

Departamento Acadêmico de Eletrônica - DAELN  
IFSC – Câmpus Florianópolis

# Eletrônica Digital I

---

## *Portas lógicas*

*Prof. Matheus Leitzke Pinto*  
*matheus.pinto@ifsc.edu.br*

# Sumário de aula

- Conceito de portas lógicas
- Porta NOT (inversora)
- Porta AND (E)
- Porta OR (OU)
- Portas NAND e NOR
- Portas XOR e XNOR
- Dupla negação



# Conceito de portas lógicas

---

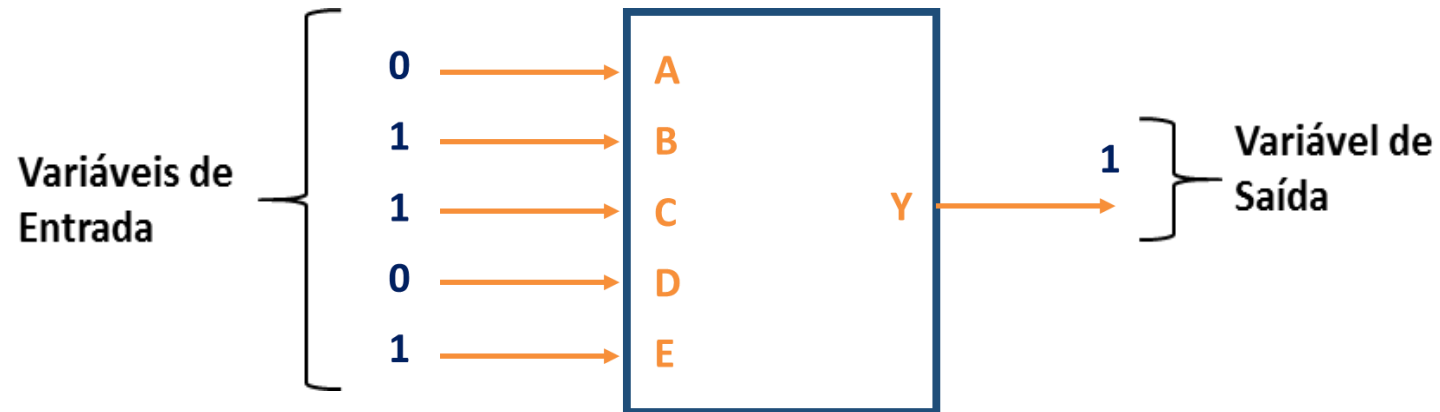
*Portas Lógicas*

# Conceito de portas lógicas

- **Portas lógicas** são os elementos básicos em eletrônica digital
- Os chips de eletrônica digital são constituídos de algumas unidades de portas lógicas, até bilhões!

# Conceito de portas lógicas

- De forma básica, uma porta lógica possui uma ou mais **entradas** e uma **saída**



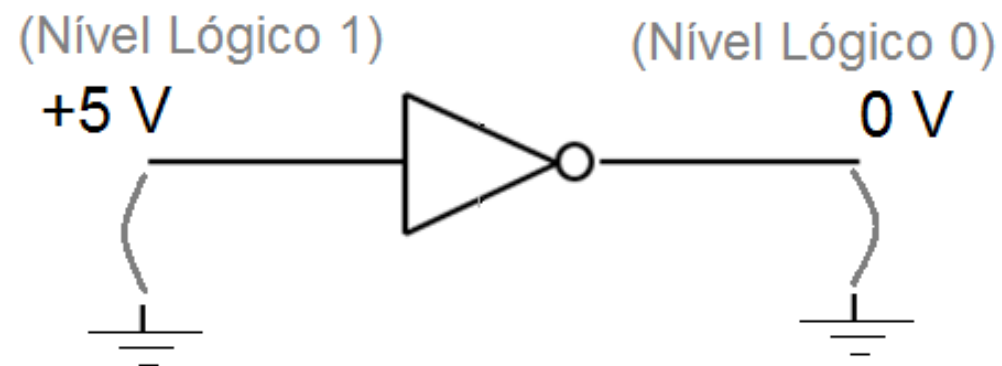
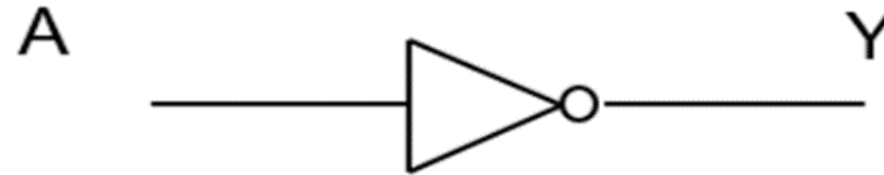
# Porta NOT (inversora)

