

# Funções e Portas Lógicas

professor  
MARCOS MOECKE

# Tipos de lógica

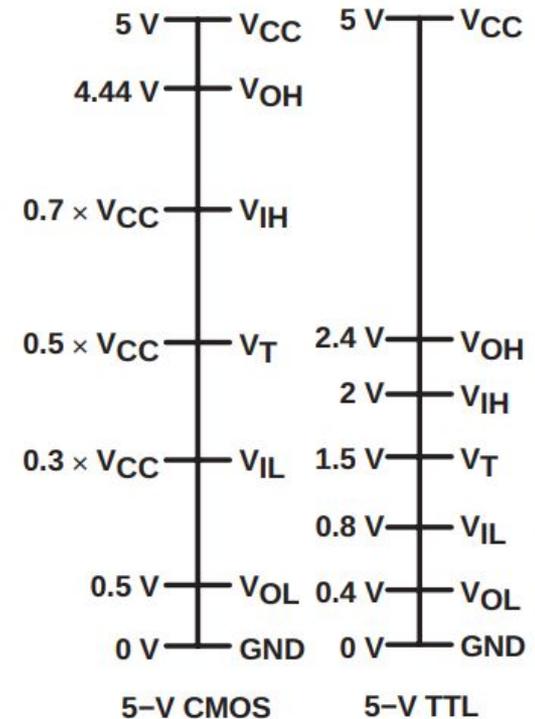
A lógica pode ser classificadas em dois tipos principais:

- **Lógica Positiva:** o valor lógico “0” é representado por uma tensão baixa (geralmente próxima do valor de terra), e o “1” é por uma tensão alta (em torno da tensão de alimentação,  $V_{cc}$ ). Ou seja, a presença de tensão representa um estado lógico verdadeiro (1), enquanto a ausência de tensão representa um estado lógico falso (0). Isso é o que normalmente se encontra em sistemas digitais modernos
- **Lógica Negativa:** o valor lógico “0” é representado por uma tensão alta, e o “1” por uma tensão baixa (que pode ser negativa). Um dos raros circuitos que utiliza a lógica negativa é a interface [RS-232](#), na qual tensões negativas representam “1” e tensões positivas representam “0”

# Tensões utilizadas e níveis lógicos

Dependendo da família lógica, diferentes tensão representam os níveis lógicos. Alguns exemplos:

- TTL (*Transistor-Transistor Logic*)  
Lógica 0: uma tensão entre 0V e 0.8V.  
Lógica 1: uma tensão entre 2V e 5V.
- CMOS (*Complementary Semiconductor*):  
Lógica 0: uma tensão entre 0V e 1/3 de  $V_{CC}$ .  
Lógica 1: uma tensão entre 2/3 de  $V_{CC}$  e  $V_{CC}$ .



