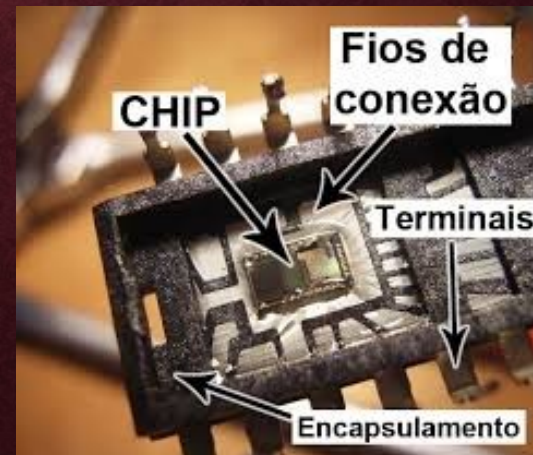
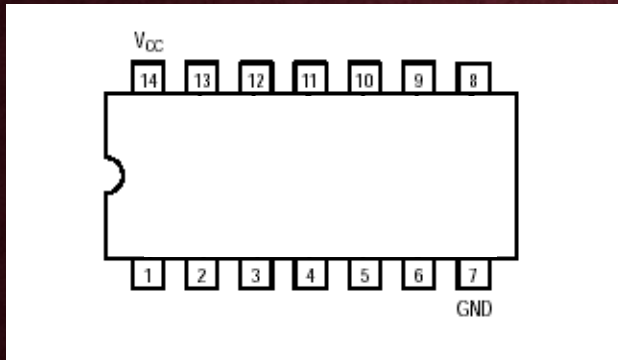


PORTAS LÓGICAS

Arquitetura de Computadores - Prof. Raul Bastos

2020_1

CIRCUITO INTEGRADO



FAMÍLIAS LÓGICAS

Os circuitos integrados digitais estão agrupados em famílias lógicas.

Famílias lógicas **bipolares**:

RTL – *Resistor Transistor Logic* – Lógica de transístor e resistência.

DTL – *Díode Transistor Logic* – Lógica de transístor e díodo.

TTL – *Transistor Transistor Logic* – Lógica transístor-transístor.

HTL – *High Threshold Logic* – Lógica de transístor com alto limiar.

ECL – *Emitter Coupled Logic* – Lógica de emissores ligados.

I²L – *Integrated-Injection Logic* – Lógica de injeção integrada.

Famílias lógicas **MOS** (**M**etal – **Ó**xido – **S**emicondutor)

CMOS – *Complementary MOS* – MOS de pares complementares NMOS/PMOS

NMOS – Utiliza só transístores MOS-FET canal N.

PMOS – Utiliza só transístores MOS-FET canal P.

Atualmente a família lógica TTL e a CMOS são as mais usadas.

TENSÕES DOS NÍVEIS LÓGICOS

Família Lógica TTL

Faixas de tensão correspondentes aos níveis lógicos de entrada:

Entre 2 e 5 Volt, nível lógico 1

Entre 0,8V e 2V o componente não reconhece os níveis lógicos 0 e 1, devendo portanto, ser evitada em projetos de circuitos digitais.

Entre 0 e 0,8 Volt, nível lógico 0

Faixas de tensão correspondentes aos níveis lógicos de saída:

Entre 2,4 e 5 Volt, nível lógico 1

Entre 0,3 e 0,5 Volt, nível lógico 0

Família Lógica CMOS

Faixa de alimentação que se estende de 3V a 15V ou 18V, dependendo do modelo.

A família CMOS possui também, uma determinada faixa de tensão para representar os níveis lógicos de entrada e de saída, porém estes valores dependem da tensão de alimentação e da temperatura ambiente.

NÍVEIS DE INTEGRAÇÃO

Os níveis de integração referem-se ao número de portas lógicas que o CI contém.

SSI (Small Scale Integration) – Integração em pequena escala: São os CI com menos de 12 portas lógicas.

MSI (Medium Scale Integration) – Integração em média escala: Corresponde aos CI que têm entre 12 a 99 portas lógicas

LSI (Large Scale Integration) – Integração em grande escala: Corresponde aos CI que têm entre 100 a 9 999 portas lógicas.

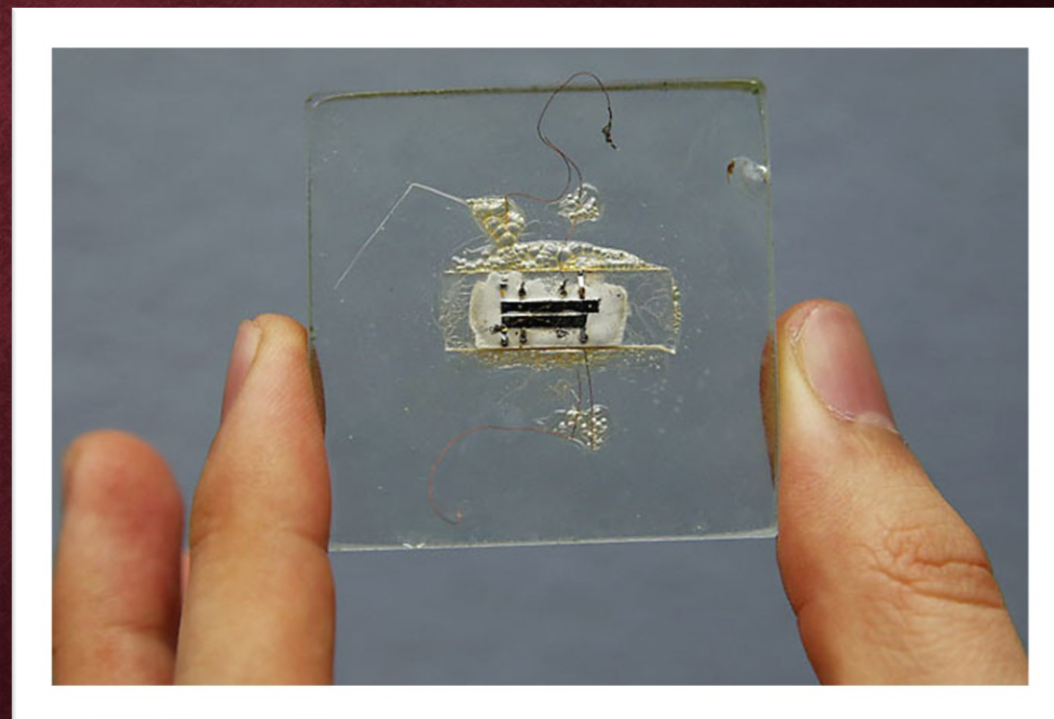
VLSI (Very Large Scale Integration) – Integração em muito larga escala: Corresponde aos CI que têm entre 10 000 a 99 999 portas lógicas.

ULSI (Ultra Large Scale Integration) – Integração em escala ultra larga: Corresponde aos CI que têm 100 000 ou mais portas lógicas.

