SISTEMAS COMPUTACIONAIS
E
SEGURANÇA

PROF. RICARDO SCHWACH
MSC, MBA, PMP, SAFe, PSM, ITIL, COBIT

PORTA LÓGICA

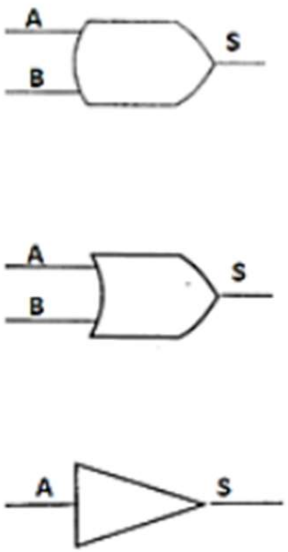


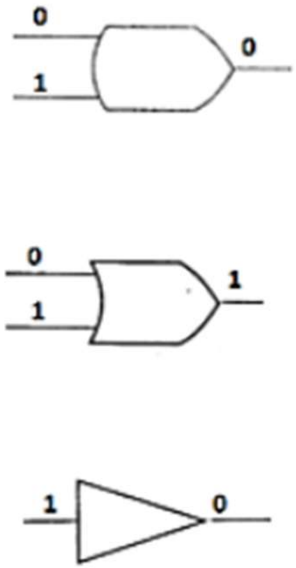
TABELA-VERDADE

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

A	S
0	1
1	0

EXEMPLO



Lógica de Boole e Portas Lógicas

PORTAS LÓGICAS

- **Portas lógicas** ou **circuitos lógicos**, são dispositivos que operam um ou mais sinais lógicos de entrada para produzir uma e somente uma saída, dependente da função implementada no circuito.
- São geralmente usadas em circuitos eletrônicos, por causa das situações que os sinais deste tipo de circuito podem apresentar: presença de sinal, ou "1"; e ausência de sinal, ou "0".
- As situações "Verdade" e "Falso" são estudadas na Lógica Matemática ou Lógica de Boole; origem do nome destas portas. O comportamento das portas lógicas é conhecido pela **Tabela Verdade** que apresenta os estados lógicos das entradas e das saídas.

Lógica de Boole e Portas Lógicas

Nos primórdios da eletrônica, os problemas eram resolvidos por meio de sistemas analógicos. Posteriormente, com o avanço da tecnologia, os problemas passaram a ser solucionados pela eletrônica digital e, desta forma, os computadores, os sistemas de controle, codificadores, decodificadores, entre outros, empregam um grupo de circuitos lógicos básicos que são conhecidos como portas "**e**", "**ou**", "**não**" e "**flip-flop**".

Combinando e utilizando adequadamente estas portas, é possível implementar as expressões geradas pela álgebra de Boole. Na álgebra de Boole há somente dois valores (estados ou símbolos), um oposto ao outro:

- 0 (zero) que representa não, falso, aparelho desligado, ausência de tensão, chave elétrica desligada.
- 1 (um) que representa sim, verdadeiro, aparelho ligado, presença de tensão, chave elétrica ligada.

Qualquer porta (função ou bloco) admite somente estes dois valores em suas entradas e saídas e, da mesma forma, uma variável booleana só aceita um dos dois valores permitidos (0 ou 1).

A tabela-verdade consiste em um mapa em que são colocadas todas as possíveis combinações (situações) com seus respectivos resultados, para uma expressão booleana qualquer.

Uma porta lógica é representada por um circuito eletrônico que recebe um ou mais sinais de entrada e produz um sinal de saída, cujo valor depende do tipo de regra lógica utilizada no referido circuito. Este conceito está detalhado na leitura a seguir:

Leia o tópico B.2 Portas e Operações Lógicas no Apêndice B - Conceito da Porta Lógica Digital (pág. 446 a 448)

https://integrada.minhabiblioteca.com.br/#/books/978-85-216-1973-4/cfi/459!/4/4@0.00:1.68

MONTEIRO, M. A. Introdução à organização de computadores. 5. ed. Rio de Janeiro: LTC, 2007.

Você Sabia?

- **Em meados do século XIX, um matemático inglês chamado George Boole, desenvolveu o sistema matemático de análise lógica. Posteriormente, no século XX, o americano Claude Shannon sugeriu que a Álgebra Booleana pudesse ser aplicada para a análise e projeto de circuitos.**

- **Portas lógicas possuem várias aplicações e grande importância no projeto de componentes da arquitetura dos computadores.**

- **É comum reunirmos várias portas lógicas e fazer combinações a fim de projetar circuitos diversos e, ainda, representá-los por meio da tabela-verdade.**

ATIVIDADE (20 min)

Fazer uma pesquisa sobre o Tema:

- Operadores lógicos na codificação de programas ou processos. Dê exemplos.

Cada linguagem de programação tem uma forma de representar os operadores lógicos. A simbologia mais encontrada são:

- **AND**, **OR** e **NOT** em linguagens como: Pascal, Visual Basic e SQL.
- &&, || e ! em linguagens como: Java e C#

► Proposição

A palavra “Proposição” vem do verbo “propor” que significa submeter à apreciação; requerer um juízo.

Trata-se de uma sentença declarativa – algo que será declarado por meio de termos, palavras ou símbolos – e cujo conteúdo poderá ser considerado verdadeiro ou falso.

▶ Proposição

A afirmação:
"Os gatos são felinos" representa uma proposição cujo valor lógico é verdadeiro.

Valor lógico - representa um dos dois possíveis juízos que se atribui a uma proposição: verdadeiro (V) ou falso (F).

► **Proposição**

A frase "Tenha um ótimo dia!"

É uma sentença para a qual se **não** se pode atribuir um valor lógico.

Para o estudo de **lógica** - interessa apenas as sentenças **declarativas** para as quais é possível atribuir o juízo de verdadeiro ou falso.

Proposição

As proposições são representadas por letras minúsculas, como por exemplo:

p

q

r

a

s

etc

Proposição

p: Rex é um cachorro.

q: 987 > 789.

r: Patrícia tomou café na semana passada.

Ao se afirmar que é verdade que Rex é um cachorro (proposição p acima), diz-se que o valor lógico de **p** é verdadeiro.

A proposição **q** também é verdadeira.

Raciocínio Lógico

O Raciocínio Lógico está sedimentado sobre alguns princípios:

Princípio da Identidade
Uma proposição verdadeira é verdadeira; uma proposição falsa é falsa.

Princípio da Não-Contradição
Nenhuma proposição poderá ser verdadeira e falsa ao mesmo tempo.

Princípio do Terceiro Excluído
Uma proposição ou será verdadeira, ou será falsa: não há outra possibilidade.

▶ Raciocínio Lógico

Proposições podem ser:

- Simples
- Compostas

► Raciocínio Lógico

Proposições Simples

São aquelas que vêm sozinhas, desacompanhadas de outras proposições.

Todo homem é mortal.

François é um gato de duas cores.

Raciocínio Lógico

- Rex é um cahorro da raça pastor alemão **e** Gutti é um cachorro brincalhão.
- Paulo vai ao shopping **ou** Laís vai ao teatro.
- **Ou** Caio é paulista, **ou** é mineiro.
- **Se** eu comprar o apartamento **então** não irei viajar.
- Abrirei uma escola **se e somente se** eu ganhar na loteria.

Nas sentenças apresentadas, estão em destaque **conectivos lógicos** que poderão estar presentes em uma proposição composta.

Conectivos lógicos são expressões que servem para unir duas ou mais proposições.

Conectivos Lógico

Uma proposição composta é verdadeira ou falsa, dependendo:

- do valor lógico das proposições componentes
- do tipo de conectivo que as une.

► Conectivos Lógico

Conectivo “e”: (conjunção)

Proposições compostas em que está presente o conectivo “e” são ditas conjunções.

Símbolo $\wedge$

▶ Conectivos Lógico

Conectivo “e”: (conjunção)

Rex é um cahorro da raça pastor alemão **e** Gutti é um cachorro brincalhão.

Representando em proposição

p = Rex é um cahorro da raça pastor alemão

e

q = Gutti é um cachorro brincalhão

Representação final....... $p \wedge q$

► Conectivos Lógico

Conectivo “e”: (conjunção)

$p \land q$

Uma conjunção só será verdadeira, se ambas as proposições componentes forem também verdadeiras

Conectivos Lógico

Ou Seja,

A sentença *"Rex é um cahorro da raça pastor alemão* **e** *Gutti é um cachorro brincalhão."*

Só será **verdadeira** se for verdade, ao mesmo tempo, p e q.

E para a sentença ser falsa?

Uma (ou mais) proposições componentes sendo falsa, e a conjunção será – toda ela – falsa.

Conectivos Lógico

Modo de representar:

Escrita formalizada = $p \wedge q$

Tabela verdade - serve para verificar as possíveis combinações

Conjuntos – representação gráfica

Conectivos Lógico

p = Rex é um cachorro da raça pastor alemão

e

q = Gutti é um cachorro brincalhão

Rex é um cahorro da raça pastor alemão	Gutti é um cachorro brincalhão	Rex é um cahorro da raça pastor alemão e Gutti é um cachorro brincalhão
p	q	p∧q
V	V	V

Conectivos Lógico

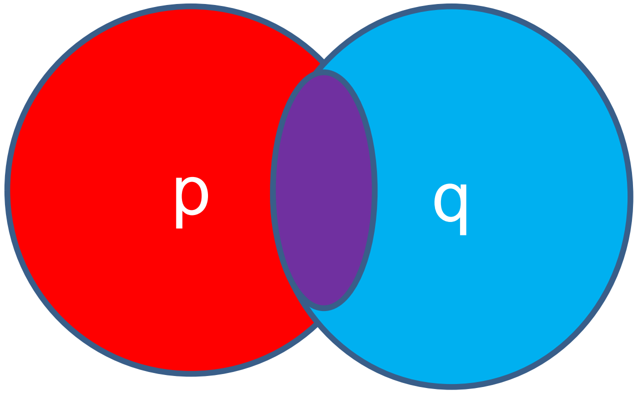
Mas como faço as possibilidades?

p	q	p∧q
V	V	V
V	F	F
F	V	F
F	F	F

Conectivos Lógico

Representando com conjuntos

A conjunção “p e q” corresponderá à interseção do conjunto p com o conjunto q.



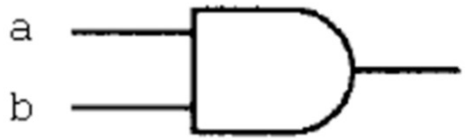
Conectivos Lógico

Representação

Porta AND

Entradas = a,b

Saída = ab



Atividade (5 min)

Representar a funcionalidade da **Porta Lógica AND** utilizando:

Simuladores:

- Falstad
- EasySim
- Logic Gate Simulator
- Logisim
- .....