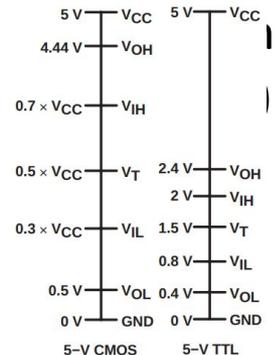
GND - terra do

$V_{CC}$  - tensão de alimentação positiva do circuito

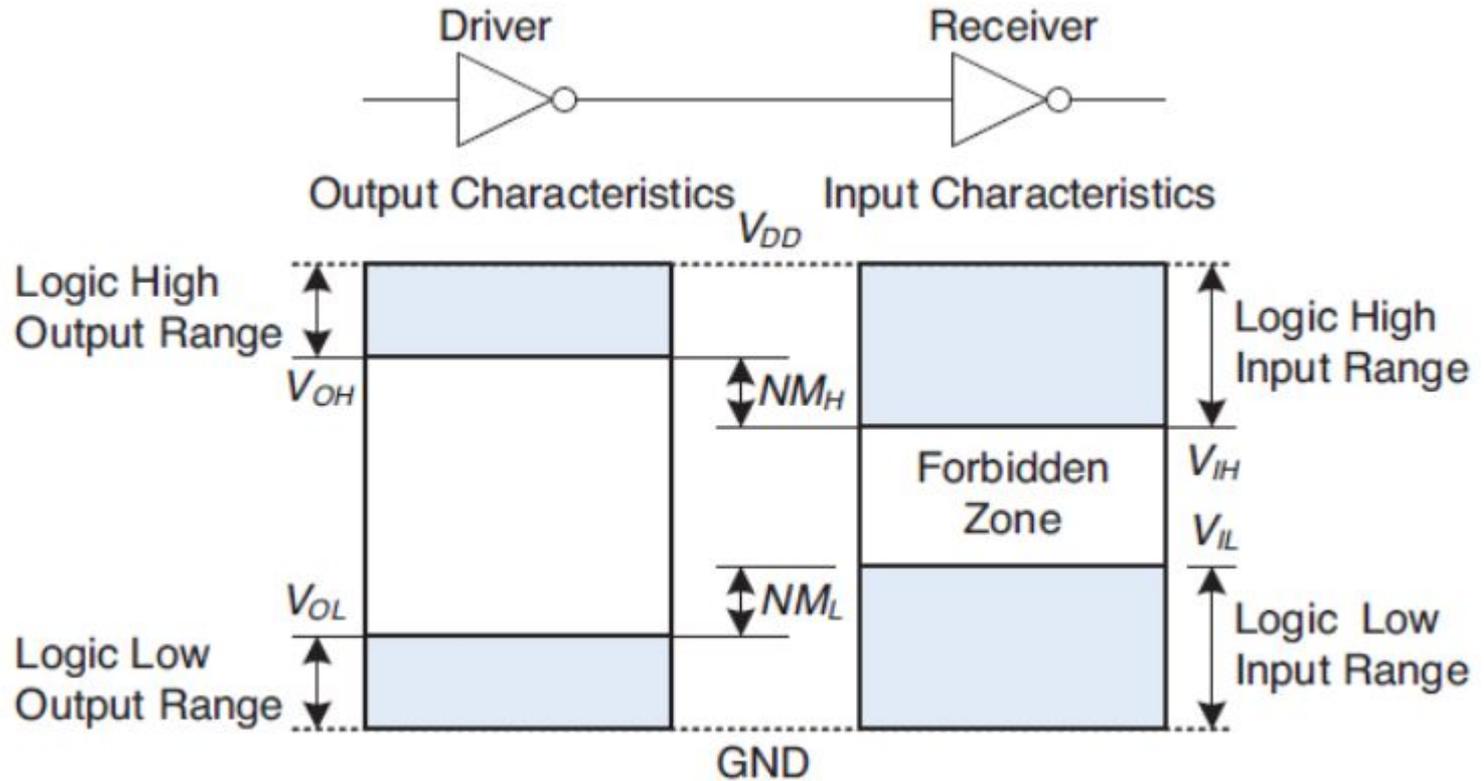
# Limites das tensões utilizadas e níveis lógicos

Note que os limites de tensão que garantem a interpretação correta dos sinais lógicos “0” e “1” em um dispositivo digital muda entre entrada e saída, visando tolerar uma margem de ruído no sinal.

- **VOL (*Output Low Voltage*):** é a tensão máxima na saída de um dispositivo digital quando este está em estado lógico 0 (baixo).
- **VIL (*Input Low Voltage*):** é a tensão máxima que um dispositivo digital interpreta como nível lógico 0 (baixo) em sua entrada.
- **VOH (*Output High Voltage*):** é a tensão mínima na saída de um dispositivo digital quando este está em estado lógico 1 (alto).
- **VIH (*Input High Voltage*):** é a tensão mínima que um dispositivo digital interpreta como nível lógico 1 (alto) em sua entrada.

