# Combinação de Portas Lógicas

Prof. Gustavo Girão
**girao@imd.ufrn.br**

Baseado no material do Prof. Ricardo Weber (UFRGS)

# Apresentação

- Nesta aula veremos alguns tipos de portas lógicas obtidas a partir da combinação das portas lógicas básicas (OR, AND e NOT): portas NAND, NOR, XOR e XNOR.
- Estudaremos também porque todos os circuitos lógicos podem ser implementados utilizando apenas portas lógicas NAND e NOR.

# Objetivos

- Criar novas portas lógicas a partir das portas lógicas básicas;
- Testar a universalidade das portas lógicas NAND e NOR;
- Aprender a montar o circuito XOR e XNOR utilizando portas lógicas
- básicas e portas NAND e NOR.

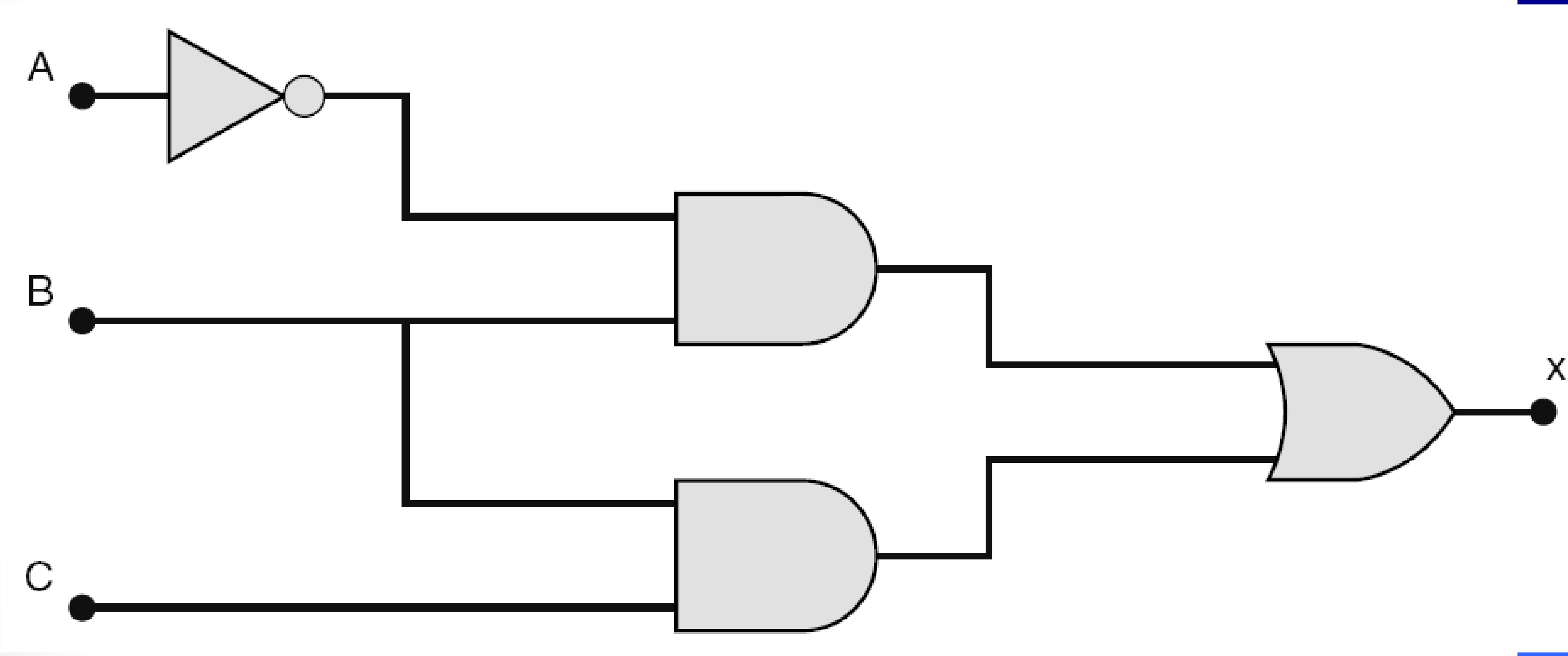
# Avaliando as Saídas dos Circuitos Lógicos

- Regras para avaliação de uma expressão booleana:
  - Executar todas as inversões de termos individuais.
  - Realizar todas as operações dentro de parêntesis.
  - Realizar a operação AND antes de uma operação OR, a menos que os parêntesis indiquem o contrário.
  - Sempre que uma expressão tiver uma barra sobre ela, realizar as operações no interior da expressão e depois inverter o resultado.

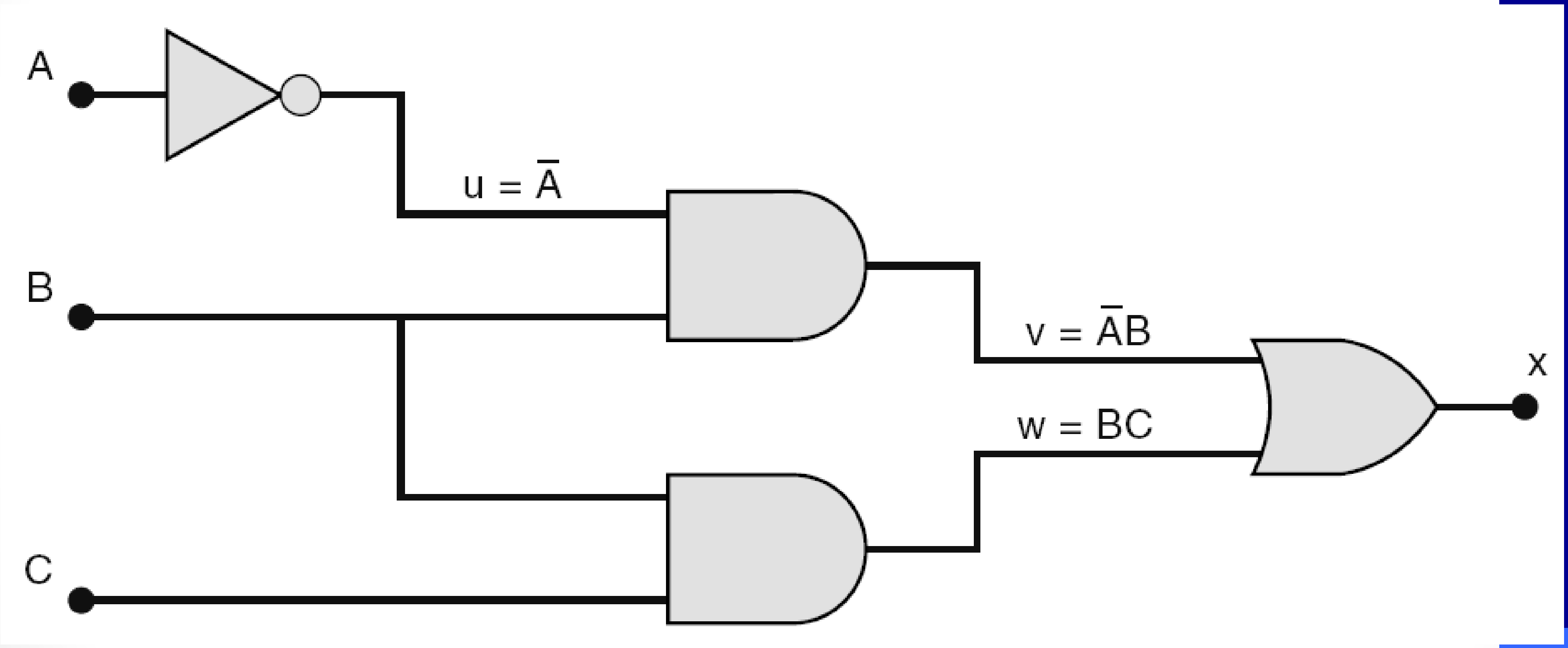
# Avaliando as Saídas dos Circuitos Lógicos

- Uma maneira de analisar um circuito composto por várias portas lógicas é usar uma tabela-verdade.
- Ela permite analisar uma porta ou uma combinação lógica de uma só vez.
- Ela também permite verificar novamente seu trabalho.
- Ao terminar, você tem um quadro de enorme benefício para solucionar o circuito lógico.

# Avaliando as Saídas dos Circuitos Lógicos

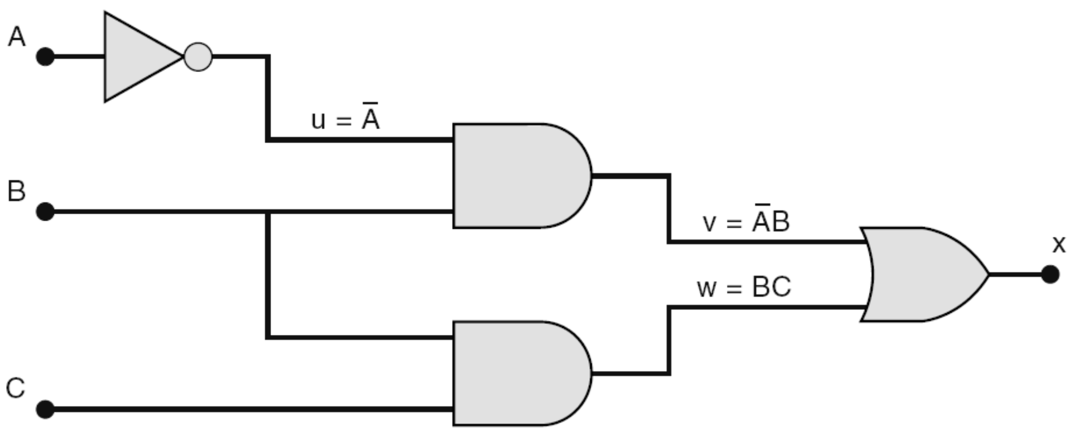


# Avaliando as Saídas dos Circuitos Lógicos



# Avaliando as Saídas dos Circuitos Lógicos

- O primeiro passo, após listar todas as combinações de entradas, é criar uma coluna na tabela-verdade para cada sinal intermediário (nó).