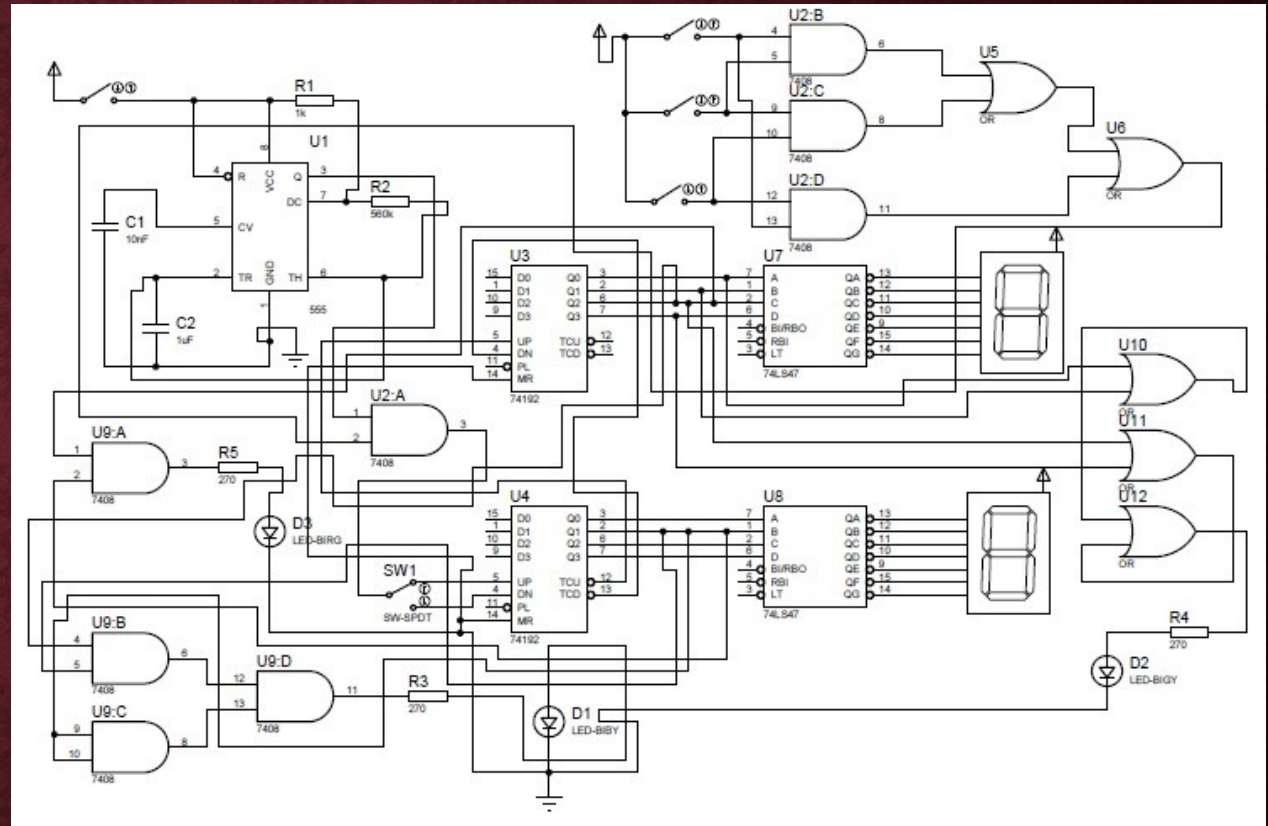
CIRCUITO INTEGRADO / CI / MICROCHIP (1959)

O primeiro circuito integrado, ou CI, foi fabricado pela Texas Instruments e apresentado em 6 de Fevereiro de 1959.

O desenvolvimento do circuito integrado evoluiu muito nas décadas de 60 e 70, até o surgimento dos microprocessadores, no início dos anos 1980.



<http://gg.gg/j73nn>



**PARA ME ENVIAR FOTO OU O TEU ARQUIVO CONTENDO AS RESPOSTAS
DOS EXERCÍCIOS, NO ENDEREÇO ACIMA E CONTÉM TAMBÉM UM
SIMULADOR PRA SER INSTALADO**

Arquitetura de Computadores – Prof. Raul Bastos

INTRODUÇÃO

PORTAS LÓGICAS (GATES)

As operações de um computador resumem-se na combinação de operações aritméticas básicas: **somar, complementar, comparar e mover bits.**

“Quem” realiza estas complicadíssimas operações são circuitos eletrônicos conhecidos como circuitos lógicos ou *Gates*.

Computadores digitais (binários) são construídos com circuitos eletrônicos digitais - as portas lógicas (circuitos lógicos).

A lógica é a base da eletrônica digital e da informática.

HISTÓRICO

Esta surgiu na **Grécia** antiga com a contribuição de três filósofos:

Sócrates - Pelo sua investigação, se duas verdades são alcançadas individualmente, ao juntá-las tem-se uma **única** verdade.

Platão - Platão (seguidor de Sócrates) escreveu vários de seus diálogos e desenvolveu sua filosofia abrangendo a ética, a política e o conhecimento, tendo como princípio o método da investigação.

Aristóteles - Aristóteles, baseado nos diálogos escritos por Platão, observou que a linguagem deve ter uma estrutura lógica, para que leve, necessariamente, a uma verdade.

Histórico

Os sistemas lógicos estão calcados na **álgebra** dos chaveamentos ou **álgebra de Boole**, instituída pelo matemático inglês George Boole (1815–864) e que admite apenas duas grandezas: **falso** ou **verdadeiro**, representados por 0 e 1 respectivamente.

Esses sinais binários são representados por níveis de tensão nos **circuitos** do computador.

Um computador pode ser projetado e/ou descrito em diversos níveis de **abstração**. Assim podemos descrever inteiramente um computador através de equações **booleanas** ou o seu equivalente em portas lógicas **E**, **OU** e **NOT**.

A ÁLGEBRA BOOLEANA

Boole construiu sua lógica a partir de símbolos, representando as expressões por letras e ligando-as através de conectivos - símbolos algébricos.

A álgebra booleana nos permite descrever algebricamente as relações entre as saídas e as entradas dos circuitos lógicos.

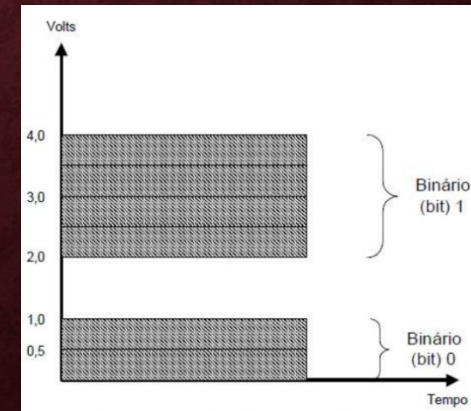
As portas lógicas são os blocos fundamentais a partir dos quais todos os circuitos digitais são construídos.