Conectivos Lógico

Conectivo "ou": ***(disjunção)***

Recebe o nome de **DISJUNÇÃO** toda proposição composta em que as partes estejam unidas pelo conectivo **ou**.

Símbolo - ∨

Paulo vai ao shopping **ou** Laís vai ao teatro

Representação formal: **p** ∨ **q**.

- Uma disjunção será falsa quando as duas partes que a compõem forem ambas falsas.
- Nos demais casos, a disjunção será verdadeira.

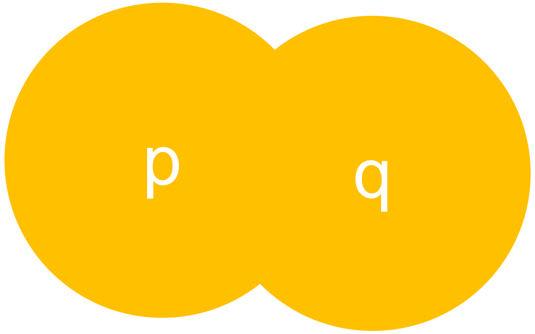
Conectivos Lógico

Representação Tabela Verdade

p	q	p q
V	V	V
V	F	V
F	V	V
F	F	F

Conectivos Lógico

Representação - Conjunto



Conectivos Lógico

Representação

Porta lógica OR

Entradas = a, b

Saída = a+b

Atividade (5 min)

Representar a funcionalidade da **Porta Lógica OR** utilizando:

Simuladores:

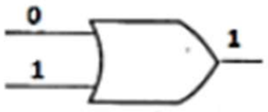
- Falstad
- EasySim
- Logic Gate Simulator
- Logisim
- .....

Lógica de Boole e Portas Lógicas

PORTA LÓGICA | TABELA-VERDADE | EXEMPLO

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1



A	S
0	1
1	0

▶ Conectivos Lógico

Conectivo “Ou ... ou ...”: (disjunção exclusiva)

Há uma diferença com o anterior, vamos analisar:

a) Ganharei um sorvete ou um chocolate.

b) Ou ganharei um sorvete ou um chocolate.

a) 1ª parte sendo verdade, não impede a 2ª parte de ser verdade também, ou seja, posso ganhar o sorvete e ganhar o chocolate.

b) Se for verdade a 1ª parte não será possível a 2ª parte também ser verdade. E vice-versa.

Conectivos Lógico

Símbolo $\veebar$

A sentença

Ou ganharei um sorvete **ou** um chocolate.

Esta sentença é denominada de DISJUNÇÃO EXCLUSIVA.

- Uma disjunção exclusiva só será verdadeira se obedecer à mútua exclusão das sentenças.
- Só será verdadeira se houver uma das sentenças verdadeira e a outra falsa. Nos demais casos, a disjunção exclusiva será falsa.

Conectivos Lógico

Tabela Verdade

p	q	$p \veebar q$
V	V	F
V	F	V
F	V	V
F	F	F

▶ Conectivos Lógico

Representação

Porta Lógica EXCLUSIVE OR

Entrada = a,b

Saída = a $\oplus$ b

▶ Conectivos Lógico

Conectivo "Se ... então ...": *(condicional)*

Se chover então o chão ficará molhado.

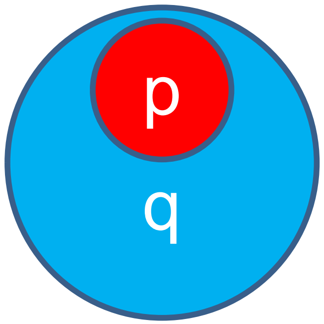
Se amanhecer chovendo então não irei à praia.

▶ **Conectivos Lógico**

Símbolo $\Rightarrow$

P $\longrightarrow$ **q**

Representação em conjunto



Corresponderá à inclusão do conjunto p no conjunto q (p está contido em q)

Conectivos Lógico

NEGAÇÃO “ não”

Negação da Proposição Simples

No caso de uma proposição simples, é só colocar palavra não antes da sentença, e já a tornamos uma negativa.

João é médico.

Negativa: João não é médico.

Maria é estudante.

Negativa: Maria não é estudante.

Conectivos Lógico

Negação da Proposição Simples

No caso a sentença original já seja uma negativa (já traga a palavra não), então para negar a negativa, teremos que excluir a palavra não.

João não é médico.

Negativa: João é médico.

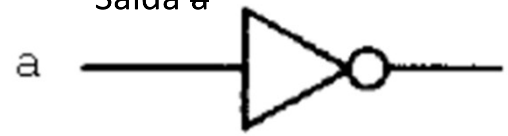
Maria não é estudante.

Negativa: Maria é estudante.

▶ Conectivos Lógico

Negação da Proposição Simples

Porta lógica NOT
Entrada a
Saída ~~a~~



Símbolo

Uma pequena cantoneira (¬) ou

Um sinal de til (~), antecedendo a sentença

Formalização

~p

Conectivos Lógico

Negação da Proposição Composta

Pode ocorrer em qualquer proposição da sentença.

Negação de uma proposição conjuntiva: ~(p e q)

Para negar uma proposição no formato de conjunção (p e q), deve-se fazer o seguinte:

1. Negaremos a primeira parte (~p);

2. Negaremos a segunda parte (~q);

3. Troca-se E por OU.

Conectivos Lógico

Negação da Proposição Composta

Exemplo: “Não é verdade que João é médico e Pedro é dentista”.

1. Nega-se a primeira parte (~p) = João não é médico;
2. Nega-se a segunda parte (~q) = Pedro não é dentista;
3. Troca-se E por OU, e o resultado final será o seguinte:

“João não é médico ou Pedro não é dentista”.

Traduzindo para a linguagem da lógica:

$$\sim(p \wedge q) = \sim p \vee \sim q$$

Atividade (5 min)

Representar a funcionalidade da **Porta Lógica AND e OR** utilizando **NOT**:

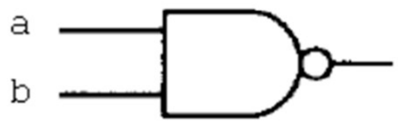
Simuladores:

- Falstad
- EasySim
- Logic Gate Simulator
- Logisim
- .....

Porta **NAND** (**NÃO-E**)

Conectivos Lógicos

Porta lógica – NAND



Entrada ab
Saída ~~ab~~

Porta **NAND (NÃO-E)**

Conectivos Lógico

Negação da Proposição Composta

Negação de uma proposição disjuntiva: ~(p ou q)

Para negar uma proposição no formato de disjunção (p ou q), faremos o seguinte:

1. Negaremos a primeira parte (~p);

2. Negaremos a segunda parte (~q);

3. Troca-se OU por E.

► Conectivos Lógico

"Pedro é dentista ou Paulo é engenheiro".

1. Nega-se a primeira parte (~p) = Pedro não é dentista;

2. Nega-se a segunda parte (~q) = Paulo não é engenheiro;

3. Troca-se OU por E, e o resultado final será o seguinte:

"Pedro não é dentista e Paulo não é engenheiro".

$$\sim(p \vee q) = \sim p \wedge \sim q$$

Atividade (5 min)

Representar a funcionalidade da **Porta Lógica NAND**:

Simuladores:

- Falstad
- EasySim
- Logic Gate Simulator
- Logisim
- .....

Lógica de Boole e Portas Lógicas

Conectivos Lógicos

Porta lógica – NOR

Entrada a b
Saída ~~a + b~~

▶ Conectivos Lógico

Negação de uma proposição condicional

$\sim(p \rightarrow q)$

Como é que se nega uma condicional?

► Conectivos Lógico

Negação de uma proposição condicional

1º) Mantém-se a primeira parte;

2º) Nega-se a segunda parte.

Conectivos Lógico

Negação de uma proposição condicional

Exemplo: “Se chover, então levarei o guarda-chuva”.

1º) Mantendo a primeira parte: “Chover”

2º) Negando a segunda parte: “não levarei o guarda-chuva”.

▶ Conectivos Lógico

Negação de uma proposição condicional

Resultado final: “Chove e eu não levo o guarda-chuva”.

Formalizando

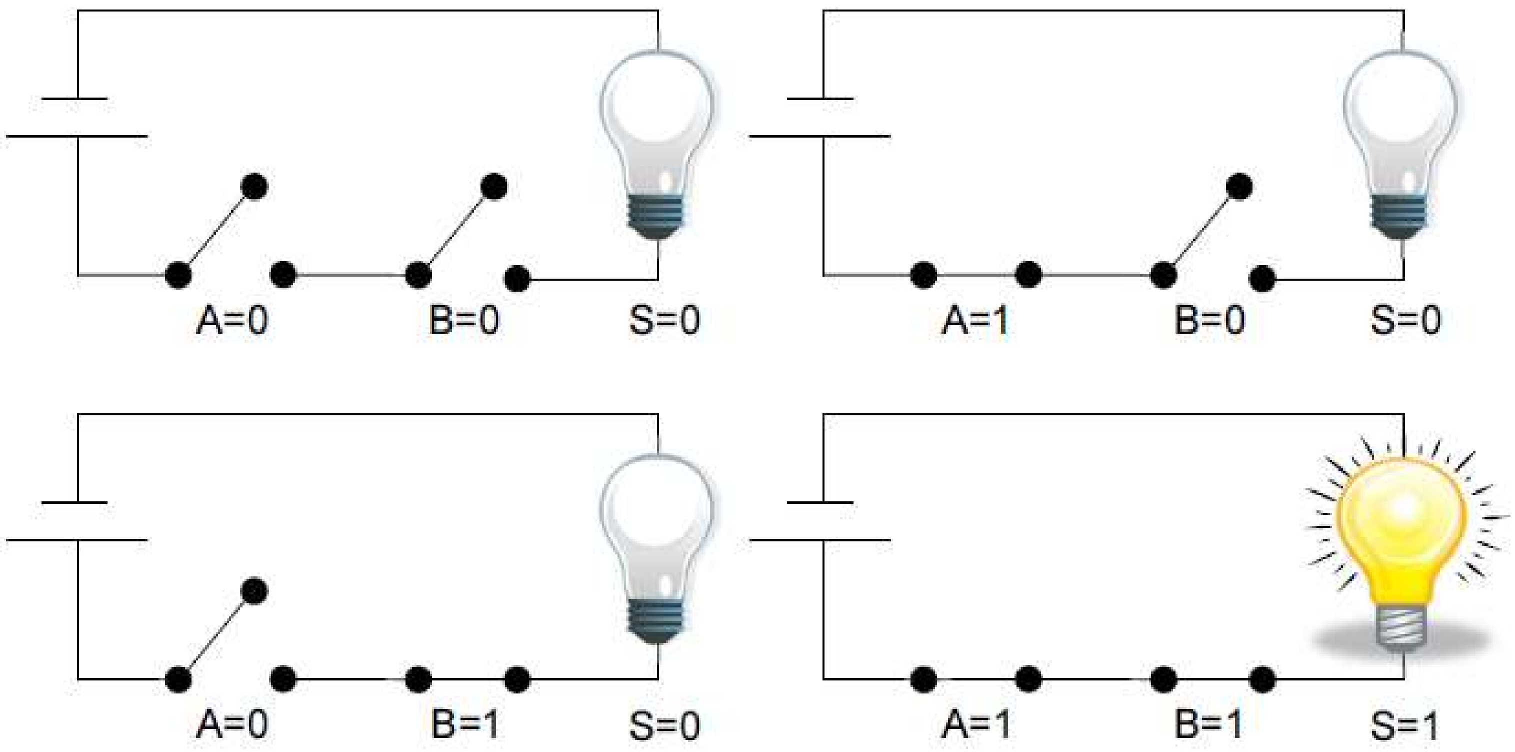
$$\sim(p \rightarrow q) = p \wedge \sim q$$

