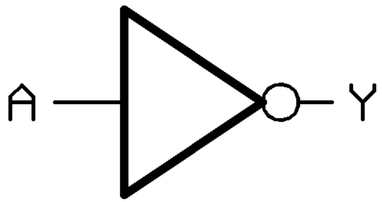
Portas Lógicas Básicas

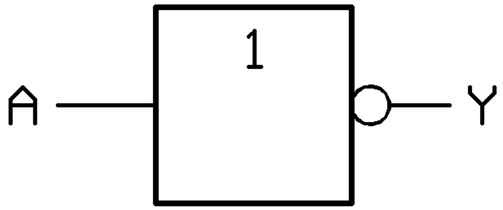
Circuitos Digitais I

Porta NOT (Inversora)

- Símbolo ANSI:



- Símbolo IEEE



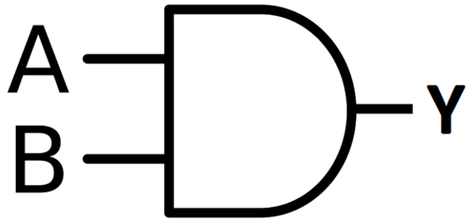
- Tabela Verdade:

Entrada A	Saída Y
0	1
1	0

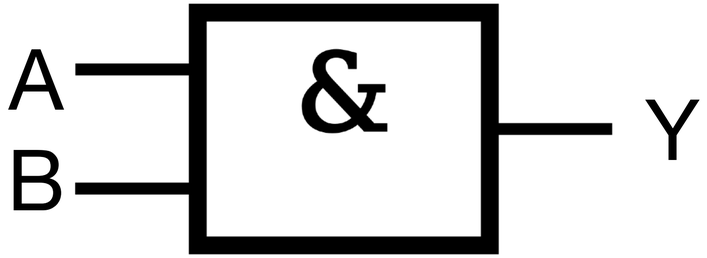
- Equação (booleana): $Y = \overline{A}$

Porta AND

- Símbolo ANSI:



- Símbolo IEEE:



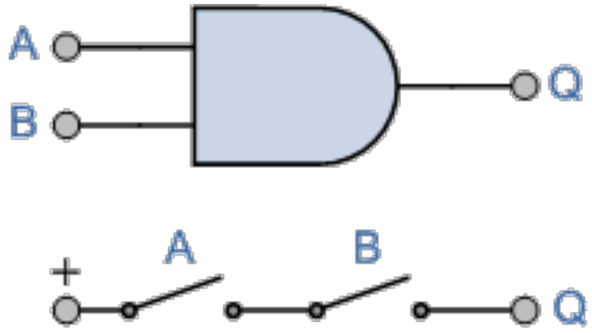
- Tabela verdade

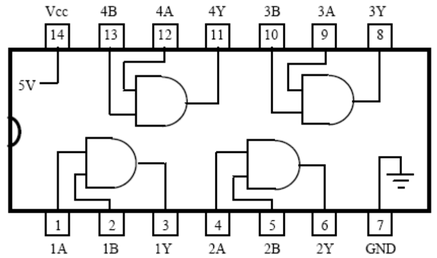
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Obs.: entradas $\geq$ 1 e saídas = 1.

- Equação: $Y = A \cdot B$

- Note:

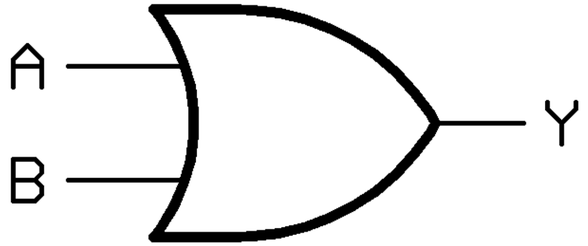




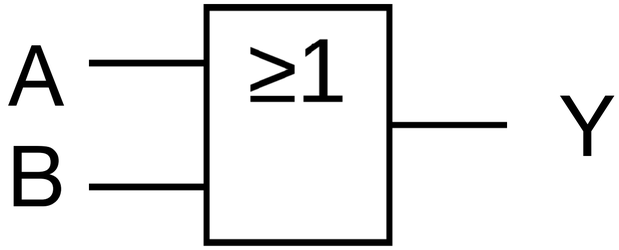
74LS08

Porta OR

- Símbolo ANSI:



- Símbolo IEEE:



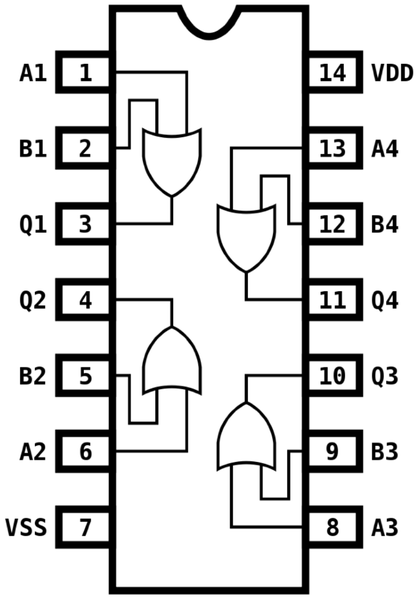
- Tabela verdade

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

Obs.: entradas $\geq$ 1 e saídas = 1.

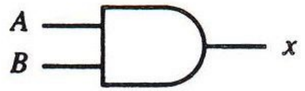
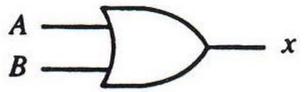
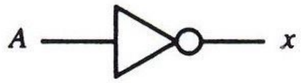
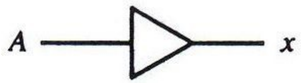
- Equação: $Y = A + B$

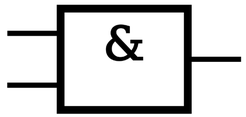
- Note:

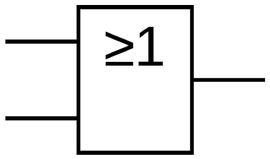


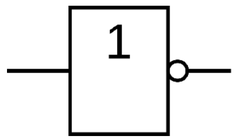
74LS32

Primeiro Resumo

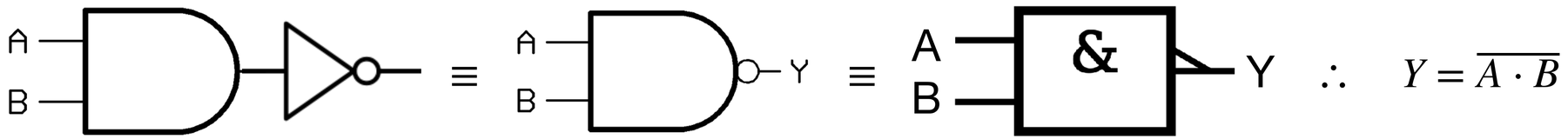
Name	Graphic symbol	Algebraic function	Truth table
AND		$x = A \cdot B$ or $x = AB$	A B | x 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$x = A + B$	A B | x 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$x = A'$	A | x 0 | 1 1 | 0
Buffer		$x = A$	A | x 0 | 0 1 | 1

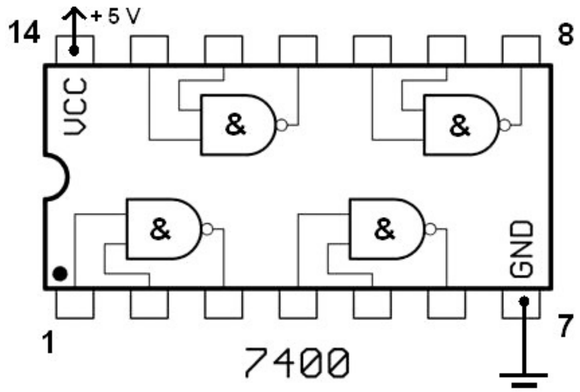


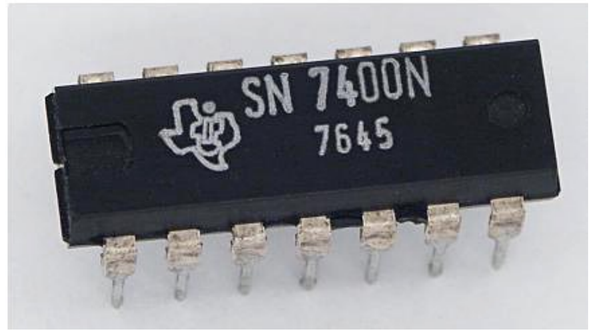


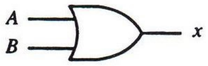
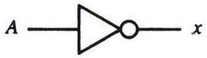
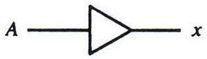


