A.A=A, depois colocamos o termo AC em evidência, depois temos $(B + \overline{B}) = 1$, por fim colocamos o termo A em evidência, o que nos resulta na expressão mais simplificada.

Outro exemplo:

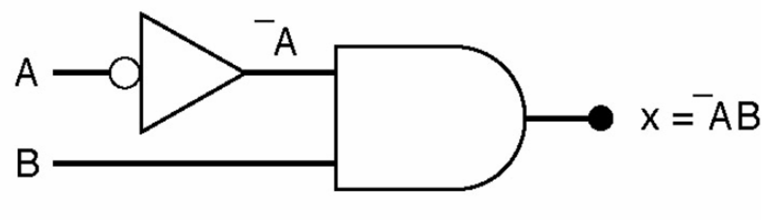
$$z = A\overline{B}\overline{C} + A\overline{B}C + ABC$$
$$z = A\overline{B}(\overline{C} + C) + ABC$$
$$z = A(\overline{B} + BC)$$
$$z = A(\overline{B} + C)$$

Comentários:
Colocamos o termos $A\overline{B}$ em evidência para os dois primeiros termos, $(\overline{C} + C) = 1$, depois colocamos A em evidência pelo teorema já visto podemos simplificar a expressão.

PROJETANDO CIRCUITOS LÓGICOS COMBINACIONAIS:

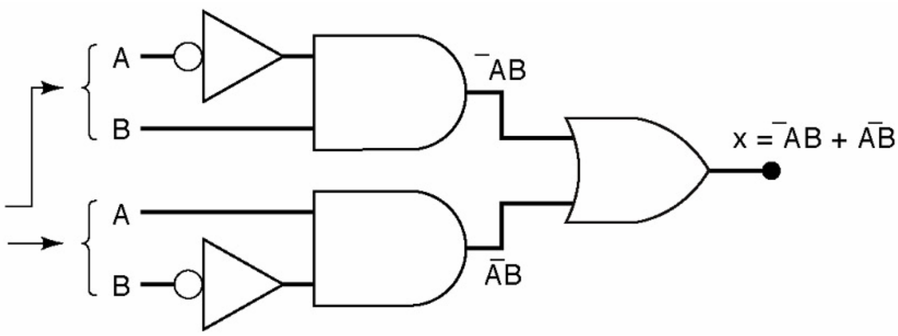
Podemos particularmente projetar um circuito lógico para que a saída tenha determinados níveis em função da combinação dos níveis das entradas. Podemos utilizar a tabela verdade para o projeto.
Veja o exemplo abaixo, só na condição A=0 e B=1 desejamos que a saída seja “1”:

A	B	x
0	0	0
0	1	1
1	0	0
1	1	0



Neste outro exemplo desejamos que a saída tenha nível alto somente quando o nível da entrada A for diferente do nível da entrada B:

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0



PROCEDIMENTO DO PROJETO:

Uma vez identificada a tabela verdade, podemos, usando as três operações básicas (AND, OR e NOT) descrever a expressão algébrica na forma de soma de produtos, veja abaixo passo-a-passo:
Passo 1: Construa a tabela verdade com base na sua necessidade.

A	B	C	X	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}BC$
1	0	0	0	
1	0	1	1	$A\overline{B}C$
1	1	0	1	$AB\overline{C}$
1	1	1	1	ABC