A álgebra booleana é uma ferramenta usada na descrição, na análise, no projeto e na implementação dos circuitos digitais.

A ÁLGEBRA BOOLEANA

Trabalha com apenas duas grandezas: falso ou verdadeiro. As duas grandezas são representadas por 0 (falso) e 1 (verdadeiro).

Nota: nos circuitos lógicos do computador, os sinais binários são representados por quantidades físicas, sinais elétricos, os quais são gerados e mantidos internamente ou recebidos de elementos externos, em dois níveis de intensidade, cada um correspondendo a um valor binário, como exemplificado na figura a seguir:



PORTAS E OPERAÇÕES LÓGICAS

Enquanto cada elemento lógico ou condição é representado por um valor “0” ou “1”, faz-se necessário que tenhamos meios de combinar diferentes sinais lógicos ou condições para gerar um resultado útil.

Uma *porta lógica* (“gate”) é um circuito eletrônico, portanto uma peça de hardware, que se constitui no elemento básico e mais elementar de um sistema de computação. Grande parte do hardware do sistema é fabricado através da adequada combinação de milhões desses elementos.

TABELA VERDADE

Técnica que serve para determinar como a saída de um circuito lógico depende dos níveis lógicos presentes nas entradas do circuito.

A tabela verdade relaciona todas as possíveis combinações para os níveis lógicos presentes nas entradas.

Se o número de entradas de um circuito for N , o número de linhas da tabela verdade será 2^N .

Geralmente a tabela verdade é construída com as entradas variando em sequência, numa contagem binária crescente.

Uma tabela verdade possui, então, tantas linhas de informação quantas são as possíveis combinações de valores de entrada que se tenha.

SÍMBOLOS GRÁFICOS

Assim como na álgebra comum, é necessário definir símbolos matemáticos e gráficos para representar as operações lógicas (e os operadores lógicos).

A figura mostra os símbolos matemáticos e gráficos referentes às operações lógicas (portas) que iremos analisar:

VARIÁVEL BOOLEANA

Exemplos:

-Lâmpada:

acesa (1) ou apagada (0)

-Chave:

fechada (1) ou aberta (0)

-Verdadeiro (1) ou Falso(0)

Representação:

– Expressão Lógica

– Tabela Verdade

– Símbolos
(portas lógicas)

PORTA NOT (INVERSOR)

Porta NOT é um circuito lógico que requer apenas um valor na entrada: inverte o sinal de entrada (executa a negação do sinal de entrada), ou seja, se o sinal de entrada for 0 ela produz uma saída 1, se a entrada for 1 ela produz uma saída 0.

PORTA LÓGICA NOT (NEGAÇÃO)

Símbolo antigo



Expressão da função

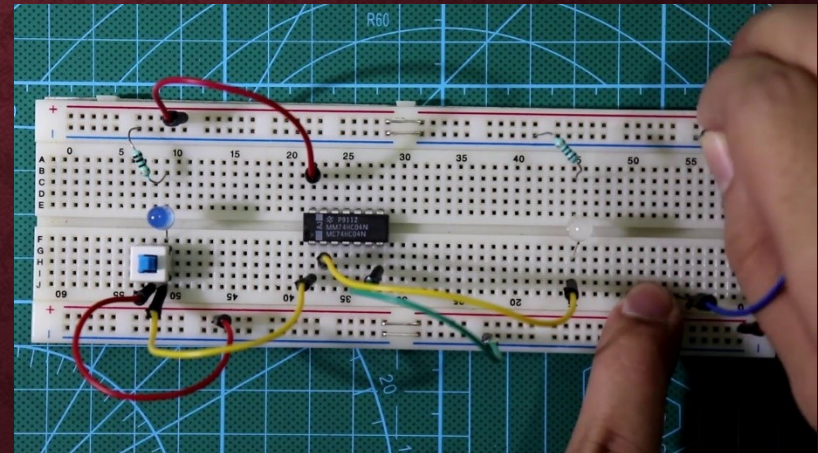
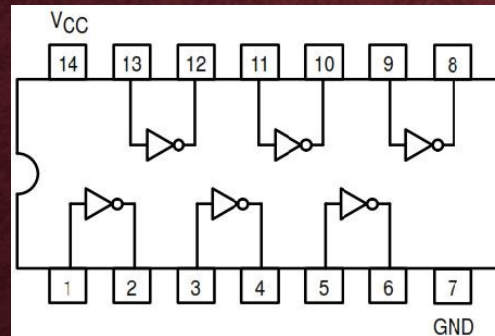
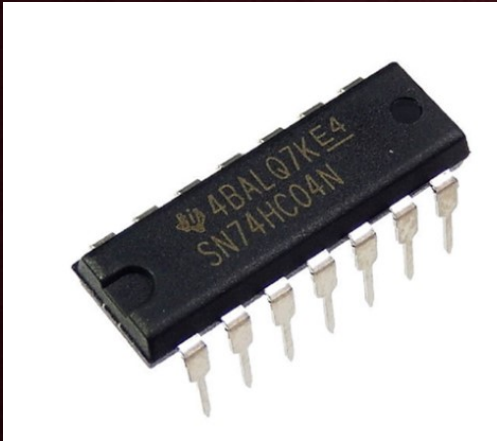
$$S = \overline{A}$$

Tabela de verdade

A	
0	1
1	0

O nível lógico da saída (**S**) é a negação do nível lógico da entrada (**A**).

NOT – (NÃO): INVERSORA



<https://www.youtube.com/watch?v=0IoNwTy9l4g>

CI 7408 - AND

A importância dessas portas lógicas está no fato de representarem os elementos básicos de construção da maioria dos circuitos digitais práticos.

Quando se deseja construir um circuito lógico (ou digital) relativamente simples, usa-se uma placa de circuito impresso com soquetes sobre os quais insere-se um circuito integrado (CI) digital.

A maioria dos CI's já são padronizados, e os mais comuns pertencem à série denominada 7400. Os mais simples utilizam a tecnologia de Integração em Pequena Escala (SSI - Small Scale Integration).

Como um exemplo, veja o CI 7408, que contém quatro portas AND, cuja relação de suas entradas e saídas com os pinos de seu encapsulamento pode ser vista no diagrama.

OPERAÇÃO LÓGICA OU PORTA AND (E)

A porta AND combina dois ou mais sinais de entrada de forma equivalente a um circuito em série, para produzir um único sinal de saída, ou seja, ela produz uma saída 1, se todos os sinais de entrada forem 1; caso qualquer um dos sinais de entrada for 0, a porta AND produzirá um sinal de saída igual a zero.

Não existe limite para o número de entradas que podem ser aplicadas em uma porta AND. Porém, por razões práticas, portas AND comerciais são usualmente manufaturadas com 2, 3 ou 4 portas.

Um circuito integrado (CI) padrão contém 14 ou 16 pinos, para praticidade de tamanho e manipulação. Um CI de 14 pinos, por exemplo, pode conter 4 portas de 2 entradas ou 3 portas de 3 entradas ou 2 portas de 4 entradas, sendo reservadas duas portas para as conexões de força.