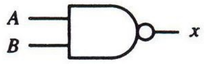
Variações





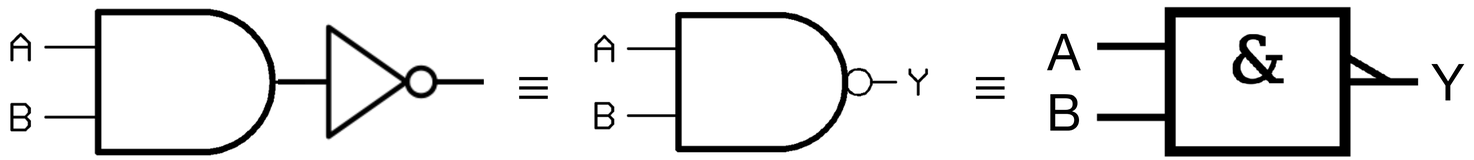


Name	*Graphic symbol*	*Algebraic function*	*Truth table*
AND		$x = A \cdot B$ or $x = AB$	A B | x 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$x = A + B$	A B | x 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$x = A'$	A | x 0 | 1 1 | 0
Buffer		$x = A$	A | x 0 | 0 1 | 1

Name	*Graphic symbol*	*Algebraic function*	*Truth table*
NAND		$x = (AB)'$	A B | x 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$x = (A + B)'$	A B | x 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0

Variações

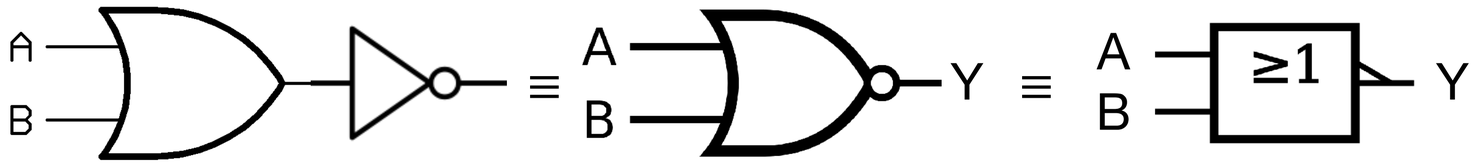
AND + NOT ≡ **NAND** ≡   **Equação NAND:** $\therefore \quad Y = \overline{A \cdot B}$



NAND

A	B	Y	AND
0	0	$Y = \overline{0 \cdot 0} = \overline{0} = 1$	0
0	1	$Y = \overline{0 \cdot 1} = \overline{0} = 1$	0
1	0	$Y = \overline{1 \cdot 0} = \overline{0} = 1$	0
1	1	$Y = \overline{1 \cdot 1} = \overline{1} = 0$	1

OR + NOT ≡ **NOR** ≡   **Equação NOR:** $\therefore \quad Y = \overline{A + B}$

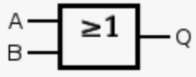


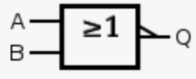
NOR

A	B	Y	OR
0	0	$Y = \overline{0 + 0} = \overline{0} = 1$	0
0	1	$Y = \overline{0 + 1} = \overline{1} = 0$	1
1	0	$Y = \overline{1 + 0} = \overline{1} = 0$	1
1	1	$Y = \overline{1 + 1} = \overline{1} = 0$	1

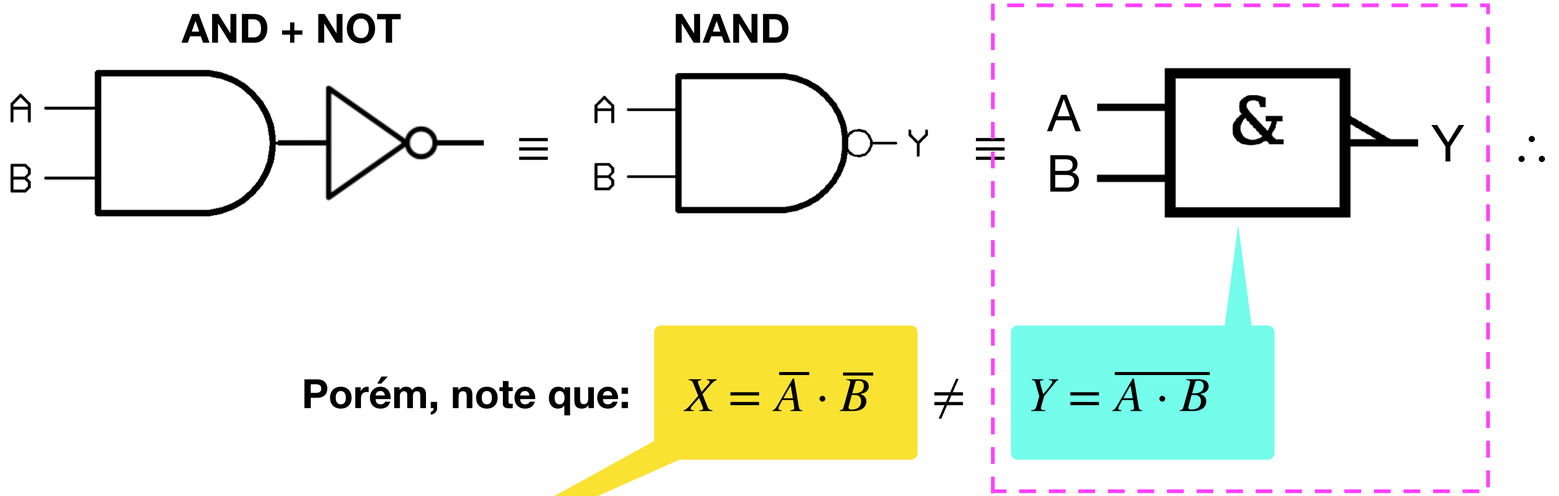
Primeiro Resumo

Portas Lógicas Básicas

Tipo	IEEE Std 91/91a 1991	IEC 60617-12 1997	Álgebra Booleana	Tabela verdade
Buffer			A	INPUT: A; OUTPUT: Q 0 → 0 1 → 1
NOT (inverter)			$\overline{A}$ or $\neg A$	INPUT: A; OUTPUT: Q 0 → 1 1 → 0
AND			$A \cdot B$ or $A \wedge B$	INPUT: A, B; OUTPUT: Q 0 0 → 0 0 1 → 0 1 0 → 0 1 1 → 1
OR			$A + B$ or $A \vee B$	INPUT: A, B; OUTPUT: Q 0 0 → 0 0 1 → 1 1 0 → 1 1 1 → 1

Tipo	IEEE Std 91/91a 1991	IEC 60617-12 1997	Álgebra Booleana	Tabela verdade
NAND			$\overline{A \cdot B}$ or $A \uparrow B$	INPUT: A, B; OUTPUT: Q 0 0 → 1 0 1 → 1 1 0 → 1 1 1 → 0
NOR			$\overline{A + B}$ or $A \downarrow B$	INPUT: A, B; OUTPUT: Q 0 0 → 1 0 1 → 0 1 0 → 0 1 1 → 0

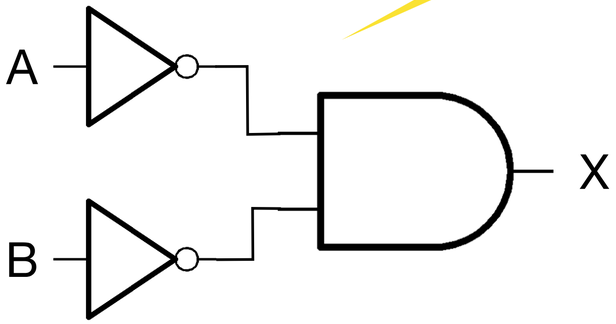
Atenção



Equação NAND:

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

Porém, note que: $X = \overline{A} \cdot \overline{B} \neq Y = \overline{A \cdot B}$