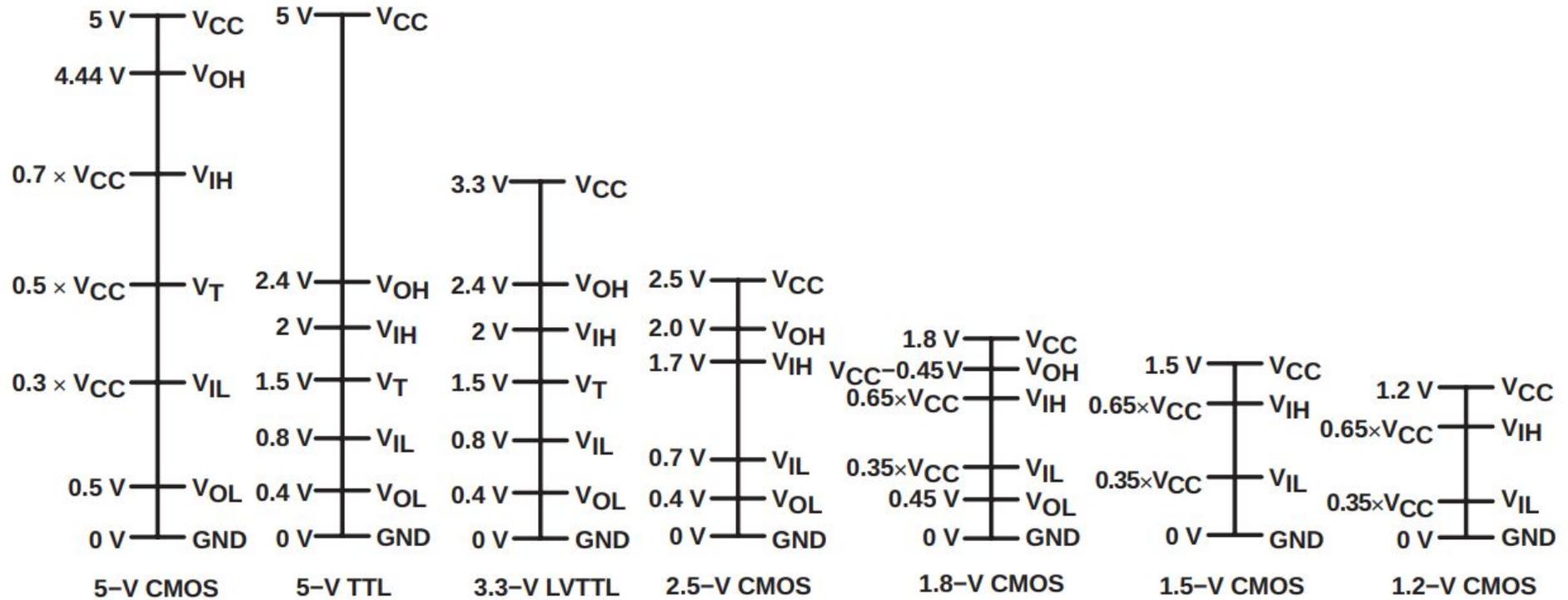


# Margem de ruído



$$NML = V_{IL} - V_{OL}$$
$$NMH = V_{OH} - V_{IH}$$

# Outras famílias lógicas



# Exercício

Determine a margem de ruído das famílias lógicas CMOS e TTL mostradas na figura anterior.

- 5V CMOS       $NML =$  \_\_\_\_\_       $NMH =$  \_\_\_\_\_
- 5V TTL       $NML =$  \_\_\_\_\_       $NMH =$  \_\_\_\_\_
- 3.3V TTL       $NML =$  \_\_\_\_\_       $NMH =$  \_\_\_\_\_
- 2.5V CMOS       $NML =$  \_\_\_\_\_       $NMH =$  \_\_\_\_\_
- 1.8V CMOS       $NML =$  \_\_\_\_\_       $NMH =$  \_\_\_\_\_

# Funções lógicas (portas lógicas)

São 4 funções

- Inversora (NÃO / NOT)
- E / AND
- OU / OR
- XOU / XOR (OU exclusivo)

Pela associação da função inversora a cada função pode ser obter ainda:

- NÃO E / NAND
- NÃO OU / NOR
- NÃO XOU / XNOR (coincidência)

# Tabela-verdade, símbolo, operador lógico

*O funcionamento de cada uma dessas funções pode ser descrito pelas tabelas-verdade.*

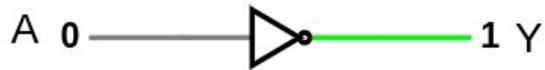
- Na tabela verdade, é feita a descrição do valor de saída de uma porta, para todas as possíveis combinações de entrada.
- O símbolo é uma notação que se usa nos diagramas esquemáticos para representar as portas lógicas
- O operador é um símbolo matemático utilizado para representar as funções lógicas.