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

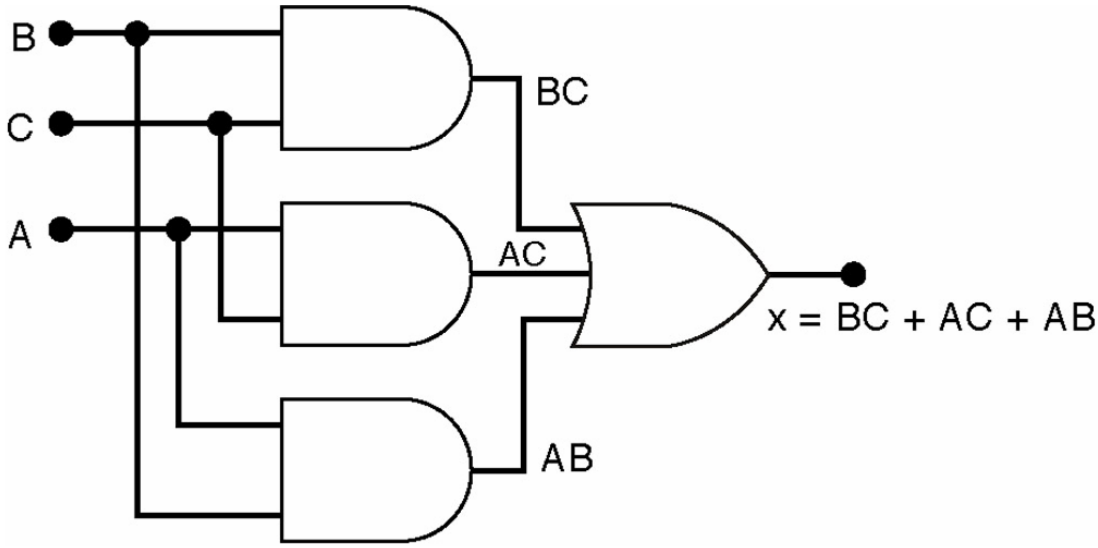
Passo 2: Escreva o termos AND para cada caso em que a saída for igual a “1”, de forma que as variáveis de entrada resultem nível “1” na saída.

Passo 3: Escreva a expressão na forma de soma de produtos.

Passo 4: Simplifique a expressão.

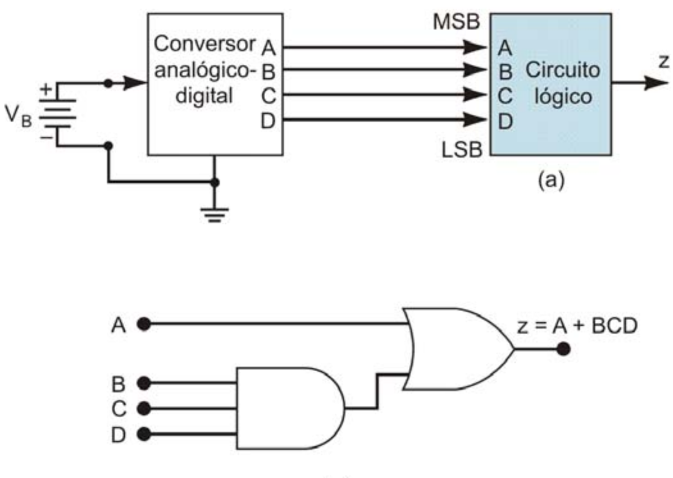
$$x = \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$
$$x = BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$$
$$x = BC + AC + AB$$

Passo 5: Implemente o circuito:



Veja a figura abaixo onde um um conversor analógico-digital está monitorando a tensão dc de uma bateria de 12 V de um veículo. A saída do conversor é um número binário de quatro bits, ABCD que corresponde a tensão da bateria em degraus de 1 V, sendo a variável A a mais significativa. As saídas binárias do conversor são as entradas de um circuito que gera uma saída em nível ALTO sempre que o valor binário for maior que 0110 = 6, ou seja, quando a tensão da bateria for maior do que 6 V. Projete este circuito lógico.
A tabela verdade é mostrada ao lado do circuito, bem como o circuito simplificado, vejamos a simplificação:

$$z = \overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\,\overline{D} + AB\overline{C}D + ABC\overline{D} + ABCD$$
$$z = \overline{A}BCD + A\overline{B}\,\overline{C}(\overline{D} + D) + A\overline{B}C(\overline{D} + D) + AB\overline{C}(\overline{D} + D) + ABC(\overline{D} + D)$$
$$z = \overline{A}BCD + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC$$
$$z = \overline{A}BCD + A\overline{B}(\overline{C} + C) + AB(\overline{C} + C)$$
$$z = \overline{A}BCD + A\overline{B} + AB$$
$$z = \overline{A}BCD + A(\overline{B} + B)$$
$$z = \overline{A}BCD + A$$
$$z = BCD + A$$



(a)

(c)

