Circuito Lógico

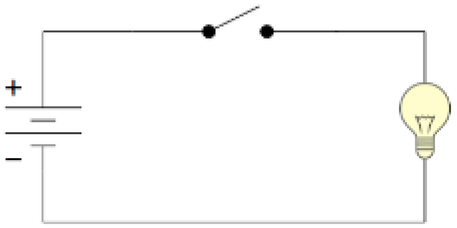
Curso ILC

Circuito Lógico

Uma chave pode estar em uma de duas possíveis posições:

- Chave fechada: corrente pode passar.
- Chave aberta: há interrupção de corrente.

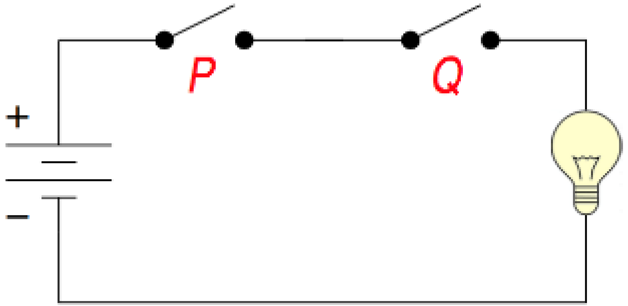
Exemplo de uma chave num circuito:

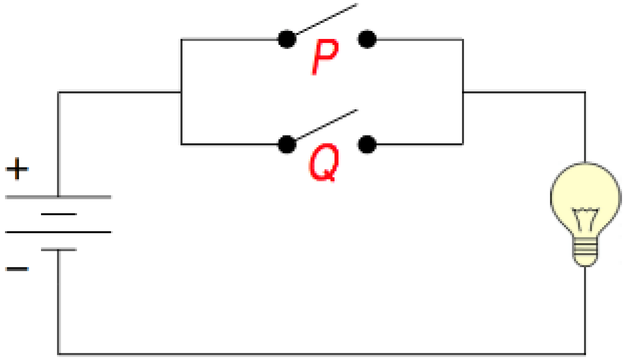


- Lâmpada acende sse corrente passa por ela.
- Isto acontece, sse a chave está fechada.

Circuito Lógico: aplicação

Sejam os circuitos abaixo e os possíveis comportamentos:





Chaves		Lâmpada
P	Q	Estado
Fechada	Fechada	Acesa
Fechada	Aberta	Apagada
Aberta	Fechada	Apagada
Aberta	Aberta	Apagada

Chaves		Lâmpada
P	Q	Estado
Fechada	Fechada	Acesa
Fechada	Aberta	Acesa
Aberta	Fechada	Acesa
Aberta	Aberta	Apagada

– $P \wedge Q$ (conjunção);
– $P \vee Q$ (disjunção).

No projeto de circuitos digitais, os valores lógicos **verdadeiro** e **falso** são normalmente substituídos pelos símbolos **1** e **0**.

➜ Estes símbolos são chamados de *bits* (*binary digits*).

Circuito Lógico: Aplicação

Axiomas da álgebra de Boole

1. As variáveis booleanas só podem assumir os valores 0 ou 1, ou seja:

$$\boldsymbol{x} = 0 \text{ ou } \boldsymbol{x} = 1$$

2. O complemento ou negação de uma variável $\boldsymbol{x}$, simbolizado por $\overline{\boldsymbol{x}}$ é obtida tal que:

Se: $\boldsymbol{x} = 0$, então: $\overline{\boldsymbol{x}} = 1$
Se: $\boldsymbol{x} = 1$, então: $\overline{\boldsymbol{x}} = 0$

3. A operação AND representada pelo símbolo: $\cdot$, ou ^, corresponde à multiplicação lógica sendo definida pelas equações:

$$0 \cdot 0 = 0 \qquad 1 \cdot 0 = 0$$

$$0 \cdot 1 = 0 \qquad 1 \cdot 1 = 1$$

Axiomas da álgebra de Boole

4. A operação OR representada pelo símbolo +, ou V, corresponde à soma lógica sendo definida pelas equações:

$$0 + 0 = 0 \qquad 1 + 0 = 1$$
$$0 + 1 = 0 \qquad 1 + 1 = 1$$

5. A operação XOR representada pelo símbolo $\oplus$, corresponde à multiplicação lógica sendo definida pelas equações:

$$0 \oplus 0 = 0 \qquad 1 \oplus 0 = 1$$
$$0 \oplus 1 = 1 \qquad 1 \oplus 1 = 0$$

Portas lógicas

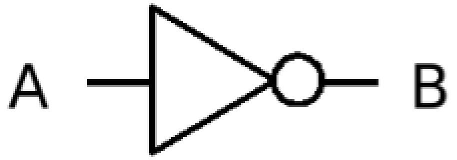
- **Portas Lógicas:** são elementos de circuitos lógicos básicos que realizam as Operações Booleanas básicas:

    ✓ Inversão (Negação), AND, OR, e XOR.

As portas lógicas básicas são as seguintes:

Porta Inversora ou NOT

Realiza a operação de complemento ou negação, sendo o símbolo e a tabela verdade como mostrado a seguir:



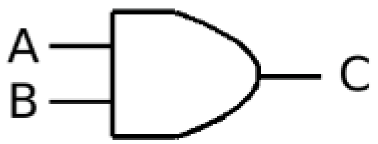
A	B
1	0
0	1

Portas lógicas

Porta AND

Realiza a operação AND (multiplicação lógica) entre duas ou mais variáveis binárias.

O símbolo da porta AND de 2 entradas e a sua tabela verdade são mostrados a seguir:

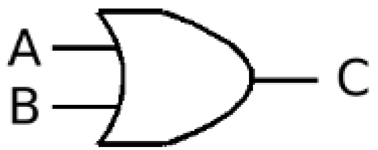


A	B	C = A·B
0	0	0
0	1	0
1	0	0
1	1	1

Porta OR

Realiza a operação OR (soma lógica) entre duas ou mais variáveis binárias.