*Nota: Utilizamos o simulador Falstad para verificar o funcionamento das portas.*

# Resumo 1 (diretas)

## Porta INVERSORA

$$Y = \bar{A}, Y = A'$$



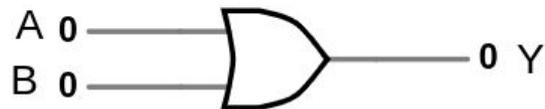
## Porta E / AND

$$Y = A.B, Y = AB$$



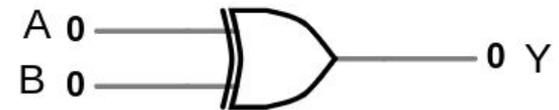
## Porta OU / OR

$$Y = A + B$$



## Porta XOU / XOR

$$Y = A \oplus B$$



# Resumo 2 (invertidas)

Porta **BUFFER**

$$Y = A$$



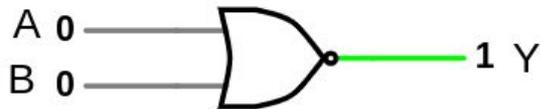
Porta **NÃO E / NAND**

$$Y = \overline{A \cdot B}, \quad Y = \overline{AB}$$



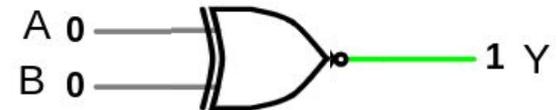
Porta **NÃO OU / NOR**

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



Porta **XNOU / XNOR**

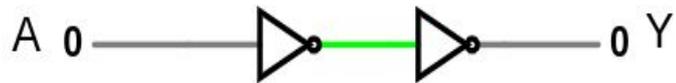
$$Y = \overline{A \oplus B}$$



# Resumo 3 (equivalentes)

Porta **BUFFER**

$$Y = A$$



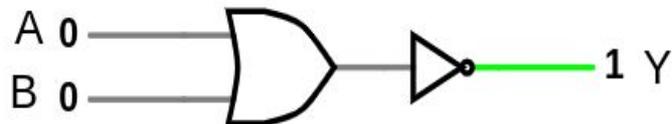
Porta **NÃO E / NAND**

$$Y = \overline{A \cdot B}, \quad Y = \overline{AB}$$



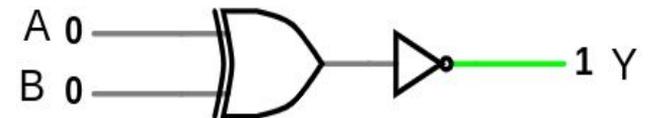
Porta **NÃO OU / NOR**

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



Porta **XNOU / XNOR**

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



# Tabela-verdade

Organização da tabela-verdade com duas entradas A e B e saída Y

valor	A	B	Y
0	0	0	
1	0	1	
2	1	0	
3	1	1	

Organização da tabela-verdade com três entradas A, B e C e saída Y

valor	A	B	C	Y
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	

# Tabela-verdade

AND			
valor	A	B	Y
0	0	0	
1	0	1	
2	1	0	
3	1	1	

NAND			
valor	A	B	Y
0	0	0	
1	0	1	
2	1	0	
3	1	1	

OR			
valor	A	B	Y
0	0	0	
1	0	1	
2	1	0	
3	1	1	

NOR			
valor	A	B	Y
0	0	0	
1	0	1	
2	1	0	
3	1	1	

XOR			
valor	A	B	Y
0	0	0	
1	0	1	
2	1	0	
3	1	1	

XNOR			
valor	A	B	Y