PORTA LÓGICA AND (E)

Símbolo antigo



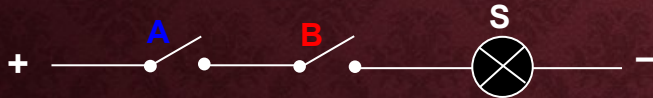
Expressão da função

$$S = A \cdot B$$

Tabela de verdade

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

Analogia da porta lógica AND com um circuito eléctrico:



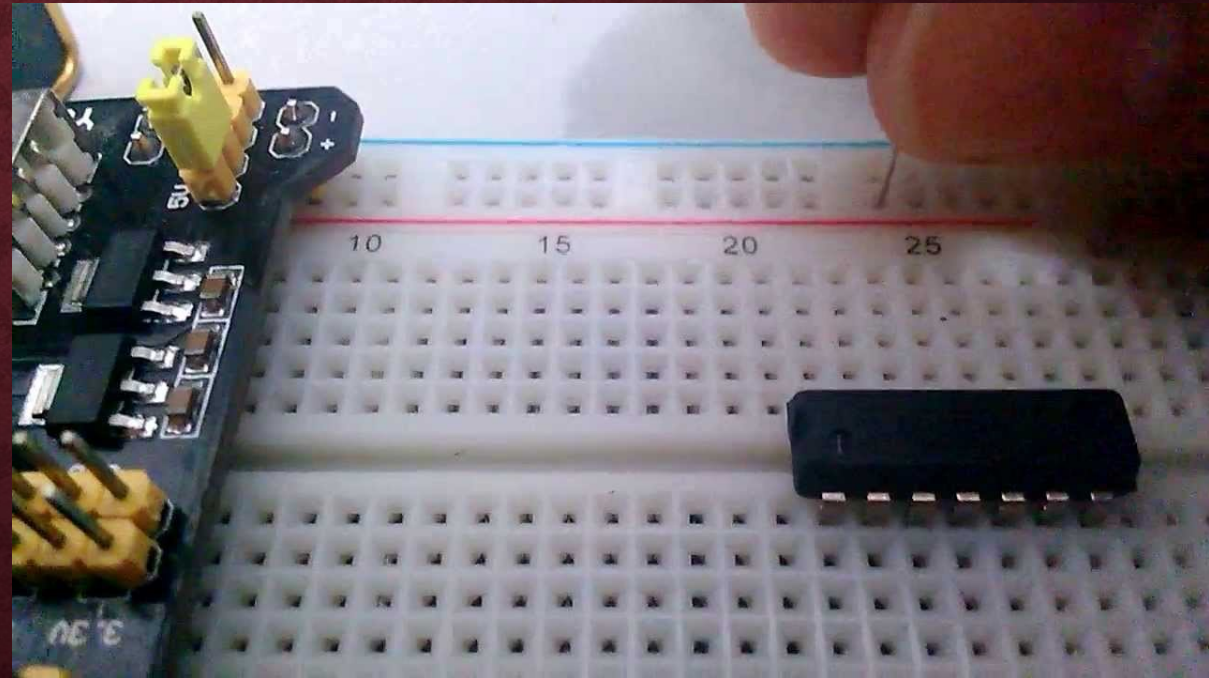
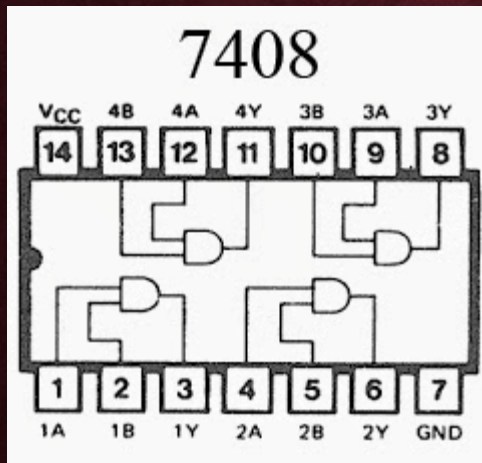
Quando as duas entradas (A e B) são zero (interruptores desligados) a saída (S) também é zero (lâmpada apagada).

Quando uma só das entradas é 1 (um só interruptor ligado) a saída (S) é zero (lâmpada apagada).

Quando as duas entradas (A e B) são 1 (os dois interruptores ligados) a saída (S) também é 1 (lâmpada acesa),

CONCLUSÃO: Só temos o nível lógico 1 na saída quando todas as entradas forem 1 (neste caso, A e B)

PORTA AND



<https://www.youtube.com/watch?v=M7IvSy57W8Y>

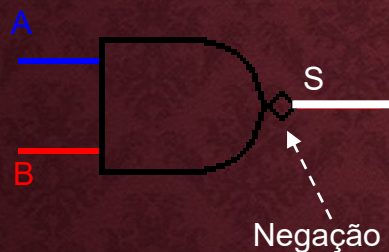
PORTA NAND – NOT AND

A porta NAND equivale a uma porta AND seguida por uma porta NOT, isto é, ela produz uma saída que é o inverso da saída produzida pela porta AND. Esta porta produzirá uma saída falsa se e somente se todas as entradas forem verdade.

Esta porta é muito comum de ser encontrada em complexos circuitos lógicos, visto que é possível simplificar a fabricação de circuitos lógicos, reduzindo a quantidade de componentes eletrônicos, se usarmos apenas circuitos NAND.

PORTA LÓGICA NAND (NÃO E)

Símbolo antigo



Expressão da função

$$S = \overline{A \times B}$$

Tabela de verdade

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

A porta lógica NAND é uma porta lógica AND com a saída negada.

Pode observar-se que os níveis lógicos da saída (**S**) da tabela de verdade NAND é a negação dos níveis lógicos da saída (**S**) da tabela de verdade AND.

PORTA OR (OU)

A porta OR é definida para produzir um resultado verdade (=1) na sua saída, se pelo menos uma das entradas for verdade.

Tal como a porta AND, a porta OR pode ter qualquer número de entradas. Porém, portas OR comerciais são limitadas a 2, 3 e 4 entradas, tais como as portas AND.

Esta definição pode ser expressa pela tabela verdade e símbolos mostrados na figura:

PORTA LÓGICA OR (OU)

Símbolo antigo



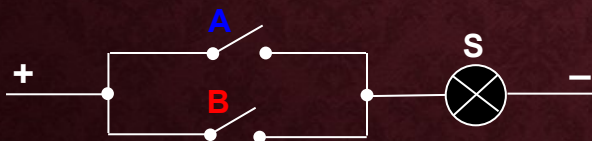
Expressão da função

$$S = A + B$$

Tabela de verdade

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

Analogia da porta lógica OR com um circuito eléctrico:



Quando as duas entradas (A e B) são zero (interruptores desligados) a saída (S) também é zero (lâmpada apagada).