O símbolo da porta OR de 2 entradas e a sua tabela verdade são mostrados a seguir:



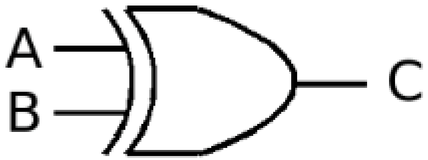
A	B	C=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Portas lógicas

Porta XOR

Realiza a operação XOR (soma exclusiva) entre duas ou mais variáveis binárias.

O símbolo da porta XOR de 2 entradas e a sua tabela verdade são mostrados a seguir:

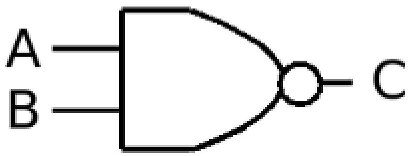


A	B	C=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Porta NAND

Realiza a operação NAND (negação da multiplicação lógica) entre duas ou mais variáveis binárias.

O símbolo da porta NAND de 2 entradas e a sua tabela verdade são mostrados a seguir:



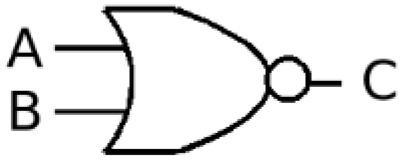
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Portas Lógicas

NOR

Realiza a operação NOR (negação da soma lógica) entre duas ou mais variáveis binárias.

O símbolo da porta NOR de 2 entradas e a sua tabela verdade são mostrados a seguir:

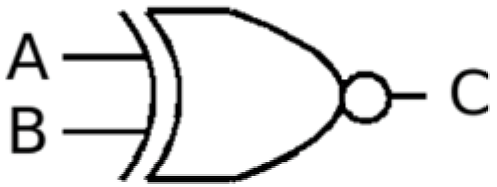


A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

XNOR

Realiza a operação XNOR (negação da soma exclusiva) entre duas ou mais variáveis binárias.

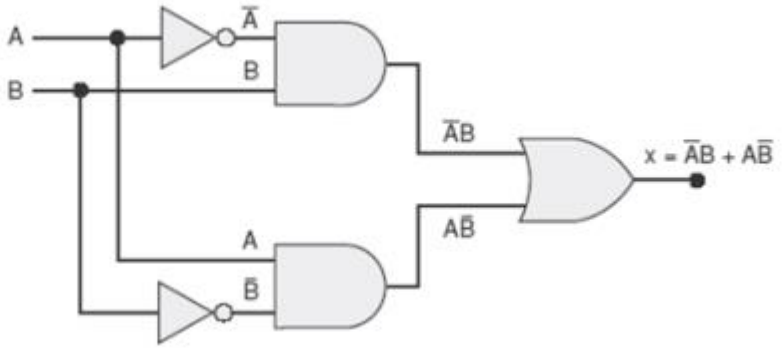
O símbolo da porta XNOR de 2 entradas e a sua tabela verdade são mostrados a seguir:



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

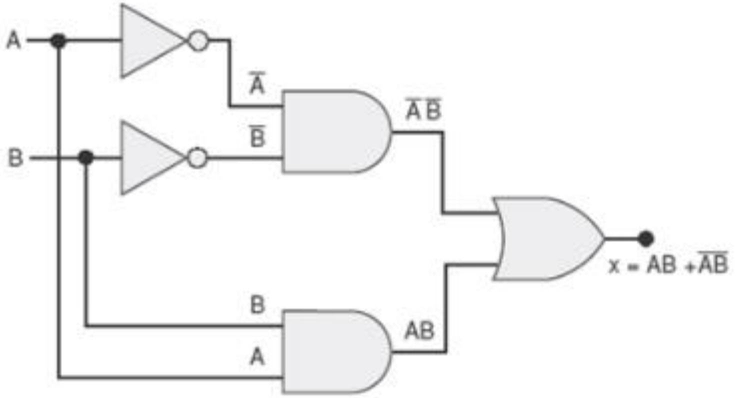
Portas lógicas

XOR com portas NOT/AND/OR



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

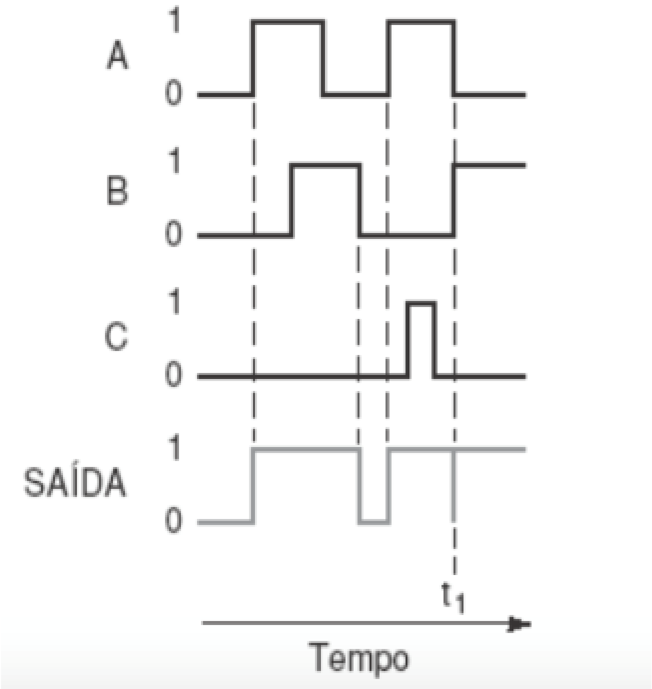
XNOR com portas NOT/AND/OR



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

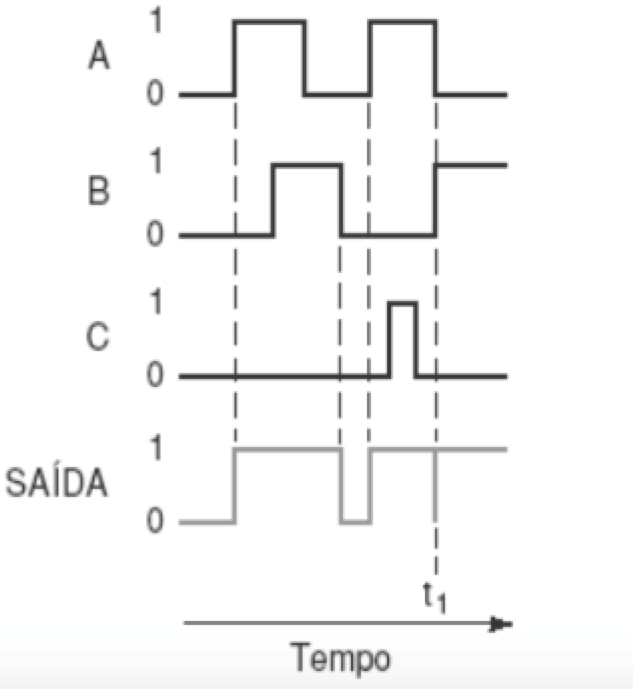
Exercicio

Observe o comportamento abaixo e desenhe o circuito

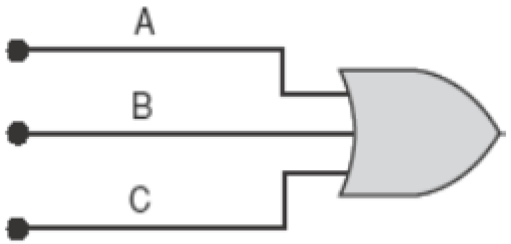


Exercicio

Observe o comportamento abaixo e desenhe o circuito



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



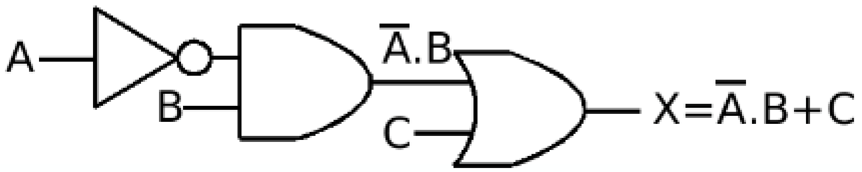
A + B + C

Como exemplo: o circuito abaixo pode ser expresso pela expressão booleana: $\boldsymbol{x} = \overline{\boldsymbol{A}}.\ \boldsymbol{B} + \mathbf{C}$

Qualquer circuito lógico, por mais complexo que seja, pode ser completamente descrito através das operações booleanas OR, AND, e NOT.

Circuitos Lógicos

Como exemplo: o circuito abaixo pode ser expresso pela expressão booleana: $\boldsymbol{x = \overline{A}.\,B + C}$



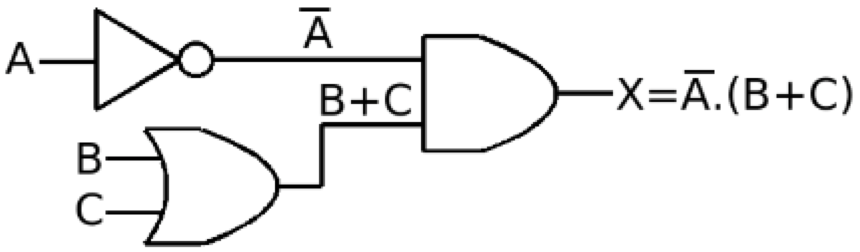
Qualquer circuito lógico, por mais complexo que seja, pode ser completamente descrito através das operações booleanas OR, AND, e NOT.

Circuitos Lógicos

Já a expressão booleana: $\overline{A}.\,(B + C)$ descreve o circuito:

Circuitos Lógicos

Já a expressão booleana: $\overline{A}.\,(B + C)$ descreve o circuito:



Observe que as expressões: $\overline{A}.\,B + C$ e $\overline{A}.\,(B + C)$, diferem apenas na presença dos parênteses, o que deixa claro que a operação AND precede as demais operações, exceto se a precedência for quebrada pelos parênteses.

Circuitos Lógicos

A) $X = \overline{A}\ .\ B\ .\ C\ .\ (\overline{A + D})$

B) $X = [D + \overline{(A + B).C}].E$

Exercícios: mostre os circuitos resultantes

Avaliação das expressões boolenas

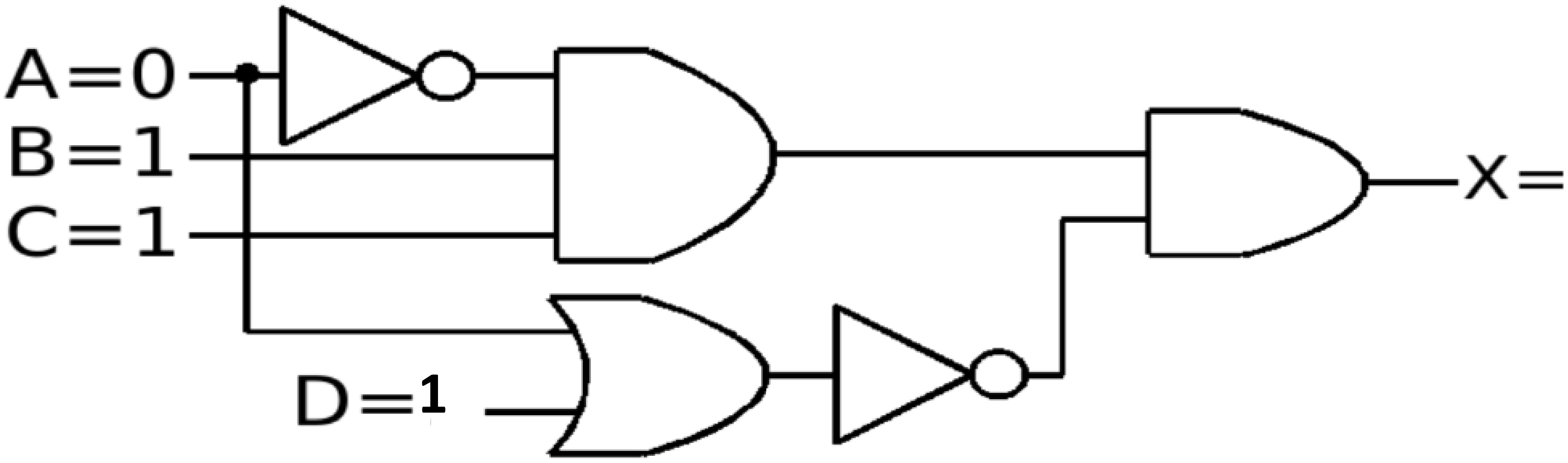
a) Na expressão: $X = \overline{\boldsymbol{A}}\ .\ \boldsymbol{B}\ .\ \mathbf{C}\ .\ (\overline{\boldsymbol{A}+\boldsymbol{D}})$, substituindo-se as variáveis A, B, C, e D, por 0, 1, 1, e 1, respectivamente, obtém-se:

$$X = \overline{\mathbf{0}}\ .\ \mathbf{1}\ .\ \mathbf{1}\ .\ (\overline{\mathbf{0}+\mathbf{1}}) = \mathbf{1.1.1}.(\overline{\mathbf{0}+\mathbf{1}}) = \mathbf{1.1.1}.(\overline{\mathbf{1}}) = \mathbf{1.1.1.0} = \mathbf{0}$$

