Circuitos Lógicos
Portas Lógicas

Prof.: Daniel D. Silveira

Álgebra de Boole

- George Boole desenvolveu um sistema de análise lógica por volta de 1850
- Este sistema é conhecido atualmente como álgebra de Boole
- A álgebra de Boole expressa a operação de um circuito na forma de uma operação algébrica
- Na álgebra Booleana, as constantes e variáveis podem ter apenas 2 valores: 0 ou 1 (níveis lógicos)

Variáveis Lógicas

- As variáveis lógicas assumem estados distintos, e podem representar situações da vida real

Nível Lógico 0	**Nível Lógico 1**
Falso	Verdadeiro
Desligado	Ligado
Baixo	Alto
Nao	Sim
Chave aberta	Chave Fechada

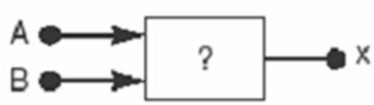
- A álgebra booleana tem apenas três operações básicas: AND (E), OR (OU), NOT (NÃO)

Tabela verdade

- Técnica para determinar como a saída lógica de um circuito depende dos níveis lógicos presentes nas entradas do circuito

Entradas / Saída

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



(a)

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

(b)

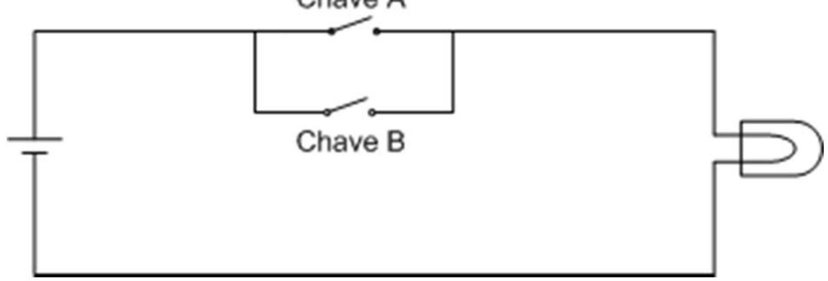
A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)

A operação OR (OU)

- Representada algebricamente como:

$$S=A+B \text{ (leia-se A OU B)}$$



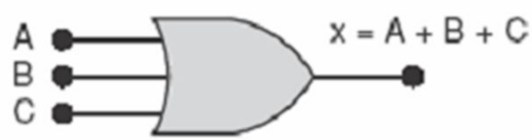
OR

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

(a)

Porta OR

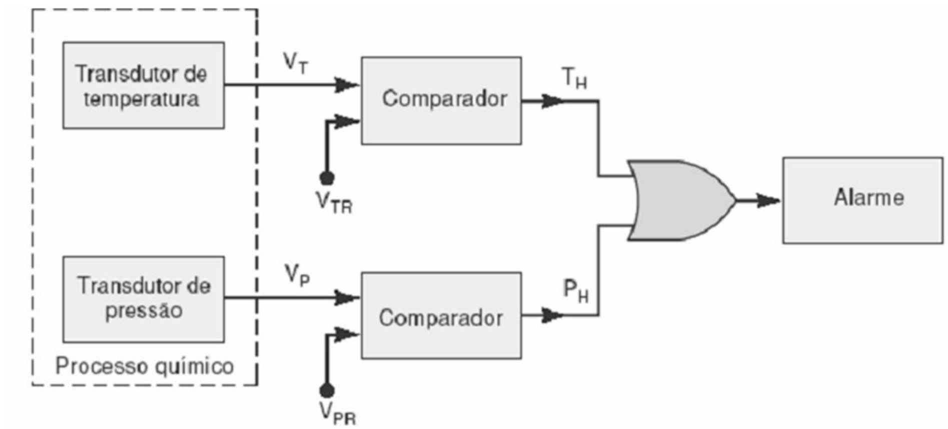
(b)