Atividade (5 min)

Representar a funcionalidade da **Porta Lógica NOR**:

Simuladores:

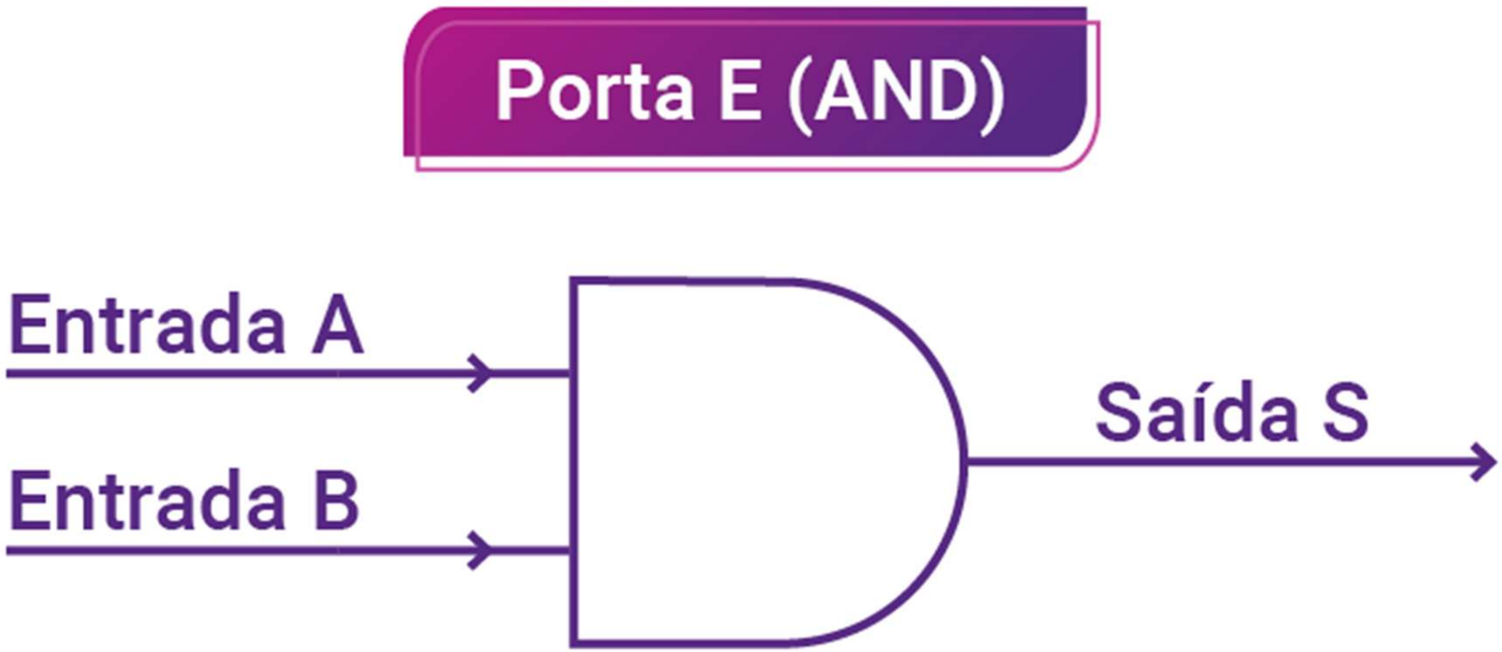
- Falstad
- EasySim
- Logic Gate Simulator
- Logisim
- .....


A=0
B=0
S=0
A=1
B=0
S=0
A=0
B=1
S=0
A=1
B=1
S=1

Porta AND ou Porta E

Porta Lógica E (AND)
A porta **E** (**AND**) é um circuito que executa a função **E**. Esta função é chamada de conectivo **E** ou ***conjunção***.

Representação:

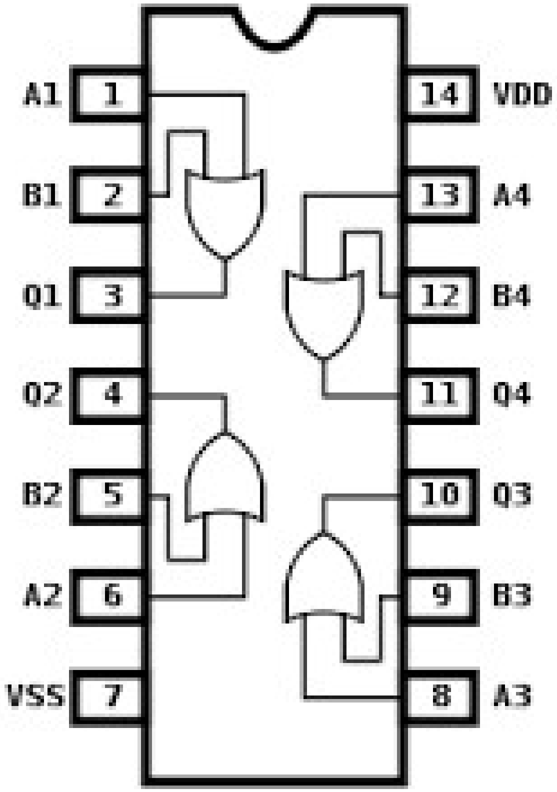


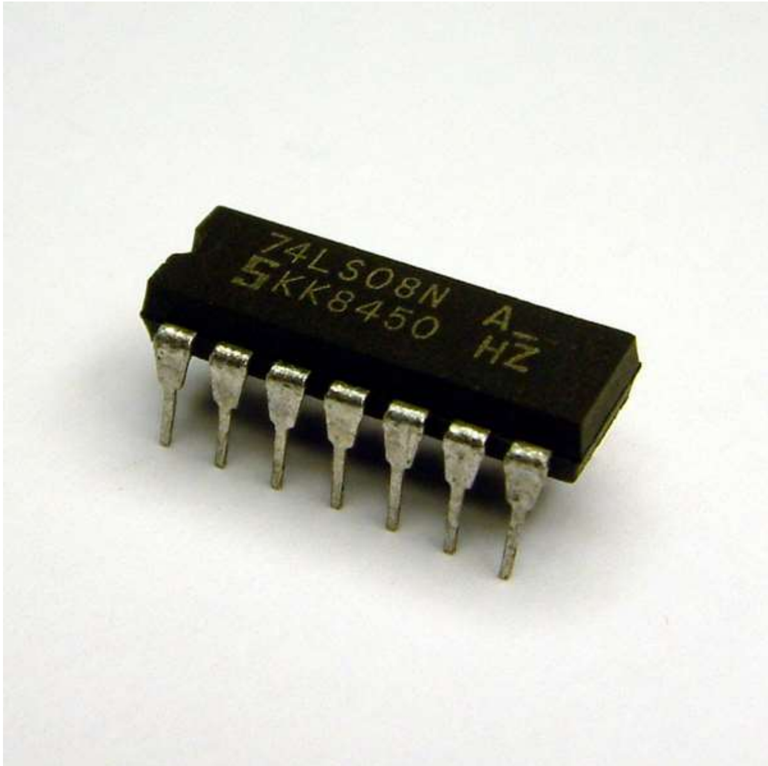
Porta AND ou Porta E

A expressão S = A e B é denotada por S = A ^ B, ou S = A & B, ou apenas S = A.B.

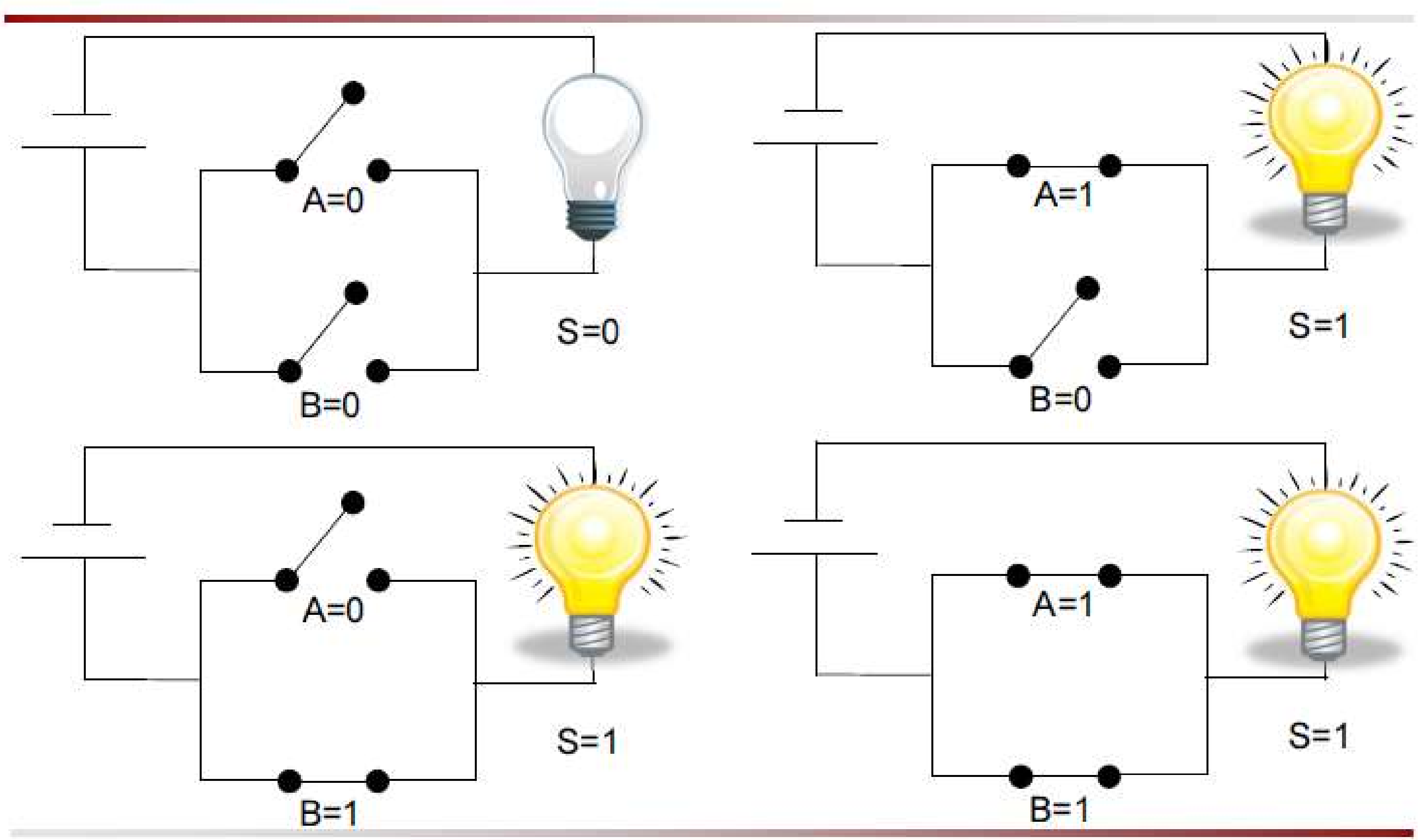
A porta E (AND) executa a tabela-verdade da função E: a saída será 1 somente se ambas as entradas forem iguais a 1. Nos demais casos, a saída será 0. A Tabela exibe a tabela-verdade da função E:

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1


A1
1
B1
2
Q1
3
Q2
4
B2
5
A2
6
VSS
7
14
VDD
13
A4
12
B4
11
Q4
10
Q3
9
B3
8
A3

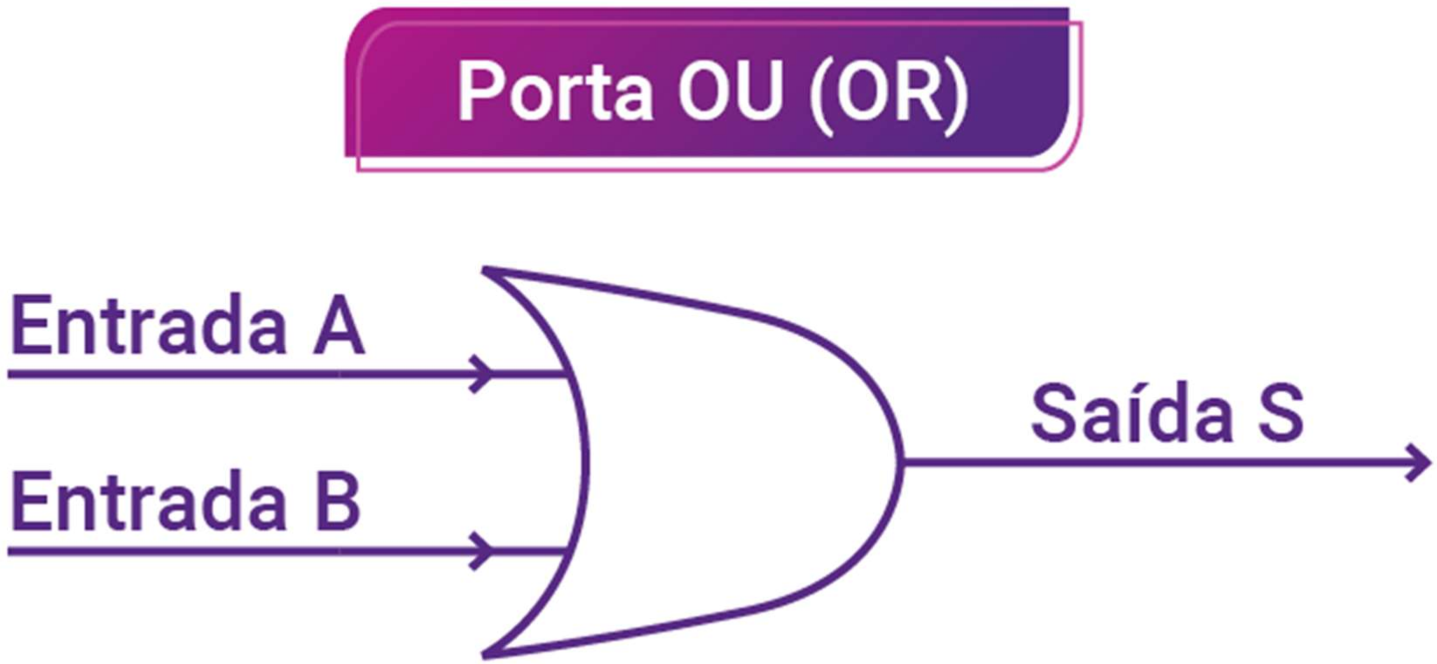

74LS08N
SKK8450
A
HZ

Porta **OR** ou Porta **OU**



A porta OU (OR) é um circuito que executa a função OU. Esta função é chamada de disjunção.

Representação:



- Operações lógicas com bits
  - **OR ou OU**
    - operação que aceita dois operandos
      - operando são binários simples (base 2)
    - operação OR é
      - 0 or 0 = 0
      - 0 or 1 = 1
      - 1 or 0 = 1
      - 1 or 1 = 1
    - Em português:
      - “se o primeiro operando é 1 ou o segundo operando é 1 (ou os dois), o resultado é 1, senão o resultado é 0”
    - Conhecido como OR-INCLUSIVE

Porta **OR** ou Porta **OU**

A expressão S = A ou B é denotada por S = A ∨ B ou S = A + B.

A porta OU (OR) executa a tabela-verdade da função OU: a saída será 0 somente se ambas as entradas forem iguais a 0. Nos demais casos, a saída será 1. A Tabela exibe a tabela-verdade da função OU:

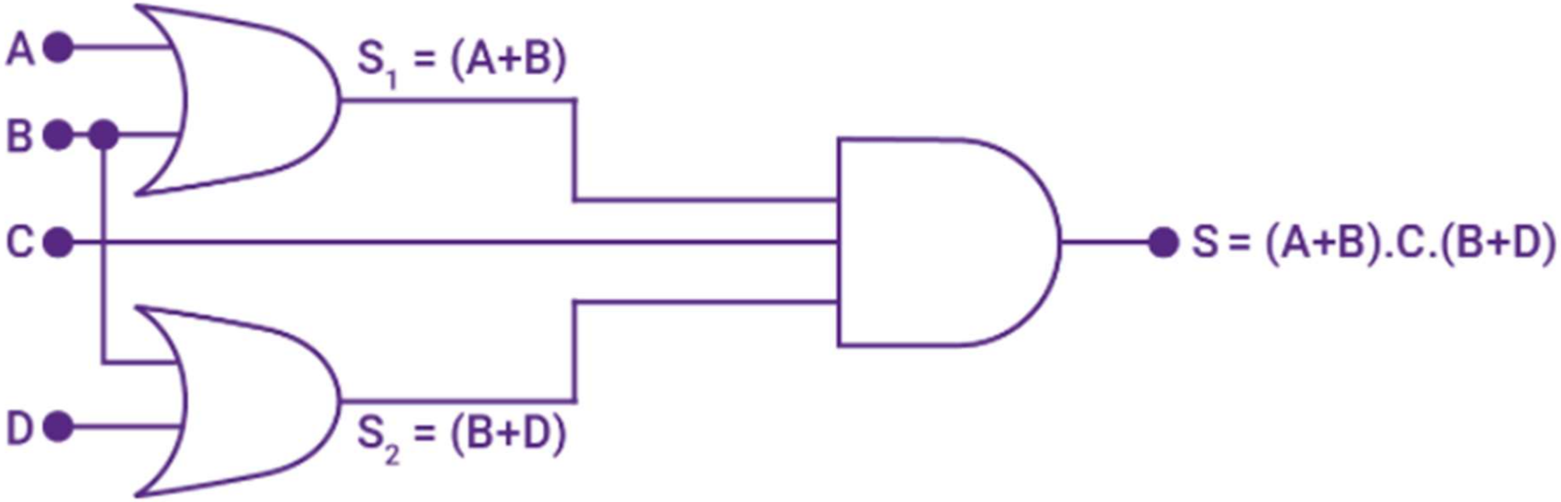
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Atividade (10 min)

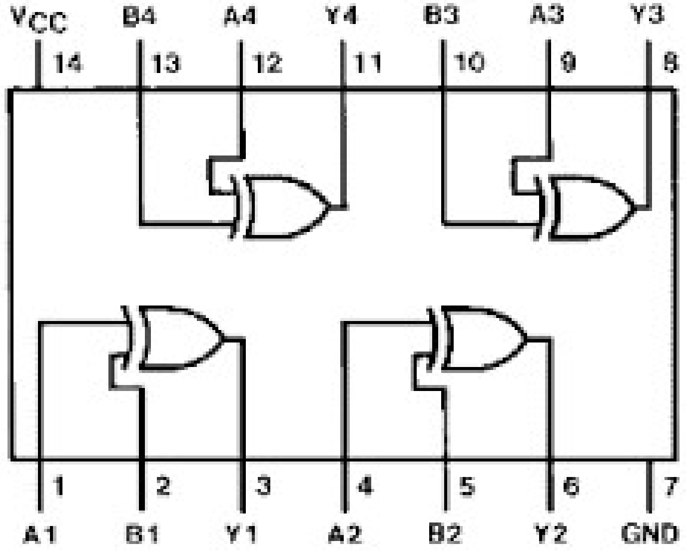
Dada a expressão booleana S = (A+B).C.(B+D), representar o circuito lógico correspondente.

Resposta

Dada a expressão booleana S = (A+B).C.(B+D), representar o circuito lógico correspondente.