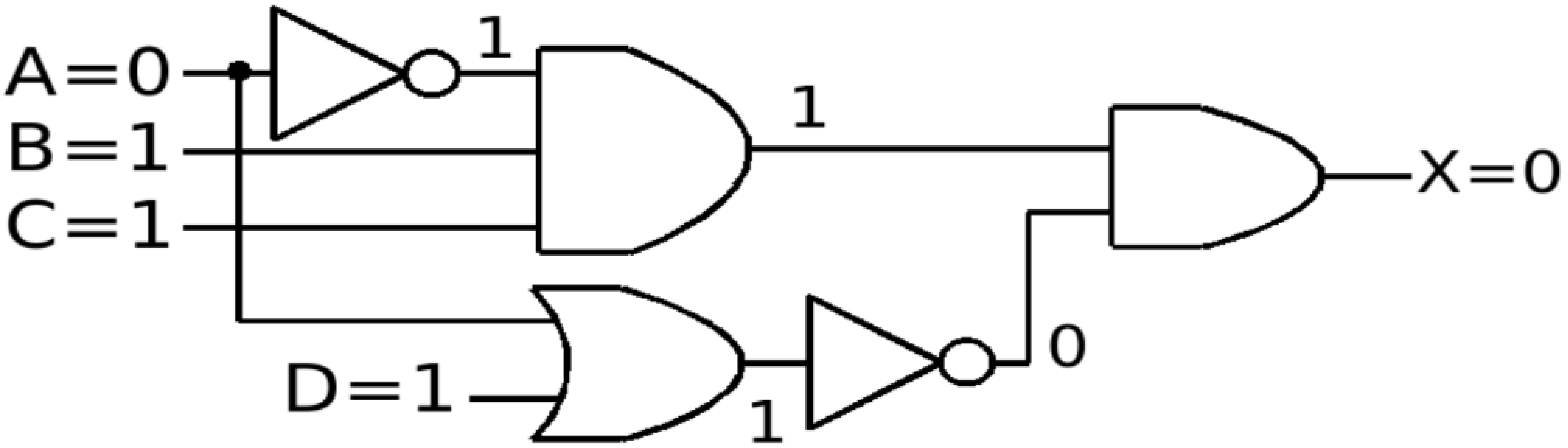
b) Na expressão: $X = [\boldsymbol{D} + (\overline{\boldsymbol{A}+\boldsymbol{B}).\boldsymbol{C}}].\boldsymbol{E}$, substituindo-se as variáveis A, B, C, D, e E, por 0, 0, 1, 1, e 1, respectivamente, obtém-se:

$$X = [\mathbf{1} + (\overline{\mathbf{0}+\mathbf{0}).\mathbf{1}}].\mathbf{1} = [\mathbf{1} + (\overline{\mathbf{0}.\mathbf{1}})].\mathbf{1} = [\mathbf{1}+\mathbf{1}].\mathbf{1} = \mathbf{1}.\mathbf{1} = \mathbf{1}$$

Avaliação da saída do circuito lógico

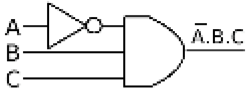


Avaliação da saída do circuito lógico

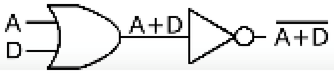


Pode-se implementar (sintetizar) um circuito lógico através da expressão booleana associada a este.
Por exemplo a expressão: $X = \overline{A}\,.\,B\,.\,C\,.\,(\overline{A+D})$, pode se implementada partindo-se das partes que a compõem: $Y = \overline{A}\,.\,B\,.\,C$ e $Z = (A + D)$,
Sintetizando-as e por fim associando as duas através da operação booleana AND, pois: $X = Y\,.\,Z$

O circuito lógico da expressão booleana: $Y = \overline{A}\,.\,B\,.\,C$, é o seguinte:



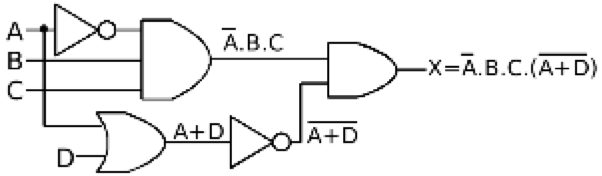
O circuito lógico da expressão booleana: $Z = (\overline{A+D})$, é o seguinte:



Por fim, as saídas dos dois blocos de circuito anteriores são entradas de uma porta AND, e a expressão booleana:

$$X = Y\,.\,Z = [\overline{A}\,.\,B\,.\,C]\,.\,(\overline{A+D})$$

Que resulta no circuito mostrado abaixo:



Implementando o circuito

1. Identidade:

$A + 0 = A$ e $A \cdot 1 = A$ (dual)

2. Elemento Nulo:

$A + 1 = 1$ e $A \cdot 0 = 0$ (dual)

3. Idempotência:

$A + A = A$ e $A \cdot A = A$ (dual)

4. Complemento:

$A + \overline{A} = 1$ e $A \cdot \overline{A} = 0$ (dual)

5. Involução:

$$\overline{(\overline{A})} = A$$

6. Comutativa:

$$A + B = B + A$$

7. Associativa:

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

8. Distributiva:

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

e

$$(A + B) \cdot (A + C) = A + A \cdot C + B \cdot A + B \cdot C \quad \text{(dual)}$$