---

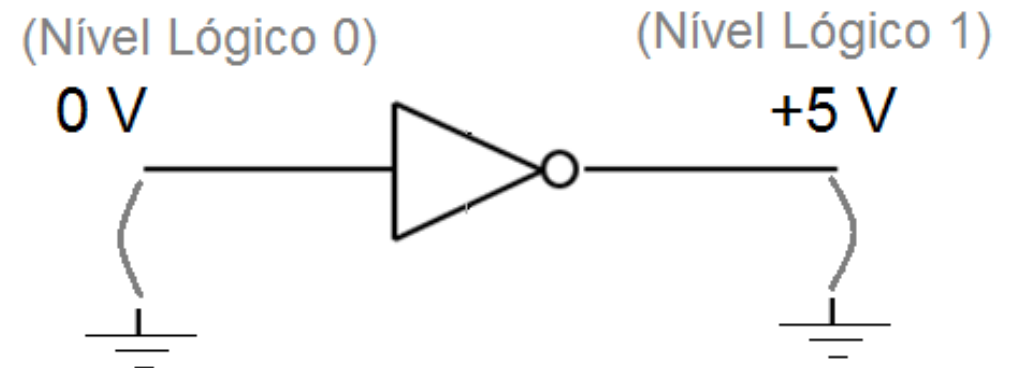
*Portas Lógicas*

# Porta NOT (inversora)

**Simbologia:**



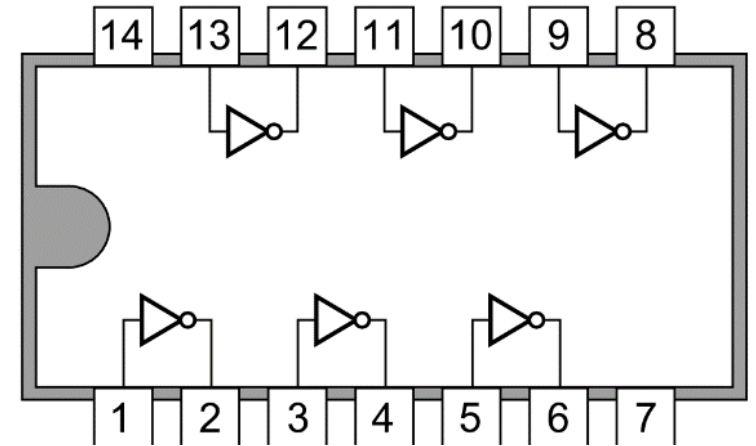
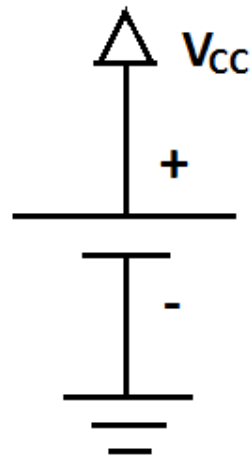
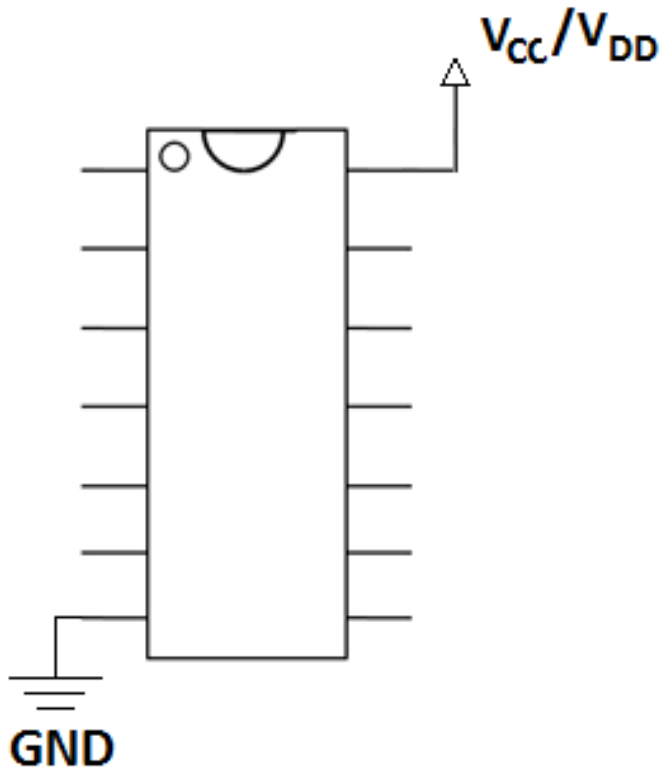
**(a)**