Quando uma só das entradas é 1 (um só interruptor ligado) a saída (S) é um (lâmpada acesa).

Quando as duas entradas (A e B) são 1 (os dois interruptores ligados) a saída (S) também é 1 (lâmpada acesa),

CONCLUSÃO: Só temos o nível lógico 0 na saída quando todas as entradas forem 0.

PORTA LÓGICA NOR (NÃO OU)

Símbolo antigo



Expressão da função

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

Tabela de verdade

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

A porta lógica NOR é uma porta lógica OR com a saída negada.

Pode observar-se que os níveis lógicos da saída (**S**) da tabela de verdade NOR é a negação dos níveis lógicos da saída (**S**) da tabela de verdade OR.

PORTA XOR – EXCLUSIVE OR

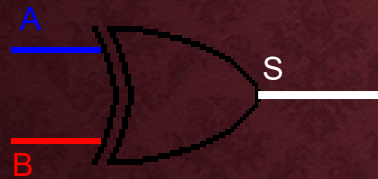
A porta (ou operação lógica) XOR, abreviação do termo EXCLUSIVE OR, pode ser considerada um caso particular da função OR, ou seja, sua definição: “a saída será verdade se exclusivamente uma ou outra entrada for verdade”.

A porta XOR compara os bits: ela produz saída 0 quando todos os bits de entrada são iguais e saída 1 quando pelo menos um dos bits de entrada é diferente dos demais.

Ao contrário das portas OR/NOR e AND/NAND, a porta XOR tem exatamente duas entradas. Com as portas XOR, é possível se fabricar um testador de igualdade entre valores, por exemplo, para testar de modo rápido se duas palavras são iguais. Se as duas palavras forem iguais, as saídas dos circuitos XOR serão todas falsas. A figura mostra o exemplo completo do teste, que inclui uma porta NOR para receber, como entrada, a saída de todos os circuitos XOR e produzir uma única saída.

PORTA LÓGICA EXCLUSIVE OR (OU EXCLUSIVO)

Símbolo antigo



Expressão da função

$$S = A \oplus B$$

Tabela de verdade

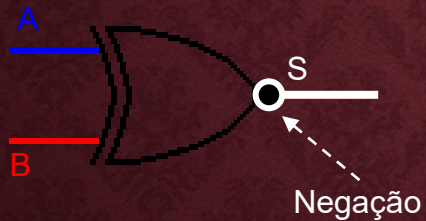
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

A saída é 1 se uma entrada é 1 ou a outra entrada é 1, mas não ambas.

De outro modo: o valor da saída (**S**) é 1 se as entradas (**A** ou **B**) são diferentes e 0 se são iguais.

PORTA LÓGICA EXCLUSIVE NOR (NÃO OU EXCLUSIVO) = FUNÇÃO COINCIDÊNCIA

Símbolo antigo



Expressão da função

$$S = A \odot B$$

Tabela de verdade

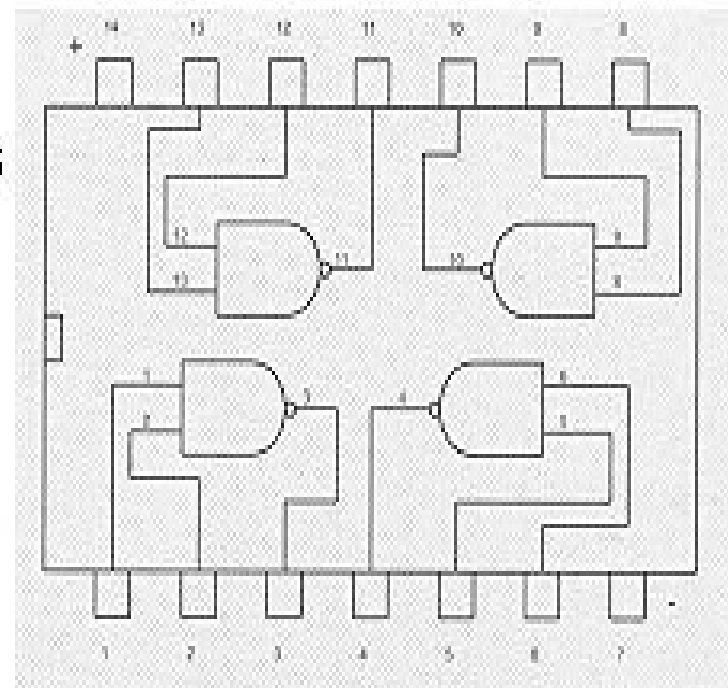
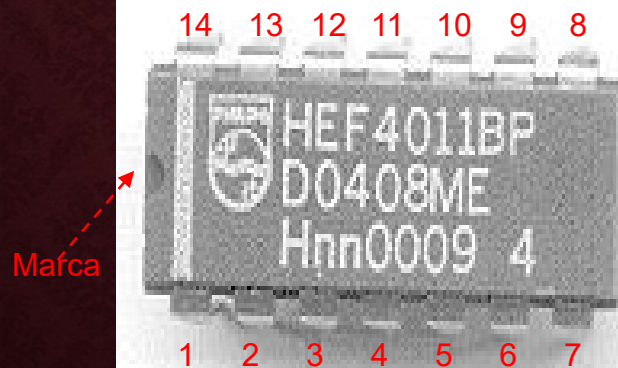
A	B	
0	0	1
0	1	0
1	0	0
1	1	1

Quando houver uma coincidência nos valores das variáveis de entrada. A notação algébrica que representa a função Coincidência é $S = A \odot B$, onde se lê: A Coincidência B.

Nota: Assume 1 quando houver coincidência entre os valores das variáveis.

CIRCUITOS INTEGRADOS DIGITAIS

CI 4011 DIGITAL - POSSUI 4 PORTAS
LÓGICAS NAND INTERNAS





OS OPERADORES LÓGICOS OU FUNÇÕES LÓGICAS BÁSICAS SÃO AS SEGUINTES:

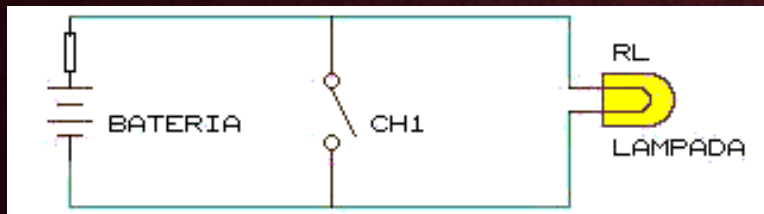


NÃO = NOT: INVERSOR

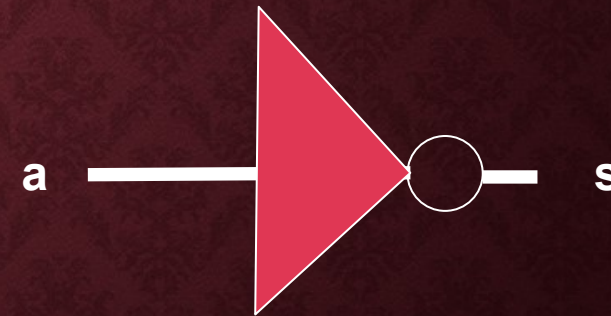
BASICAMENTE TEM-SE UMA ALTERNÂNCIA ENTRE OS DOIS VALORES.