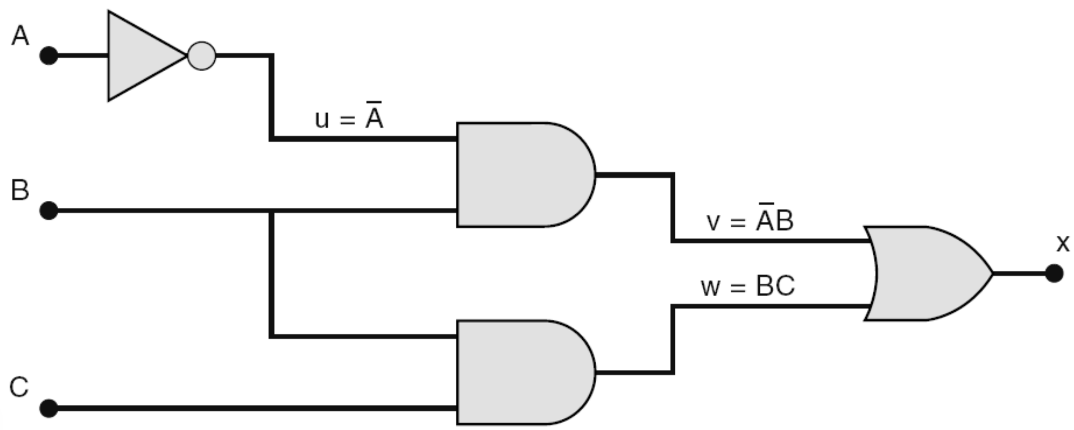



| A | B | C | $u=\overline{A}$ | $v=\overline{A}B$ | $w=BC$ | $x=v+w$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | | | |
| 0 | 0 | 1 | 1 | | | |
| 0 | 1 | 0 | 1 | | | |
| 0 | 1 | 1 | 1 | | | |
| 1 | 0 | 0 | 0 | | | |
| 1 | 0 | 1 | 0 | | | |
| 1 | 1 | 0 | 0 | | | |
| 1 | 1 | 1 | 0 | | | |

O nó *U* foi preenchido como complemento de *A.*

# Avaliando as Saídas dos Circuitos Lógicos

- O próximo passo é preencher os valores para a coluna v.

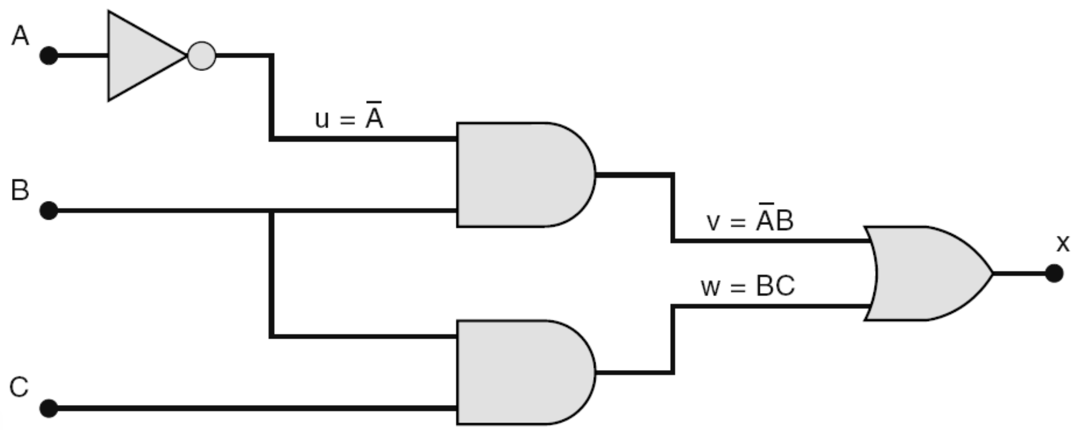



| A | B | C | u= $\overline{A}$ | v= $\overline{A}B$ | w= BC | x= v+w |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | | |
| 0 | 0 | 1 | 1 | 0 | | |
| 0 | 1 | 0 | 1 | 1 | | |
| 0 | 1 | 1 | 1 | 1 | | |
| 1 | 0 | 0 | 0 | 0 | | |
| 1 | 0 | 1 | 0 | 0 | | |
| 1 | 1 | 0 | 0 | 0 | | |
| 1 | 1 | 1 | 0 | 0 | | |

$v = \bar{A}B$ O nó *v* deve ser ALTO
quando *A* (nó *u*) é BAIXO e *B* é ALTO.

# Avaliando as Saídas dos Circuitos Lógicos

- O terceiro passo é determinar os valores do nó w, o produto lógico de BC.

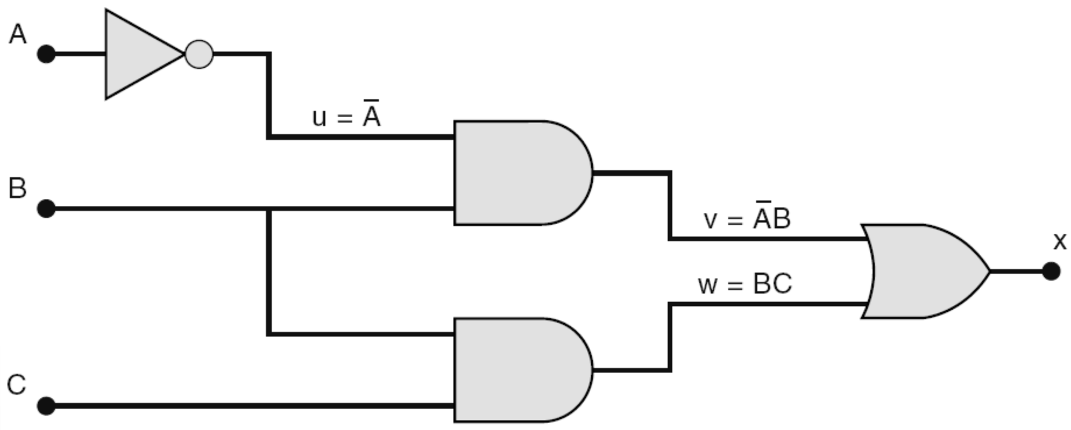



| A | B | C | u= $\overline{A}$ | v= $\overline{A}B$ | w= BC | x= v+w |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 1 | 0 | 0 | |
| 0 | 1 | 0 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 0 | 0 | 0 | |
| 1 | 1 | 0 | 0 | 0 | 0 | |
| 1 | 1 | 1 | 0 | 0 | 1 | |

A coluna é ALTO sempre que *B* é ALTO e *C* é ALTO.

# Avaliando as Saídas dos Circuitos Lógicos

- Logicamente, a etapa final é a combinação das colunas V e W para descobrir a *saída* x.




| A | B | C | $u=\overline{A}$ | $v=\overline{A}B$ | $w=BC$ | $x=v+w$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |

Desde que $x=v+w$, a saída $x$ será ALTO quando $v$ OU $w$ for ALTO.

# Avaliando as Saídas dos Circuitos Lógicos

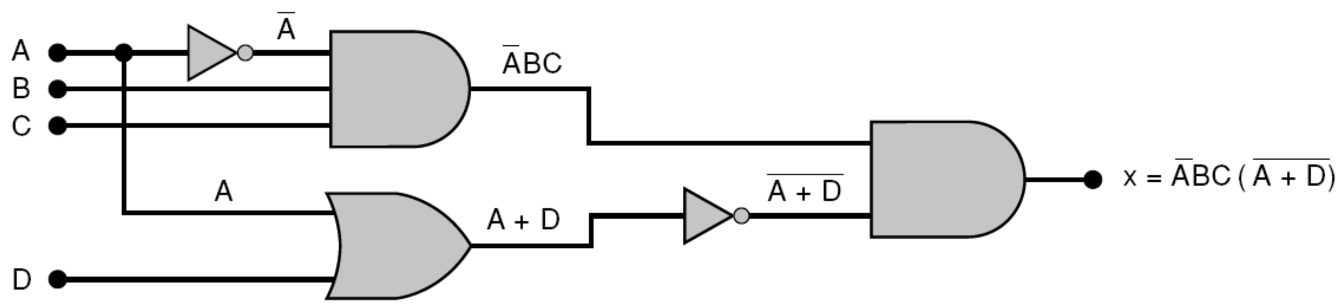
- Níveis lógicos de saída podem ser determinados diretamente a partir de um **diagrama de circuito**.

  As saídas de cada porta são percebidas até que a saída final seja encontrada.

