Acompanhamento Pet-BCC

Matéria : Circuitos Digitais

Assuntos: Circuitos Combinatórios, Álgebra de Boole e Mapa de Karnaugh.

1) Circuitos Lógicos

a) Tabela Verdade

A tabela verdade é uma tabela matemática utilizada em Lógica (e, consequentemente, em circuitos digitais) para determinar se uma forma é válida ou não.

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

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

b) Portas Lógicas

i) Operação OR (‘OU’) e a Porta OR

É a primeira das três operações booleanas básicas, que podem ser representada na forma de expressão booleana como **A + B**, em que OR é simbolizado por **'+'**. Além disso, há a porta OR, um circuito que possui duas ou mais entradas e sua saída é resultante da combinação das entradas através da operação OU. Podendo ser **saída alta (nível lógico 1, V) ou saída baixa (nível lógico 0, F).**

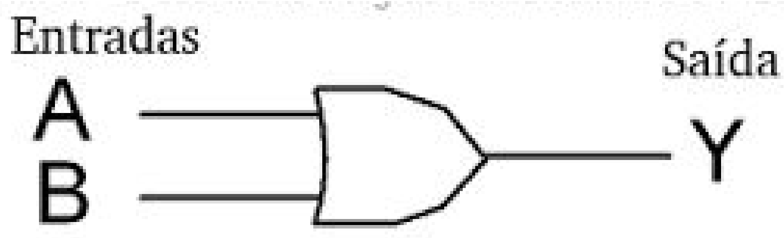


Tabela Verdade:

Entrada A	Entrada B	Saída Y
0	0	0
0	1	1
1	0	1
1	1	1

ii) Operação AND e a Porta AND

É a segunda das operações booleanas básicas, que podem ser representada na forma de expressão booleana como **A . B**, em que AND é simbolizado por **'.'**. Além disso, há a porta AND, um circuito que possui duas ou mais entradas e sua saída é resultante da combinação das entradas através da operação AND.

Tabela Verdade:

Entrada A	Entrada B	Saída Y
0	0	0
0	1	0
1	0	0
1	1	1

iii) Operação NOT (‘NÃO’) OU INVERSOR

É a terceira das três operações booleanas básicas, que podem ser representada na forma de expressão booleana como **Ā** , em que NOT é simbolizado pela barra em cima da entrada negada. Além disso, há a porta NOT, um circuito que possui apenas uma entrada e sua saída é resultante da negação da mesma.

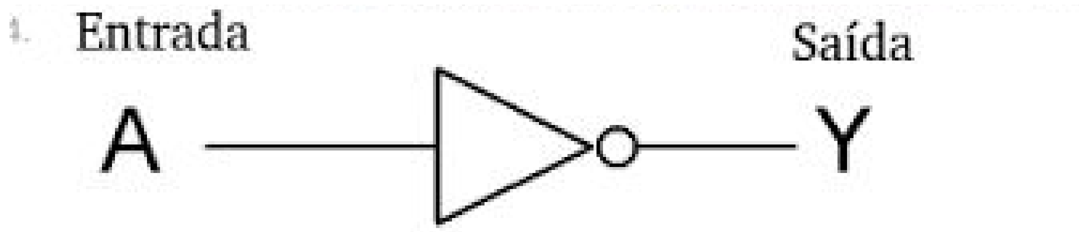


Tabela Verdade:

Entrada A	Saída Y
0	1
1	0

iv) Porta NOR ('NÃO-OU')

Combinação da operação NOT e OR. Semelhante à porta OR, diferindo apenas pelo círculo depois da porta OR, denotando a inversão. Representada pela expressão : ~(**A+B).**

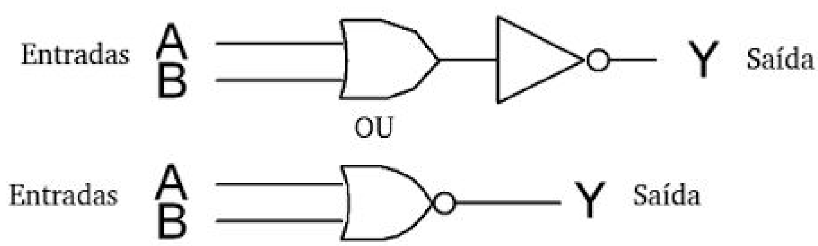


Tabela Verdade:

Entrada A	Entrada B	Saída Y
0	0	1
0	1	0
1	0	0
1	1	0

v) Porta NAND ('NÃO-AND')

Combinação da operação NOT e AND. Semelhante à porta AND, diferindo apenas pelo círculo depois da porta AND, denotando a inversão. Representada pela expressão : ~(**A.B).**