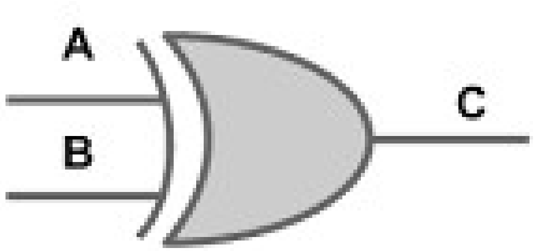


Porta **XOR** ou Porta **OU Exclusivo**



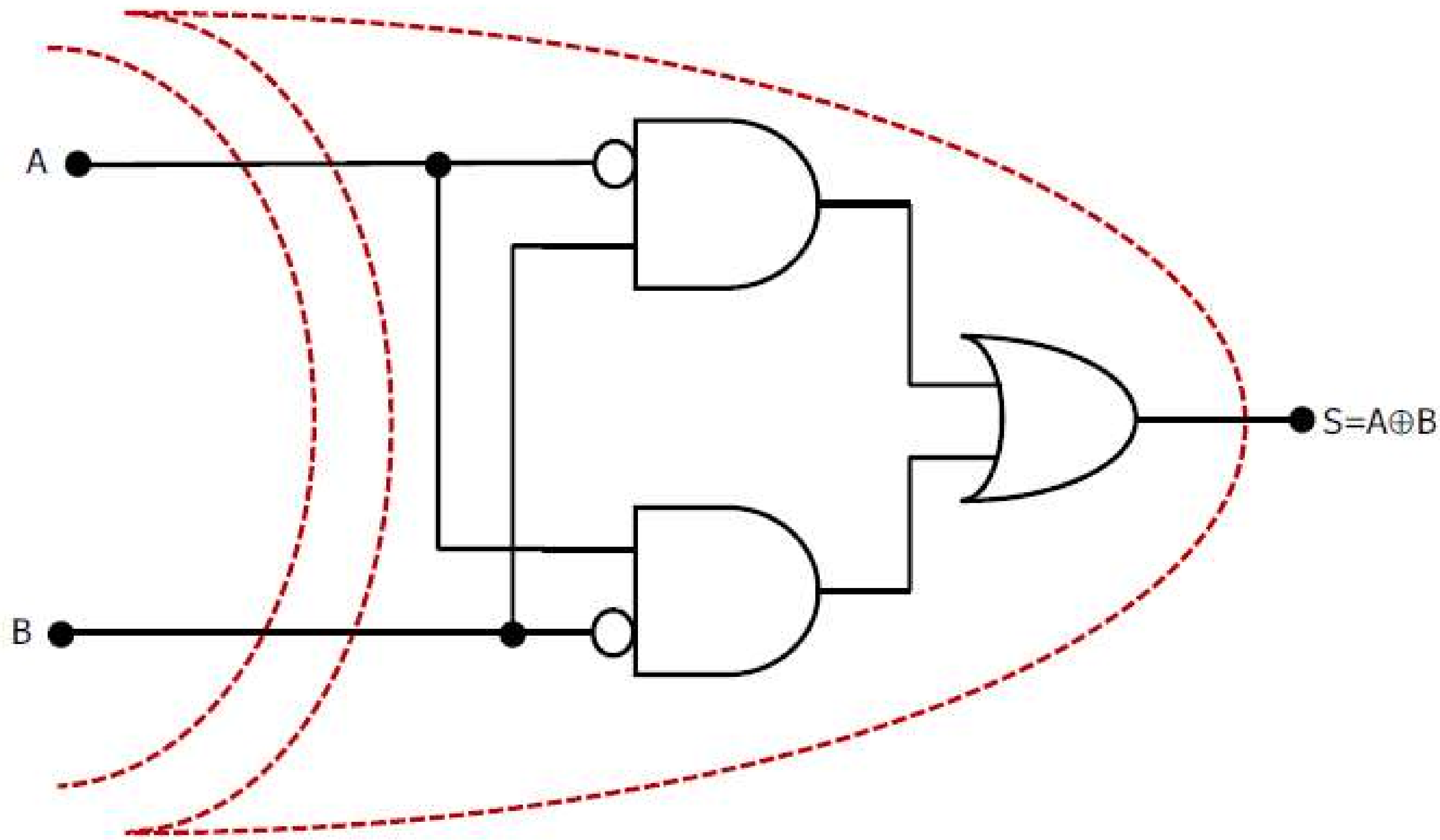
Porta **XOR** ou Porta **OU Exclusivo**

PORTA OU EXCLUSIVO (XOR) $C = A \oplus B$



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

Porta **XOR** ou Porta **OU Exclusivo**



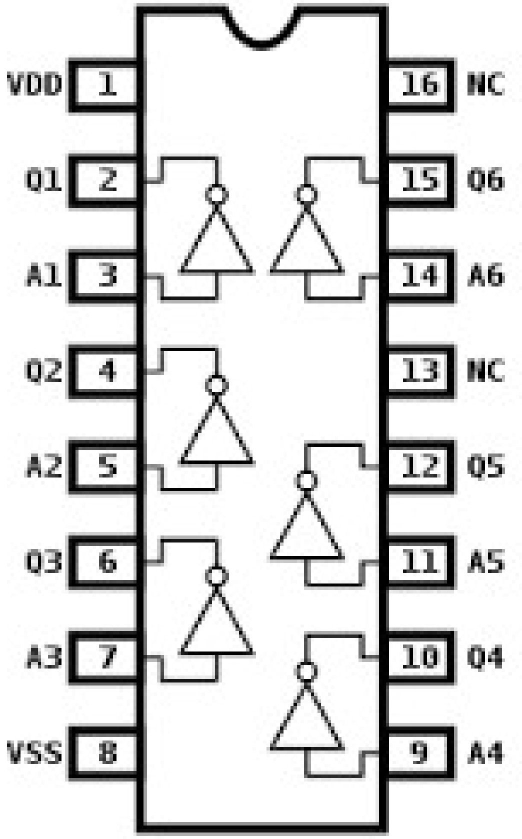
Porta **XOR** ou Porta **OU Exclusivo**

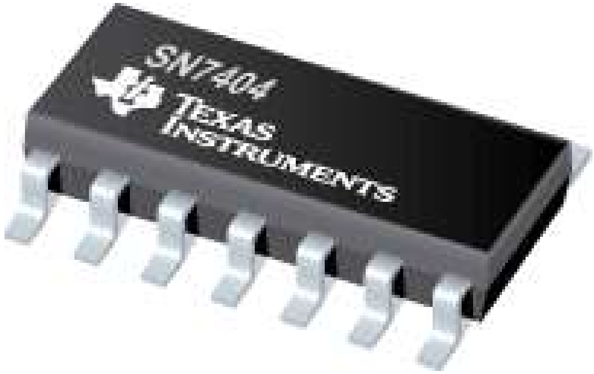
- Operações lógicas com bits
  - **XOR ou OU EXCLUSIVO**
    - operação que aceita dois operandos
      - operando são binários simples (base 2)
    - operação OR é
      - 0 xor 0 = 0
      - 0 xor 1 = 1
      - 1 xor 0 = 1
      - 1 xor 1 = 0
    - Em português:
      - “Se o primeiro operando ou o segundo operando, mas não os dois, for 1, o resultado é 1, senão o resultado é 0”

- Operações lógicas com bits
  - **XOR – Tabela Verdade**

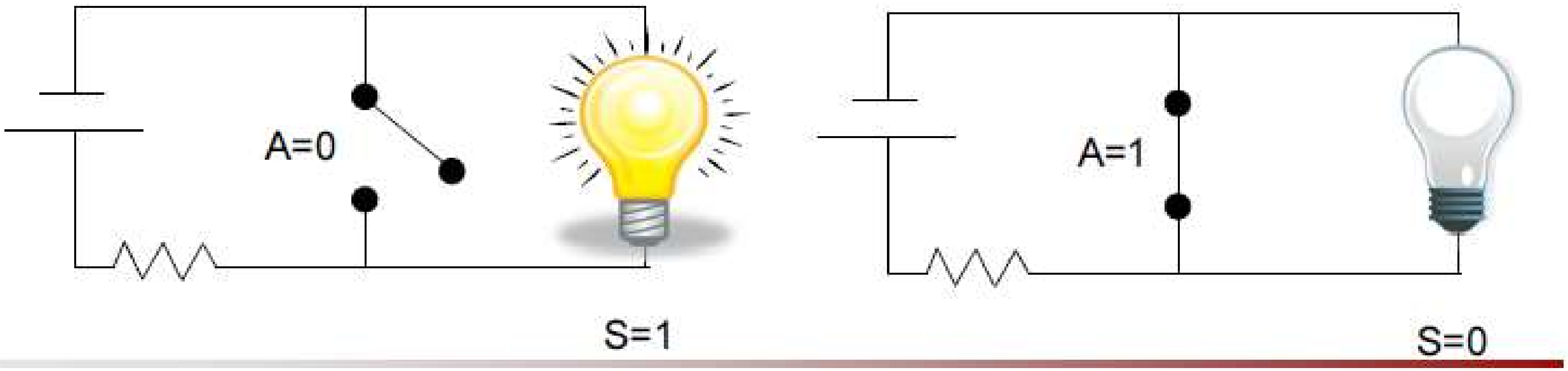
Op1	Op2	Op1 XOR Op2
0	0	0
0	1	1
1	0	1
1	1	0

Porta **NOT** ou Porta **NÃO**





Porta **NOT** ou Porta **NÃO**



Porta **NOT** ou Porta **NÃO**

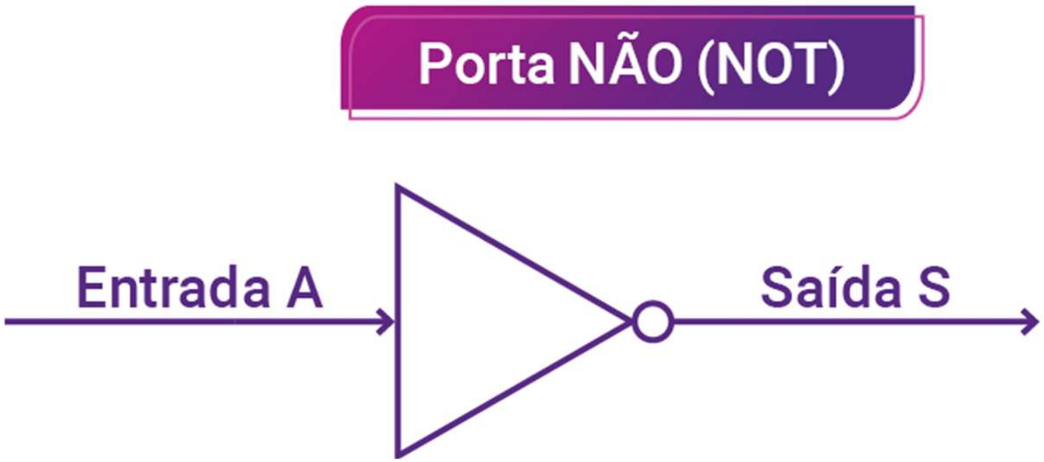
Porta Lógica NÃO (NOT)

A porta NÃO (NOT) é um circuito que executa a função NÃO. Esta função é o complemento ou a negação de uma variável binária.

- Se a variável estiver em 0, o resultado da função é 1;
- Se a variável estiver em 1, o resultado da função é 0.