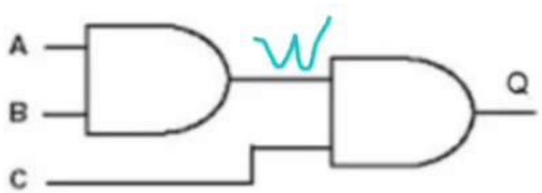
=N OR

A	B	X
0	0	$X = \overline{0} \cdot \overline{0} = 1 \cdot 1 = 1$
0	1	X=/A*/B=/0*/1=1*0=0
1	0	X=/A*/B=/1*/0=0*1=0
1	1	X=/A*/B=/1*/1=0*0=0

NAND:

A	B	Y	AND
0	0	$Y = \overline{0 \cdot 0} = \overline{0} = 1$	0
0	1	$Y = \overline{0 \cdot 1} = \overline{0} = 1$	0
1	0	$Y = \overline{1 \cdot 0} = \overline{0} = 1$	0
1	1	$Y = \overline{1 \cdot 1} = \overline{1} = 0$	1

Análise Circuitos: Exemplos:

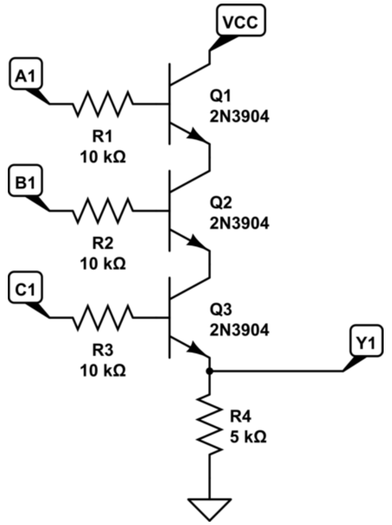


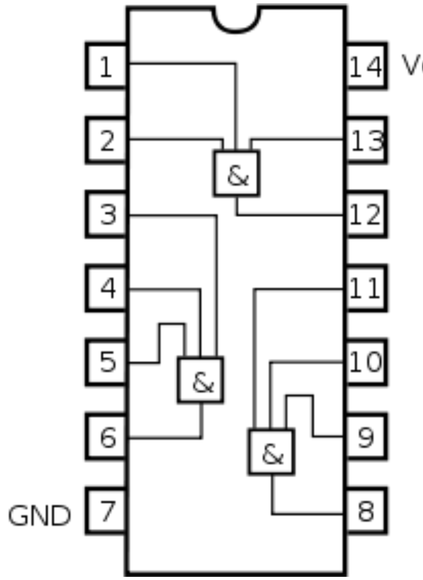
3 input AND				
Input				Output
A	B	C	W	Q
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		0
1	0	0		0
1	0	1		0
1	1	0		0
1	1	1		1

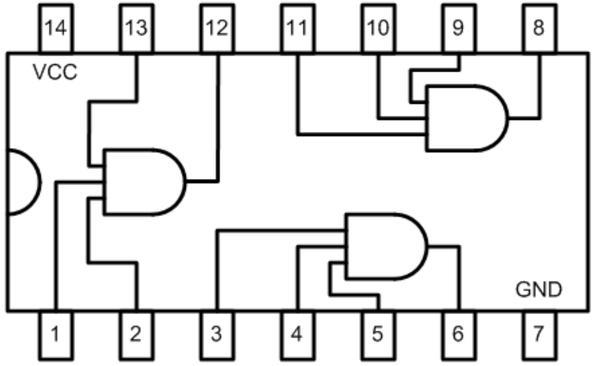
Equação ?

Q=0*B*C=0

Q=1<=1*1*1

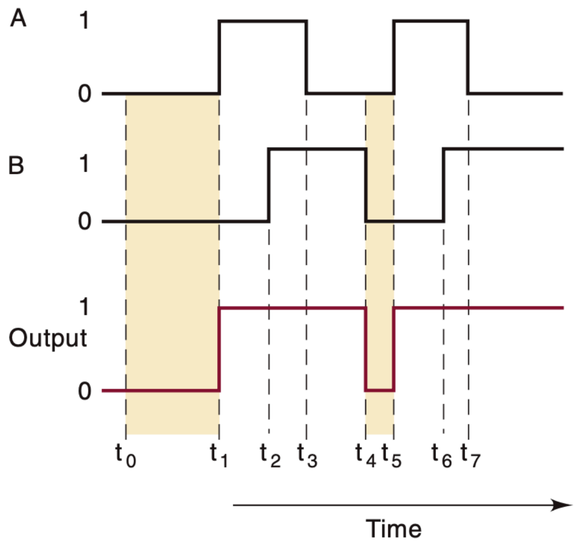




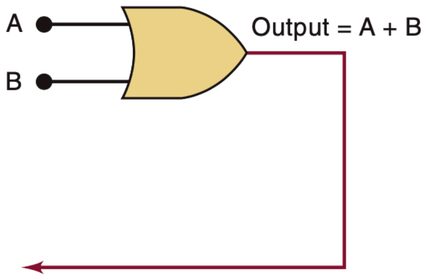


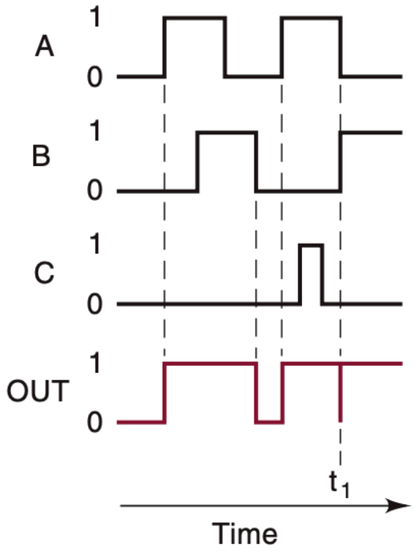
7411 Triple 3 Input AND

Análise Circuitos: Exemplos:

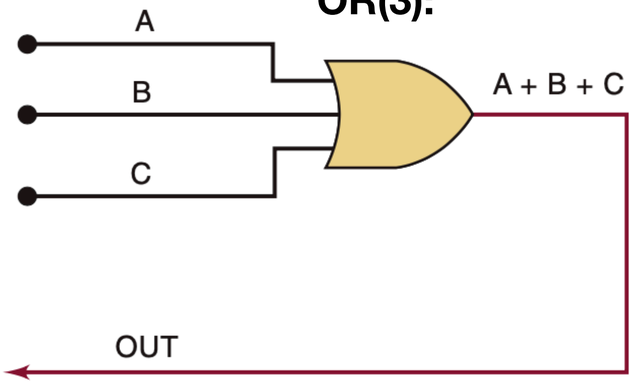


Porta OR(2):





Porta OR(3):



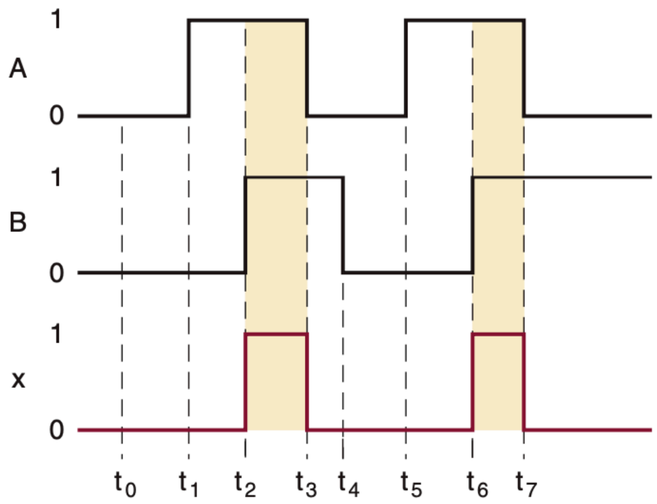
Note: basta uma entrada em nível lógico ALTO para saída comutar para nível lógico ALTO:

Álgebra de Boole:

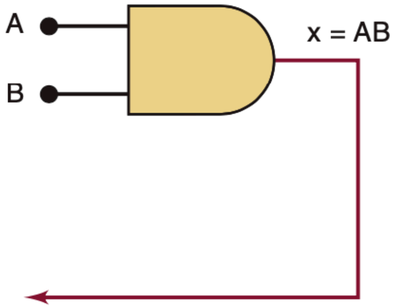
$$x + 0 = x$$

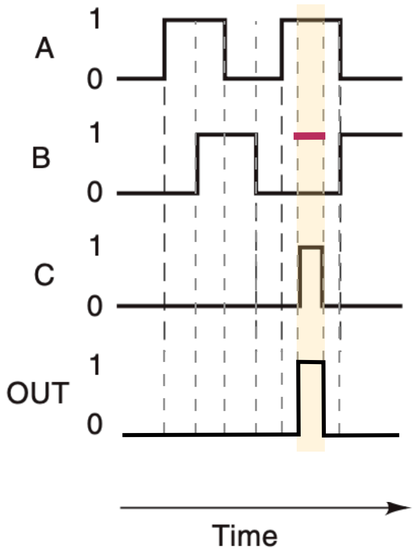
$$x + 1 = 1$$

Análise Circuitos: Exemplos:

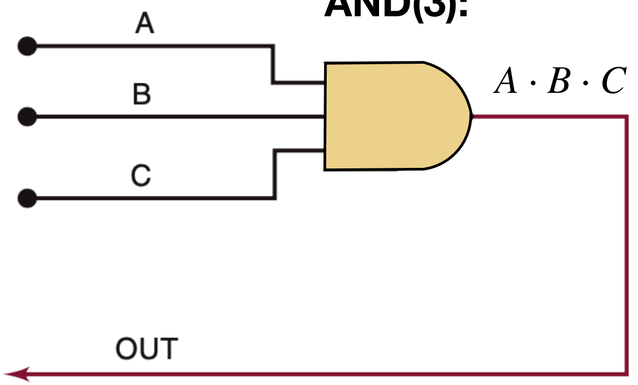


Porta AND(2):





Porta AND(3):



Note: TODAS as entradas devem estar em nível lógico ALTO para saída comutar para nível lógico ALTO:

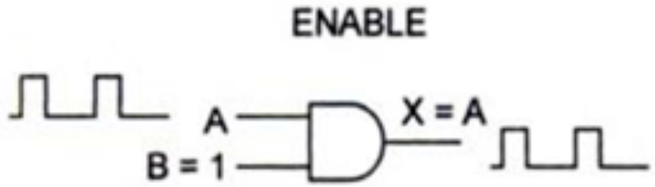
Álgebra de Boole:

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

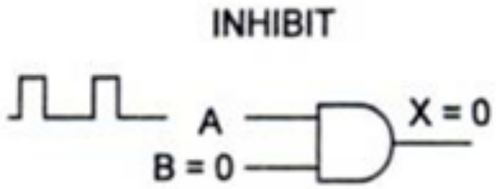
Intro à Álgebra de Boole:

Porta AND:



Note:

$$x \cdot 1 = x$$



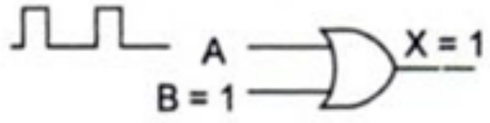
$$x \cdot 0 = 0$$

A porta AND pode funcionar como uma “chave" com Entrada de “Enable"

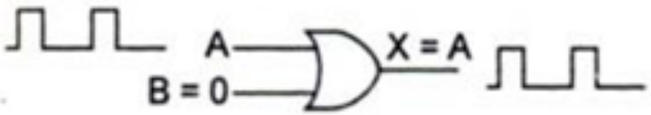
Porta OR:

Note:

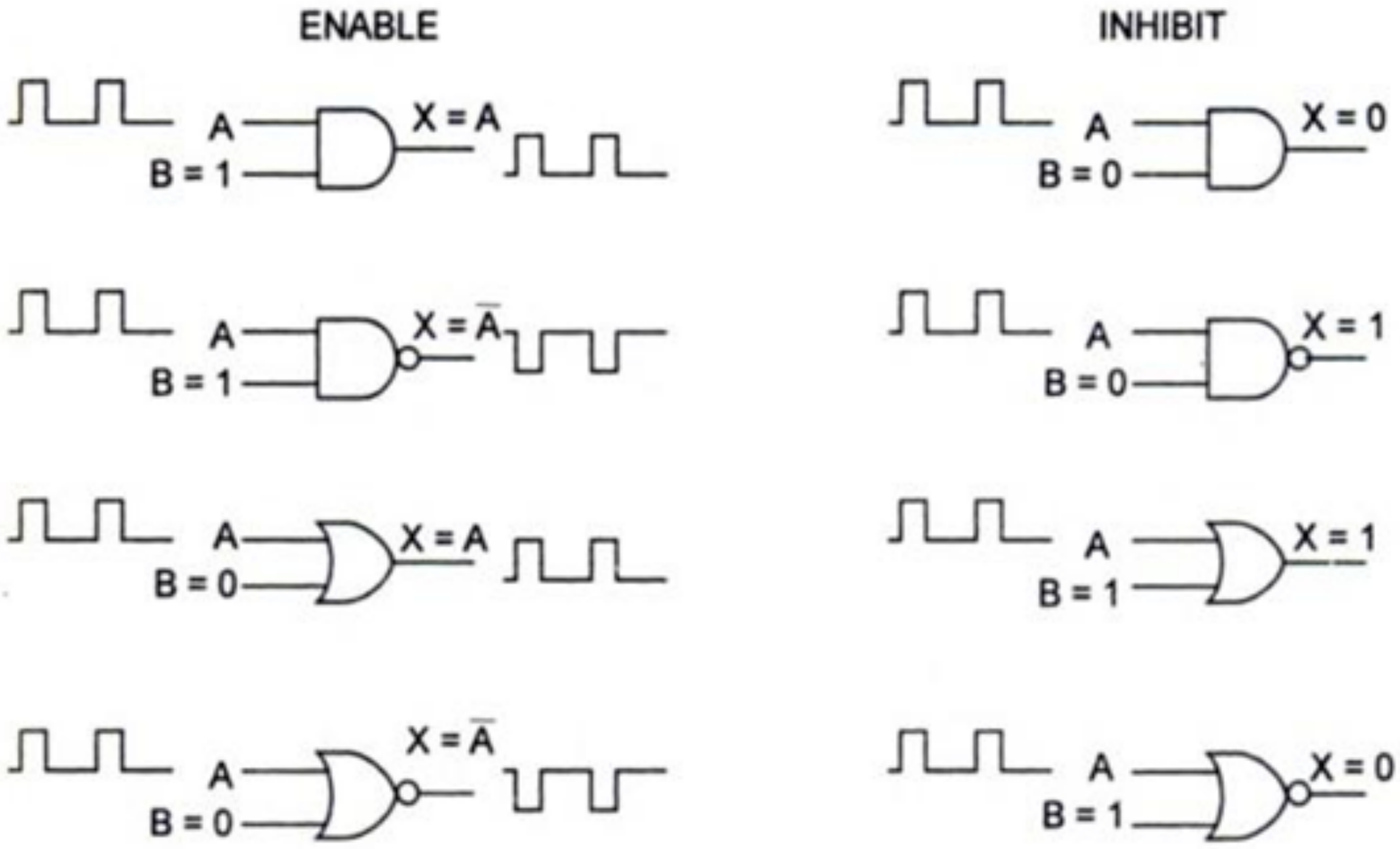
$$x + 1 = 1$$



$$x + 0 = x$$



Variação



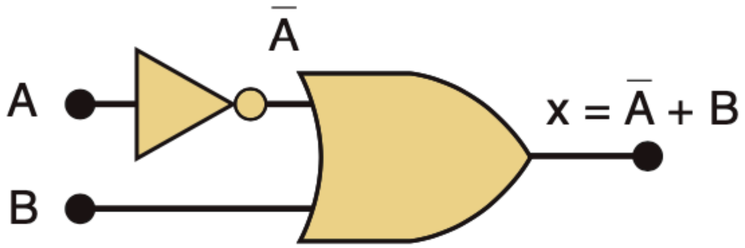
Porta AND:
Note:

$$x \cdot 0 = 0$$
$$x \cdot 1 = x$$

Porta OR:
Note:

$$x + 0 = x$$
$$x + 1 = 1$$

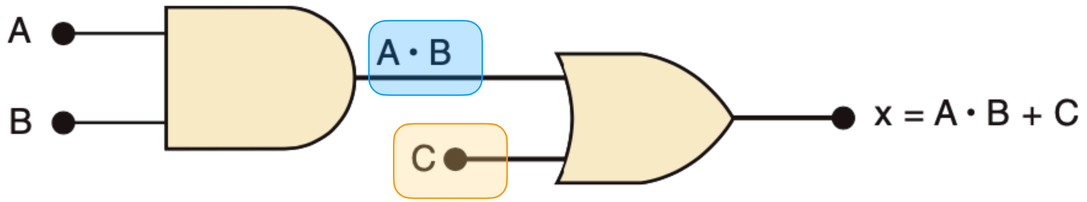
Análise Circuitos: Exemplos:



(a)

(b)

Análise Circuitos: Exemplos:

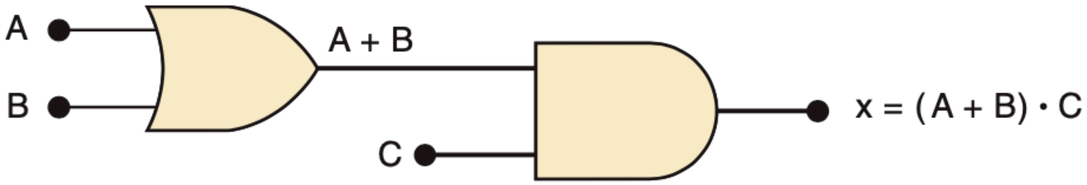


(a)

Ref	A	B	C	A · B	x
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	0
3	0	1	1	0	1
4	1	0	0	0	0
5	1	0	1	0	1
6	1	1	0	1	1
7	1	1	1	1	1

Porta OR: Basta uma entrada = “1”, Saída = “1"

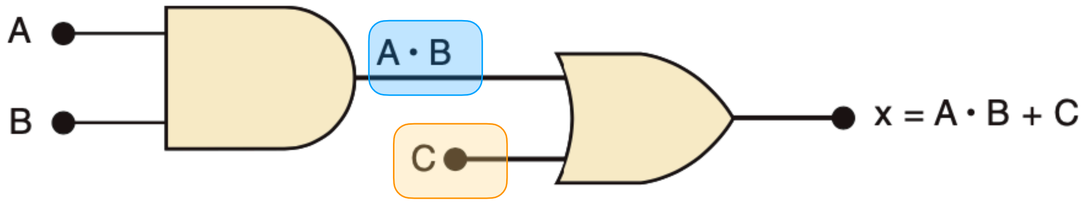
Saída da AND = “1" apenas Quando TODAS as entradas = “1".



(b)

Ref	A	B	C	A + B	x
0	0	0	0		
1	0	0	1		
2	0	1	0		
3	0	1	1		
4	1	0	0		
5	1	0	1		
6	1	1	0		
7	1	1	1		

Análise Circuitos: Exemplos:

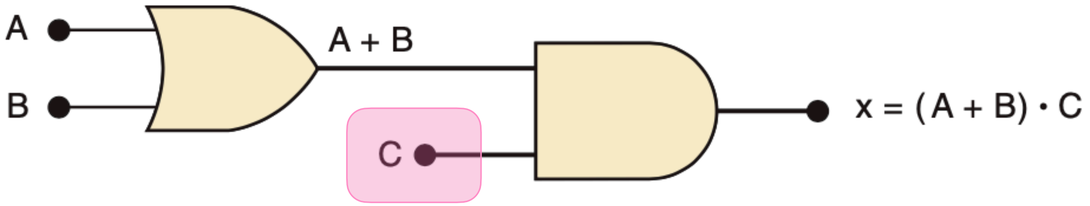


(a)

Ref	A	B	C	A · B	x
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	0
3	0	1	1	0	1
4	1	0	0	0	0
5	1	0	1	0	1
6	1	1	0	1	1
7	1	1	1	1	1

Porta OR:
Basta uma entrada = “1”,
Saída = “1"

Saída da AND = “1" apenas Quando TODAS as entradas = “1".



(b)

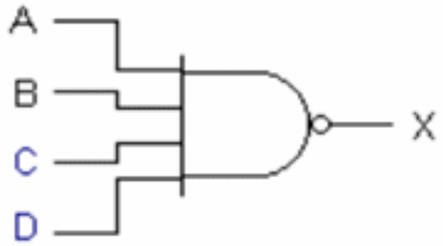
Ref	A	B	C	A + B	x
0	0	0	0		0
1	0	0	1		0
2	0	1	0		0
3	0	1	1		1
4	1	0	0		0
5	1	0	1		1
6	1	1	0		0
7	1	1	1		1

Porta AND:
Basta uma entrada = “0”,
Saída = “0”

Porta OR:
Basta uma entrada = “1”,
Saída = “1"

Exemplo:

- Que porta é esta?



$$X = \overline{ABCD}$$

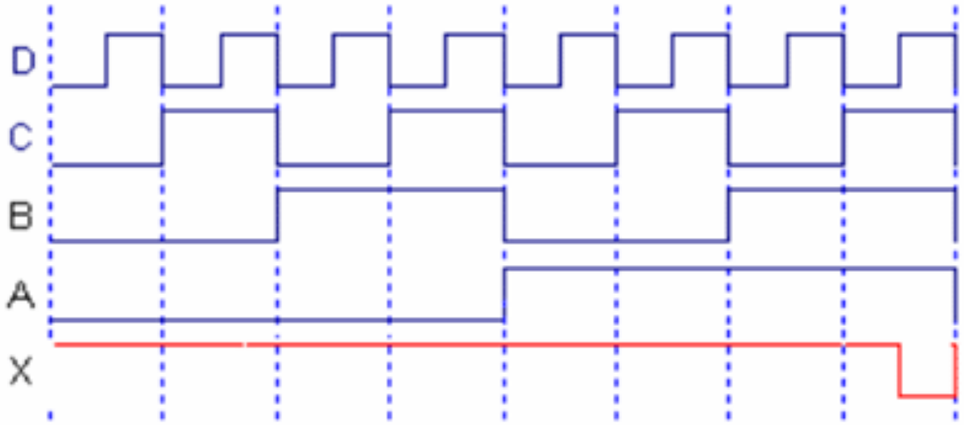
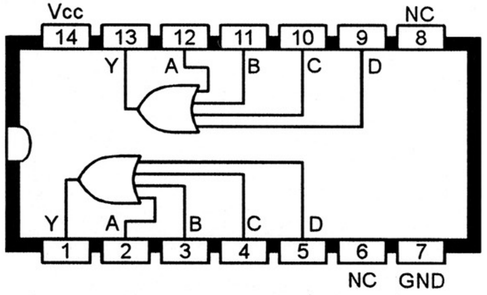


Diagrama de formas de onda

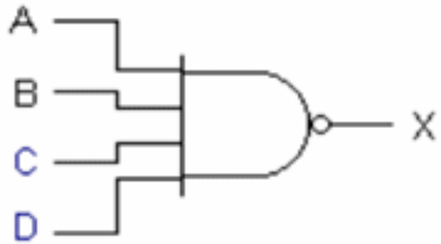
A	B	C	D	X
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0



Exemplo:

NAND(4)

- Que porta é esta?



$X = \overline{ABCD}$

Note:

$x \cdot 0 = 0$

$x \cdot 1 = x$

A	B	C	D	X
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

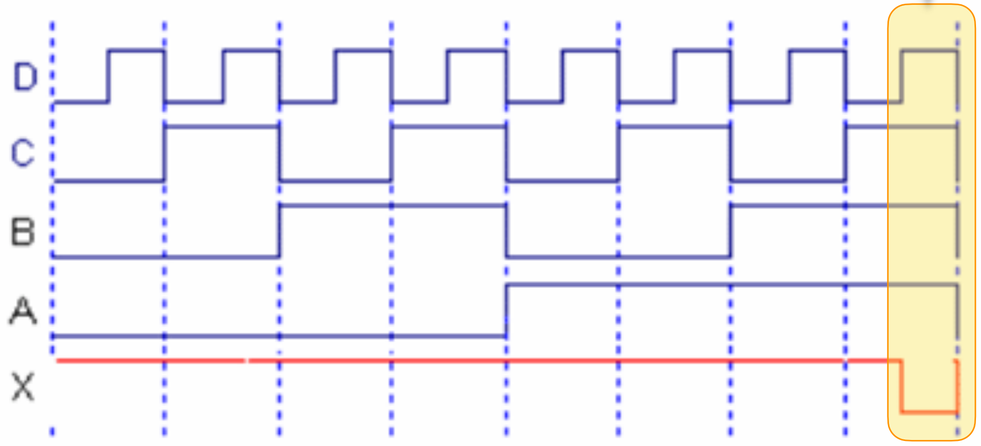
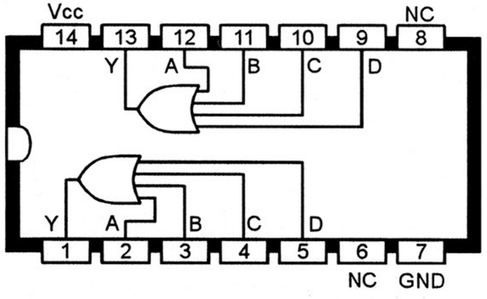