Teoremas da álgebra booleana

9. Cobertura:

$$A + A\,.\,B = A$$
e
$$A\,.\,(A + B) = A$$

10. Combinação:

$$A\,.\,B + A\,.\,\overline{B} = A$$
e
$$(A + B).\,(A + \overline{B}) = A \quad \text{(dual)}$$

$$A + \overline{A}\,.\,B = A + B$$

$$\overline{A} + A\,.\,B = \overline{A} + B$$

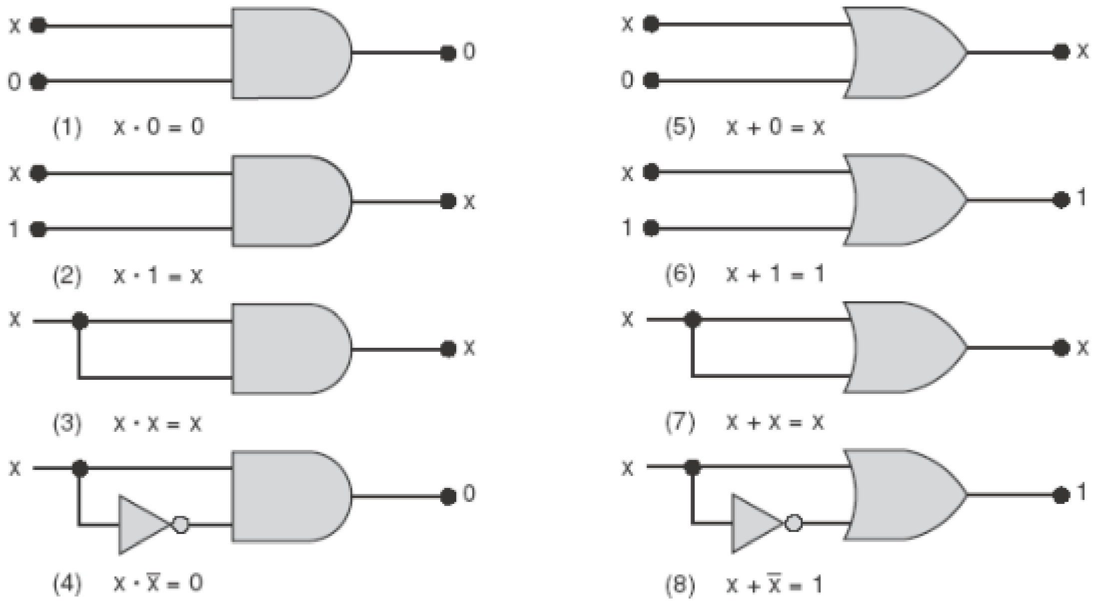
11. Consenso (Termo Fantasma):

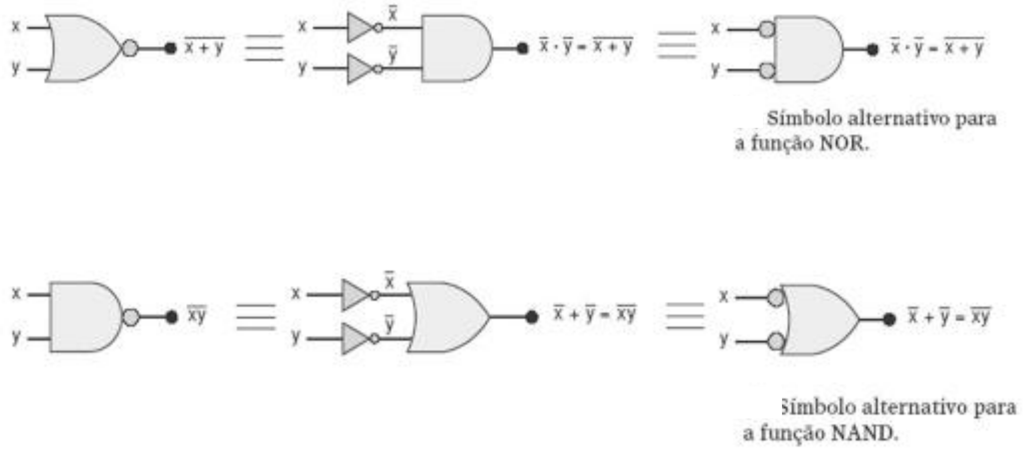
$$A\,.\,B + \overline{A}\,.\,C + B\,.\,C = A\,.\,B + \overline{A}\,.\,C$$
e
$$(A + B).\,(\overline{A} + C).\,(B + C) = (A + B)\,.\,(\overline{A} + C) \quad \text{(dual)}$$

12. De Morgan:

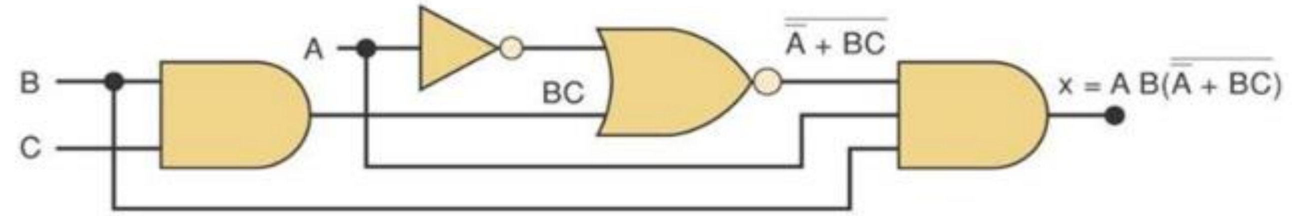
$$\overline{(A.\,B)} = \overline{A} + \overline{B}$$
e
$$\overline{(A + B)} = \overline{A}.\,\overline{B} \quad \text{(dual)}$$

Teoremas da álgebra booleana





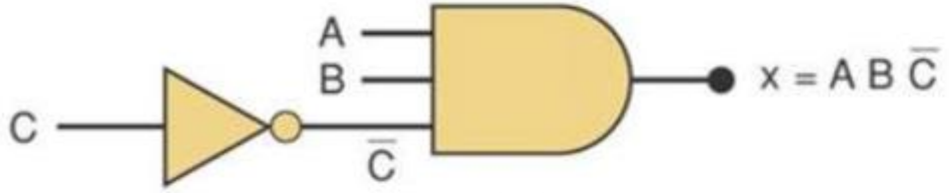
Teoremas da álgebra booleana



(a)