Por isso, essa função é também chamada de inversora.

Representação:



Porta **NOT** ou Porta **NÃO**

A expressão S = não A é denotada por S = A’ ou S = ~A.

A porta NÃO (NOT) executa a tabela verdade da função NÃO: a saída será 1 se a entrada for 0 e a saída será 0 se a entrada for 1. A Tabela exibe a tabela-verdade da função NOT:

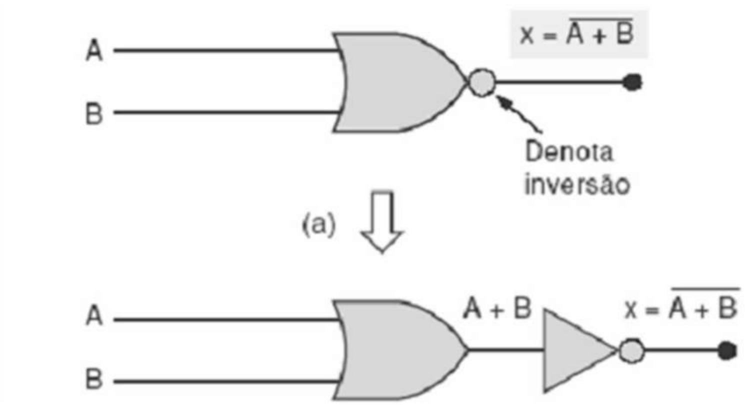
A	A'
0	1
1	0

Questão

Qual será a tabela verdade de uma porta OR com uma função NOT ligada à sua saída? E para uma porta AND?

Porta **NOR** ou Porta **NÃO OU**

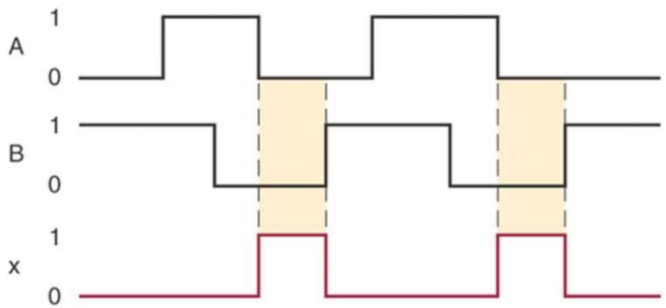
Combinação da porta OU com a porta inversora (NOT)

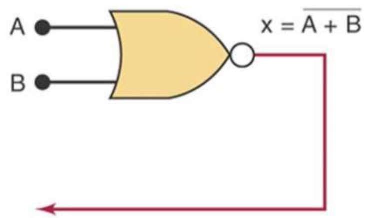


(b)

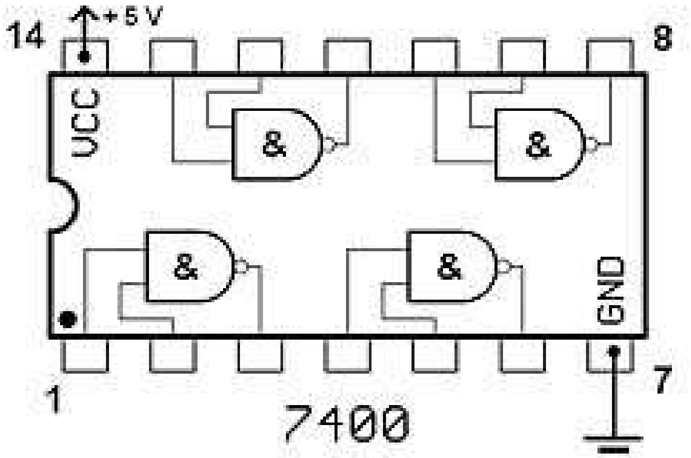
		OR	NOR
A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

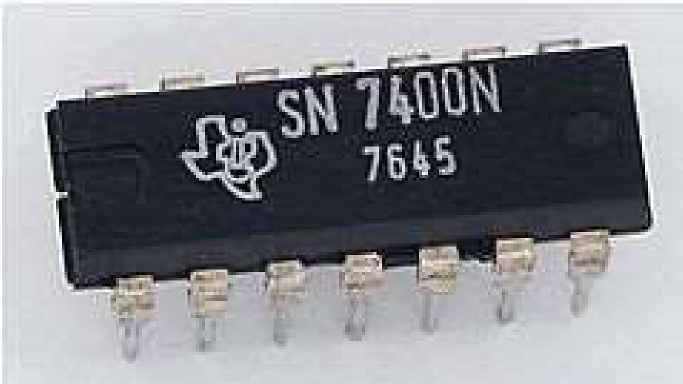
(c)

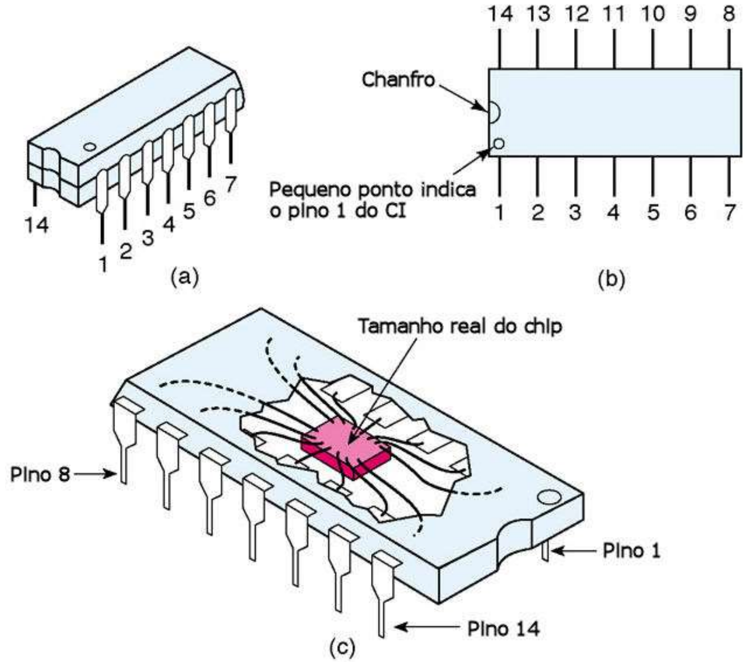




Porta **NAND** ou Porta **NE**

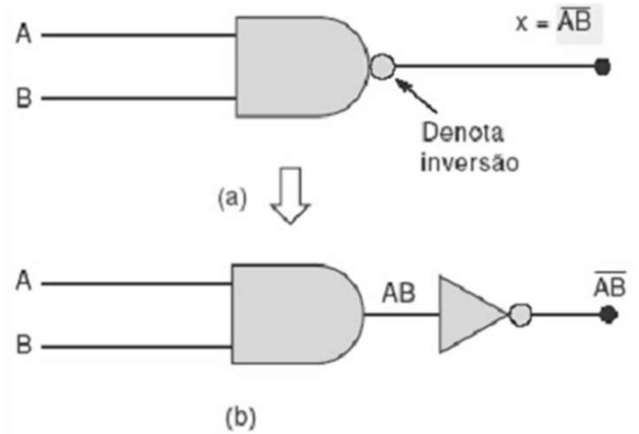






Porta **NAND** (**NÃO-E**)

Combinação da porta AND com a porta inversora (NOT)



		AND	NAND
A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(c)

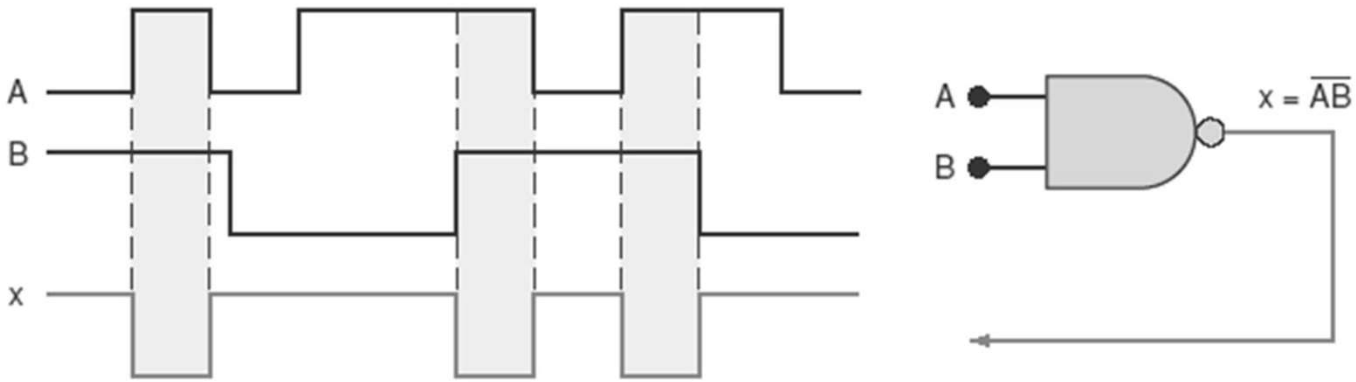


FIGURA 3.23
Exemplo 3.10.

- Operações Lógicas com números
  - As operações lógicas trabalham apenas com operandos com bit único
    - Para realizar estas operações sobre um número (8, 16, 32 bits) é necessário realizar a operação bit-a-bit

Exemplo: operação lógica AND com dois operandos de 8 bits

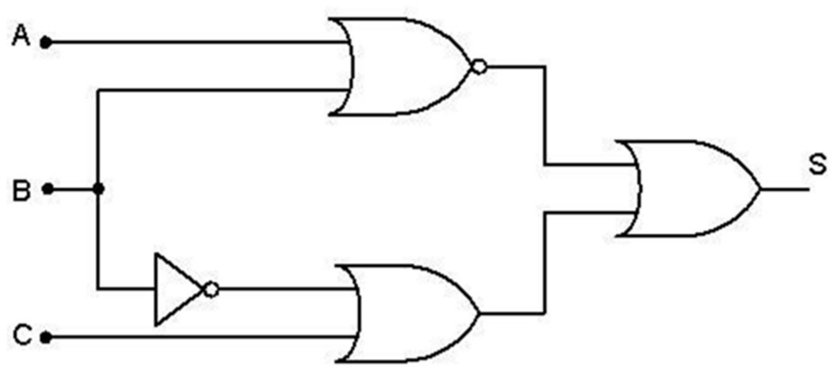
	1011 0101
AND	1110 1110
	1010 0100

  - Como as operações lógicas são definidos em termos de valores binários, deve-se converter os números decimais, hexadecimais, etc., para números binários antes de realizar as operações lógicas