	A	B	C	D	z	
(0)	0	0	0	0	0	
(1)	0	0	0	1	0	
(2)	0	0	1	0	0	
(3)	0	0	1	1	0	
(4)	0	1	0	0	0	
(5)	0	1	0	1	0	
(6)	0	1	1	0	0	
(7)	0	1	1	1	1®	$\overline{A}BCD$
(8)	1	0	0	0	1®	$A\overline{B}\,\overline{C}\,\overline{D}$
(9)	1	0	0	1	1®	$A\overline{B}\,\overline{C}D$
(10)	1	0	1	0	1®	$A\overline{B}C\overline{D}$
(11)	1	0	1	1	1®	$A\overline{B}CD$
(12)	1	1	0	0	1®	$AB\overline{C}\,\overline{D}$
(13)	1	1	0	1	1®	$AB\overline{C}D$
(14)	1	1	1	0	1®	$ABC\overline{D}$
(15)	1	1	1	1	1®	$ABCD$

(b)

Devido ao fato de termos uma expressão com número elevado de termos, a simplificação através de álgebra booleana se torna longa. O método gráfico (mapas de Karnaugh) facilita situações como esta.

MÉTODO DO MAPA DE KARNAUGH:

O mapa de Karnaugh é um método gráfico usado para simplificar uma equação lógica ou converter um tabela verdade no seu circuito lógico correspondente. Iremos chamá-lo de mapa K, embora o mapa K possa ser utilizado para simplificar uma equação lógica com qualquer número de variáveis, na prática é viável até 5 ou 6 variáveis, acima disso é interessante utilizar softwares específicos para este fim.

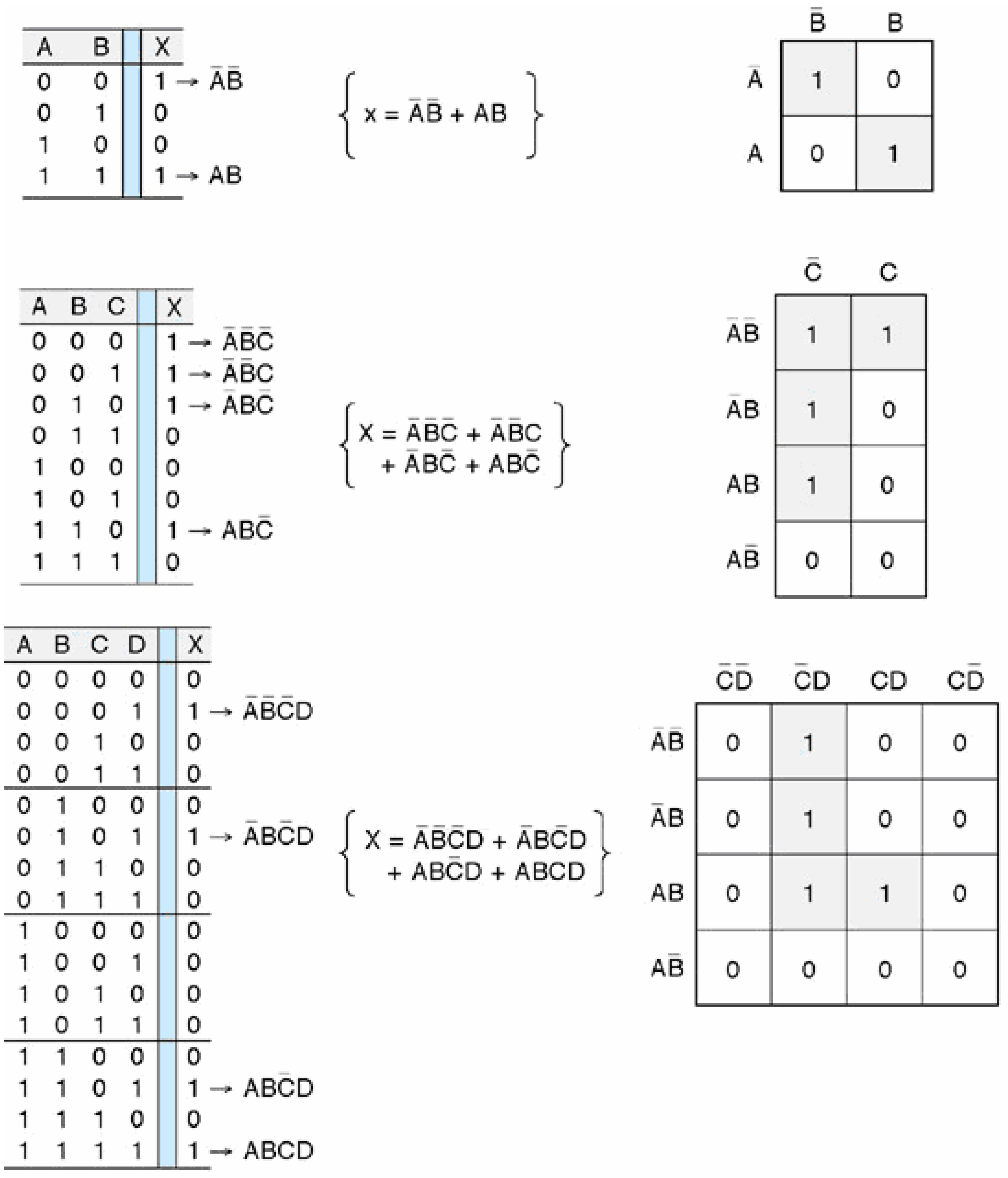
FORMATO DO MAPA DE KARNAUGH:

A tabela verdade fornece o valor da saída para cada combinação de entrada. O mapa K fornece a mesma informação em um formato diferente, cada linha na tabela verdade corresponde a um quadrado no mapa K, veja a figura que segue:

Outra característica dos mapas K é que no quadrado adjacente somente uma variável pode ser diferente, veja na figura que segue:

Para que uma única variável mude por vez a ordem por exemplo de uma tabela verdade com quatro variáveis na horizontal fica: $\overline{C}\,\overline{D}, \overline{C}D, CD, C\overline{D}$ e na vertical fica: $\overline{A}\,\overline{B}, \overline{A}B, AB, A\overline{B}$

Depois que o mapa está preenchido com os valores, a expressão de soma-de-produtos pode ser obtida fazendo-se a operação OR do termos em “1”



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

$$x = \bar{A}\bar{B} + AB$$

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

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	

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

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

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

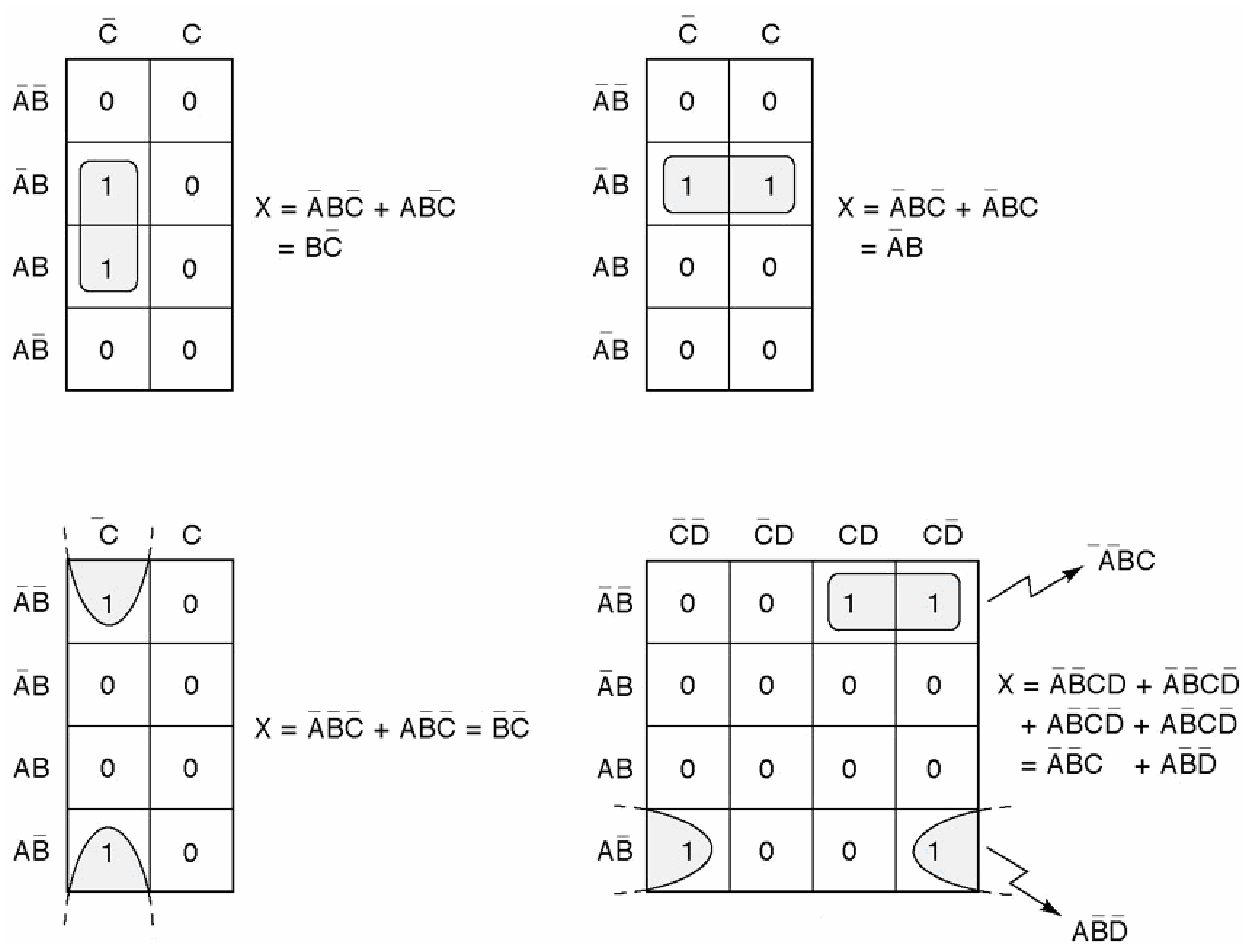
$$X = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + ABCD$$