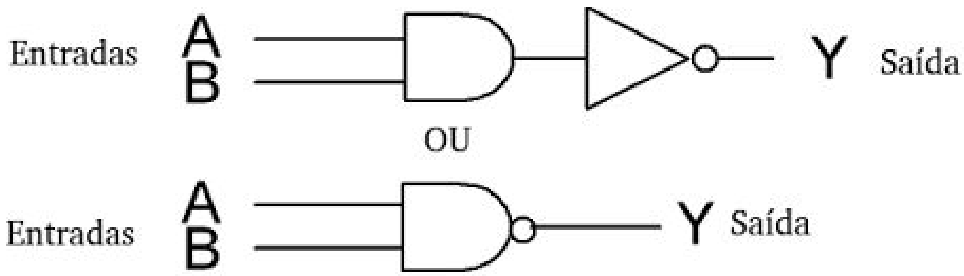


Tabela Verdade:

Entrada A	Entrada B	Saída Y
0	0	1
0	1	1
1	0	1
1	1	0

2) Circuitos Combinatórios

a) Soma-de-produtos

Os métodos de simplificação e projetos de circuitos lógicos requerem que a expressão esteja de soma-de-produtos. Consiste em dois ou mais termos AND (produtos) conectados por um operação OR.

Exemplo: ABC+A’BC

b) Produto-de-somas

Os métodos de simplificação e projetos de circuitos lógicos, às vezes, requerem que a expressão esteja de produto-de-somas. Consiste em dois ou mais termos OR (somas) conectados por um operação AND.

Exemplo: (A+B+C).(A’+B+C)

c) XOR (Exclusive-OR e Exclusive-NOR)

Circuito lógico especial, uma combinação de operações de AND, OR e NOT. Representada pela expressão: **□A . B + A . □B OU** $A \oplus B$

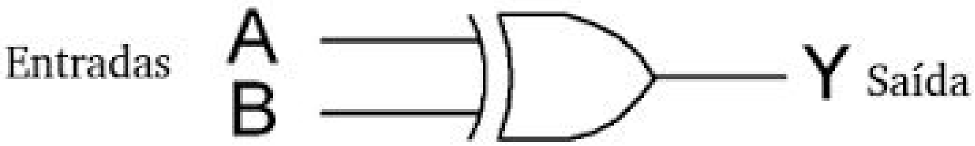


Tabela Verdade:

Entrada A	Entrada B	Saída Y
0	0	0
0	1	1
1	0	1
1	1	0

d) XNOR

Circuito lógico especial, uma combinação de operações de AND, OR e NOT. Representada pela expressão: **A . B + □A . □B ou not (**$A \oplus B$ **)**

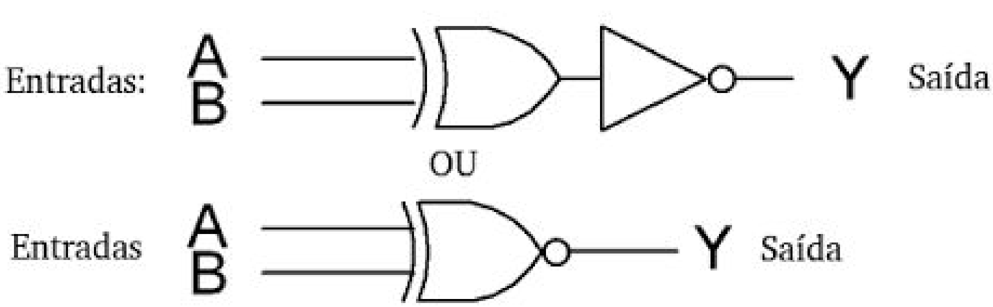


Tabela Verdade:

Entrada A	Entrada B	Saída Y
0	0	1
0	1	0
1	0	0
1	1	1

3) Álgebra de Boole

Álgebra de boole é utilizada para a simplificação de circuitos elétricos, que correspondem basicamente a expressões booleanas ou expressões lógicas. A simplificação de expressões resulta na diminuição de complexidade e consequentemente em diminuição de custo e de aumento na velocidade de processamento dos circuitos digitais. Abaixo estas as leis para simplificar as expressões:

Lei	**Nome da lei**
$\alpha \wedge \neg\alpha \equiv F$ $\alpha \vee \neg\alpha \equiv V$	Lei da contradição Lei do terceiro excluído
$\alpha \wedge V \equiv \alpha$ $\alpha \vee F \equiv \alpha$	Leis da identidade
$\alpha \wedge F \equiv F$ $\alpha \vee V \equiv V$	Leis da dominação
$\alpha \wedge \alpha \equiv \alpha$ $\alpha \vee \alpha \equiv \alpha$	Leis idempotentes
$\neg(\neg\alpha) \equiv \alpha$	Lei da dupla negação

$\alpha \wedge \beta \equiv \beta \wedge \alpha$ $\alpha \vee \beta \equiv \beta \vee \alpha$	Leis comutativas
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ $(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$	Leis associativas
$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$	Leis distributivas
$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$	Leis de De Morgan

$\alpha \vee (\alpha \wedge \beta) \equiv \alpha$ $\alpha \wedge (\alpha \vee \beta) \equiv \alpha$	Lei da absorção