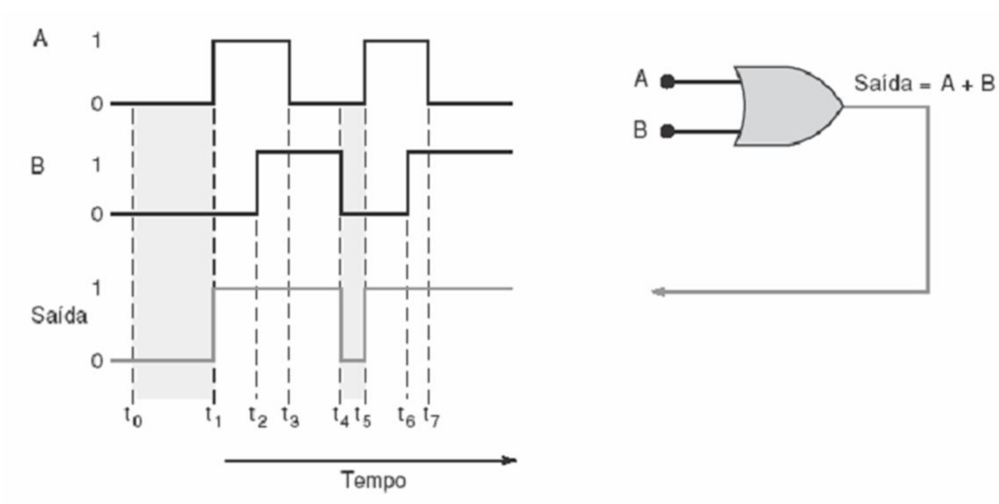


A	B	C	x = A + B + C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A operação OR (OU) – Aplicação

- Ativação de um alarme caso um sensor seja ativado

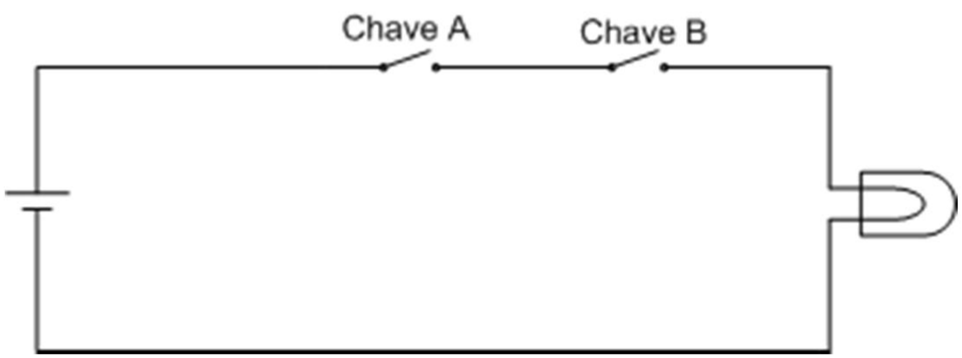




A operação AND (E)

- Representada algebricamente como:

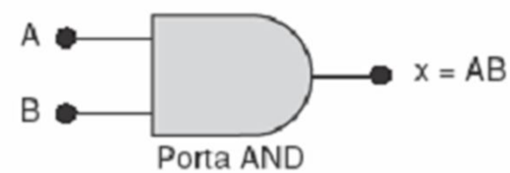
S=A.B (leia-se A **e** B)



AND

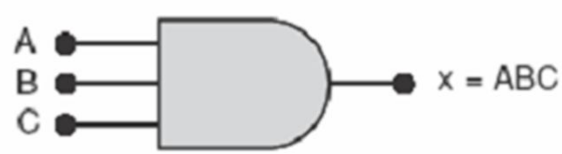
A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1

(a)



(b)

A	B	C	x = 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



A operação AND (E) – Exemplos

- Diagramas de tempo:

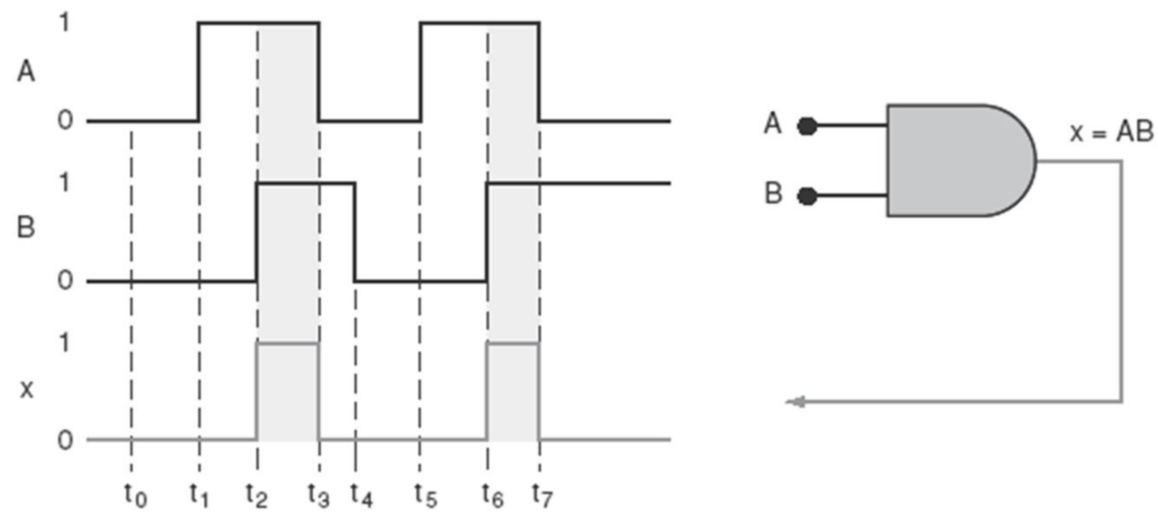


FIGURA 3.9
Exemplo 3.4.