	$\bar{C}\bar{D}$	$\bar{C}D$	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

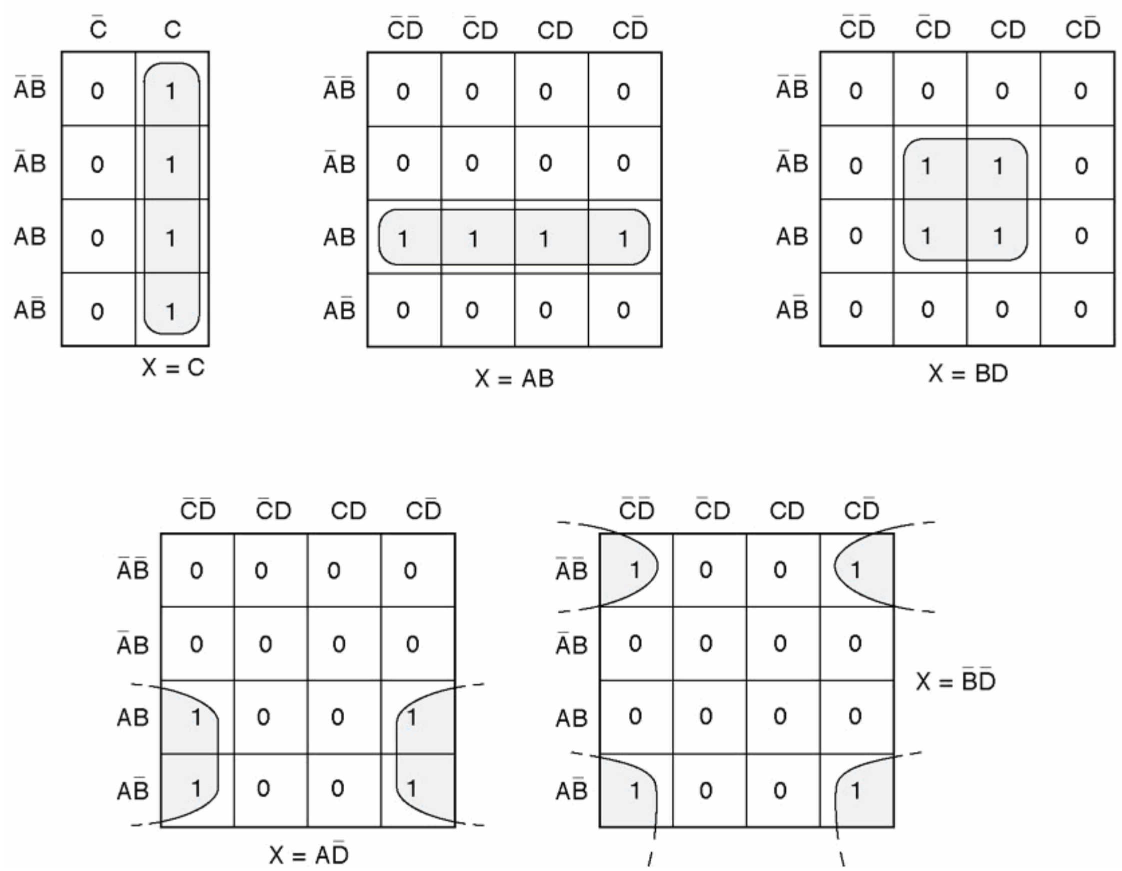
AGRUPAMENTO DOS QUADROS:

A expressão para a saída pode ser simplificada combinando adequadamente os quadros do mapa K que contém “1”, esse processo chamamos de agrupamento.

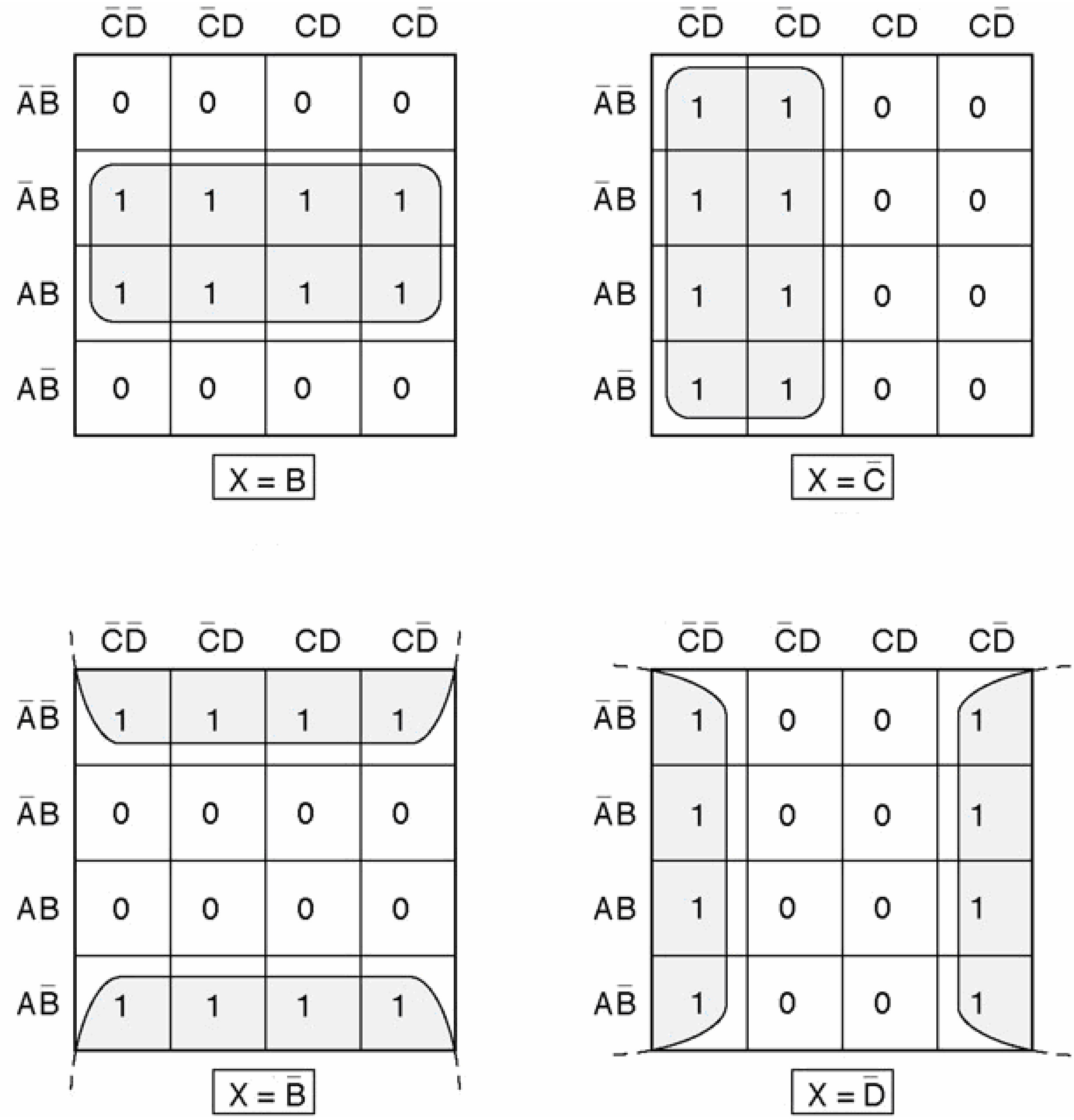
A figura abaixo mostra o agrupamento de dois quadros, sempre que quadros adjacentes forem agrupados, seja horizontalmente ou verticalmente, devemos eliminar a variável que alterou o seu estado, para um agrupamento de dois quadros iremos eliminar uma variável, o agrupamento de quatro quadros elimina duas variáveis e o agrupamento de oito quadros elimina 3 variáveis, veja abaixo:



Podemos ter agrupamentos de quatro quadros, novamente eliminamos as variáveis que mudaram de estado (condição normal e invertida):

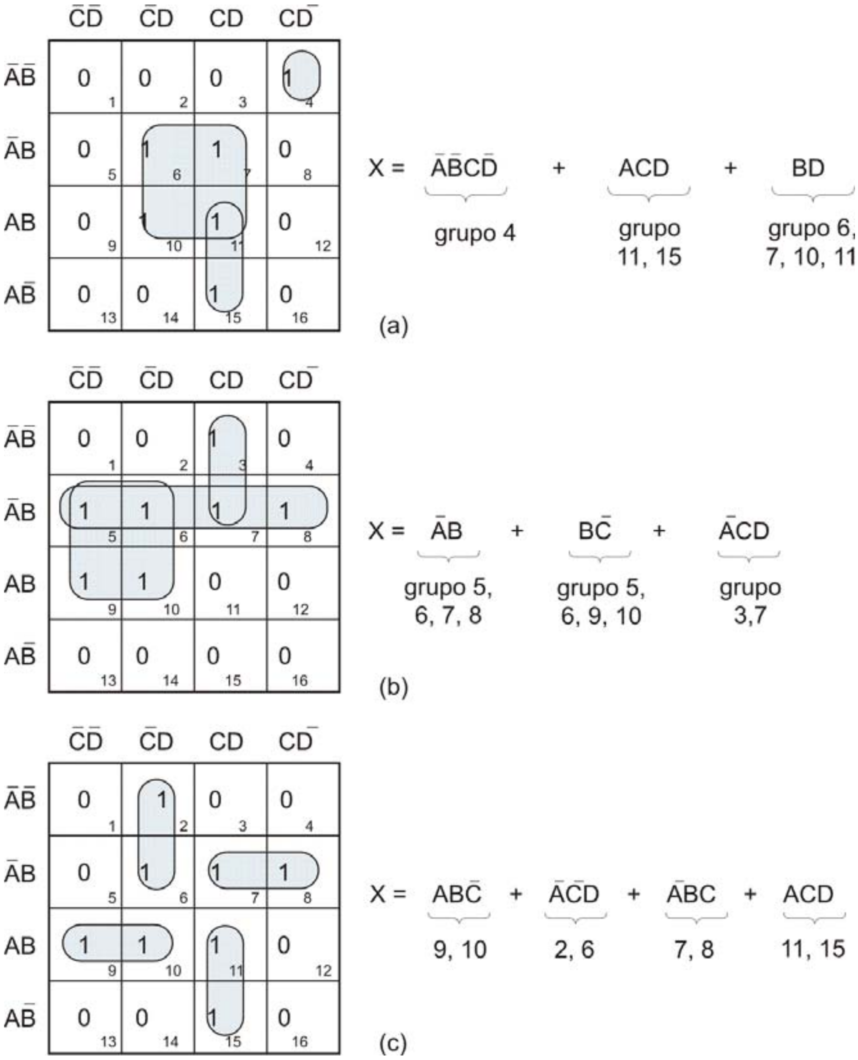


Abaixo seguem exemplos de agrupamentos de oito quadros, quando um octeto é agrupado em um mapa de quatro variáveis, três das quatro variáveis são eliminadas, uma vez que apenas uma variável permanece inalterada:



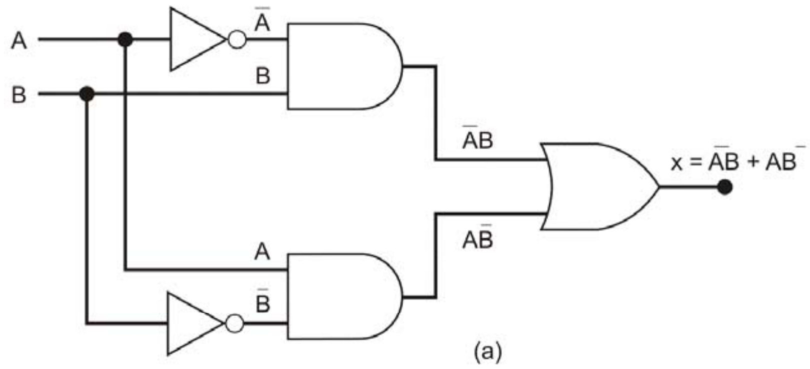
Após verificarmos os exemplos deve ficar claro que quanto maior for o número de quadros com “1” agrupados maior será a simplificação do expressão/circuito lógico.

Exemplo:
Admita que o mapa K é obtido através de uma dada tabela verdade, encontre a expressão mais simplificada:



CIRCUITOS OR-EXCLUSIVO E NOR-EXCLUSIVO:

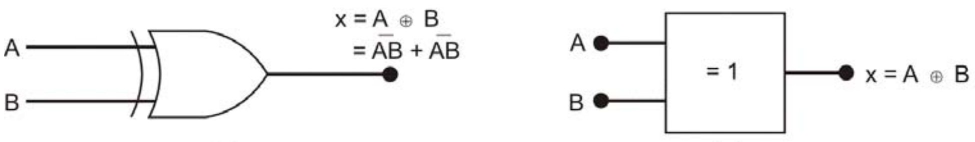
Estes circuitos aparecem em muitos sistemas digitais, o circuito OR-exclusivo produz nível alto em sua saída sempre que as suas entradas forem complementares (níveis opostos). Este circuito é muito útil em aplicações específicas e na realidade existe uma porta que realiza esta tabela verdade (TTL 74LS86), veja abaixo:



A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

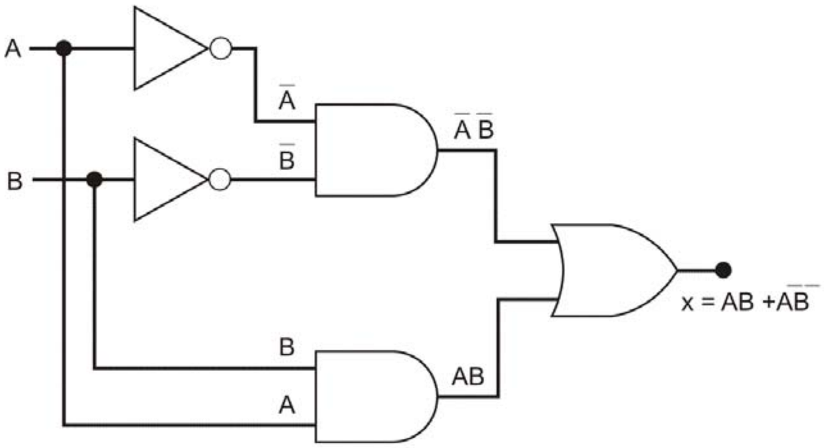
(a)

Símbolos para a porta EX-OR



(b) (c)

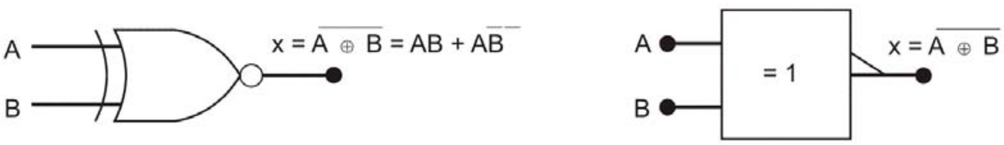
O outro circuito NOR-exclusivo gera na sua saída nível ALTO sempre que as entradas estiverem no mesmo nível lógico, também existe comercialmente CI com portas prontas para esta função (TTL 74LS266):



A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

(a)

Símbolos para a porta EX-NOR



(b) (c)