0	0	0	
1	0	1	
2	1	0	
3	1	1	

# Tabela-verdade

	AND		
valor	A	B	Y
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

	NAND		
valor	A	B	Y
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

	OR		
valor	A	B	Y
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

	NOR		
valor	A	B	Y
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0

	XOR		
valor	A	B	Y
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

	XNOR		
valor	A	B	Y
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

# Portas 3 entradas

Porta **NÃO E / NAND**

$$Y = \overline{A.B.C}$$



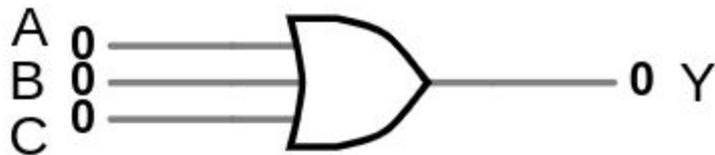
Porta **E / AND**

$$Y = A.B.C$$



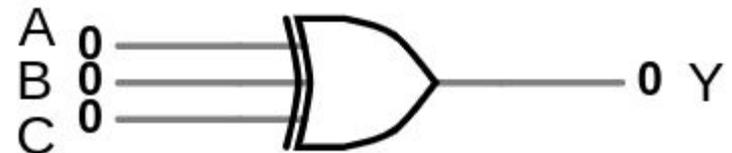
Porta **OU / OR**

$$Y = A + B + C$$



Porta **XOU / XOR**

$$Y = A \oplus B \oplus C$$



# Tabela-verdade

NAND				
valor	A	B	C	Y
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	

AND				
valor	A	B	C	Y
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	

OR				
valor	A	B	C	Y
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	

XOR				
valor	A	B	C	Y
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	

# Tabela-verdade

NAND				
valor	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

*NAND é 0 quando todas entradas são 1*

OR				
valor	A	B	C	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

*OR é 0 quando todas entradas são 0*

AND				
valor	A	B	C	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

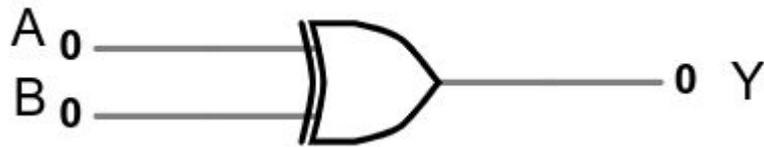
*AND é 1 quando todas entradas são 1*

XOR				
valor	A	B	C	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

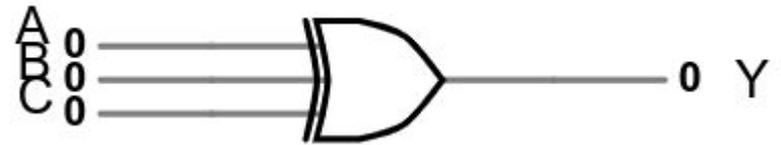
*XOR é 1 quando tem um número ímpar de entradas em 1*