- Circuito inibidor/habilitador

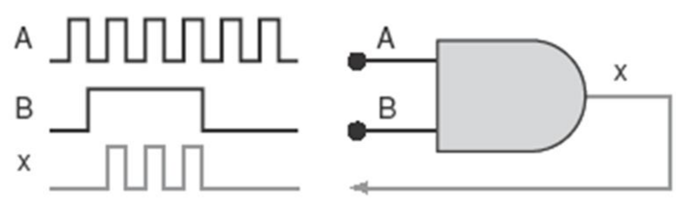
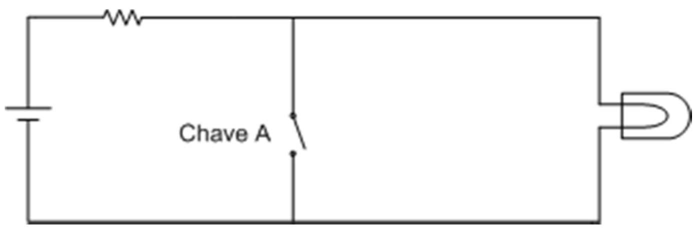


FIGURA 3.10
Exemplos 3.5A e 3.5B.

A operação NOT (NÃO) ou inversor

- Representada algebricamente como:

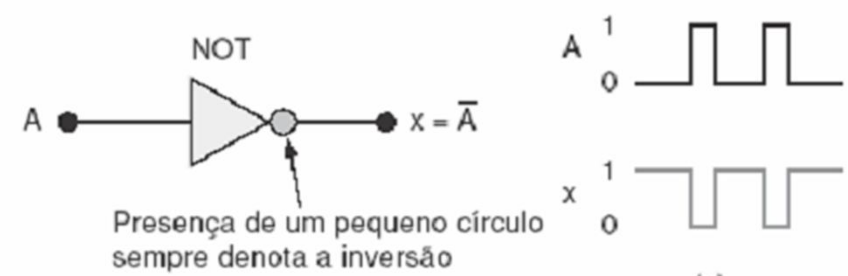
$S = \overline{A}$ ou $S = A'$ lê-se (A barra) ou (NÃO A)



NOT

A	$x = \overline{A}$
0	1
1	0

(a)

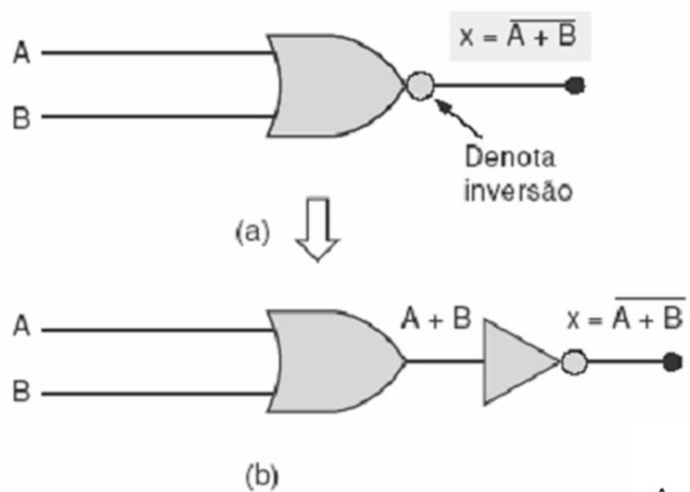


(b) (c)

- Tem apenas uma entrada
- Também conhecido como complemento

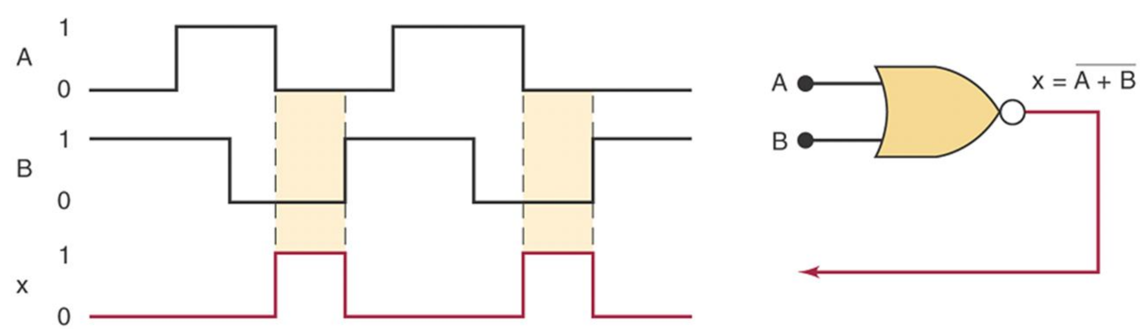
A porta NOR (NÃO-OU)