# Avaliando as Saídas dos Circuitos Lógicos

Questão para revisão

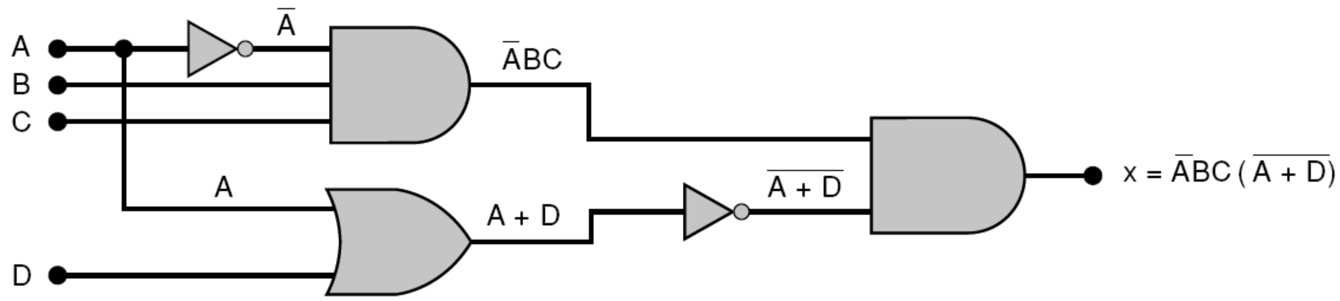
- Use a expressão para a saída x para determinar a saída do circuito na Figura abaixo para as condições: A=0, B=1, C=1 e D=0.



# Avaliando as Saídas dos Circuitos Lógicos

Questão para revisão

- Use a expressão para a saída x para determinar a saída do circuito na Figura abaixo para as condições: A=0, B=1, C=1 e D=0.

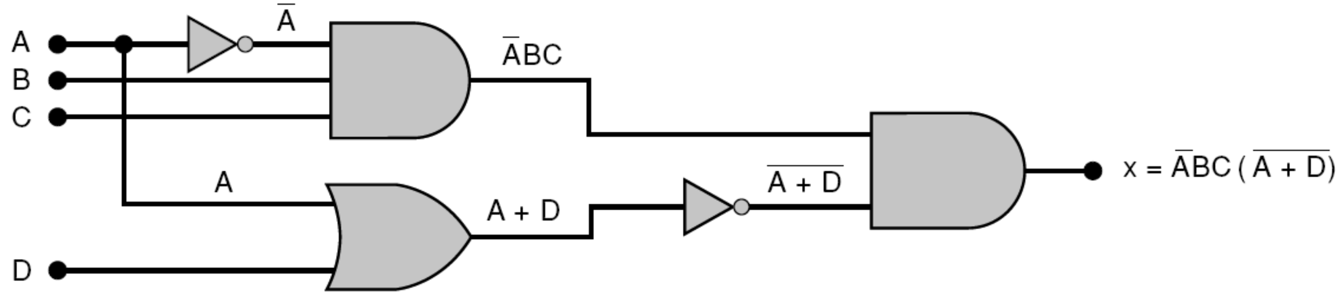



$$x = \overline{A}BC\overline{(A+D)} \Rightarrow$$

# Avaliando as Saídas dos Circuitos Lógicos

Questão para revisão

- Use a expressão para a saída x para determinar a saída do circuito na Figura abaixo para as condições: A=0, B=1, C=1 e D=0.

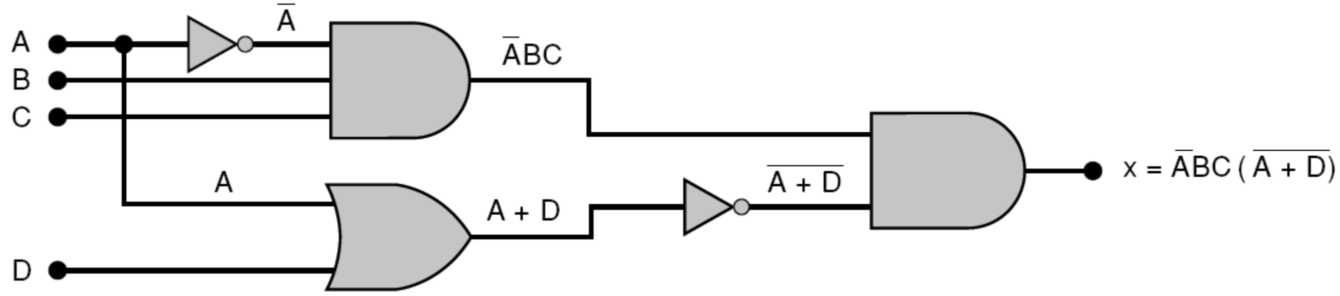



$$x = \overline{A}BC\overline{(A+D)} \Rightarrow \overline{0}\cdot 1\cdot 1\cdot\overline{(0 + 0)}$$

# Avaliando as Saídas dos Circuitos Lógicos

Questão para revisão

- Use a expressão para a saída x para determinar a saída do circuito na Figura abaixo para as condições: A=0, B=1, C=1 e D=0.



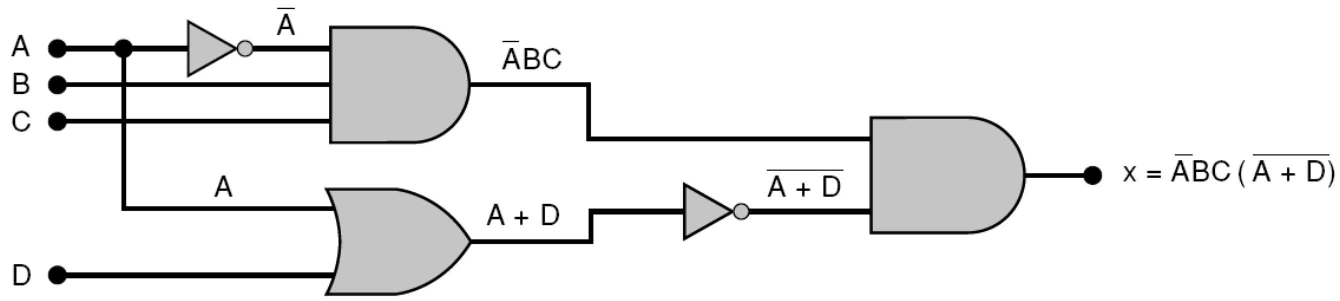

$$x = \overline{A}BC\overline{(A+D)} \Rightarrow \overline{0}\cdot 1\cdot 1\cdot \overline{(0 + 0)}$$

$$x = 1\cdot 1\cdot 1\cdot \overline{(0 + 0)}$$

# Avaliando as Saídas dos Circuitos Lógicos

Questão para revisão

- Use a expressão para a saída x para determinar a saída do circuito na Figura abaixo para as condições: A=0, B=1, C=1 e D=0.




$$x = \overline{A}BC\overline{(A+D)} \Rightarrow \overline{0}\cdot 1\cdot 1\cdot \overline{(0+0)}$$

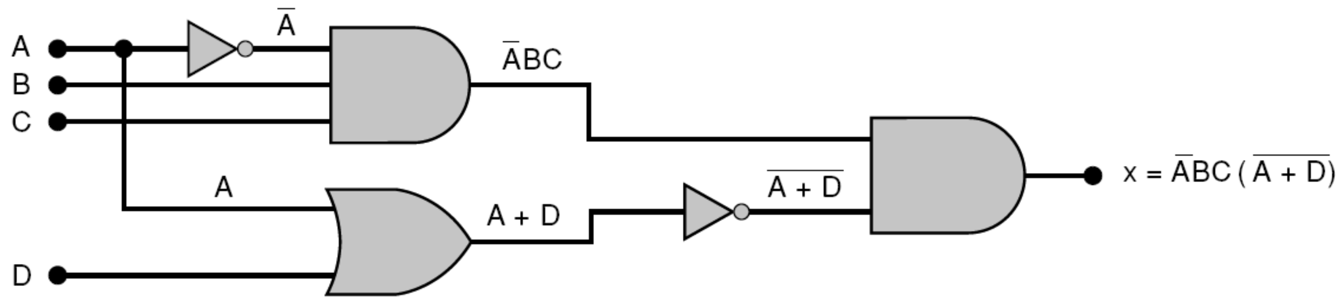
$$x = 1\cdot 1\cdot 1\cdot \overline{(0+0)}$$

$$x = 1\cdot \overline{(0)}$$

# Avaliando as Saídas dos Circuitos Lógicos

Questão para revisão

- Use a expressão para a saída x para determinar a saída do circuito na Figura abaixo para as condições: A=0, B=1, C=1 e D=0.




$$x = \overline{A}BC(\overline{A+D}) \Rightarrow \overline{0} \cdot 1 \cdot 1 \cdot (\overline{0 + 0})$$

$$x = 1 \cdot 1 \cdot 1 \cdot (\overline{0 + 0})$$

$$x = 1 \cdot (\overline{0}) = 1 \cdot 1 = 1$$

# Implementando Circuitos a partir de Expressões Booleanas

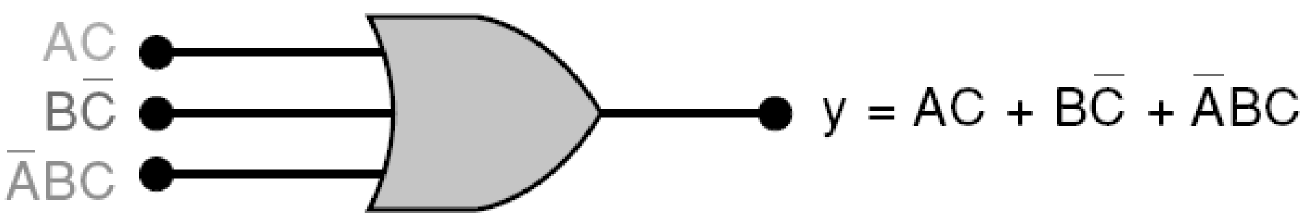
- É importante saber desenhar um circuito lógico de uma expressão booleana.