# Portas XOR (XOU)

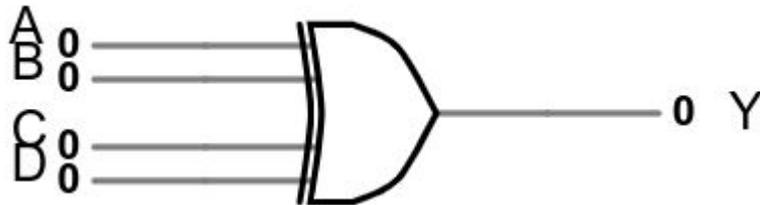
$$Y = A \oplus B$$



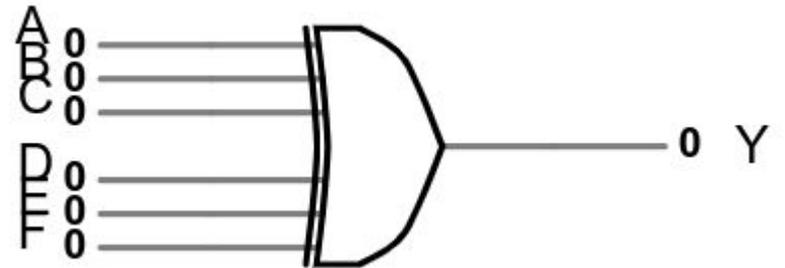
$$Y = A \oplus B \oplus C$$



$$Y = A \oplus B \oplus C \oplus D$$



$$Y = A \oplus B \oplus C \oplus D \oplus E \oplus F$$

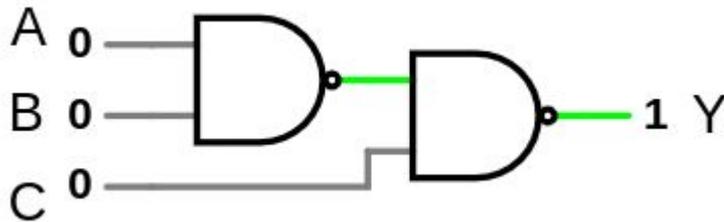


# Portas 3 entradas

É possível associar em cascata portas de 2 entradas diretas para obter uma maior equivalentes

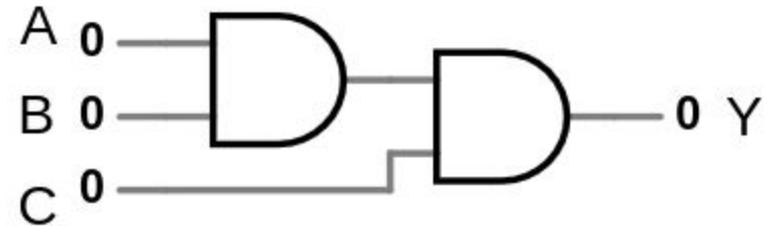
Porta **NÃO E / NAND**

$$Y = \overline{\overline{(A.B)}.C} \neq \overline{A.B.C}$$



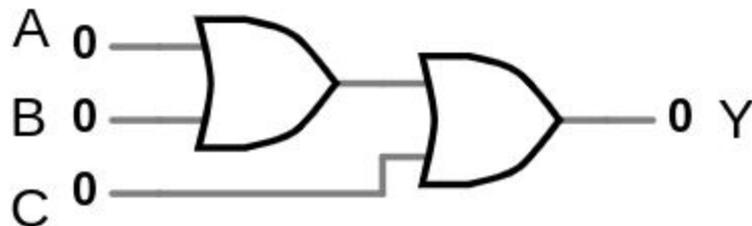
Porta **E / AND**

$$Y = A.B.C$$



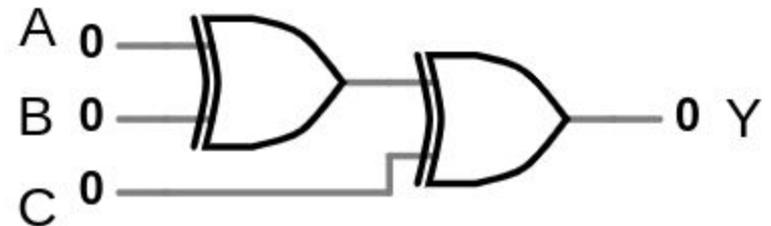
Porta **OU / OR**

$$Y = A + B + C$$



Porta **XOU / XOR**

$$Y = A \oplus B \oplus C$$



# Portas 3 entradas

É possível associar em cascata portas de 2 entradas diretas para obter uma maior equivalente