- Combinação da porta OU com a porta inversora



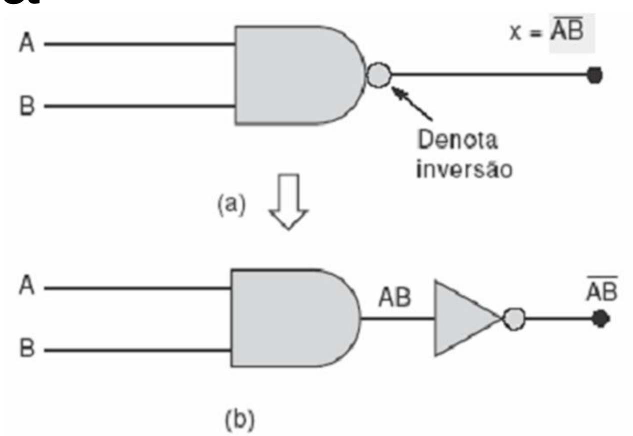
		OR	NOR
A	B	A + B	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(c)



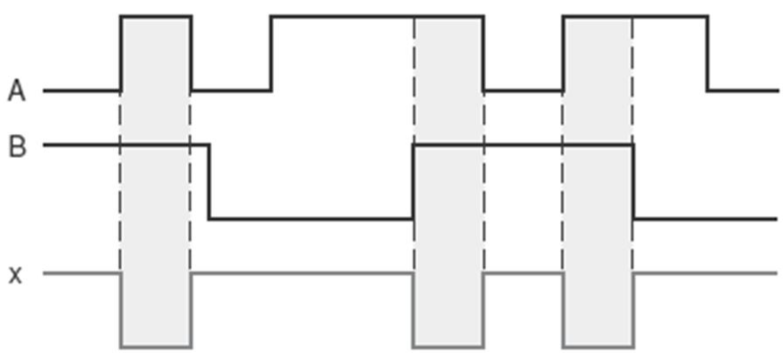
A porta NAND (NÃO-E)

- Combinação da porta AND com a porta inversora



A	B	AND: AB	NAND: $\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(c)



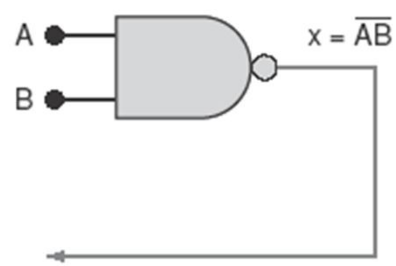


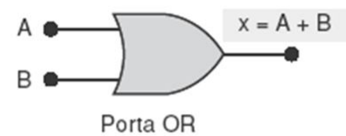
FIGURA 3.23
Exemplo 3.10.

Quadro resumo

Blocos lógicos básicos

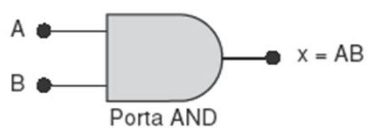
OR

A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1



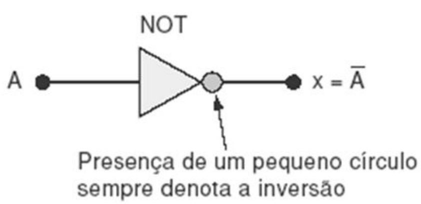
AND

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

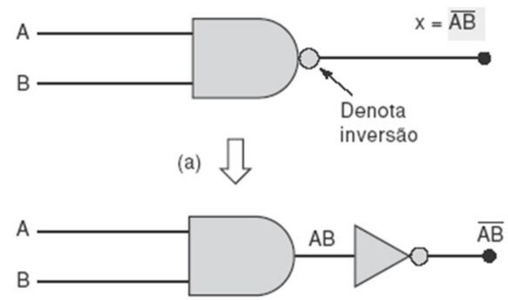


NOT

A	$x = \overline{A}$
0	1
1	0

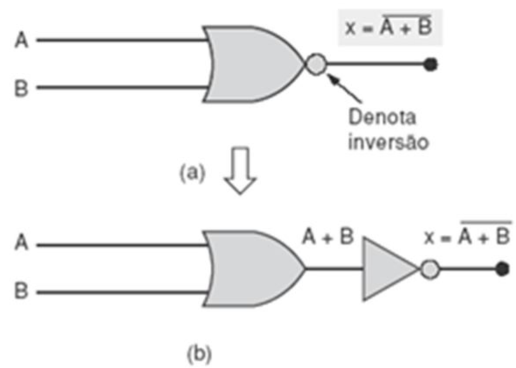


Blocos lógicos derivados



A	B	AND: AB	NAND: $\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

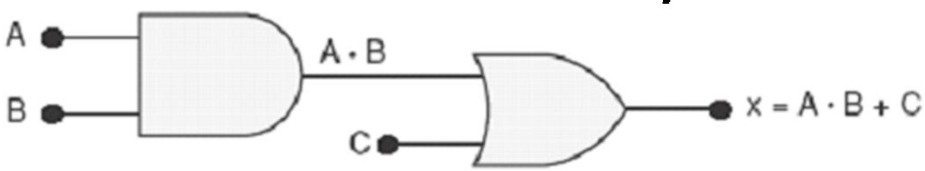
(c)