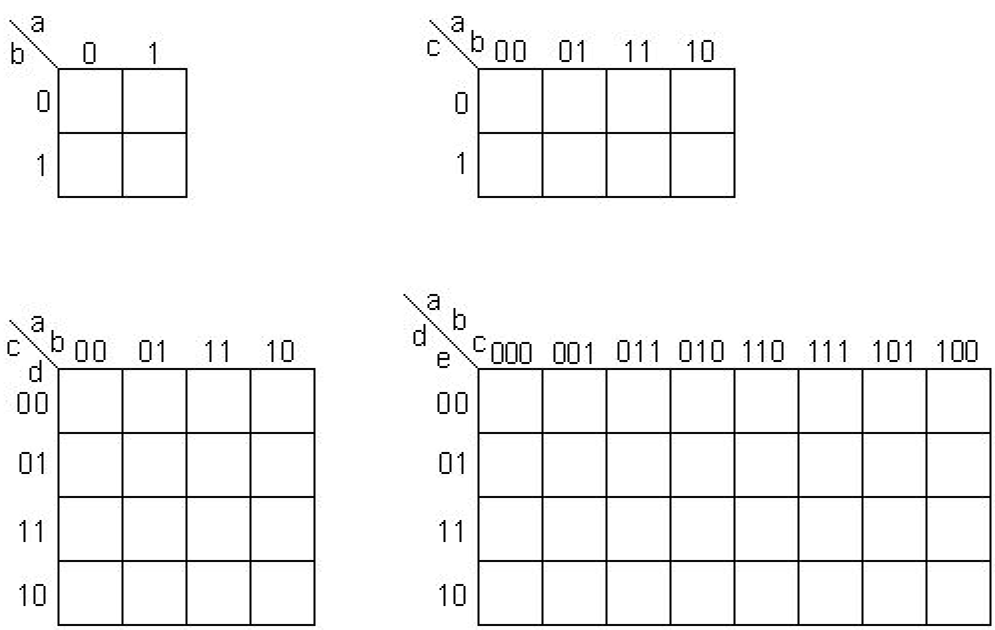
4) Mapa de Karnaugh

Além da álgebra de boole, também é possível utilizar um mapa de karnaugh para simplificar circuitos digitais.

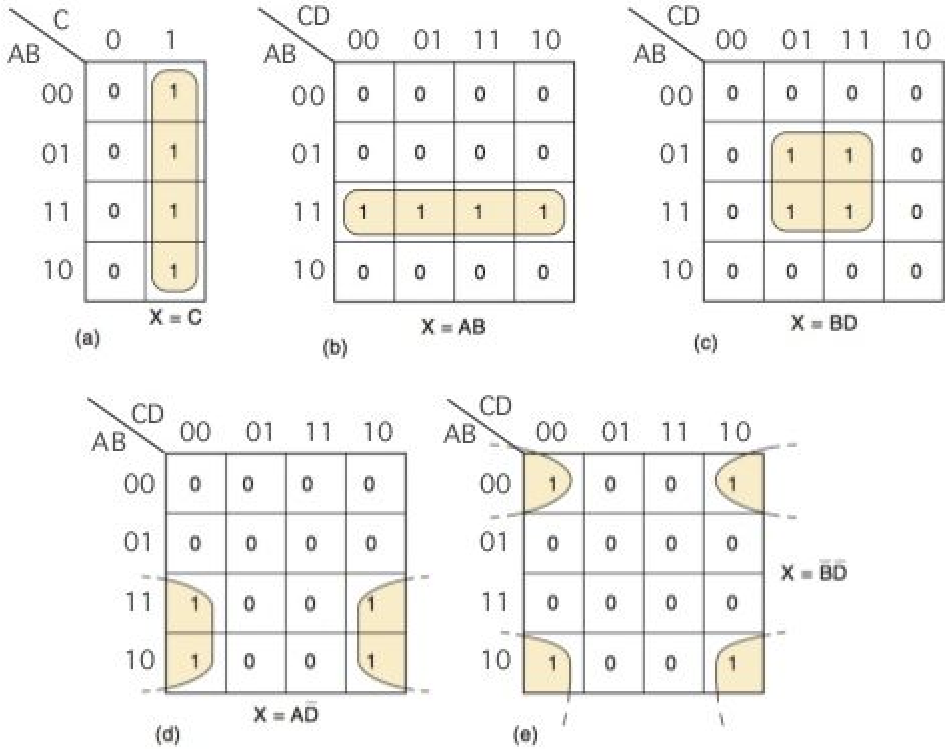
Primeiramente para utilizarmos os mapas devemos deixar a expressão booleana em um formato de somatório de produtos (Ex.: AB + CB + DC) ou produtório de somas (Ex.: (A+B)(C+B)(D+C)).

Após obter a expressão booleana no formato correto, o procedimento para utilizar o mapa de karnaugh é dividido em duas fases:

1 - Construção do mapa de células (representação da expressão)



2 - Selecionar as células vizinhas e para grupo de células representa um conjunto de somatório de produtos.


X = C
(a)
X = AB
(b)
X = BD
(c)
X = A$\bar{D}$
(d)
(e)
X = $\bar{B}\bar{D}$