**(b)**

# Porta NOT (inversora)

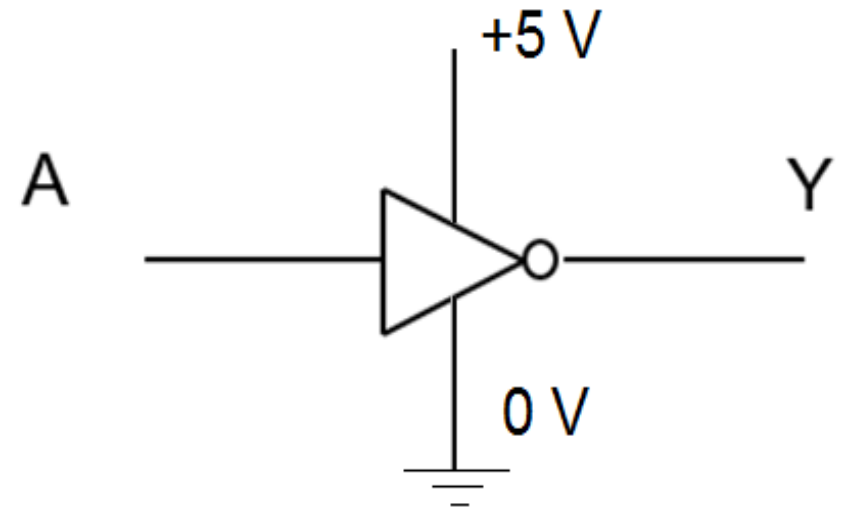
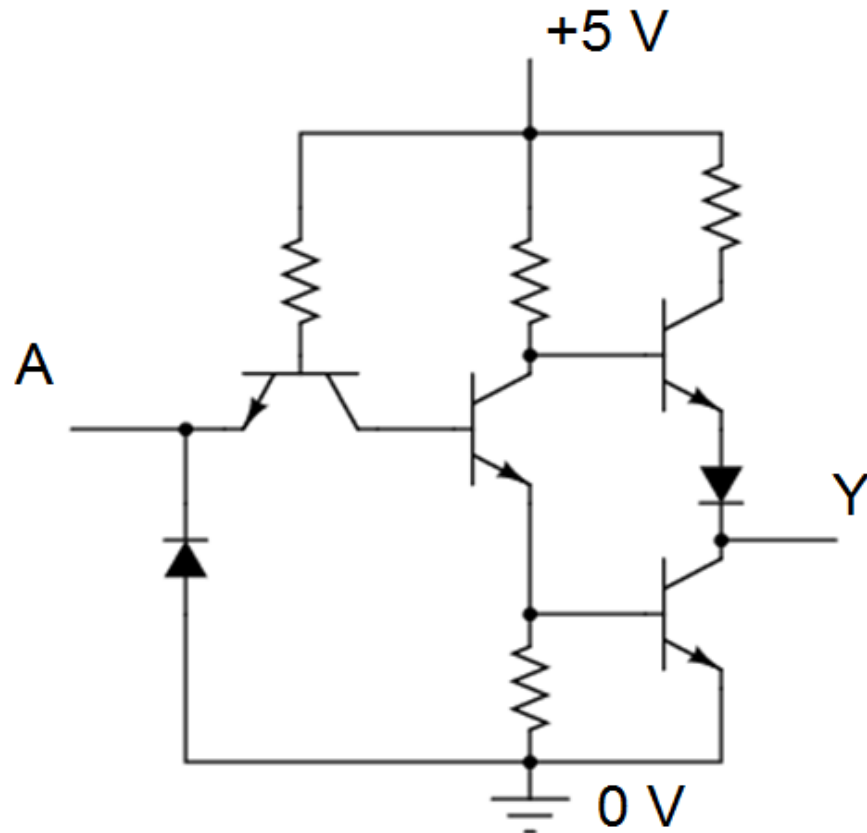
O CI 7404:





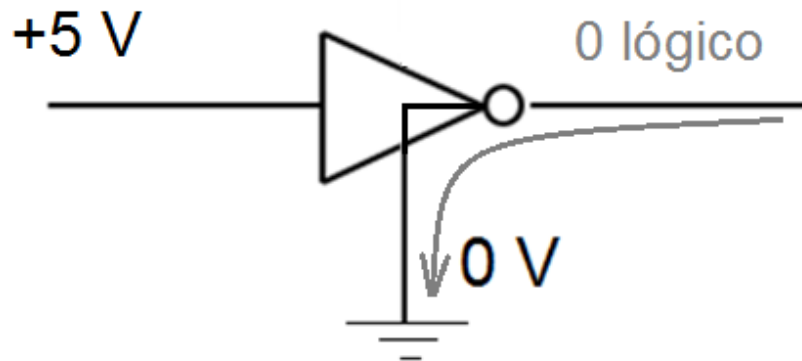
# Porta NOT (inversora)

Possível  
implementação:

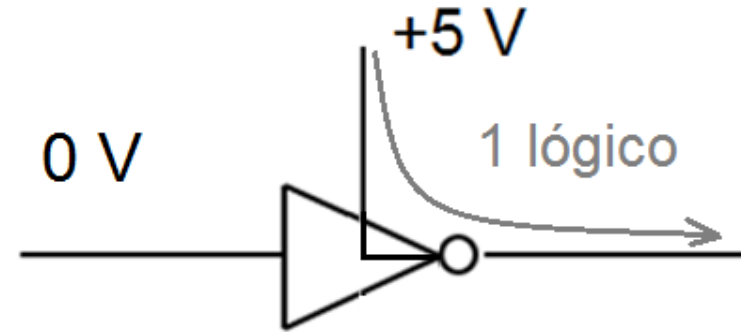