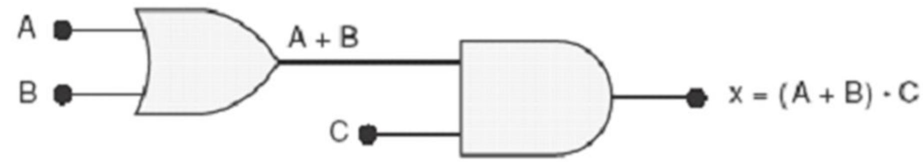


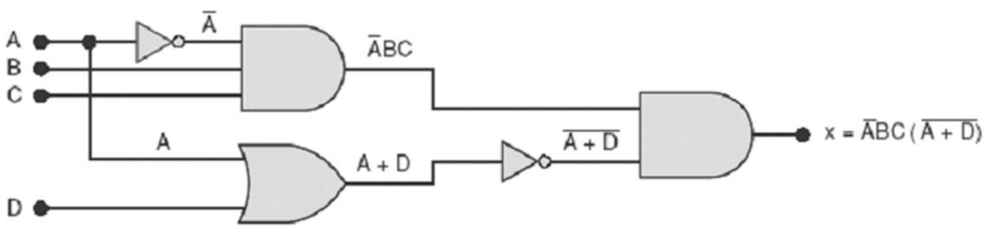
A	B	OR: $A + B$	NOR: $\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Associações de portas

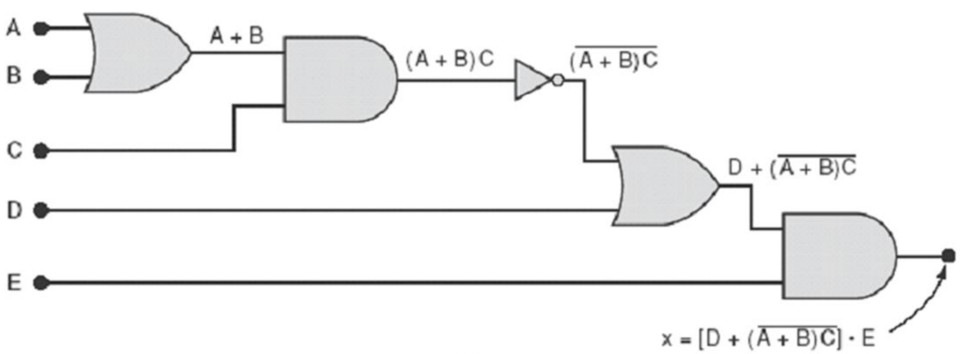
- Usando-se expressões booleanas, pode-se determinar a expressão lógica de saída







(a)



(b)

Avaliando a saída dos circuitos lógicos

- Pode-se substituir as variáveis pelos valores desejados e obter o resultado da expressão

$A = 0$, $B = 1$, $C = 1$ e $D = 1$.

$$
\begin{aligned}
x &= \bar{A}BC(\overline{A + D}) \\
&= \bar{0} \cdot 1 \cdot 1 \cdot (\overline{0 + 1}) \\
&= 1 \cdot 1 \cdot 1 \cdot (\overline{0 + 1}) \\
&= 1 \cdot 1 \cdot 1 \cdot (\bar{1}) \\
&= 1 \cdot 1 \cdot 1 \cdot 0 \\
&= 0
\end{aligned}
$$

$A = 0$, $B = 0$, $C = 1$, $D = 1$ e $E = 1$.

$$
\begin{aligned}
x &= [D + \overline{(A + B)C}] \cdot E \\
&= [1 + \overline{(0 + 0) \cdot 1}] \cdot 1 \\
&= [1 + \overline{0 \cdot 1}] \cdot 1 \\
&= [1 + \bar{0}] \cdot 1 \\
&= [1 + 1] \cdot 1 \\
&= 1 \cdot 1 \\
&= 1
\end{aligned}
$$

- Quais as saídas quando todas as entradas forem 1 para as duas expressões?