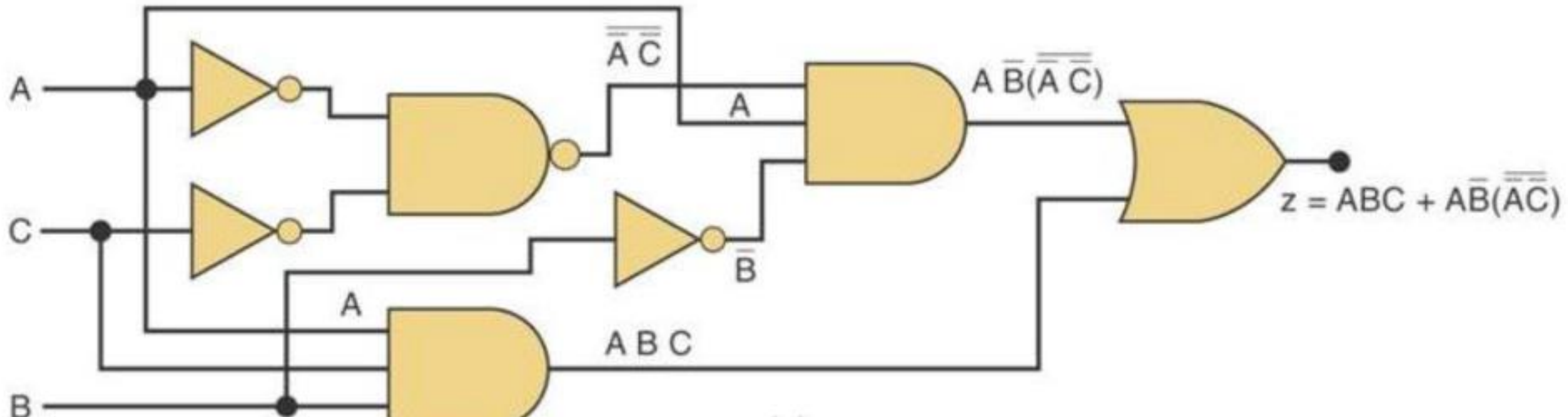
Simplificação algébrica de circuitos lógicos

$$X = A\,.\,B\,.\,(\overline{\overline{A} + BC}) = A.B.A.\overline{B.C} = A.B.A.(\overline{B} + \overline{C}) =$$
$$= A.B.A.\overline{B} + A.B.A.\overline{C} = A.B.\overline{B} + A.B.\overline{C} = 0 + A.B.\overline{C}$$
$$= A.B.\overline{C}$$



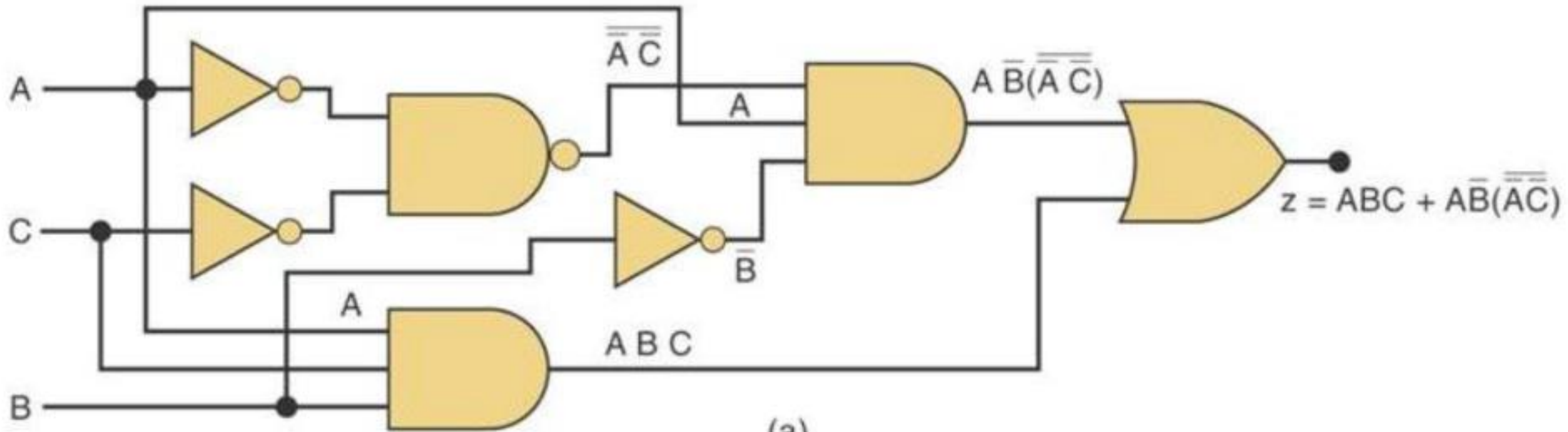
(b)

Simplificação algébrica de circuitos lógicos



(a)

Simplificação algébrica de circuitos lógicos



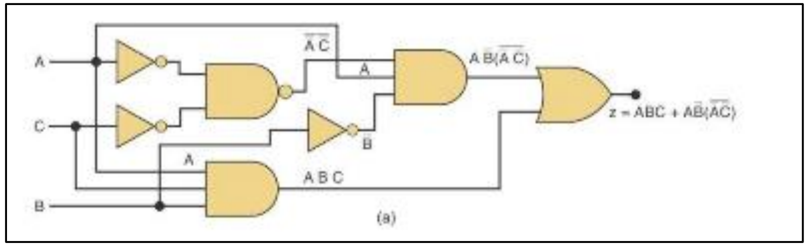
(a)

$$Z = A\,.\,B\,.\,C + A.\,\overline{B}.\,(\overline{\overline{A}.\,\overline{C}}) = A.\,B.\,C + A.\,\overline{B}.\,(\overline{\overline{A}} + \overline{\overline{C}})$$

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

$$= A.\,C.\,(B + \overline{B}) + A.\,\overline{B} = A.\,C.\,1 + A.\,\overline{B} = A.\,(\overline{B} + C)$$