# Porta NOT (inversora)

**Sentidos da  
corrente:**



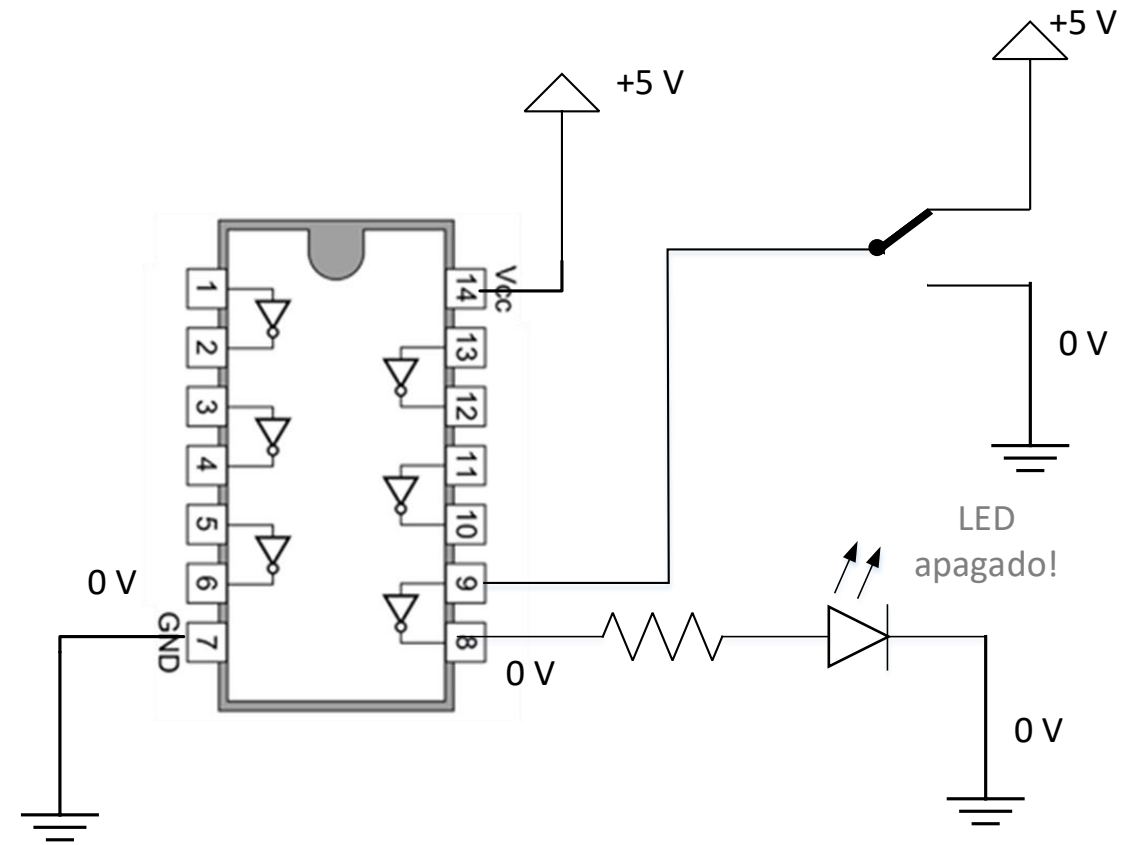
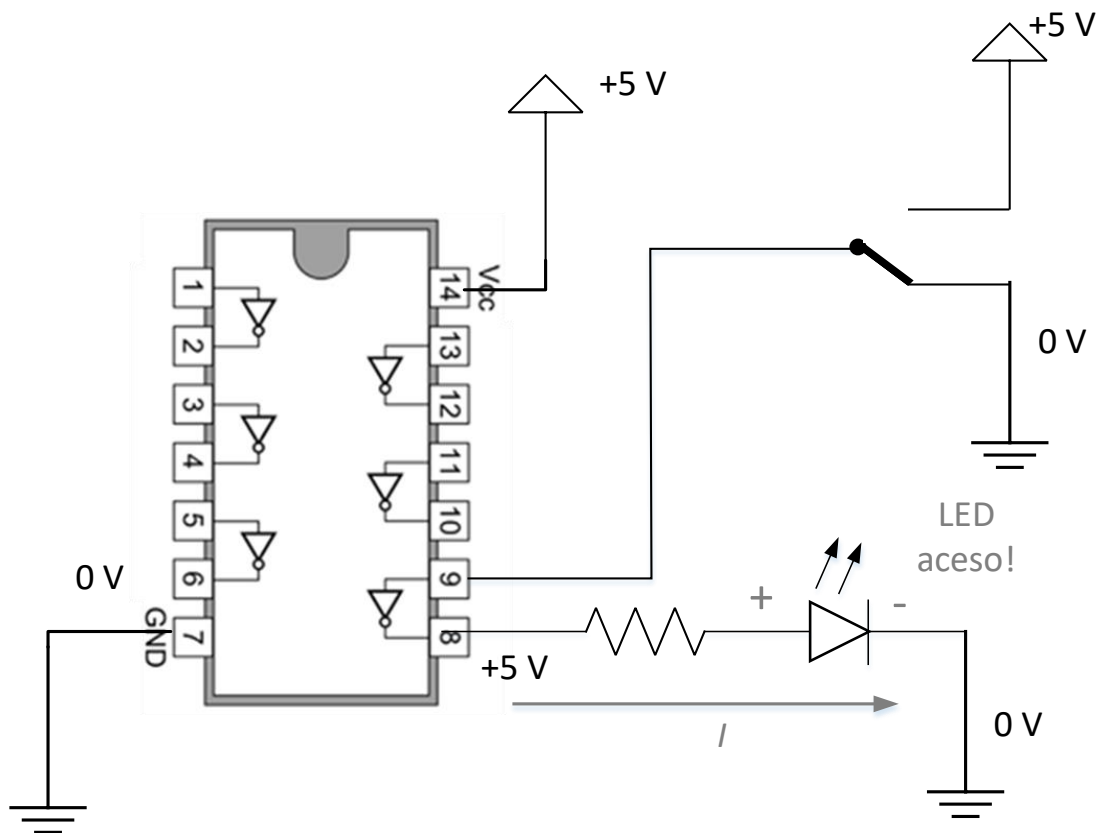
(a)



(b)