Determinando o nível lógico na saída dos diagramas

- Analisa-se a saída de cada porta separadamente

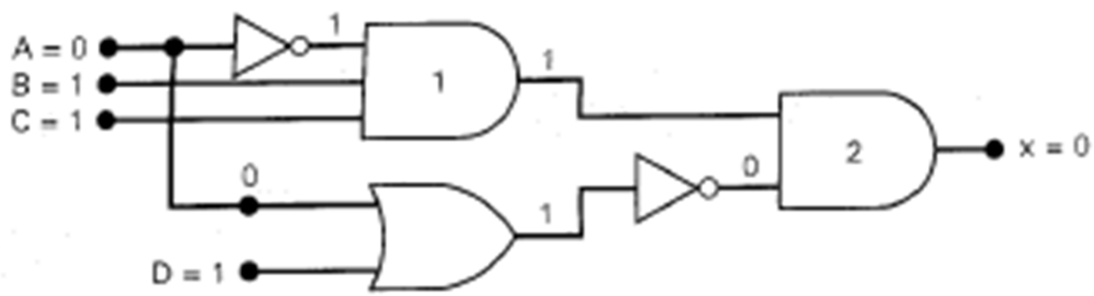
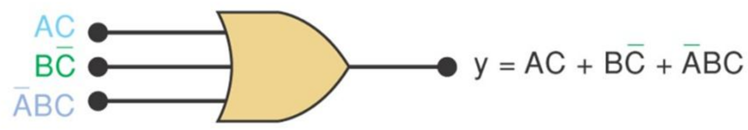


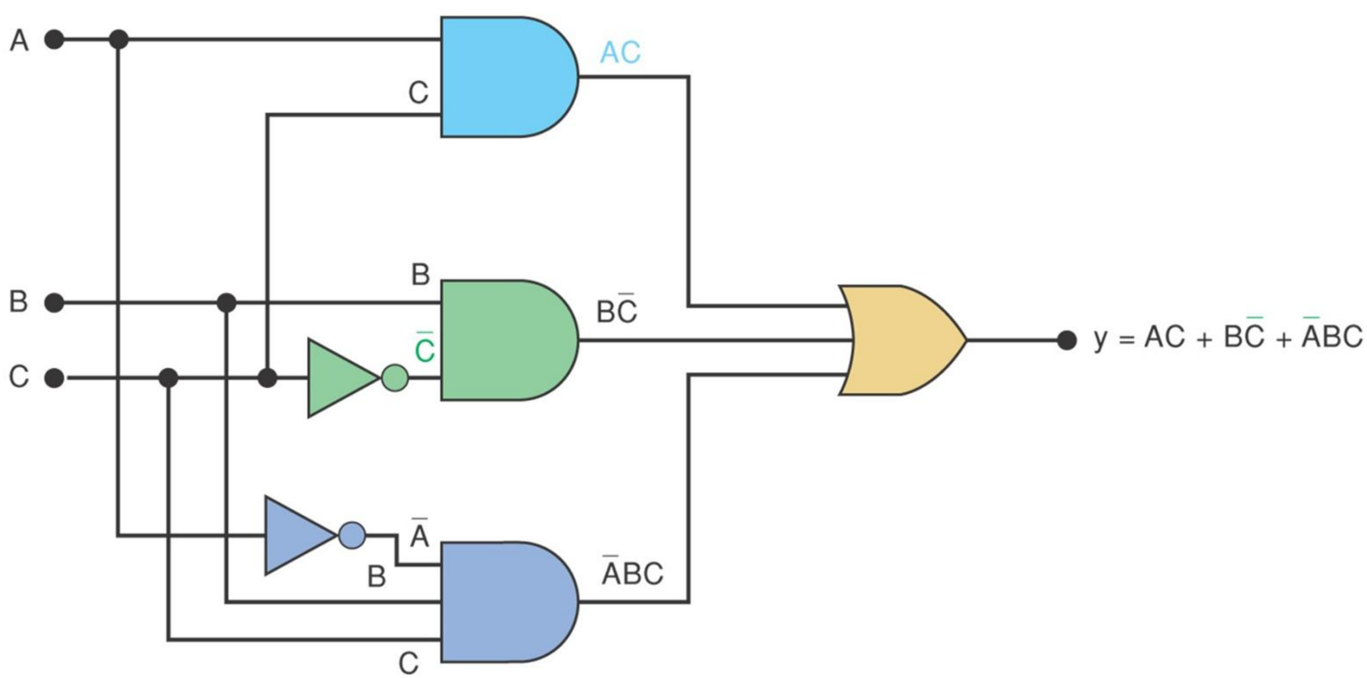
FIGURA 3.16 Determinando o nível lógico da saída a partir de um diagrama do circuito.

- Se todas as entradas estivem em nível lógico baixo, qual a saída?