  A expressão $X = A \cdot B \cdot C$ poderia ser desenhada como três entradas de uma porta AND.

  Um circuito definido por $X = \overline{A} + B$ usaria duas entradas de uma porta OR com um INVERSOR em uma das entradas.

# Implementando Circuitos a partir de Expressões Booleanas

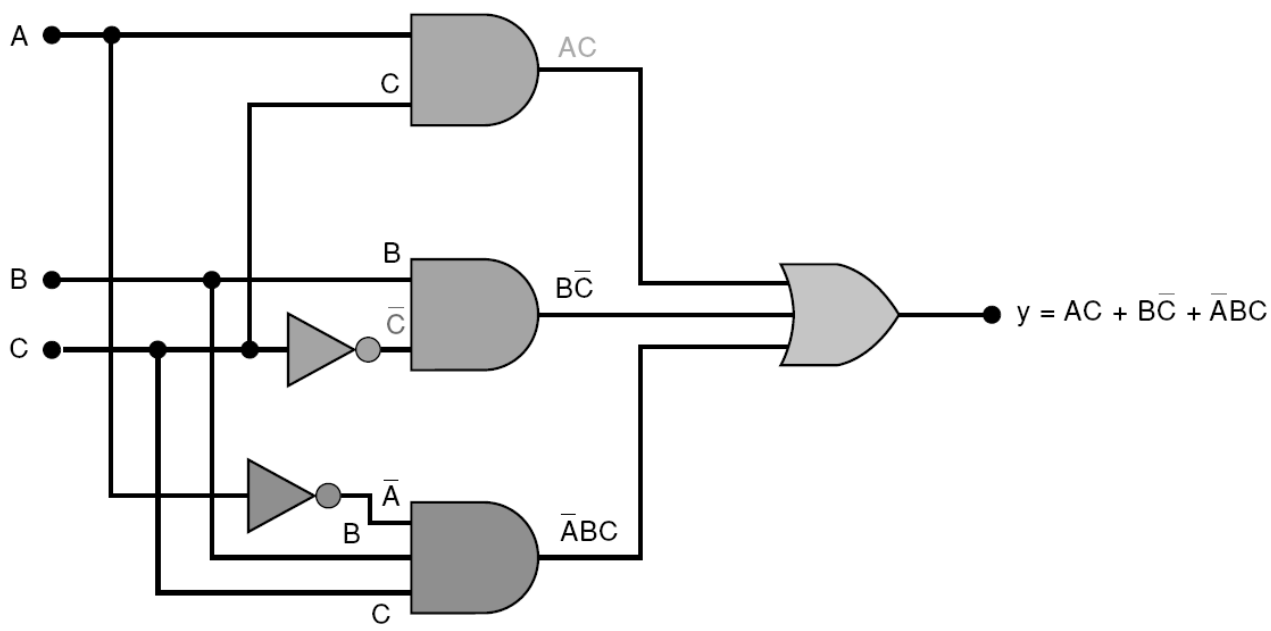
- Um circuito com saída $y = AC + B\overline{C} + \overline{A}BC$ contém três termos sobre os quais é aplicada a operação OR…




…e requer uma porta OR de três entradas.

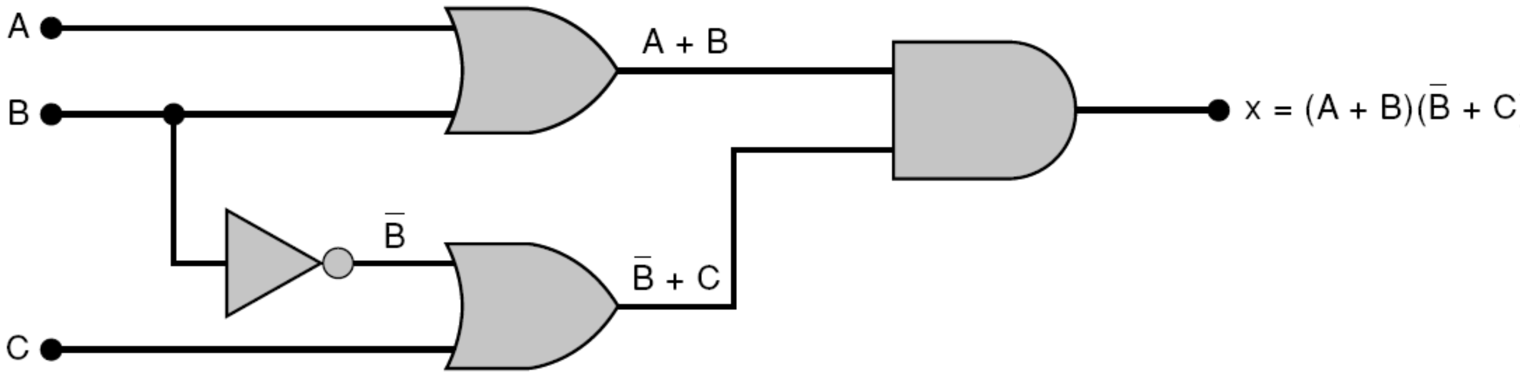
# Implementando Circuitos a partir de Expressões Booleanas

- Cada entrada da porta OR é um termo do produto AND.
- Uma porta AND com entradas adequadas pode ser usada para gerar cada um desses termos.



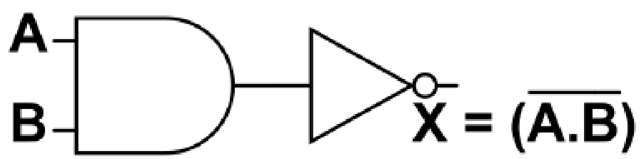
# Implementando Circuitos a partir de Expressões Booleanas

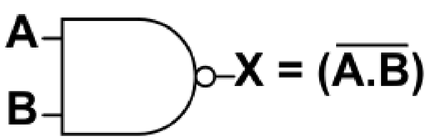
- Diagrama de circuito para implementar $x = (A + B)(\overline{B} + C)$.



# Combinação de portas lógicas

- As portas OR, AND e NOT podem ser combinadas para formar outras portas lógicas
- **Porta NAND:**
  - É uma combinação de porta AND com porta NOT

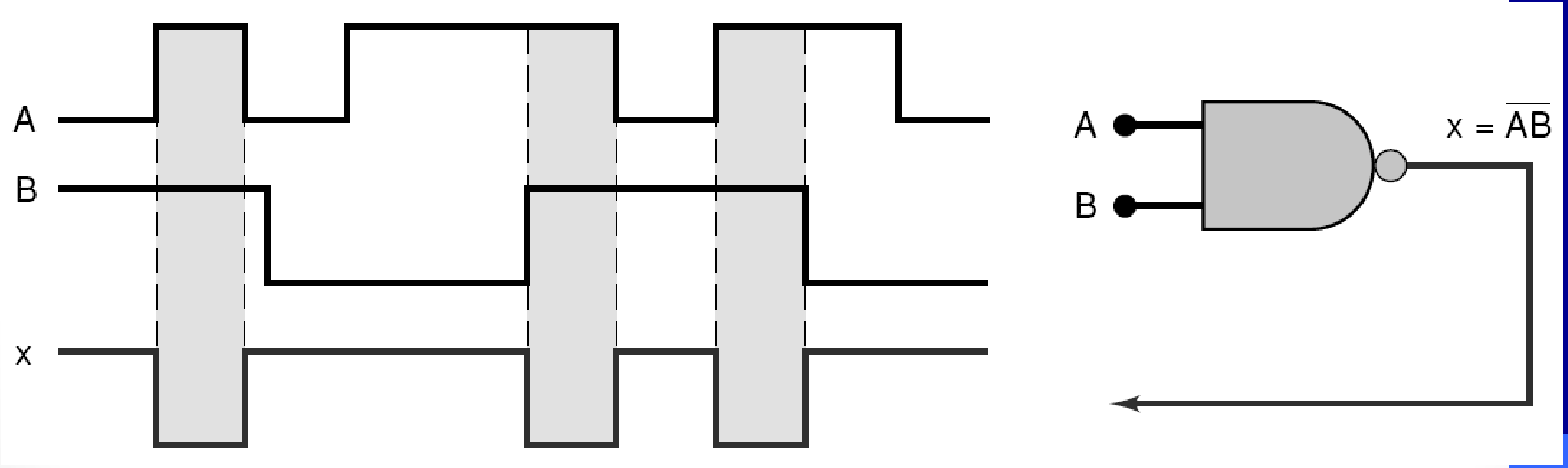



| Tabela Verdade | | |
|---|---|---|
| A | B | **X** |
| 0 | 0 | **1** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **0** |

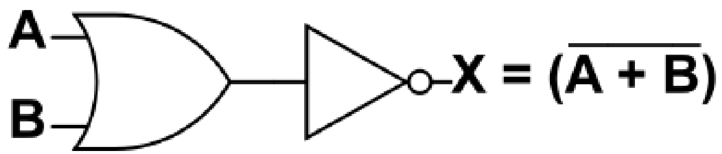
# Combinação de portas lógicas

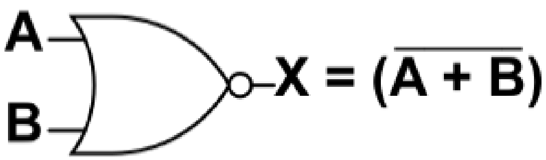
- Saída de onda de uma porta **NAND** para entrada de onda.