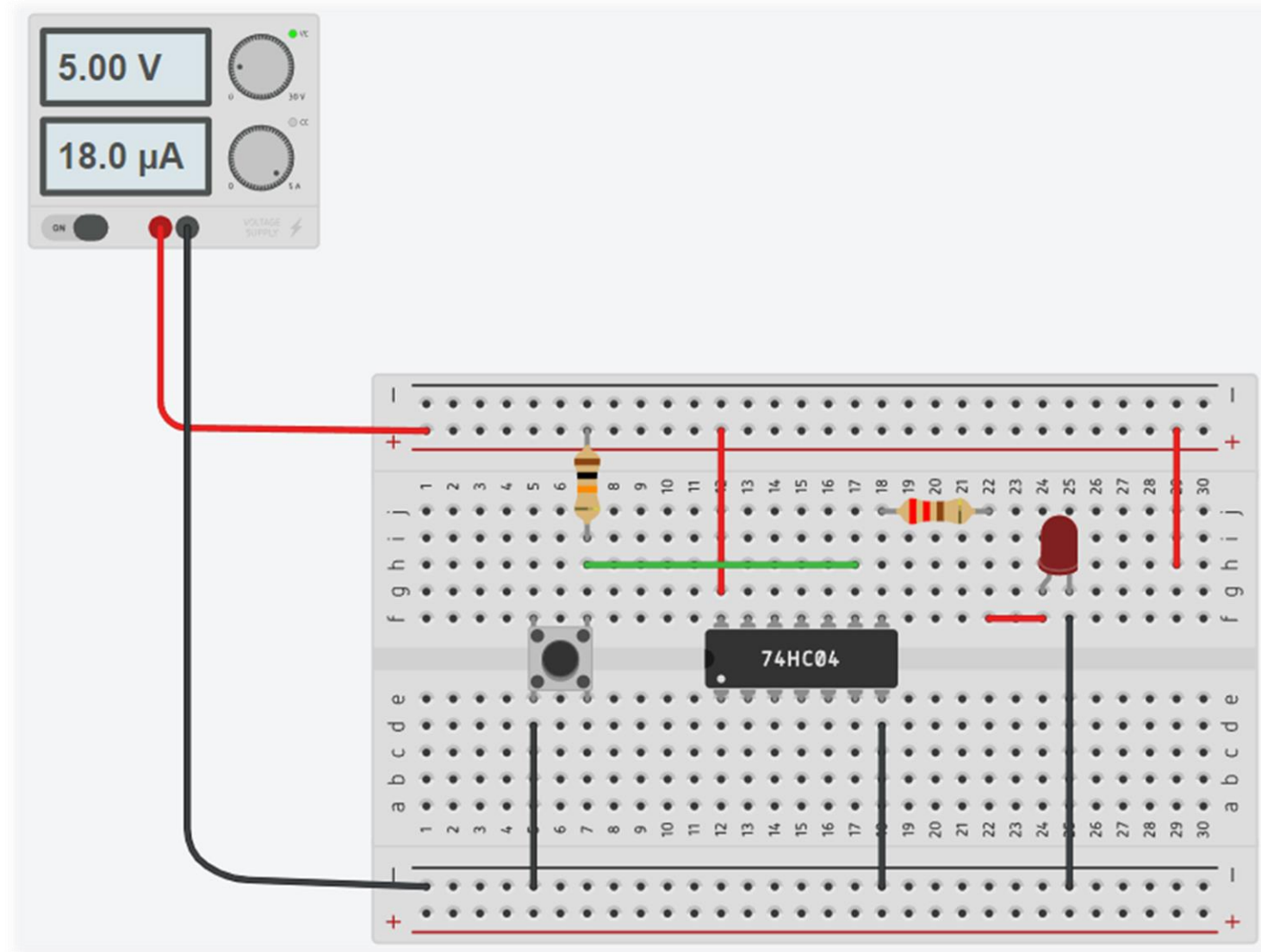
# Porta NOT (inversora)

**Exemplo de aplicação:**



# Porta NOT (inversora)

Exemplo de aplicação:



# Porta NOT (inversora)

## Função lógica:

$Y = \overline{A} \rightarrow$  onde se lê ***Y é igual à A negado.***

Se  $A = 0$ , então:

- $Y = \overline{0} = 1$

Se  $A = 1$ , então:

- $Y = \overline{1} = 0$

## Tabela verdade:

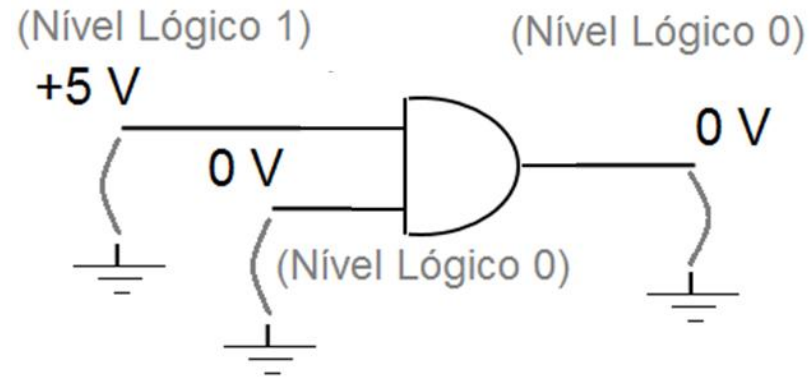
A	Y
0	1
1	0

# Porta AND (E)

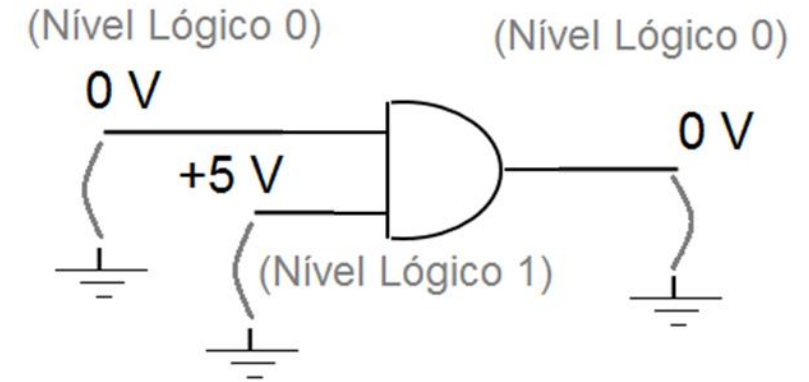
---

*Portas Lógicas*

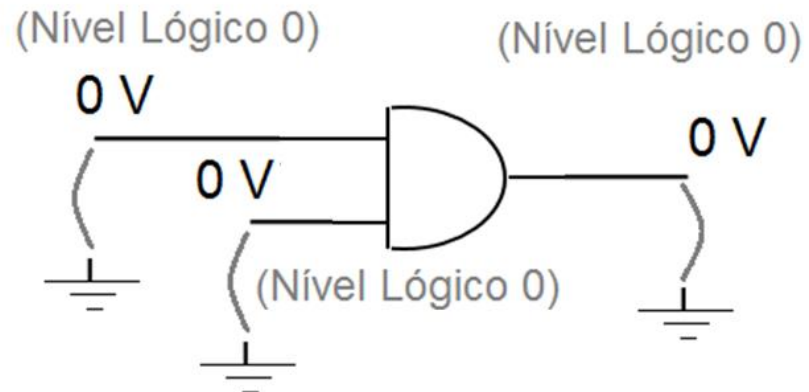
# Porta AND (E)



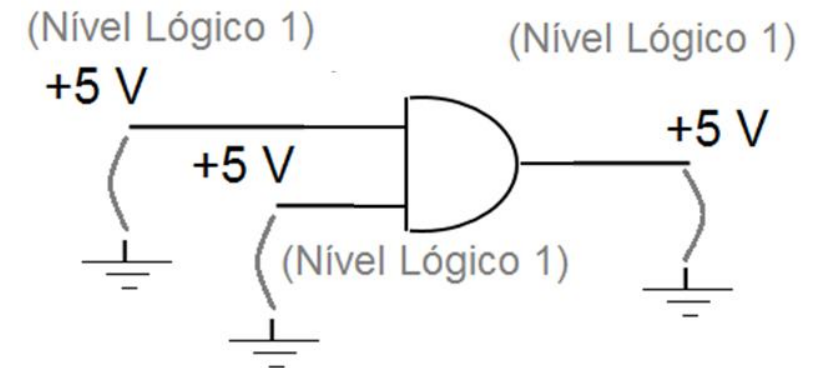
(a)



(b)

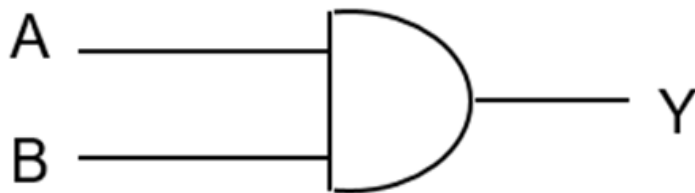


(c)



(d)

**Simbologia:**



# Porta AND (E)

**Função lógica:**

$Y = A \cdot B \rightarrow$  onde se lê ***Y é igual a A AND B.***

**Tabela verdade:**

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



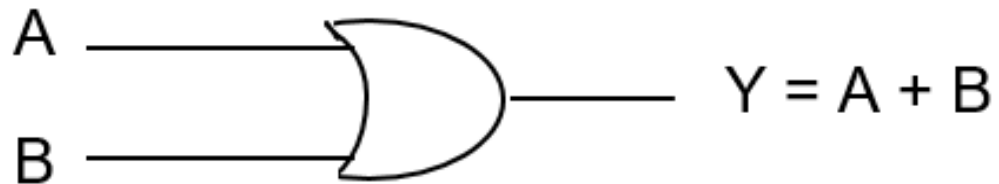
# Porta OR (OU)