Implementando circuitos a partir de expressões booleanas



(a)



(b)

Montando a tabela verdade a partir de um circuito

- Primeiro deriva-se a expressão de saída

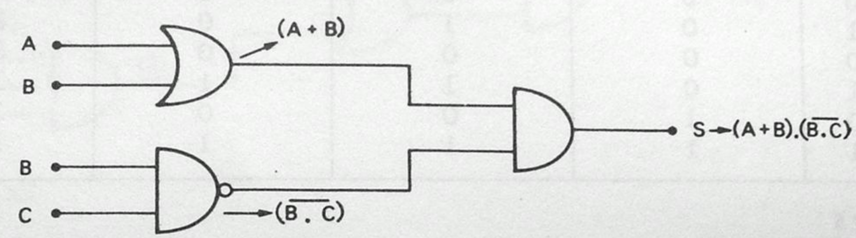


Figura 2.39

Extraímos do circuito a expressão:

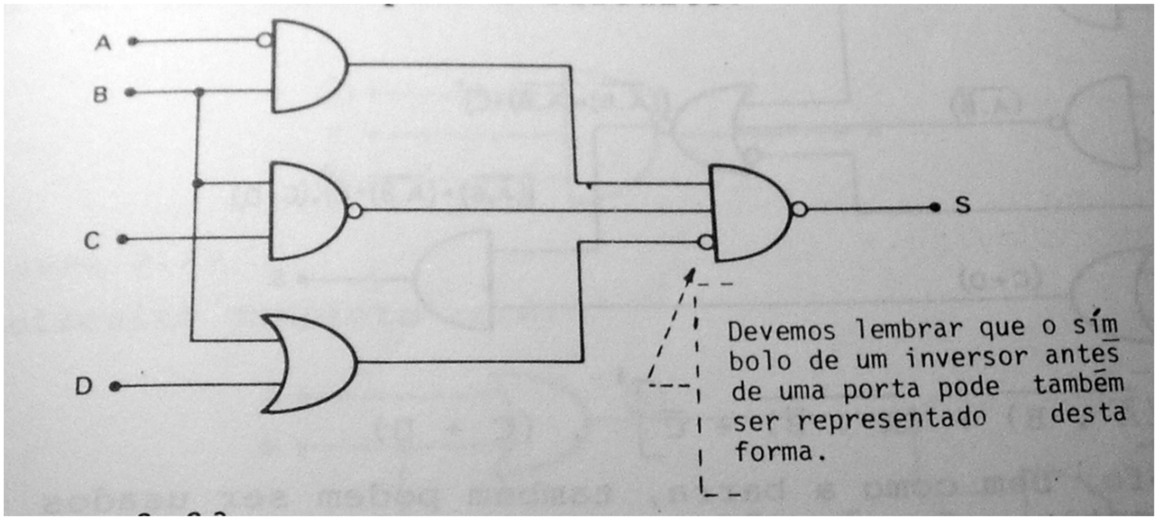
$$S = \underbrace{(A + B)}_{1^{\text{o}}\ \text{membro}} \cdot \underbrace{(\overline{B \cdot C})}_{2^{\text{o}}\ \text{membro}}$$

Seguindo o processo, montamos a tabela da verdade.

A	B	C	1º membro $A + B$	Auxiliar $B \cdot C$	2º membro $\overline{B \cdot C}$	Resultado S
0	0	0	0	0	1	0
0	0	1	0	0	1	0
0	1	0	1	0	1	1
0	1	1	1	1	0	0
1	0	0	1	0	1	1
1	0	1	1	0	1	1
1	1	0	1	0	1	1
1	1	1	1	1	0	0