Tabela -Verdade

Entrada a	Saída s
0	1
1	0



Porta NOT ou NÃO
Representação da Porta Lógica



Expressão Booleana

$$s = \bar{a}$$

$$a' = \text{NOT } a$$

E = AND

UMA FUNÇÃO É VERDADEIRA SE, E SOMENTE SE, TODOS OS TERMOS FOREM VERDADEIROS.

Tabela da Verdade

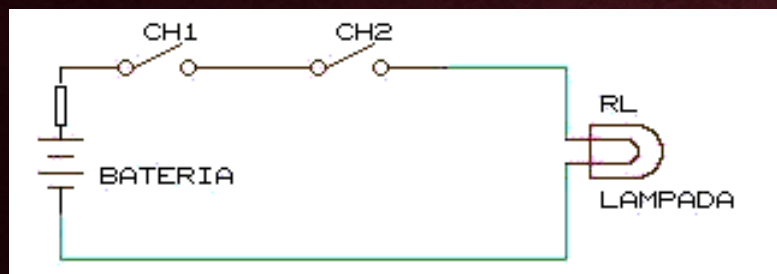
Entrada a	Entrada b	Saída s
0	0	0
0	1	0
1	0	0
1	1	1

Expressão Booleana

$$s = a . b$$

$$s = ab$$

Representação da Porta Lógica



NÃO E = NAND

EQUIVALE A UMA PORTA AND SEGUIDA DE UMA PORTA NÃO. O RESULTADO É O INVERSO DA SAÍDA DE UMA PORTA AND.

Tabela da Verdade

Entrada a	Entrada b	Saída s
0	0	1
0	1	1
1	0	1
1	1	0

Expressão Booleana

$$s = \overline{a \cdot b}$$

$$s = \overline{ab}$$

Representação da Porta Lógica



OU = OR

UMA FUNÇÃO É VERDADEIRA SE, QUALQUER UM DOS TERMOS FOR VERDADEIRO

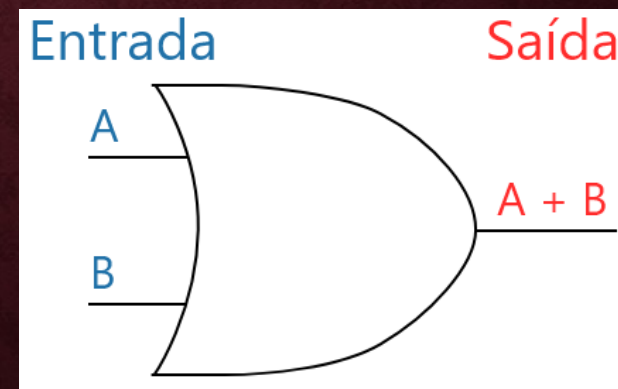
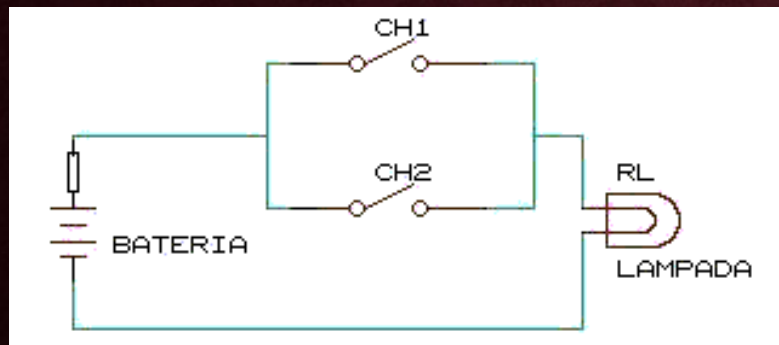
Tabela da Verdade

Entrada a	Entrada b	Saída s
0	0	0
0	1	1
1	0	1
1	1	1

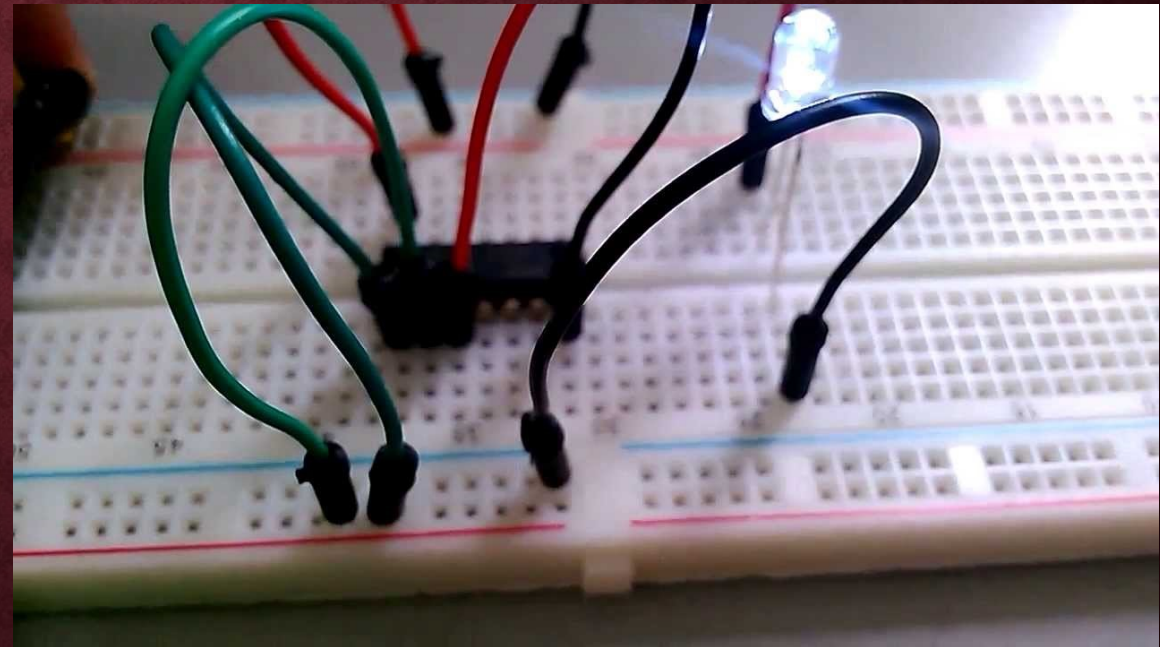
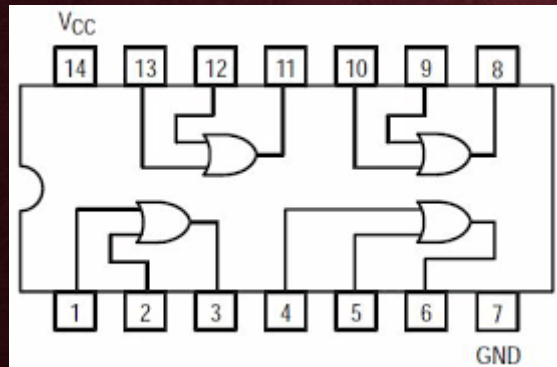
Expressão Booleana

$$s = a + b$$

Representação da Porta Lógica



PORTA OR (OU)



NOR = NOU

EQUIVALE A UMA PORTA OR SEGUIDA DE UMA PORTA NÃO. O RESULTADO É O INVERSO DA SAÍDA DE UMA PORTA OR.