# Combinação de portas lógicas

- **Porta NOR:**
  - É uma combinação de porta OR com porta NOT

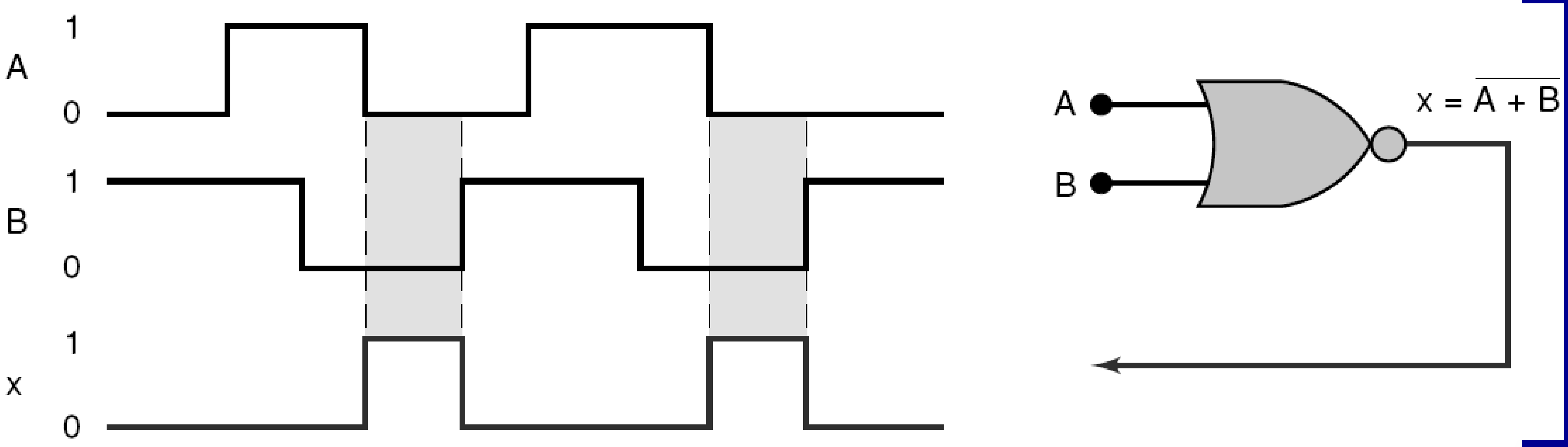



| Tabela Verdade | | |
|---|---|---|
| A | B | X |
| 0 | 0 | **1** |
| 0 | 1 | **0** |
| 1 | 0 | **0** |
| 1 | 1 | **0** |

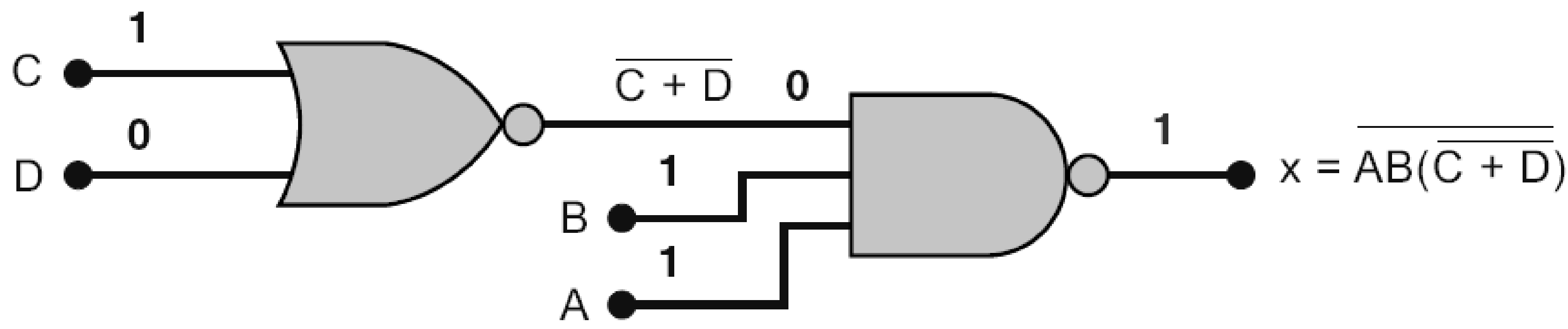
# Combinação de portas lógicas

- Saída de onda de uma porta **NOR** para entrada de onda.



# Combinação de portas lógicas

- Circuito lógico com a expressão $x = \overline{AB \bullet (\overline{C + D})}$ usando apenas NOR e NAND.



# Universalidade de portas

- As portas lógicas AND, OR e NOT podem ser implementadas utilizando apenas portas lógicas NAND ou NOR
- Por isso, podemos concluir:
  - Qualquer circuito pode ser implementado a partir de portas lógicas NAND e NOR
- Vantagem:
  - As portas NAND e NOR requerem menos transistores do que as portas AND e OR
  - E portas NAND requerem menos transistores do que as NOR
- Veremos mais disto nas próximas aulas

# Universalidade de portas

- **Porta NOT**
  - Utilizando portas NAND:
    - Vendo a tabela, considerando as entradas A e B iguais, sabe-se que só é invertida a saída nas linhas 1 e 4
  - Utilizando portas NOR:
    - Vendo a tabela, considerando as entradas A e B iguais, sabe-se que só é invertida a saída nas linhas 1 e 4

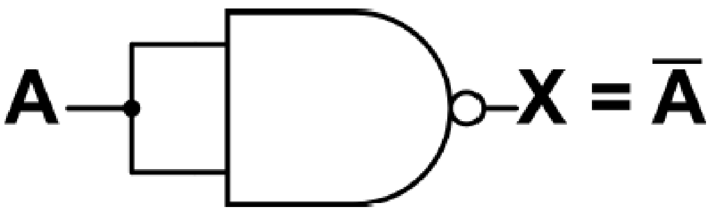
| Tabela Verdade NAND | | |
|---|---|---|
| A | B | **X** |
| 0 | 0 | **1** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **0** |

| Tabela Verdade NOR | | |
|---|---|---|
| A | B | **X** |
| 0 | 0 | **1** |
| 0 | 1 | **0** |
| 1 | 0 | **0** |
| 1 | 1 | **0** |

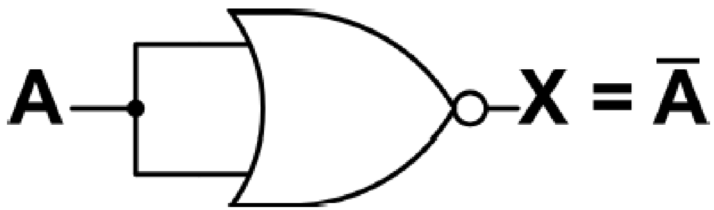
# Universalidade de portas

| Tabela Verdade NAND | | |
|---|---|---|
| A | B | **X** |
| 0 | 0 | **1** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **0** |

| Tabela Verdade NOR | | |
|---|---|---|
| A | B | **X** |
| 0 | 0 | **1** |
| 0 | 1 | **0** |
| 1 | 0 | **0** |
| 1 | 1 | **0** |



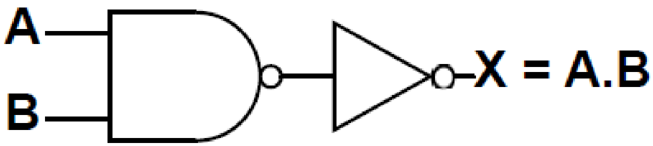

(a) Utilizando NAND



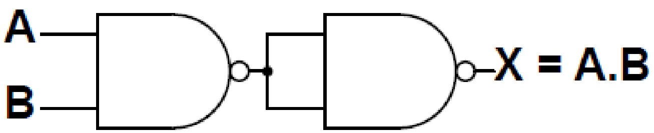

(b) Utilizando NOR

# Universalidade de portas

- Porta AND
  - Existe uma propriedade da Algebra Booleana que demonstra que $\overline{\overline{A}}$ é equivalente a A. Logo:
  - Utilizando portas NAND:
    - Coloca-se um inversor na saída da porta NAND




(c) Utilizando NAND



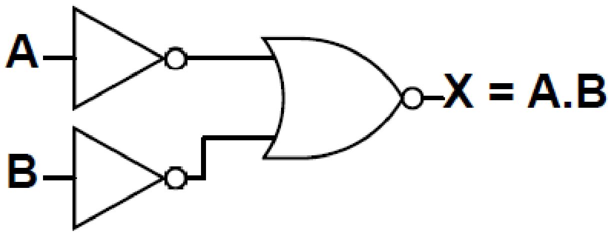

(d) Utilizando NAND

# Universalidade de portas

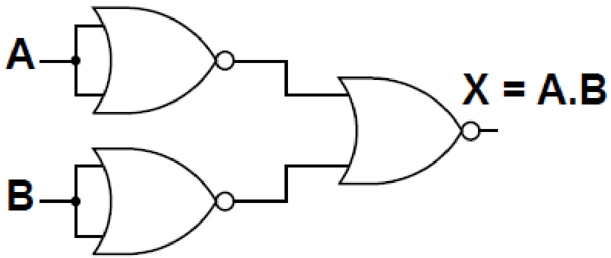
- Utilizando portas NOR:
  - Vendo as tabelas, sabe-se que as saídas são iguais a '1' nas linhas 1 e 4
  - As entradas devem ser invertidas

| Tabela Verdade AND | | |
|---|---|---|
| A | B | X |
| 0 | 0 | **0** |
| 0 | 1 | **0** |
| 1 | 0 | **0** |
| 1 | 1 | **1** |

| Tabela Verdade NOR | | |
|---|---|---|
| A | B | X |
| 0 | 0 | **1** |
| 0 | 1 | **0** |
| 1 | 0 | **0** |
| 1 | 1 | **0** |