---

*Portas Lógicas*

# Porta OR (OU)

**Simbologia:**



**Função lógica:**

$Y = A + B \rightarrow$  onde se lê ***Y é igual a A OR B.***

**Tabela verdade:**

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

# Porta NAND e NOR

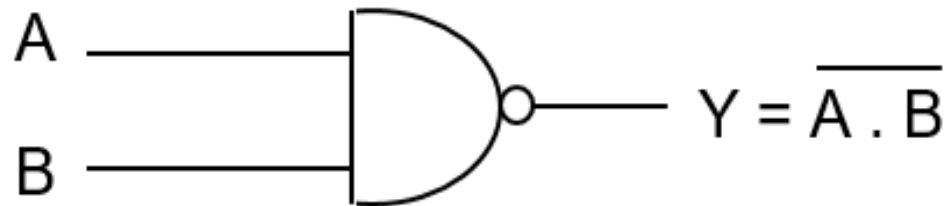
---

*Portas Lógicas*

# Porta NAND e NOR

- Porta **NÃO-E** (*NAND Gate*)

Simbologia:



Função lógica:

$Y = \overline{A \cdot B} \rightarrow$  onde se lê ***Y é igual a A NAND B.***

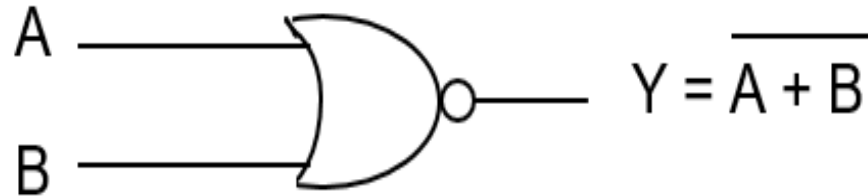
Tabela verdade:

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

# Porta NAND e NOR

- Porta **NÃO-OU** (*NOR Gate*)

Simbologia:



Função lógica:

$Y = \overline{A + B} \rightarrow$  onde se lê ***Y é igual a A NOR B.***

Tabela verdade:

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

# Porta XOR e XNOR

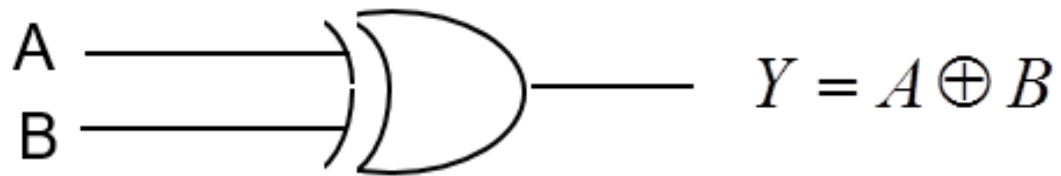
---

*Portas Lógicas*

# Porta XOR e XNOR

- Porta OU EXCLUSIVO (*XOR Gate*)

Simbologia:



Função lógica:

$Y = A \oplus B \rightarrow$  onde se lê ***Y é igual a A XOR B.***

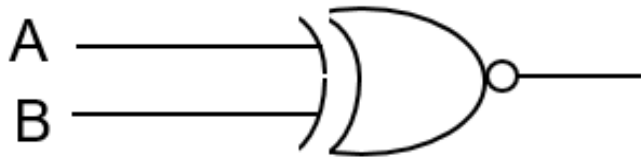
Tabela verdade:

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

# Porta XOR e XNOR

- Porta NÃO-OU EXCLUSIVO (*XNOR Gate*)

Simbologia:



$$Y = A \odot B$$

Função lógica:

$$Y = A \odot B = \overline{A \oplus B} \rightarrow \text{onde se lê } Y \text{ é igual a } A \text{ XOR } B.$$

Tabela verdade:

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



# Dupla negação

---

*Portas Lógicas*

# Dupla negação

- É importante salientar algo nesse ponto: quando utiliza-se negação duas vezes, se tem o resultado original.
- Por exemplo, uma AND com a saída negada torna-se uma NAND, mas se negarmos novamente a saída, obtém-se novamente a função AND.