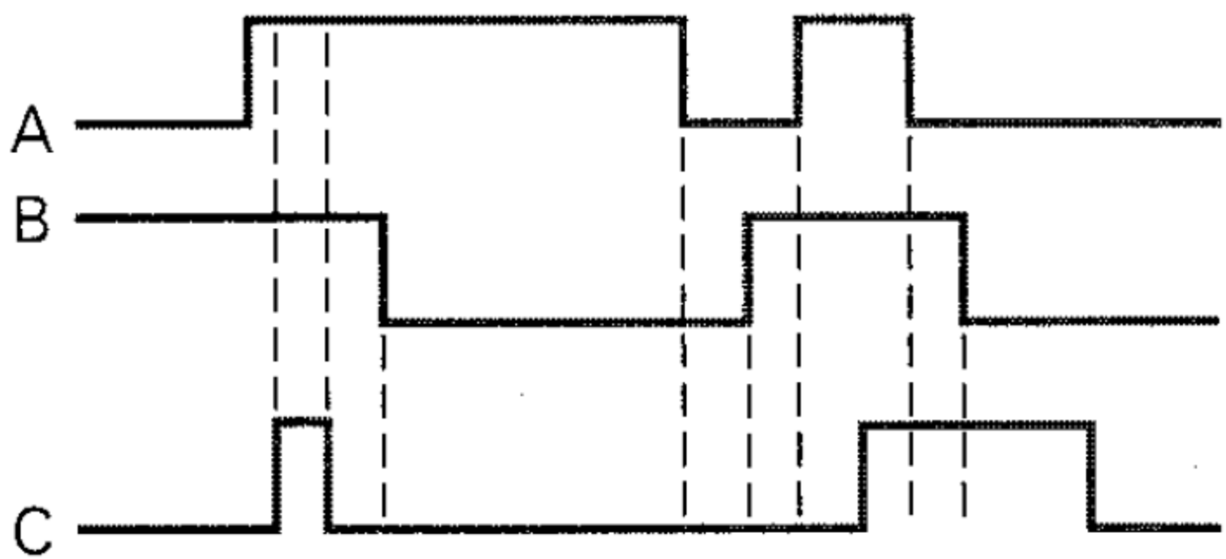
Exercícios propostos

- Desenhe o circuito que executa a função booleana S=A.B.C+(A+B).C e gere sua tabela verdade
- Escreva a expressão que representa o circuito:



Exercícios propostos

3.17 – a) Aplique as formas de onda de entrada da Figura abaixo em uma porta NOR e desenhe a forma de onda de saída.
b) Repita para a entrada C mantida permanentemente em nível BAIXO.
c) Repita para a entrada C mantida permanentemente em nível ALTO.

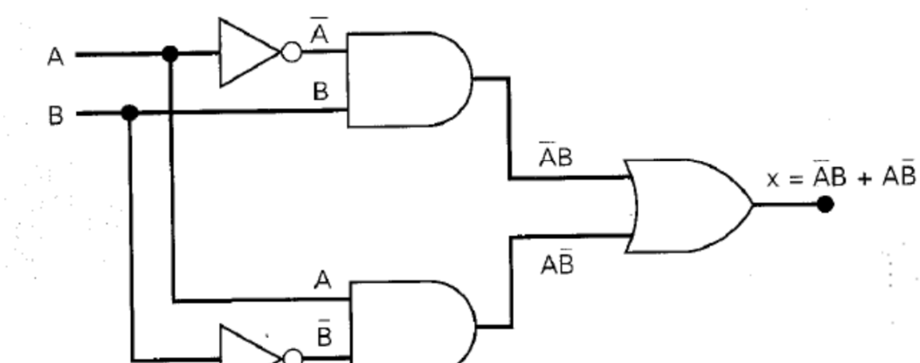


3.18 – Repita o problema 3.17 para uma porta NAND.

A porta XOR

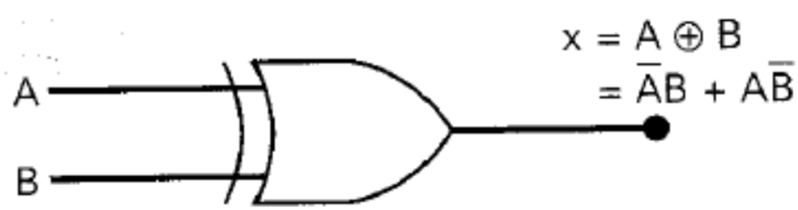
- Nível ALTO na saída ***somente*** quando as 2 entradas são diferentes entre si!
- A porta lógica que representa esta função é a XOR

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



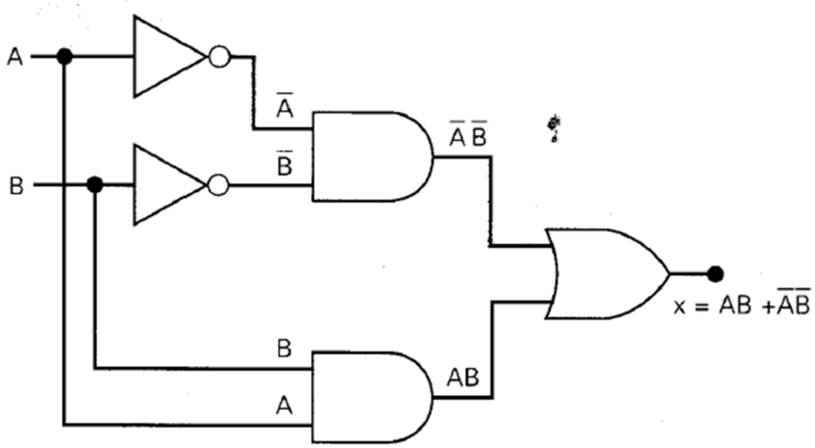
(a)

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



A porta XNOR

- Para a porta XNOR, teremos o resultado 1 ***somente*** quando as 2 entradas forem iguais (exatamente o inverso da XOR)!
- Tanto a porta XOR quanto a XNOR possuem **somente** duas entradas $x = AB + \overline{A}\,\overline{B}$



(a)

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