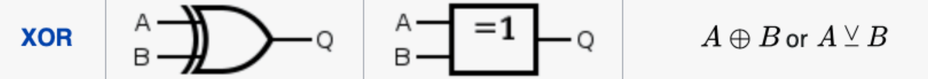
Diagrama de formas de onda



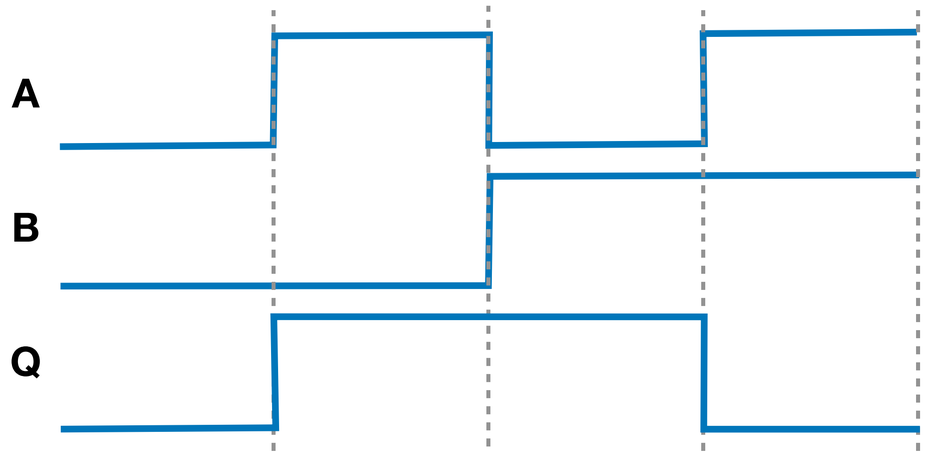
Últimas portas: XOR e NXOR

- XOR: eXclusive OR:

XOR				INPUT		OUTPUT
				A	B	Q
				0	0	0
				0	1	1
				1	0	1
				1	1	0



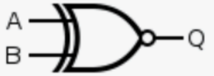
$A \oplus B$ or $A \veebar B$

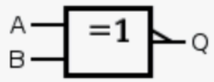


Detector de Desigualdades

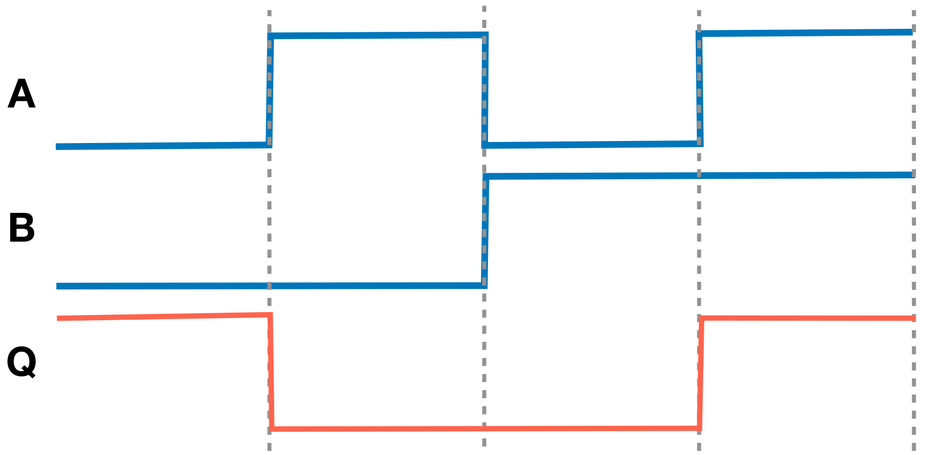
- NXOR: Negative eXclusive OR:

XNOR				INPUT		OUTPUT
				A	B	Q
				0	0	1
				0	1	0
				1	0	0
				1	1	1



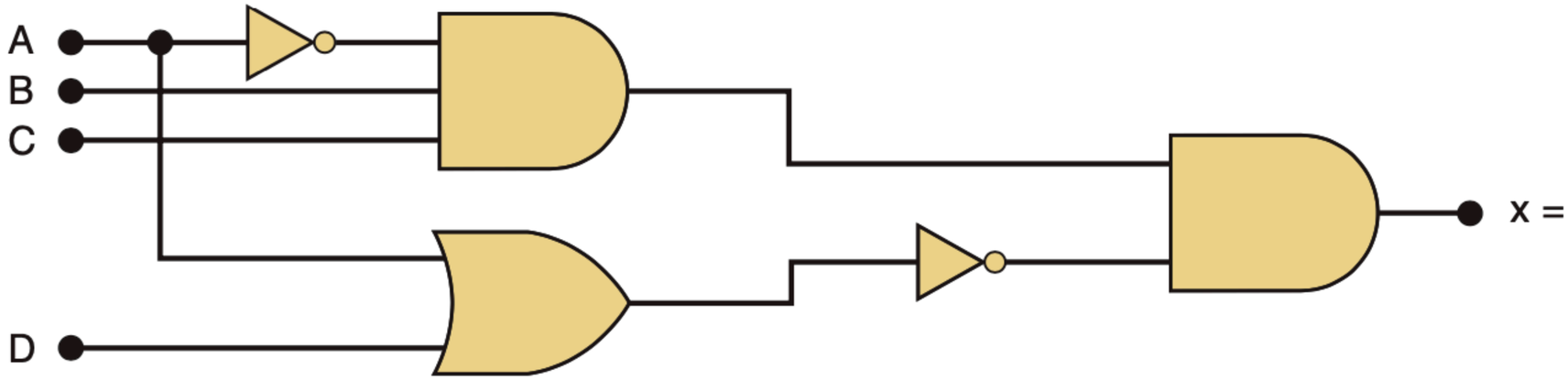


$\overline{A \oplus B}$ or $A \odot B$

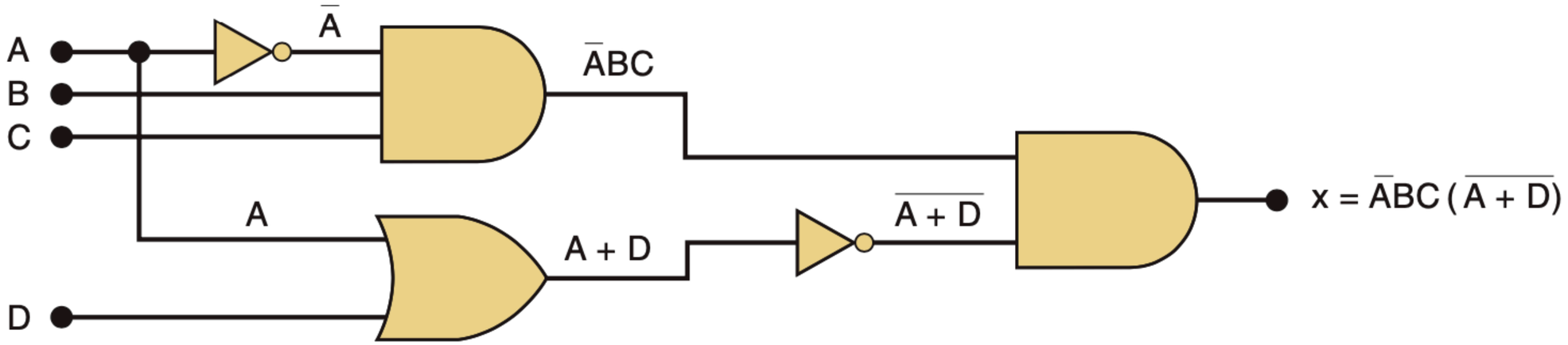


Detector de Igualdades

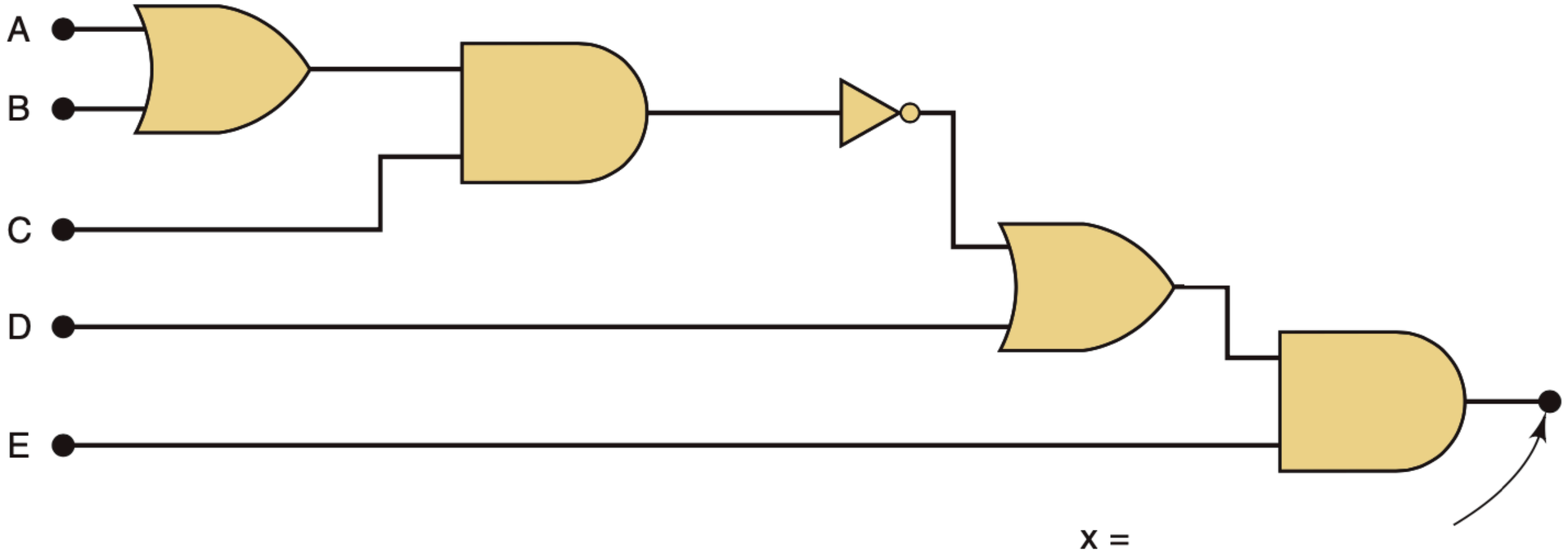
Deduzindo equações:



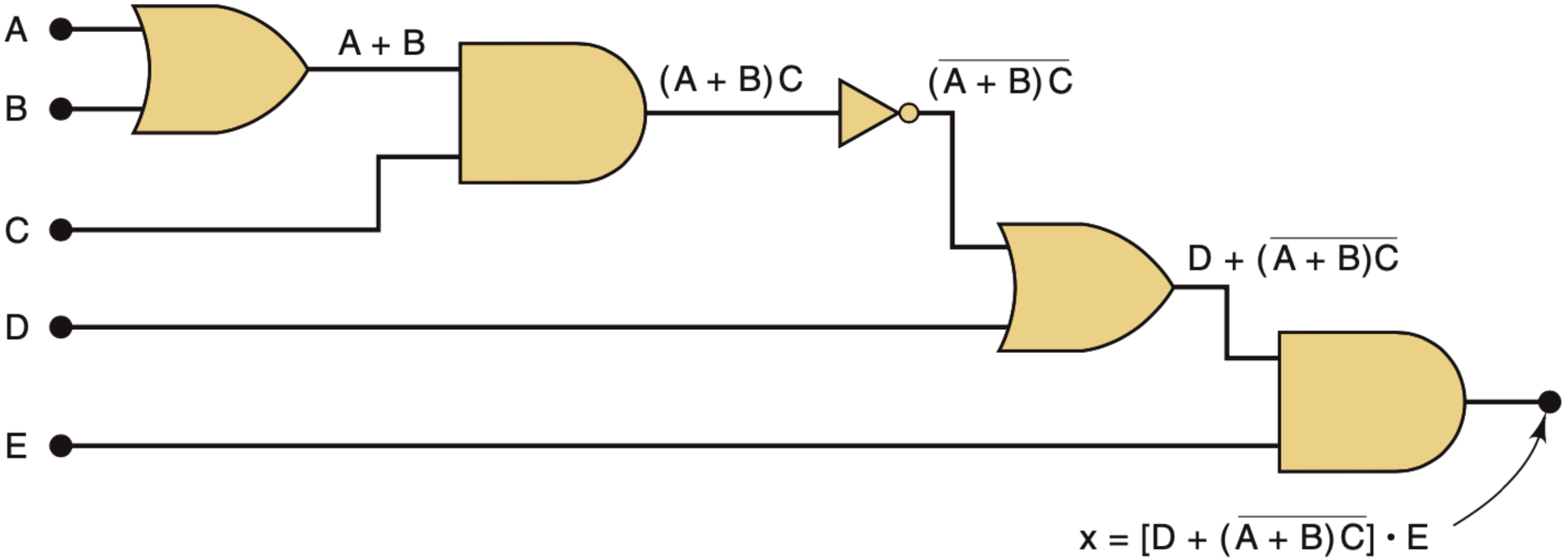
Deduzindo equações:



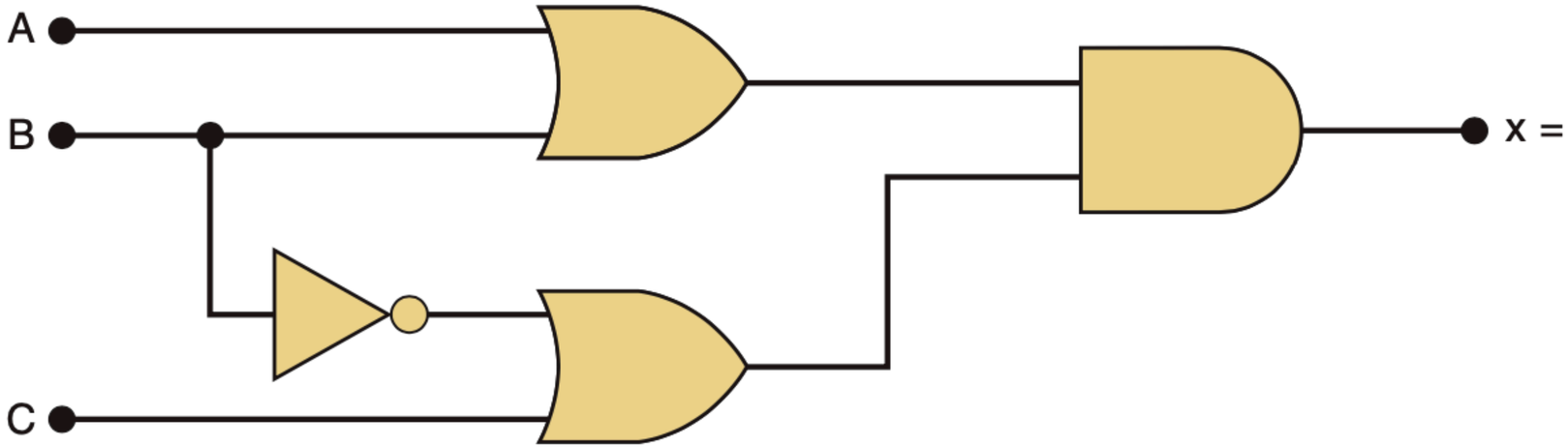
Deduzindo equações:



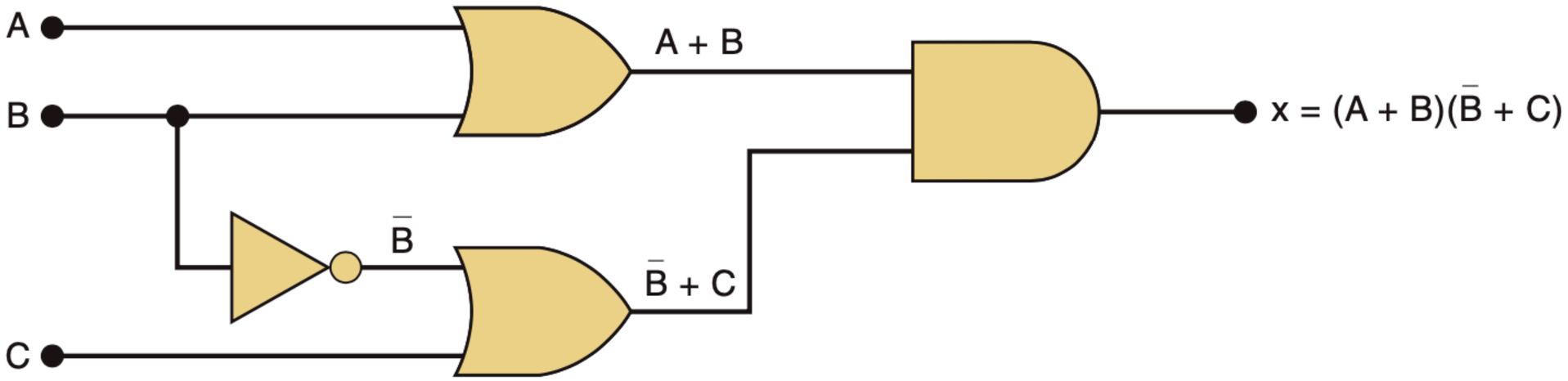
Deduzindo equações:



Deduzindo equações:



Deduzindo equações:

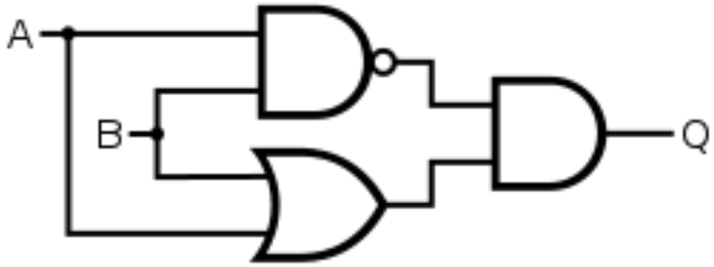


Desenhando circuitos:

- a) $x = \overline{A}\,BC\,\overline{(A + D)}$
- b) $y = AC + B\,\overline{C} + \overline{\overline{A}\,BC}$
- c) $z = \left[D + \overline{(A + B)}\,C\right]\,E$

Deduzindo expressões e tabelas verdade

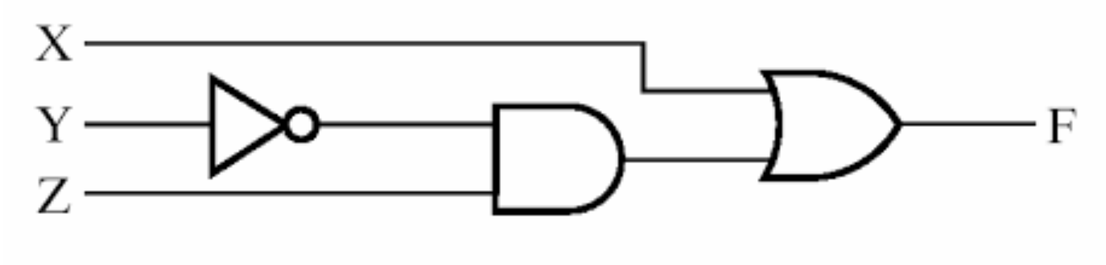
- Quando Q comuta para nível lógico ALTO?



$$Q = (A + B) \cdot (\bar{A} + \bar{B})$$

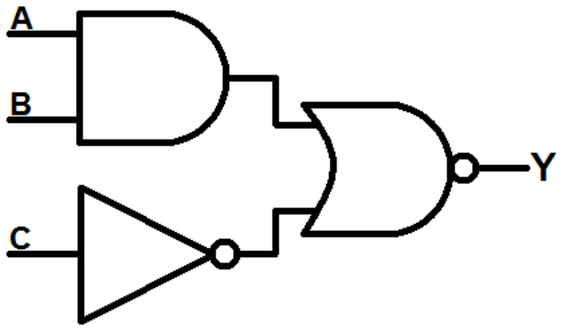
Deduzindo expressões e tabelas verdade

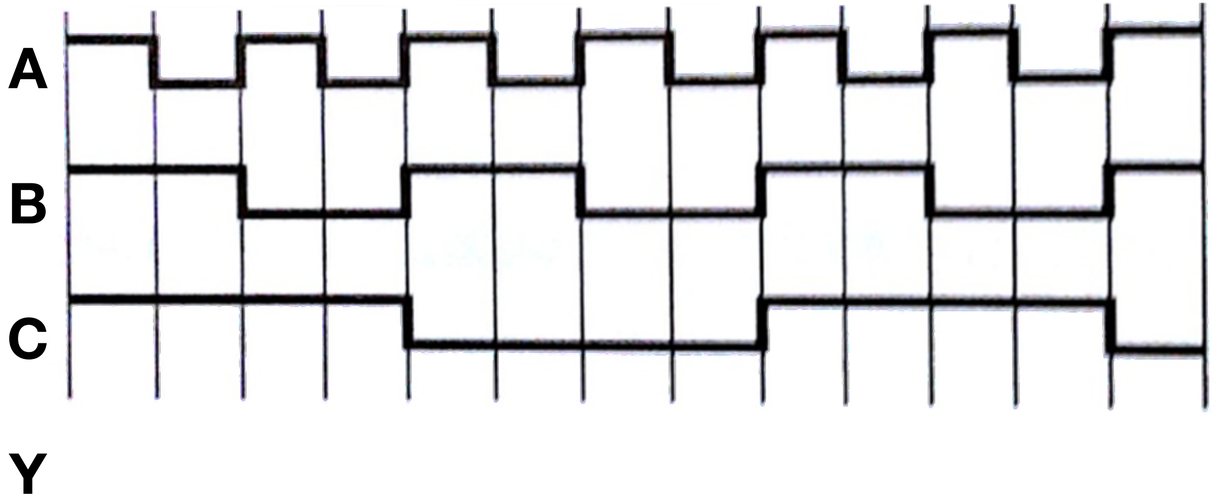
- Quando F comuta para nível lógico ALTO?



Deduzindo expressões e tabelas verdade

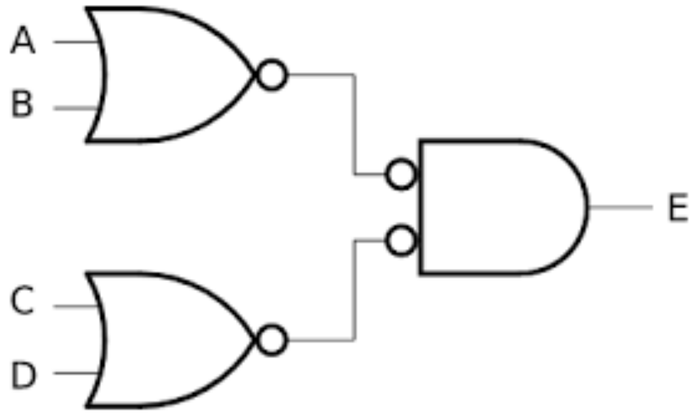
- Quando Y comuta para nível lógico ALTO?

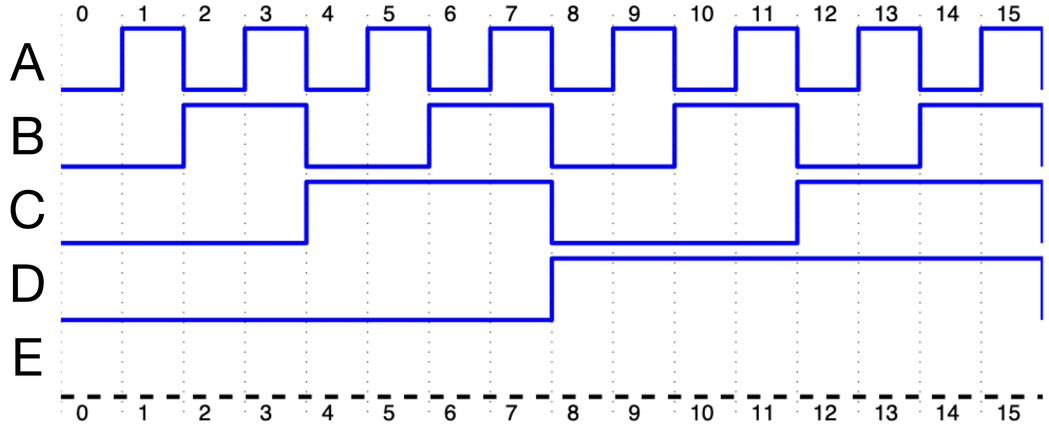




Deduzindo expressões e tabelas verdade

- Complete a forma de onda na saída E (deduza a tabela verdade para E):





Deduzindo expressões e tabelas verdade

- Quando Q comuta para nível lógico ALTO?