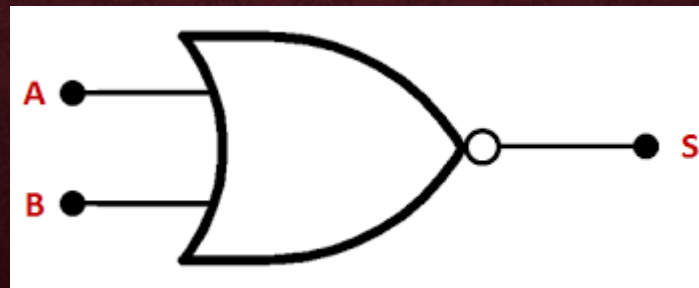
Tabela da Verdade

Entrada a	Entrada b	Saída s
0	0	1
0	1	0
1	0	0
1	1	0

Expressão Booleana

$$s = \overline{a + b}$$

Representação da Porta Lógica



XOR = OU EXCLUSIVO

A FUNÇÃO É VERDADEIRA SE, E SOMENTE SE, UM DOS TERMOS FOR VERDADEIRO

Assume 1 quando as variáveis assumirem valores diferentes entre si.

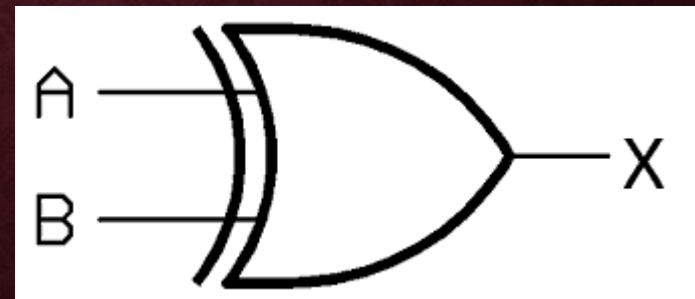
Tabela da Verdade

Entrada a	Entrada b	Saída s
0	0	0
0	1	1
1	0	1
1	1	0

Expressão Booleana

$$s = a \oplus b$$

Representação da Porta Lógica



XNOR = NOT(XOR)

**É VERDADEIRA SE, TODOS TERMOS FOREM VERDADEIROS
OU TODOS OS TERMOS FOREM FALSOS**

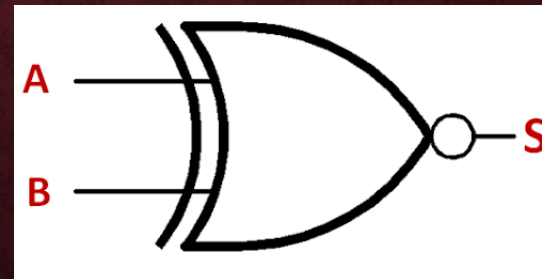
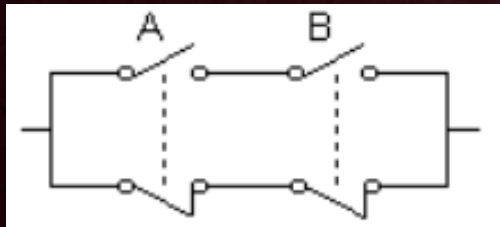
Tabela da Verdade

Entrada a	Entrada b	Saída s
0	0	1
0	1	0
1	0	0
1	1	1

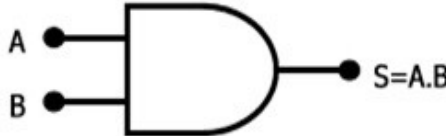
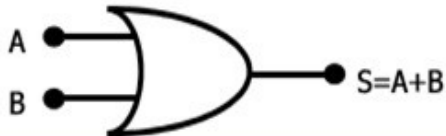
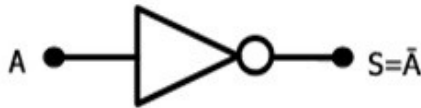

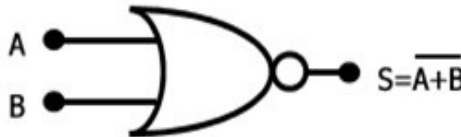

Expressão Booleana

$$S = A \odot B$$

Representação da Porta Lógica



Básicos

Nome	Símbolo Gráfico	Função Algébrica	Tabela Verdade															
E (AND)		$S=A.B$ $S=AB$	<table><tr><th>A</th><th>B</th><th>$S=A.B$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	$S=A.B$	0	0	0	0	1	0	1	0	0	1	1	1
A	B	$S=A.B$																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OU (OR)		$S=A+B$	<table><tr><th>A</th><th>B</th><th>$S=A+B$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	$S=A+B$	0	0	0	0	1	1	1	0	1	1	1	1
A	B	$S=A+B$																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NÃO (NOT) Inversor		$S=\bar{A}$ $S=A'$ $S=\neg A$	<table><tr><th>A</th><th>$S=\bar{A}$</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	$S=\bar{A}$	0	1	1	0									
A	$S=\bar{A}$																	
0	1																	
1	0																	
NE (NAND)		$S=\bar{A}.\bar{B}$ $S=(A.B)'$ $S=\neg(A.B)$	<table><tr><th>A</th><th>B</th><th>$S=\bar{A}.\bar{B}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	$S=\bar{A}.\bar{B}$	0	0	1	0	1	1	1	0	1	1	1	0
A	B	$S=\bar{A}.\bar{B}$																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOU (NOR)		$S=\overline{A+B}$ $S=(A+B)'$ $S=\neg(A+B)$	<table><tr><th>A</th><th>B</th><th>$S=\overline{A+B}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	$S=\overline{A+B}$	0	0	1	0	1	0	1	0	0	1	1	0
A	B	$S=\overline{A+B}$																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR		$S=A\oplus B$	<table><tr><th>A</th><th>B</th><th>$S=A\oplus B$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	$S=A\oplus B$	0	0	0	0	1	1	1	0	1	1	1	0
A	B	$S=A\oplus B$																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