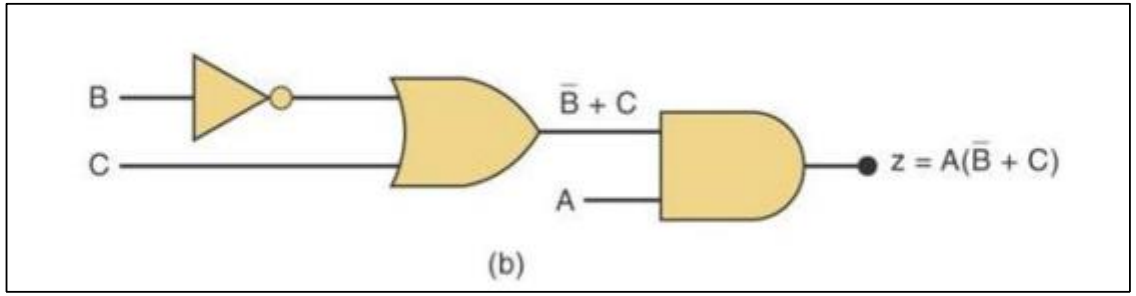
Simplificação algébrica de circuitos lógicos



(a)

(b)

$$
\begin{aligned}
Z &= A.B.C + A.\overline{B}.(\overline{\overline{A}.\overline{C}}) = A.B.C + A.\overline{B}.(\overline{\overline{A}} + \overline{\overline{C}}) \\
&= A.B.C + A.\overline{B}.(A + C) = A.B.C + A.\overline{B} + A.\overline{B}.C \\
&= A.C.(B + \overline{B}) + A.\overline{B} = A.C.1 + A.\overline{B} = A.(\overline{B} + C)
\end{aligned}
$$



(b)

Mapa de Karnaugh

- **Método gráfico usado para simplificar uma equação lógica ou converter uma tabela verdade no seu circuito logico correspondente**

- **Sair do circuito idealizado para o circuito otimizado que você vai realmente construir**

- **A partir da tabela verdade que determina o comportamento, chega-se ao circuito a ser construído que reproduz tabela verdade**

Considerações importantes

- Na construção da tabela, manter a distancia de Hamming de 1, isto é, mudar apenas 1 bit em cada passo (onde as pessoas erram mais;)

- Aprendendo por exemplos

EXEMPLO: Tabela Verdade

A	B	Y
1	1	1
1	0	0
0	1	0
0	0	1

DA TABELA VERDADE PARA MAPA DE KARNAUGH

A	B	Y
1	1	1
1	0	0
0	1	0
0	0	1

A\B	0	1
0	1	
1		1

DA TABELA VERDADE PARA MAPA DE KARNAUGH

A	B	Y
1	1	1
1	0	0
0	1	0
0	0	1

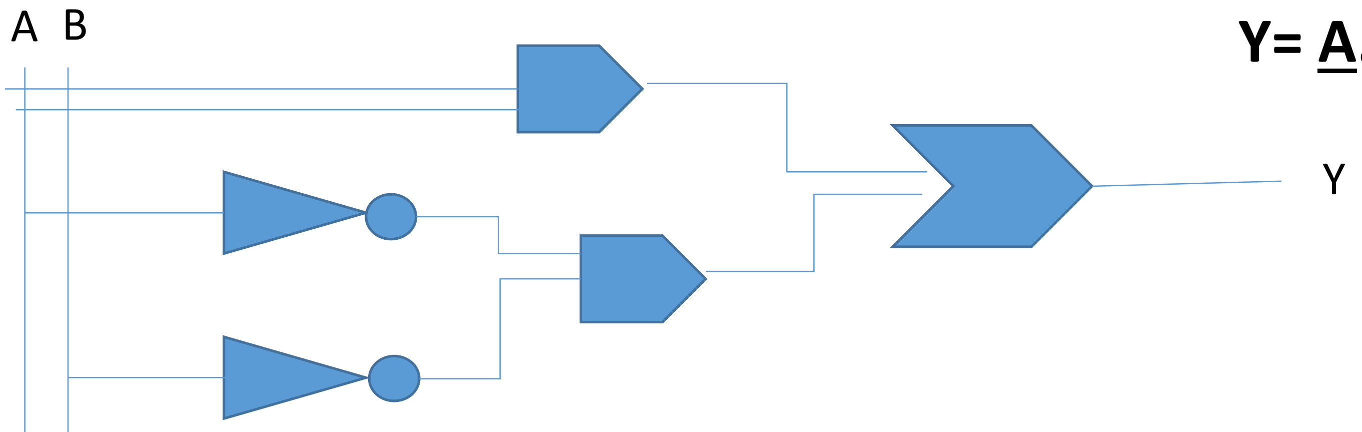
A\B	0	1
0	1	
1		1

$\mathbf{Y = \underline{A}.\underline{B} + A.B}$

DA TABELA VERDADE PARA MAPA DE KARNAUGH

A	B	Y
1	1	1
1	0	0
0	1	0
0	0	1

A\B	0	1
0	1	
1		1



Y= $\underline{A}.\underline{B}$ +A.B

Exemplo: Considere a tabela verdade abaixo

A	B	C	SAIDA
0	0	0	1
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	1

Exemplo: Construa o mapa de Karnaugh

A	B	C	SAIDA
0	0	0	1
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	1

A \BC	"00"	"01"	"11"	"10"
"0"	1	1	1	
"1"			1	1

Exemplo: Grupe (grupos de 2^n)

A	B	C	SAIDA
0	0	0	1
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	1

A \BC	"00"	"01"	"11"	"10"
"0"	1	1	1	
"1"			1	1

Exemplo: quem muda não entra no circuito

A	B	C	SAIDA
0	0	0	1
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	1

A \BC	"00"	"01"	"11"	"10"
"0"	1	1	1	
"1"			1	1

$\underline{A}$ QUER DIZER “NOT A”