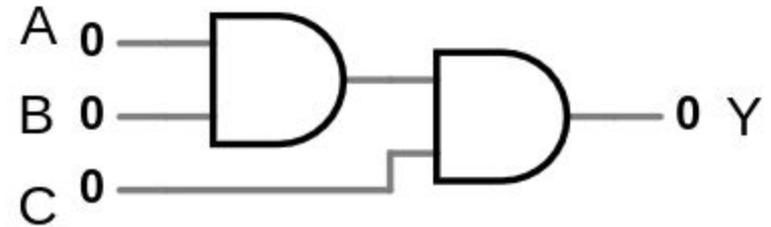
Porta **NÃO E / NAND**

$$Y = \overline{\overline{A.B}.C} \neq \overline{A.B.C}$$



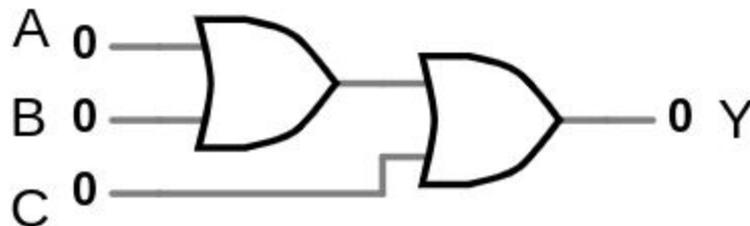
Porta **E / AND**

$$Y = A.B.C$$



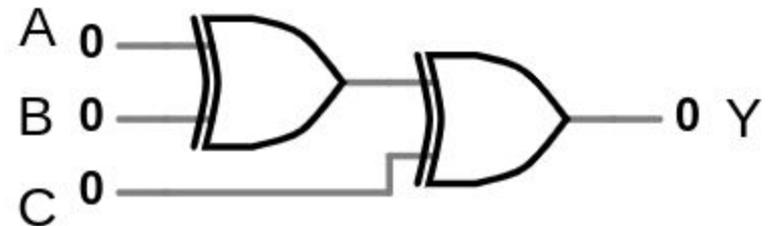
Porta **OU / OR**

$$Y = A + B + C$$



Porta **XOU / XOR**

$$Y = A \oplus B \oplus C$$



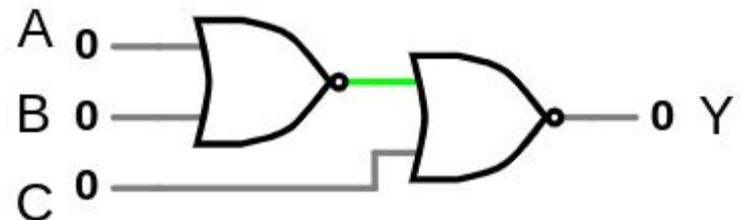
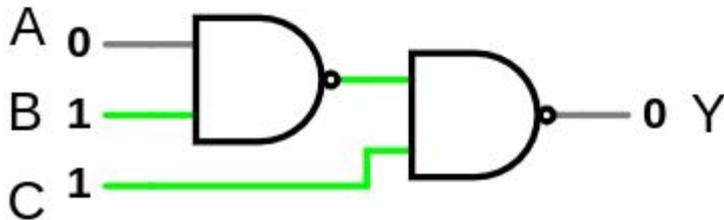
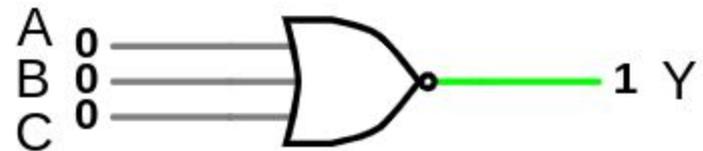
# Portas 3 entradas (inversas)

Não se pode associar em cascata portas inversas de 2 entradas para obter uma maior equivalentes

Porta NÃO E / NAND



Porta NÃO OU / NOR



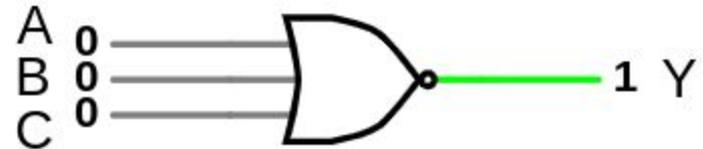
# Portas 3 entradas (inversas)

Não se pode associar em cascata portas inversas de 2 entradas para obter uma maior equivalentes