(e) Utilizando NOR



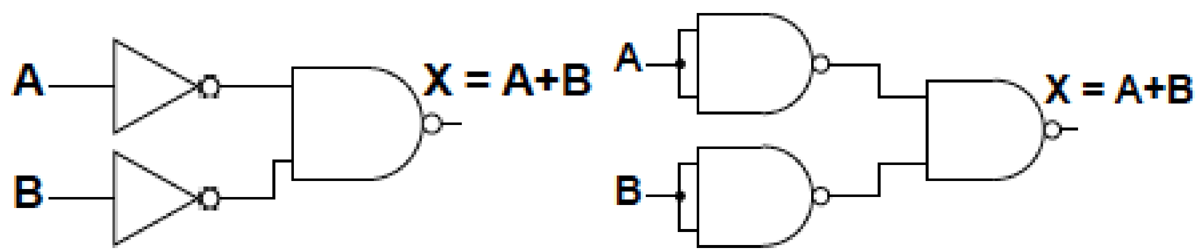
(f) Utilizando NOR

# Universalidade de Portas

- Porta OR
  - Utilizando portas NAND:
  - Vendo as tabelas, sabe-se que as saídas são iguais a ’0’ nas linhas 1 na tabela do OR e 4 na tabela da NAND
  - As entradas devem ser invertidas

| Tabela Verdade OR | | |
|---|---|---|
| A | B | X |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

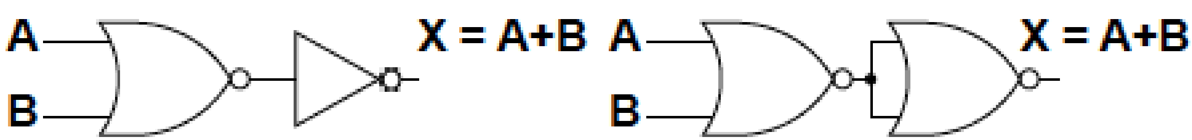
| Tabela Verdade NAND | | |
|---|---|---|
| A | B | X |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




(g) Utilizando NAND (h) Utilizando NAND

# Universalidade de Portas

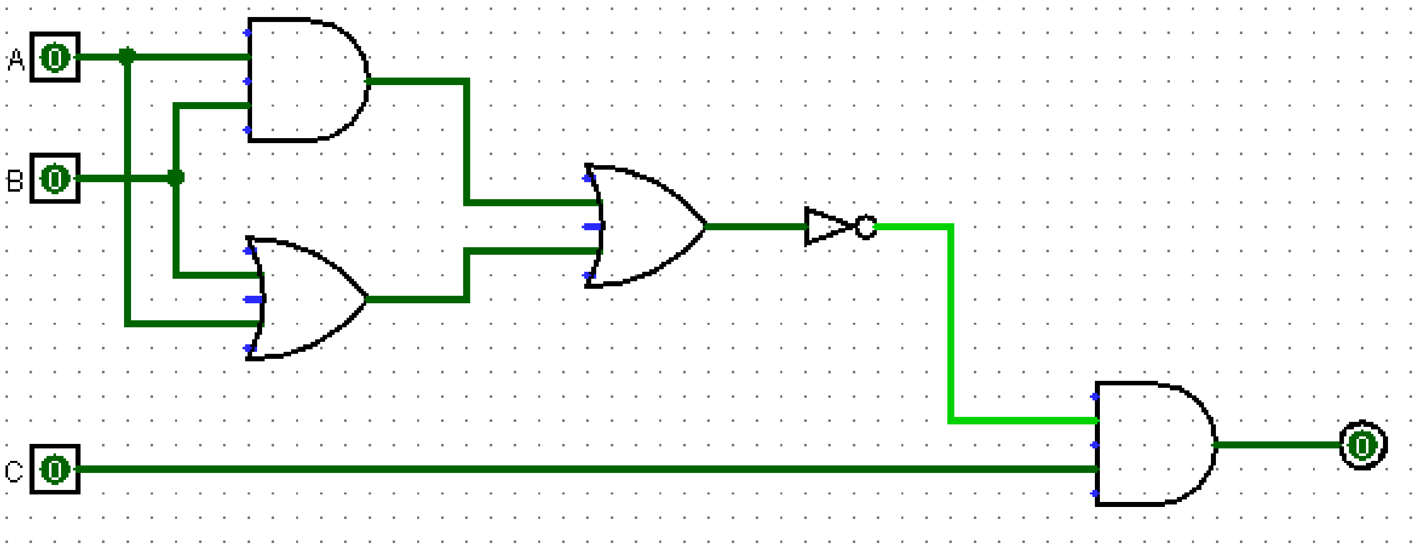
- Porta OR
  - Utilizando portas NOR:
    - Coloca-se um inversor na saída da porta NOR