Circuito= $\underline{A} . \underline{B} + B .C + A.B$

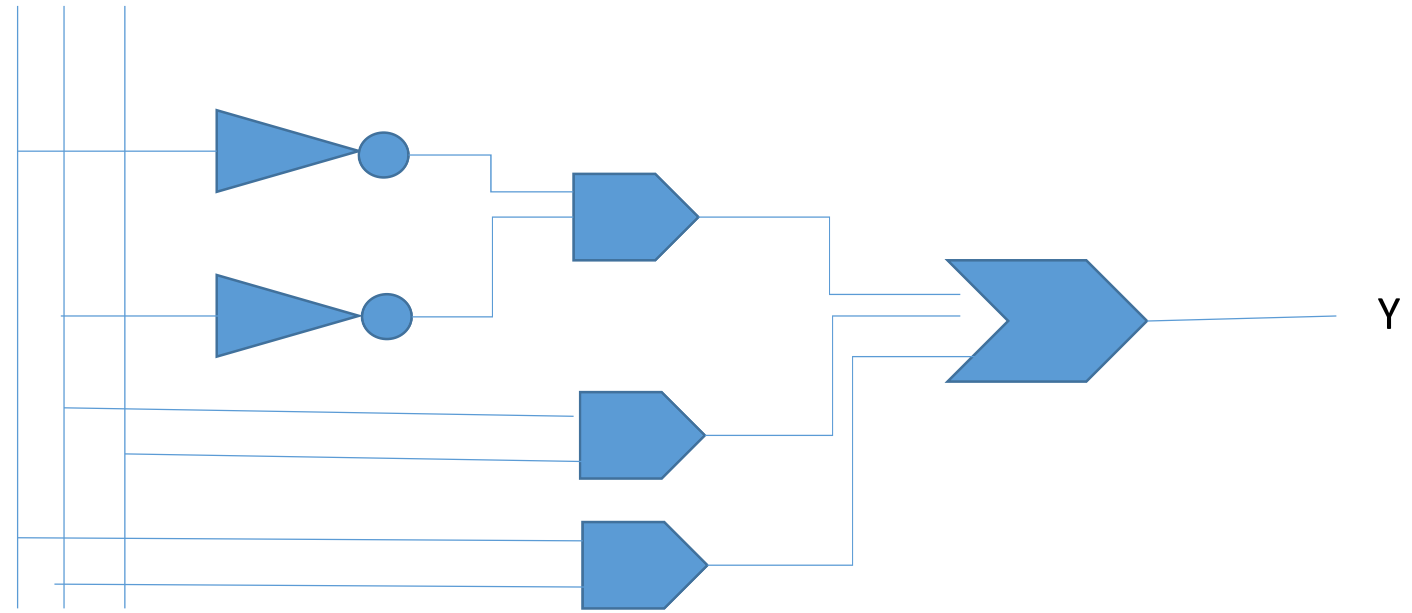
Exemplo: quem muda não entra no circuito

A	B	C	Y
0	0	0	1
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	1

A \BC	"00"	"01"	"11"	"10"
"0"	1	1	1	
"1"			1	1

A B C

$Y = \underline{A} \cdot \underline{B} + B \cdot C + A \cdot B$



EXEMPLO #2: DADA A TABELA VERDADE

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

EXEMPLO #2: DADA A TABELA VERDADE

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

AB\CD	"00"	"01"	"11"	"10"
"00"	1	1		1
"01"	1	1		1
"11"			1	
"10"	1			1

EXEMPLO #2: DADA A TABELA VERDADE

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

AB\CD	"00"	"01"	"11"	"10"
"00"	1	1		1
"01"	1	1		1
"11"			1	
"10"	1			1

$Y=\underline{A}.\underline{C}+\underline{A}.\underline{D}+A.\underline{B}.\underline{D}+A.B.C.D$

EXERCICIO em sala

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

EXERCICIO → mapa de karnaugh

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

A\BC	"00"	"01"	"11"	"10"
0	1	1		1
1				1

EXERCICIO → mapa de karnaugh

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

A\BC	"00"	"01"	"11"	"10"
0	1	1		1
1				1

$Y=\underline{A}.\underline{B} + \underline{A}.\underline{C} + B.\underline{C}$

EXERCICIO em sala

- Use o mapa de Karnaugh para simplificar $y = \bar{C}(\bar{A}\bar{B}\bar{D} + D) + A\bar{B}C + \bar{D}.$

EXERCICIO em sala

- Use o mapa de Karnaugh para simplificar

$$y = \bar{C}(\bar{A}\bar{B}\bar{D} + D) + A\bar{B}C + \bar{D}.$$

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

EXERCICIO em sala

- Use o mapa de Karnaugh para simplificar

$$y = \bar{C}(\bar{A}\bar{B}\bar{D} + D) + A\bar{B}C + \bar{D}.$$

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

AB\CD	"00"	"01"	"11"	"10"
"00"	1	1	0	1
"01"	1	1	0	1
"11"	1	1	0	1
"10"	1	1	1	1

EXERCICIO em sala

- Use o mapa de Karnaugh para simplificar

$$y = \overline{C}(\overline{A}\overline{B}\overline{D} + D) + A\overline{B}C + \overline{D}.$$

AB\CD	"00"	"01"	"11"	"10"
"00"	1	1	0	1
"01"	1	1	0	1
"11"	1	1	0	1
"10"	1	1	1	1

EXERCICIO em sala

$Y = \underline{C} + A.\underline{B} + \underline{D}$

- Use o mapa de Karnaugh para simplificar

$$y = \overline{C}(\overline{A}\,\overline{B}\,\overline{D} + D) + A\overline{B}C + \overline{D}.$$

AB\CD	"00"	"01"	"11"	"10"
"00"	1	1	0	1
"01"	1	1	0	1
"11"	1	1	0	1
"10"	1	1	1	1

$Y = \underline{C} + A.\underline{B} + \underline{D}$

Exercícios para casa

- Determine a expressão mínima para os mapas abaixo:

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

(a)

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

(b)

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

(c)

Exercícios para casa

- Simplifique as expressões usando o mapa de Karnaugh

$$y = \overline{(C + D)} + \bar{A}C\bar{D} + A\bar{B}\bar{C} + \bar{A}\bar{B}CD + AC\bar{D}$$

$$x = AB\,(\overline{\bar{C}D}) + \bar{A}BD + \bar{B}\bar{C}\bar{D}$$