Porta NÃO E / NAND



Porta NÃO OU / NOR



# Transistores MOSFET

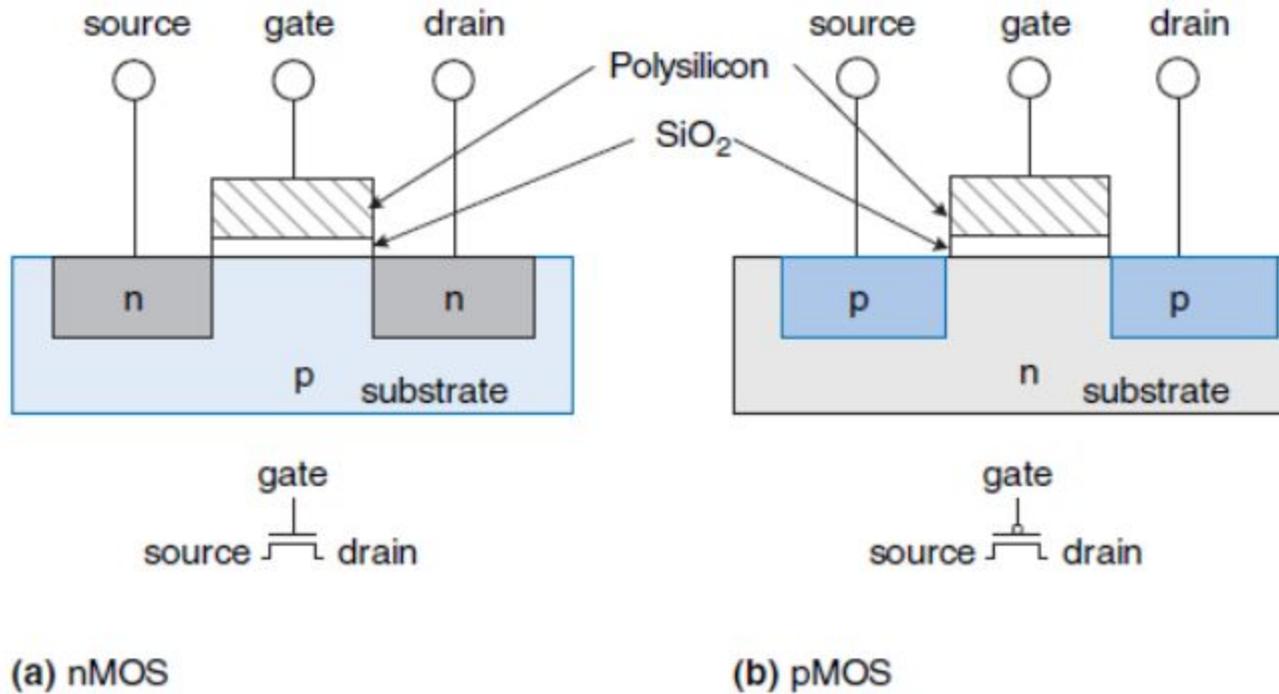
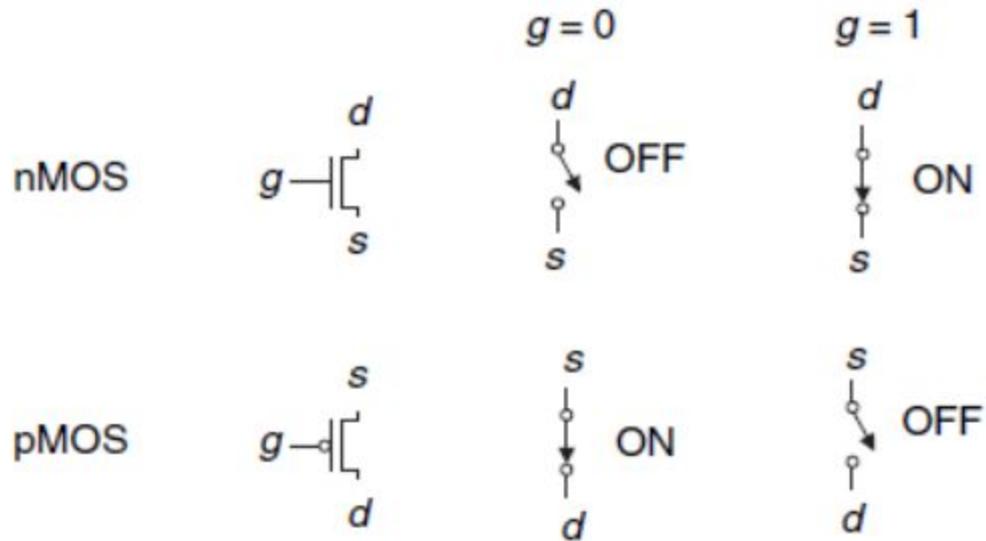


Figura 1.29 Transistores nMOS e pMOS

# Transistores MOSFET



**Figura 1.31** Modelo de interrupto dos MOSFET