RESUMO

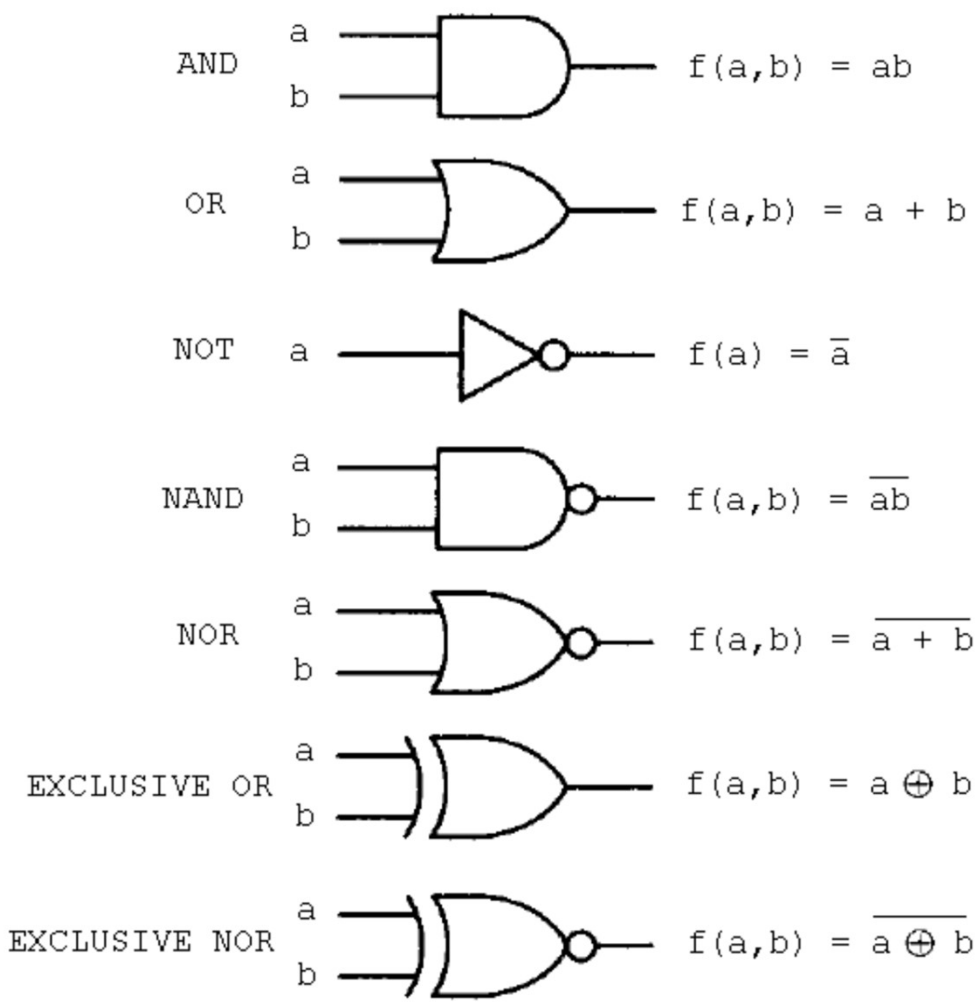
Quando aplicamos as lógicas em eletrônica, denominamos portas lógicas.

Sendo que a saída de uma porta pode ser a entrada de outra.



RESUMO

Portas Lógicas



Das portas Lógicas em bits 1 e 0.

AND

Entradas

a,b

Saída

c

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

Das portas Lógicas em bits 1 e 0.

OR

Entradas

a,b

Saída

c

a	b	c
0	0	0
0	1	1
1	0	1
1	1	1

Das portas Lógicas em bits 1 e 0.

NOT

Entradas

a

Saída

c

a	c
0	1
1	0

Das portas Lógicas em bits 1 e 0.

NOR

Entradas

a,b

Saída

c

a	b	c
0	0	1
0	1	0
1	0	0
1	1	0

Das portas Lógicas em bits 1 e 0.

AND

Entradas

a,b

Saída

c

a	b	c
0	0	1
0	1	1
1	0	1
1	1	0

RESUMO

Das portas Lógicas em bits 1 e 0.

XOR

Entradas

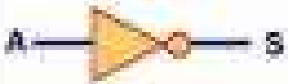
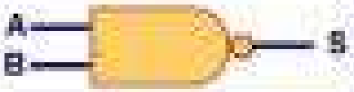
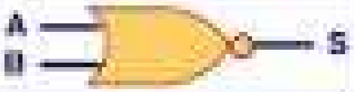
a,b

Saída

c

a	b	c
0	0	0
0	1	1
1	0	1
1	1	0

Portas Lógicas - Símbolos

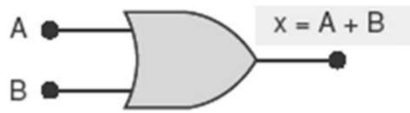
NOME	Símbolo Gráfico	Símbolo Algébrico
NOT		$S = \overline{A}$ ou $S = A'$
AND		$S = A \cdot B$ ou $S = AB$
OR		$S = A + B$
NAND		$S = (\overline{A B})$
NOR		$S = (\overline{A + B})$
XOR		$S = A \oplus B$

Quadro Resumo

- Blocos Lógicos básicos

OR

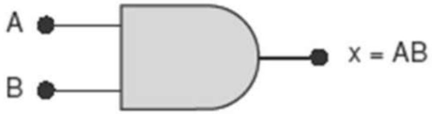
A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1



Porta OR

AND

A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1



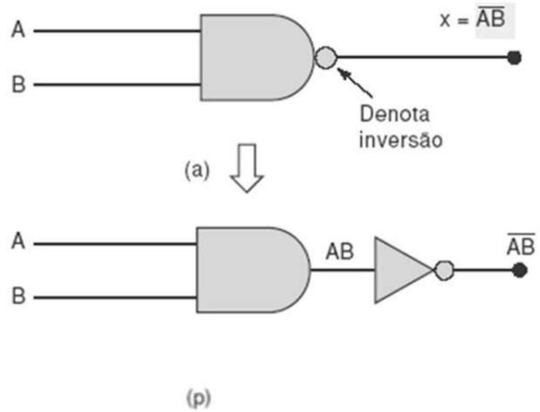
Porta AND

NOT

A	x = $\overline{A}$
0	1
1	0

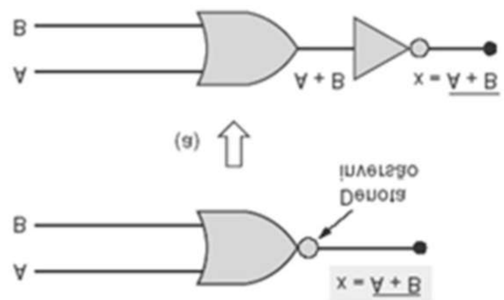
NOT

- Blocos Lógicos derivados



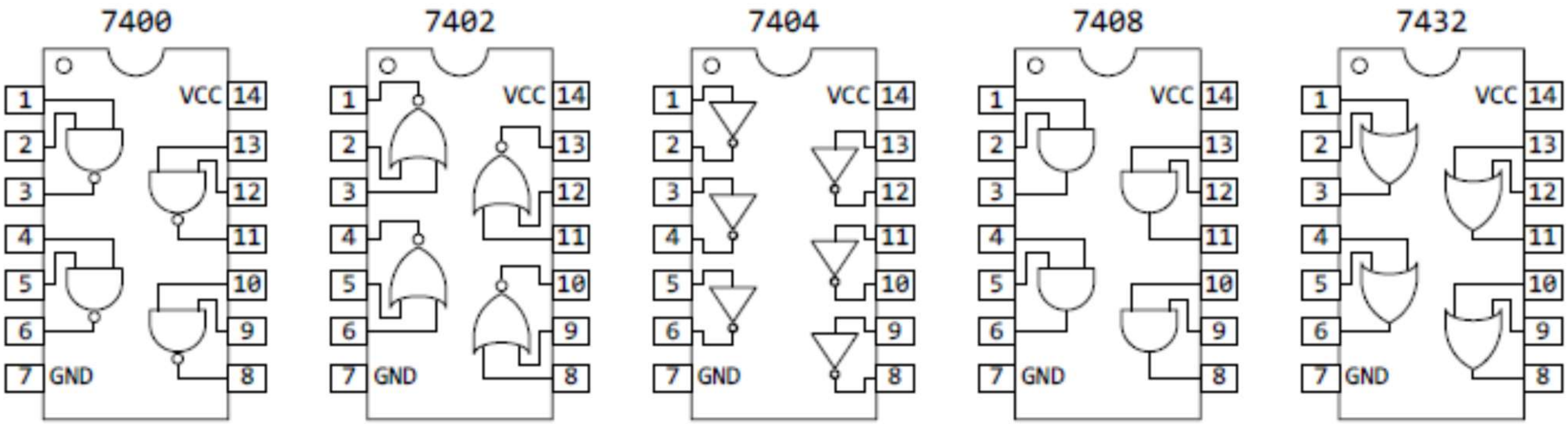
		AND	NAND
A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(c)