(i) Utilizando NOR (j) Utilizando NOR

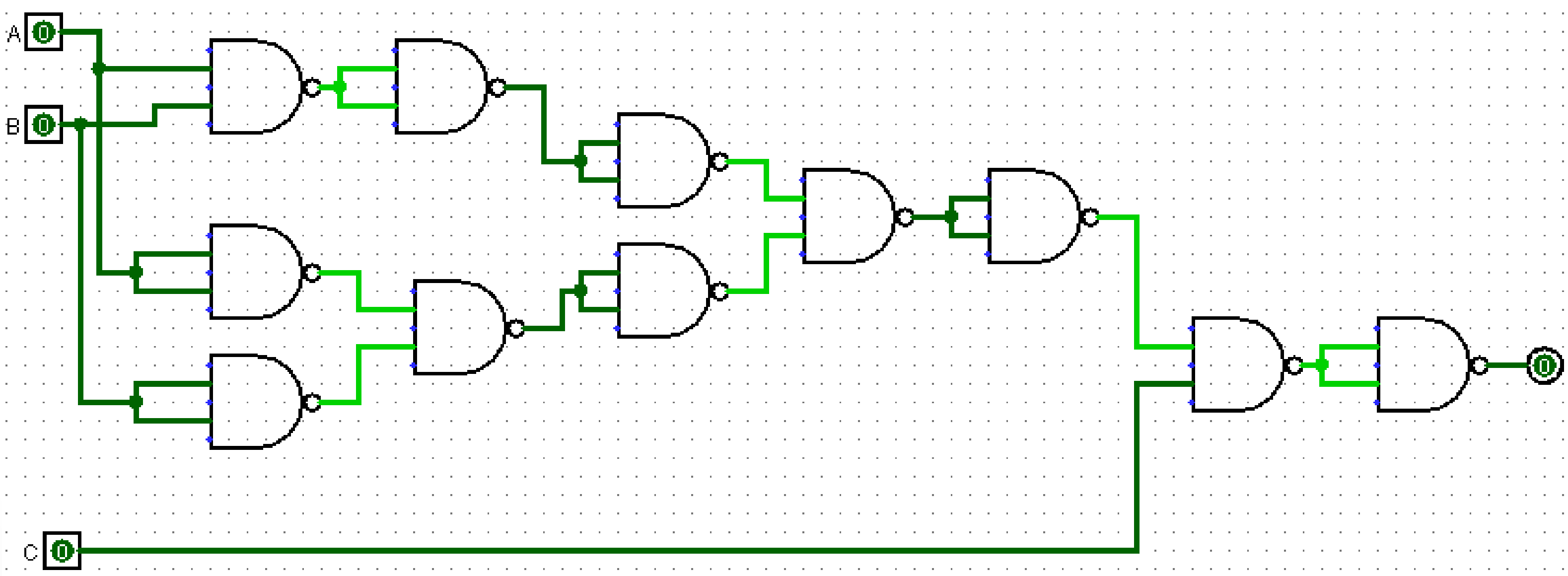
# Universalidade de Portas

- Redesenhe o circuito usando apenas portas NAND



# Universalidade de Portas

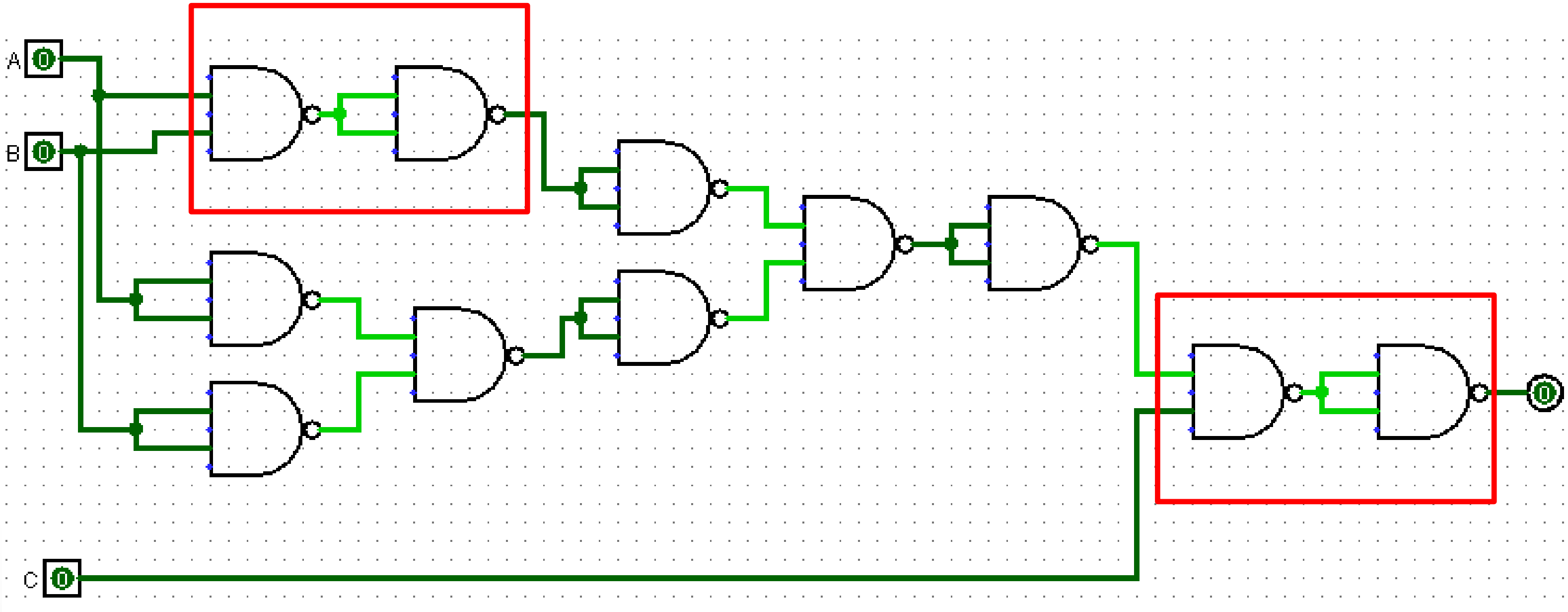
- Redesenhe o circuito usando apenas portas NAND




IMD0121

# Universalidade de Portas

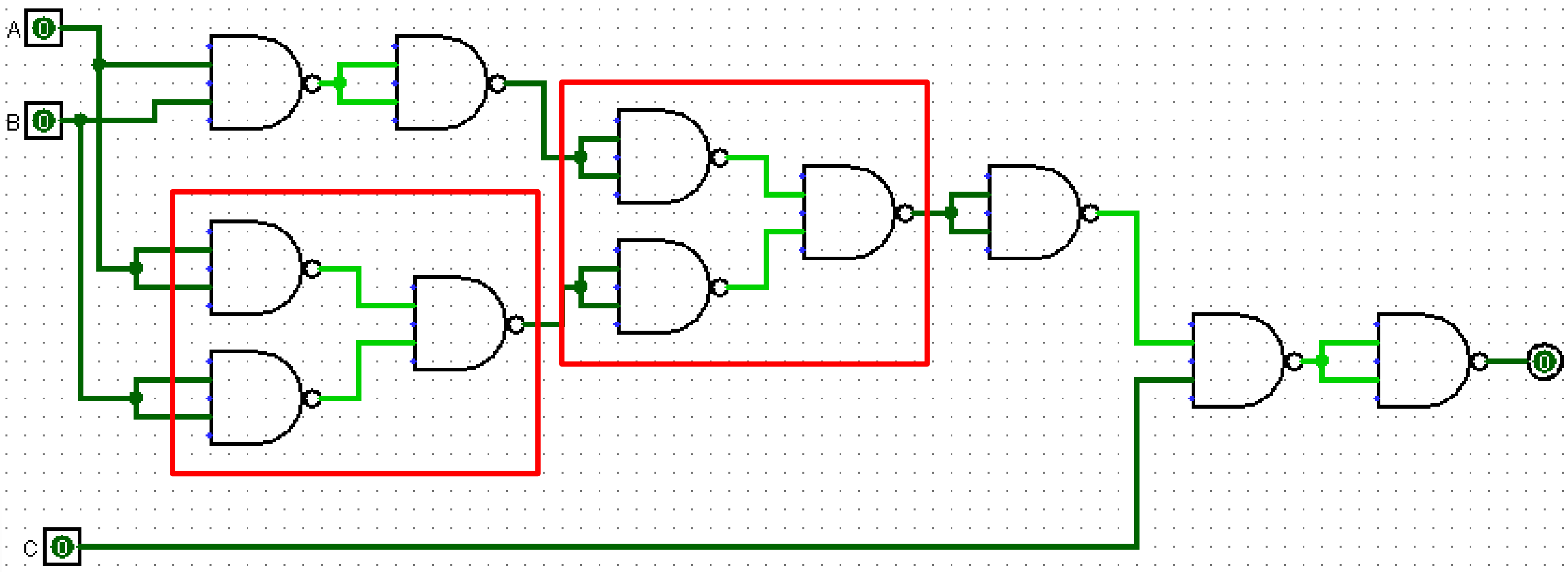
- Redesenhe o circuito usando apenas portas NAND




IMD0121

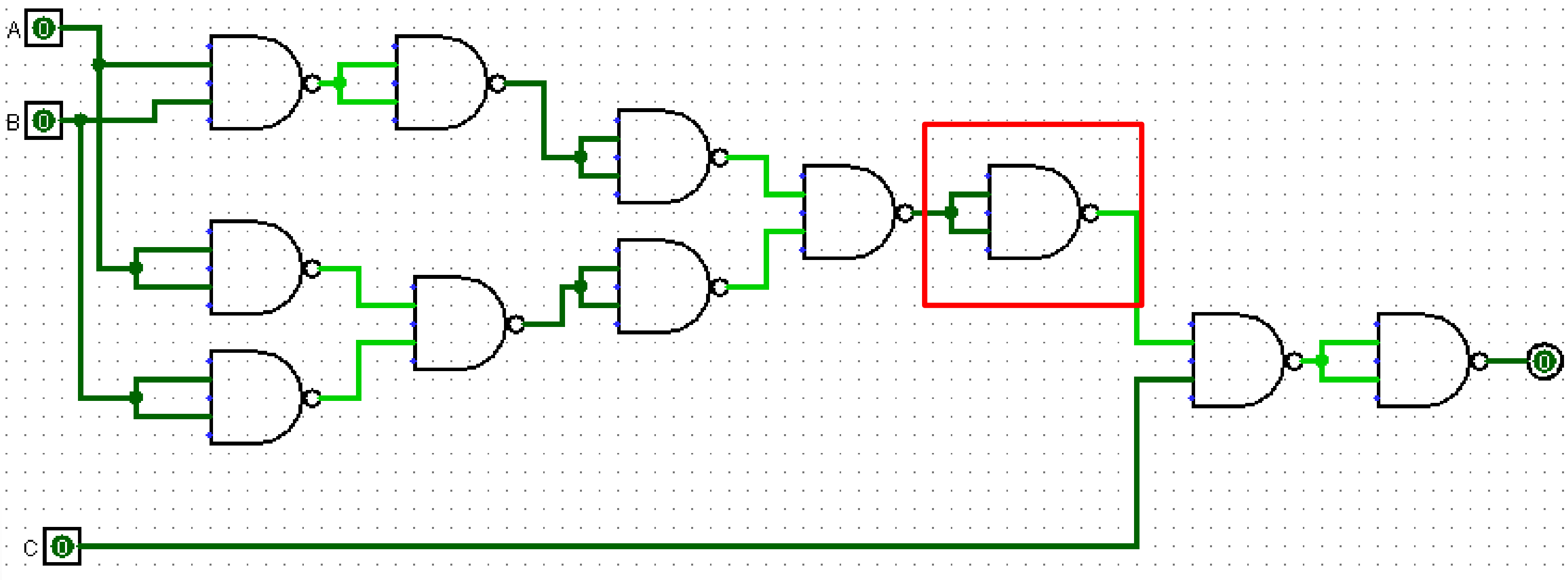
# Universalidade de Portas

- Redesenhe o circuito usando apenas portas NAND



# Universalidade de Portas

- Redesenhe o circuito usando apenas portas NAND

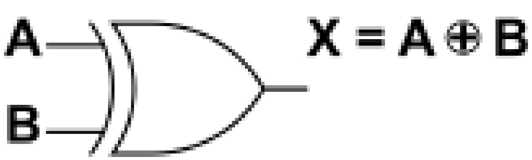



IMD0121

# Combinação de portas lógicas

- Porta XOR:
  - Conhecido como OU exclusivo
  - Na porta XOR de duas entradas
    - Saída = ’1’: entradas diferentes
    - Saída = ’0’: entradas iguais

$$X = A \oplus B$$




| Tabela Verdade | | |
|---|---|---|
| A | B | X |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Combinação de portas lógicas

- Porta XOR:
  - Como representar a porta XOR com portas lógicas básicas?
  - Conhecemos a tabela-verdade.
  - Como definir a equação booleana a partir da tabela-verdade?