DESCREVENDO CIRCUITOS LÓGICOS ALGEBRICAMENTE

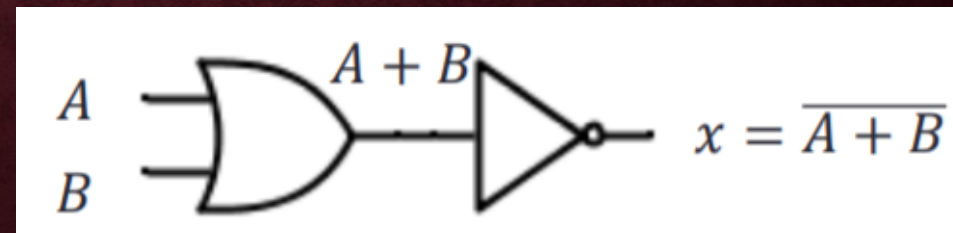
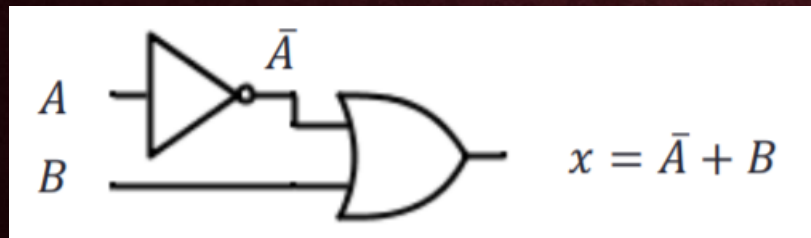
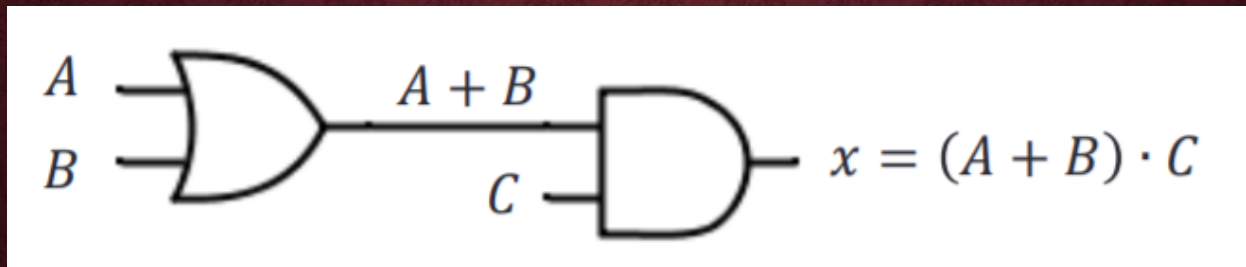
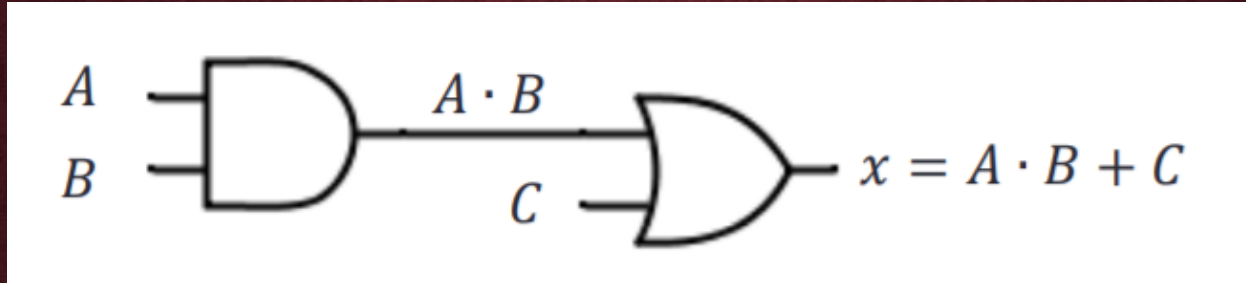
As portas AND, OR e NOT são os blocos fundamentais dos circuitos digitais. Portanto qualquer circuito pode ser descrito através dessas três funções.

Quando combinamos portas lógicas, uma expressão booleana será a variável de entrada de alguma porta lógica.

Ao combinar expressões AND e OR, a operação AND é realizada primeiro, como na álgebra comum para a operação de multiplicação.

Parêntesis são usados para se alterar a precedência.

EXEMPLOS

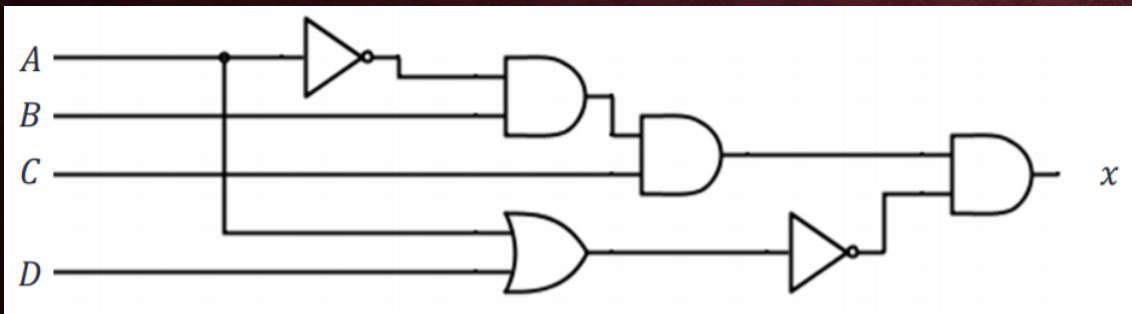


EXERCÍCIO

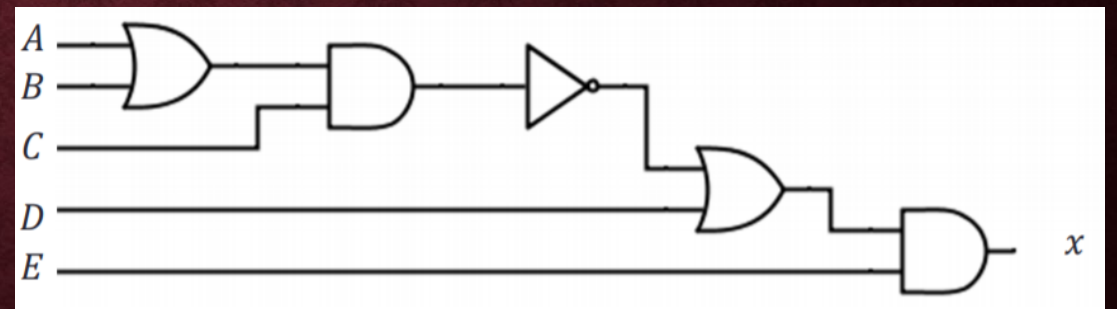
Nos circuitos abaixo:

- (a) escreva a expressão booleana para a saída x .
- (b) troque cada porta AND por porta OR e cada porta OR por porta AND e escreva novamente a expressão booleana para a saída x para os dois circuitos.

1.



2.



Atividade.



<https://qrqo.page.link/fygl6>



<https://qrqo.page.link/v52Gg>



<https://qrqo.page.link/fWj8x>