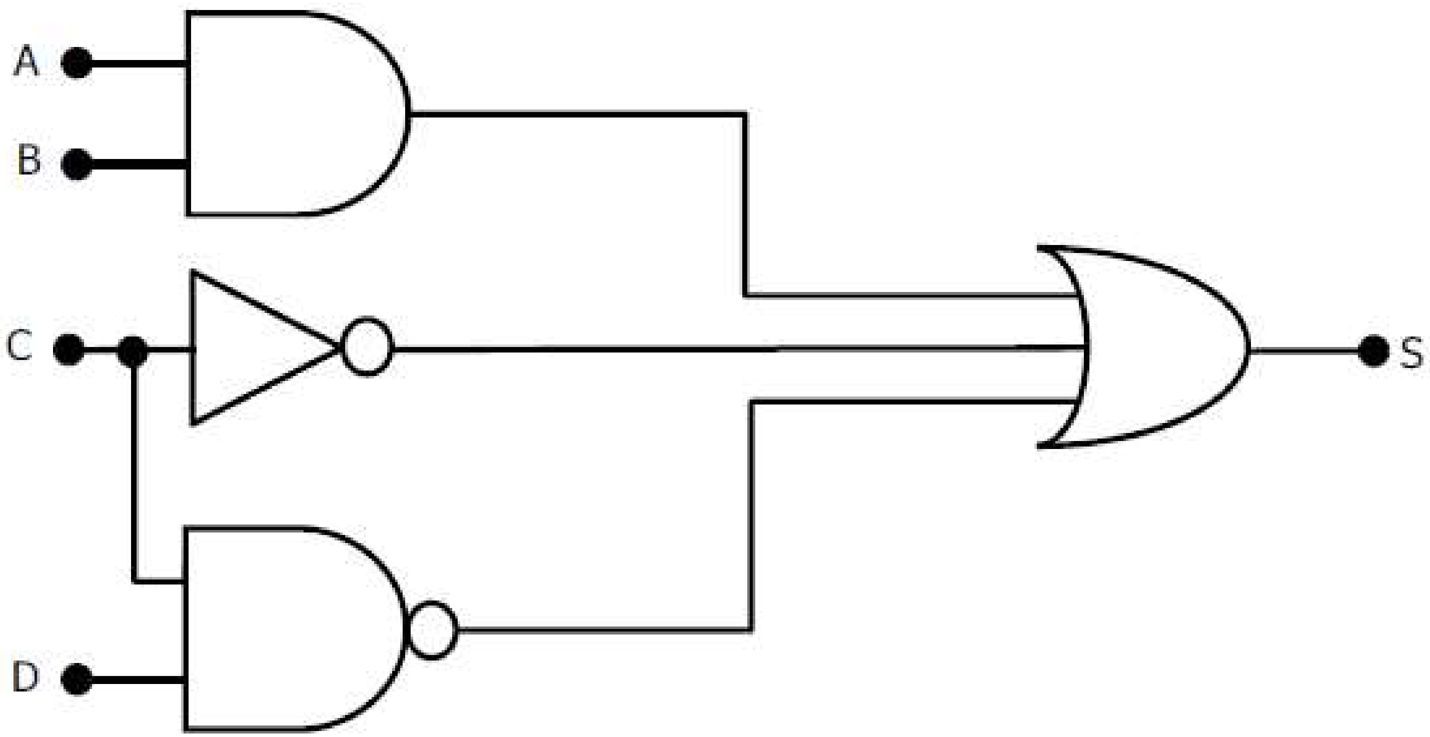
		OR	NOR
A	B	A + B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Circuitos Integrados



Exercício 1

- Escreva a expressão booleana executada pelo circuito abaixo



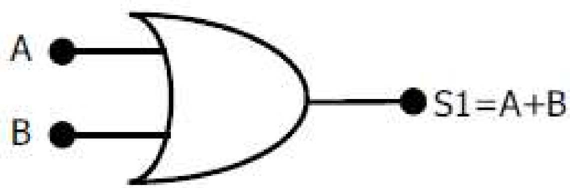
Exercício 2

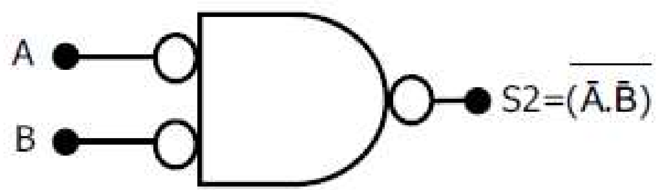
- Desenhe o circuito lógico cuja expressão característica é

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

Exercício 3

- Prove, usando tabela verdade, que os seguintes blocos lógicos são equivalentes

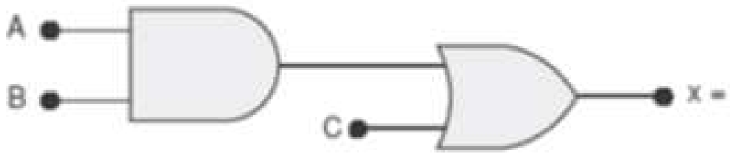




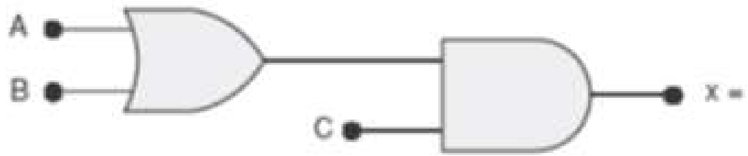
Exercício 4

- Obtenha a expressão de X:

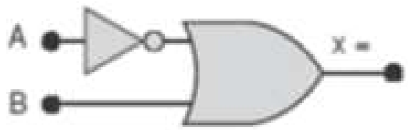
A)

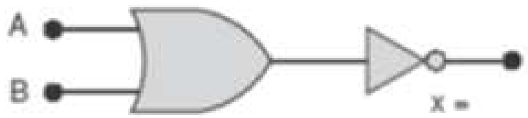


B)



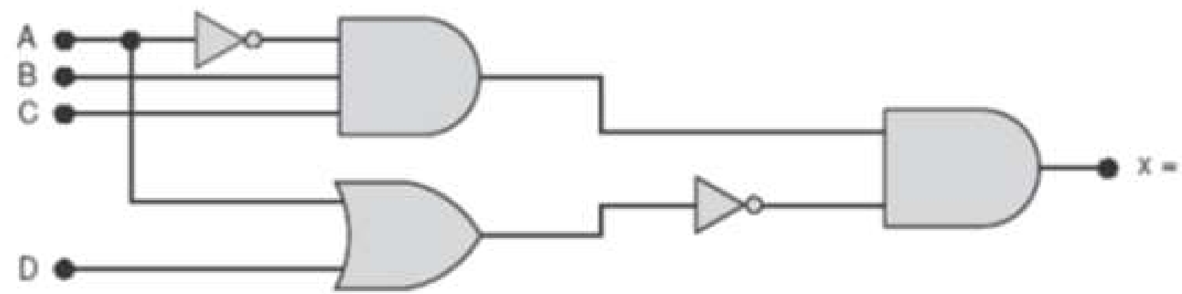
C)



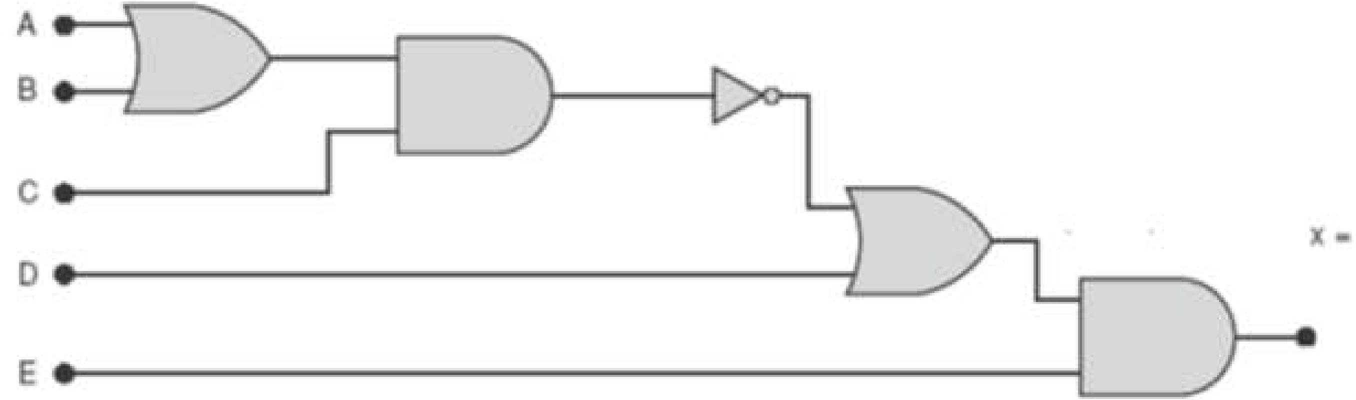


- Obtenha a expressão de X:

A)

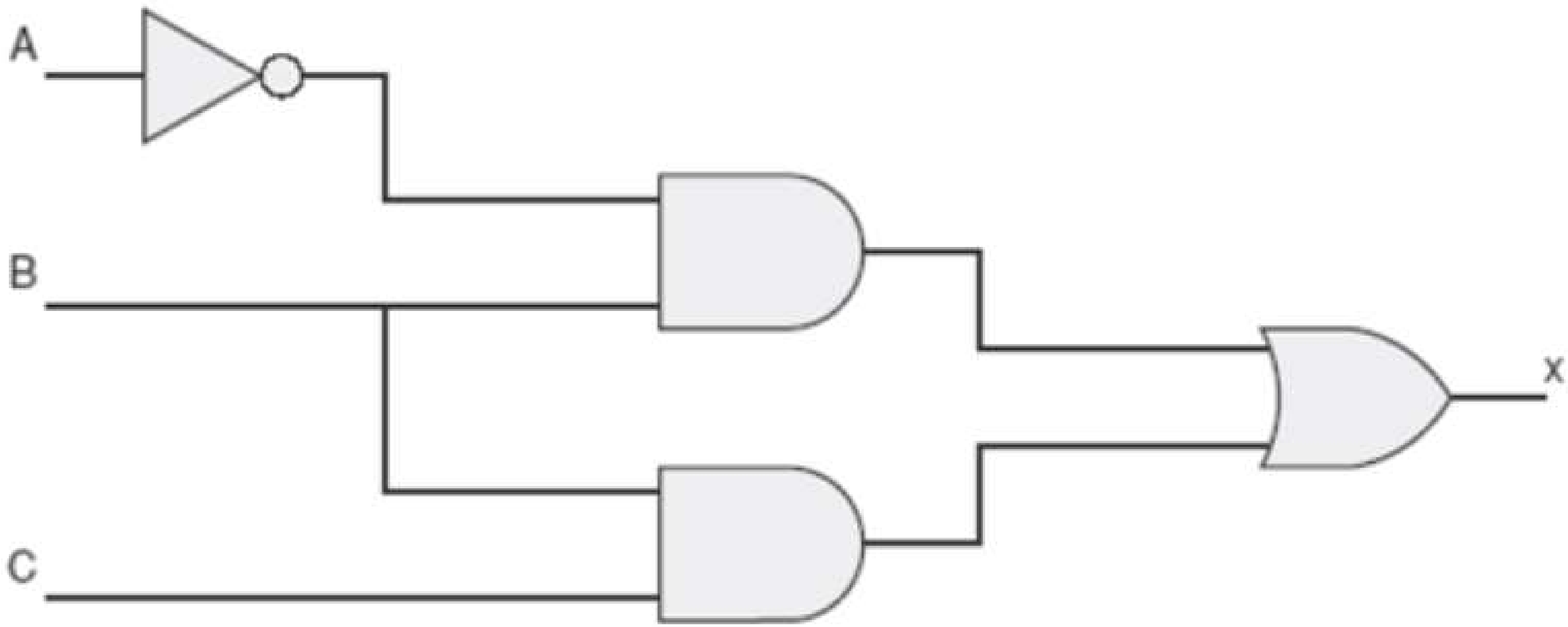


B)



Exercício 6

- Obter a Tabela Verdade do circuito:



Exercício 7

- Construir um circuito lógico a partir da expressão:

$$y = AC + B\overline{C} + \overline{A}BC$$

Referências Bibliográficas

Cleone Silva de Lima, Fundamentos de Lógica e Algoritmos – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte.

Obrigado