# Combinação de portas lógicas

- Porta XOR:
  - Como representar a porta XOR com portas lógicas básicas?
  - Conhecemos a tabela-verdade.
  - Como definir a equação booleana a partir da tabela-verdade?
  - Utilizando MAXTERMOS ou MINTERMOS

| Tabela Verdade | | | | |
|---|---|---|---|---|
| A | B | X | Mintermos | Maxtermos |
| 0 | 0 | 0 | $A.B$ | $A + B$ |
| 0 | 1 | 1 | $A.B$ | $A + B$ |
| 1 | 0 | 1 | $A.B$ | $A + B$ |
| 1 | 1 | 0 | $A.B$ | $A + B$ |

IMD0121

# Combinação de portas lógicas

- Porta XOR:
  - Como representar a porta XOR com portas lógicas básicas?
  - Conhecemos a tabela-verdade.
  - Como definir a equação booleana a partir da tabela-verdade?
  - Utilizando MAXTERMOS ou MINTERMOS

| Tabela Verdade | | | | |
|---|---|---|---|---|
| A | B | X | Mintermos | Maxtermos |
| 0 | 0 | 0 | $A.B$ | $A + B$ |
| 0 | 1 | 1 | $A.B$ | $A + B$ |
| 1 | 0 | 1 | $A.B$ | $A + B$ |
| 1 | 1 | 0 | $A.B$ | $A + B$ |

Mintermos: $X = \bar{A}.B + A.\bar{B}$

Maxtermos: $X = (A + B).(\bar{A} + \bar{B})$

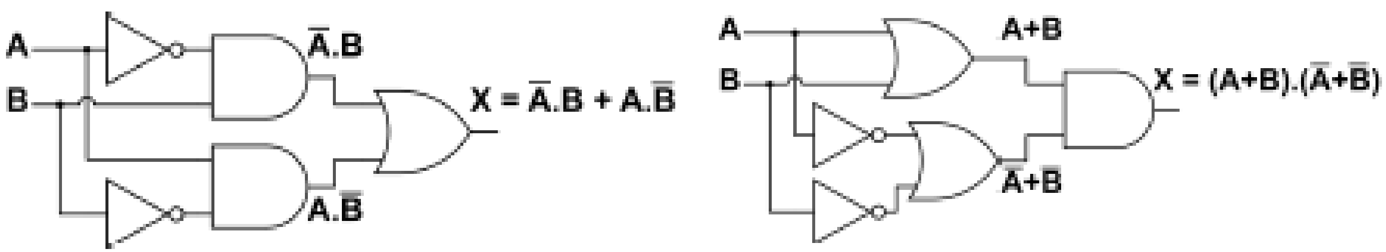
IMD0121

# Combinação de portas lógicas

- Porta XOR

Mintermos: $X = \bar{A}.B + A.\bar{B}$

Maxtermos: $X = (A + B).(\bar{A} + \bar{B})$




(k) Utilizando mintermos

(l) Utilizando maxtermos

Como implementamos uma XOR com NANDS? E Com NORs?

# Combinação de portas lógicas

- Porta XOR
  - Como funciona a porta XOR com mais de duas entradas?
  - Funciona como um detector de paridades
    - Saída = ’1’: o número de entradas iguais a ’1’ é ímpar
    - Saída = ’0’: o número de entradas iguais a ’1’ é par
  - O termo OU exclusivo funciona apenas para portas XOR com duas entradas
  - Como fica a tabela verdade?

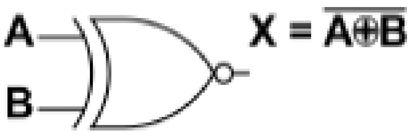
# Combinação de Portas Logicas

| Tabela Verdade | | | |
|---|---|---|---|
| A | B | C | X |
| 0 | 0 | 0 | ? |
| 0 | 0 | 1 | ? |
| 0 | 1 | 0 | ? |
| 0 | 1 | 1 | ? |
| 1 | 0 | 0 | ? |
| 1 | 0 | 1 | ? |
| 1 | 1 | 0 | ? |
| 1 | 1 | 1 | ? |

# Combinação de portas lógicas

- Porta XNOR:
  - Na porta XNOR de duas entradas
    - Saída = ’0’: entradas diferentes
    - Saída = ’1’: entradas iguais

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




| Tabela Verdade | | |
|---|---|---|
| A | B | X |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Combinação de portas lógicas

- Porta XNOR:
  - Como representar a porta XNOR com portas lógicas básicas?
  - Conhecemos a tabela-verdade.
  - Como definir a equação booleana a partir da tabela-verdade?
  - Utilizando MAXTERMOS ou MINTERMOS

| Tabela Verdade | | | | |
|---|---|---|---|---|
| A | B | X | Mintermos | Maxtermos |
| 0 | 0 | 1 | $A.B$ | $A + B$ |
| 0 | 1 | 0 | $A.B$ | $A + B$ |
| 1 | 0 | 0 | $A.B$ | $A + B$ |
| 1 | 1 | 1 | $A.B$ | $A + B$ |

# Combinação de portas lógicas

- Porta XNOR:
  - Como representar a porta XNOR com portas lógicas básicas?
  - Conhecemos a tabela-verdade.
  - Como definir a equação booleana a partir da tabela-verdade?
  - Utilizando MAXTERMOS ou MINTERMOS

| Tabela Verdade | | | | |
|---|---|---|---|---|
| A | B | X | Mintermos | Maxtermos |
| 0 | 0 | 1 | $A.B$ | $A+B$ |
| 0 | 1 | 0 | $A.B$ | $A+B$ |
| 1 | 0 | 0 | $A.B$ | $A+B$ |
| 1 | 1 | 1 | $A.B$ | $A+B$ |

Mintermos: $X = \bar{A}.\bar{B} + A.B$

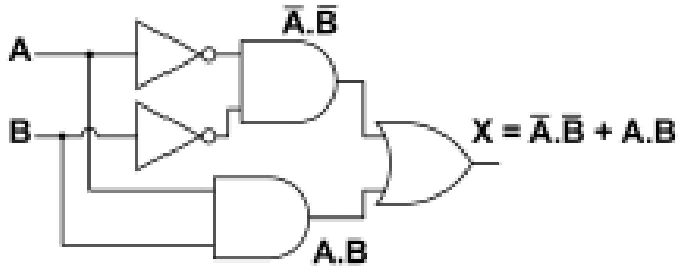
Maxtermos: $X = (A + \bar{B}).(\bar{A} + B)$

IMD0121

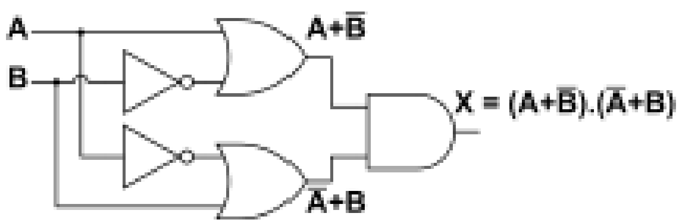
# Combinação de portas lógicas

Mintermos: $X = \bar{A}.\bar{B} + A.B$

Maxtermos: $X = (A + \bar{B}).(\bar{A} + B)$




(m) Utilizando mintermos

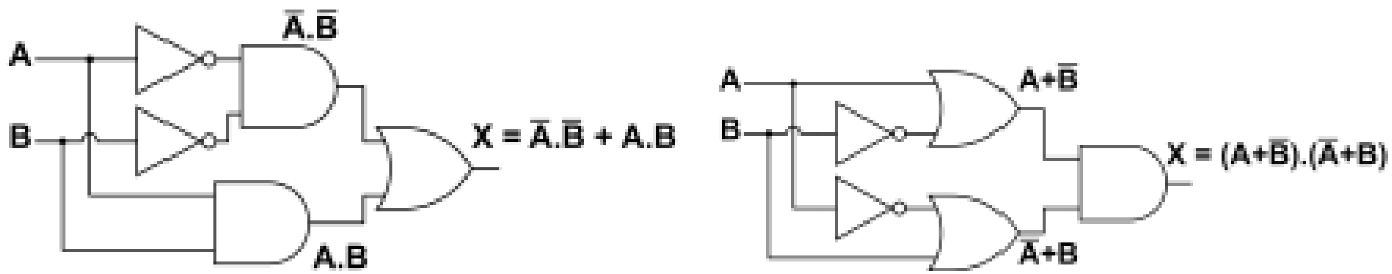



(n) Utilizando maxtermos

# Combinação de portas lógicas

Mintermos: $X = \bar{A}.\bar{B} + A.B$

Maxtermos: $X = (A + \bar{B}).(\bar{A} + B)$



(m) Utilizando mintermos (n) Utilizando maxtermos

IMD0121

# Combinação de portas lógicas

- Porta XNOR
  - Como funciona a porta XNOR com mais de duas entradas?
  - Funciona como um detector de paridades
    - Saída = ’0’: o número de entradas iguais a ’1’ é ímpar
    - Saída = ’1’: o número de entradas iguais a ’1’ é par
  - Como fica a tabela verdade com três entradas?

# Combinação de portas lógicas

| Tabela Verdade | | | |
|---|---|---|---|
| A | B | C | X |
| 0 | 0 | 0 | ? |
| 0 | 0 | 1 | ? |
| 0 | 1 | 0 | ? |
| 0 | 1 | 1 | ? |
| 1 | 0 | 0 | ? |
| 1 | 0 | 1 | ? |
| 1 | 1 | 0 | ? |
| 1 | 1 | 1 | ? |

# Combinação de portas lógicas

| Tabela Verdade | | | |
|---|---|---|---|
| A | B | C | X |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

# Referências


- **STALLINGS, William. Arquitetura e organização de computadores. 10. ed. São Paulo: Pearson, 2017. 814 p.**
  - **Capítulo 9**

- TOCCI, Ronald J; Widmer, Neal S. Sistemas Digitais: principios e Aplicações. 11. ed. São Paulo SP: Pearson, 2011, 817 p. ISBN 9788576050957
  - Capitulo 1

- PATTERSON, David A; HENNESSY, John L. Organização e projeto de computadores: A interface HARDWARE/SOFTWARE. Rio de Janeiro: Elsevier, 2005, 3ª edição.

# Combinação de Portas Lógicas

Prof. Gustavo Girão
**girao@imd.ufrn.br**

Baseado no material do Prof. Ricardo Weber (UFRGS)