

ELETRÔNICA DIGITAL 1

#FÁCILDEMAIS

Antonio Tadeu Pellison



EDITORA CONHECIMENTO LIVRE

Antonio Tadeu Pellison

Eletrônica Digital 1 #FácilDemais

1ª ed.

Piracanjuba-GO
Editora Conhecimento Livre
Piracanjuba-GO

1ª ed.

Dados Internacionais de Catalogação na Publicação (CIP)

Pellison, Antonio Tadeu
P168E Eletrônica Digital 1 #FácilDemais

/ Antonio Tadeu Pellison. – Piracanjuba-GO

Editora Conhecimento Livre, 2025

57 f.: il

DOI: 10.37423/2025.edcl1078

ISBN: 978-65-5367-602-2

Modo de acesso: World Wide Web

Incluir Bibliografia

1. portas-lógicas 2. circuitos-combinacionais 3. práticas-digitais I. Pellison, Antonio Tadeu II. Título

CDU: 620

<https://doi.org/10.37423/2025.edcl1078>

O conteúdo dos artigos e sua correção ortográfica são de responsabilidade exclusiva dos seus respectivos autores.

EDITORIA CONHECIMENTO LIVRE

Corpo Editorial

MSc Edson Ribeiro de Britto de Almeida Junior

MSc Humberto Costa

MSc Thays Merçon

MSc Adalberto Zorzo

MSc Taiane Aparecida Ribeiro Nepomoceno

PHD Willian Douglas Guilherme

MSc Andrea Carla Agnes e Silva Pinto

Dr. Walmir Fernandes Pereira

MSc Edisio Alves de Aguiar Junior

MSc Rodrigo Sanchotene Silva

MSc Adriano Pereira da Silva

MSc Frederico Celestino Barbosa

MSc Guilherme Fernando Ribeiro

MSc. Plínio Ferreira Pires

MSc Fabricio Vieira Cavalcante

PHD Marcus Fernando da Silva Praxedes

MSc Simone Buchignani Maigret

Dr. Adilson Tadeu Basquerote

Dra. Thays Zigante Furlan

MSc Camila Concato

PHD Miguel Adriano Inácio

Dra. Anelisa Mota Gregoleti

PHD Jesus Rodrigues Lemos

MSc Gabriela Cristina Borborema Bozzo

MSc Karine Moreira Gomes Sales

Dr. Saulo Cerqueira de Aguiar Soares

MSc Pedro Panhoca da Silva

MSc Helton Rangel Coutinho Junior

MSc Carlos Augusto Zilli

MSc Euvaldo de Sousa Costa Junior

Dra. Suely Lopes de Azevedo

Dr. Francisco Odecio Sales

MSc Ezequiel Martins Ferreira

MSc Eliane Avelina de Azevedo Sampaio

MSc Carlos Eduardo De Oliveira Gontijo

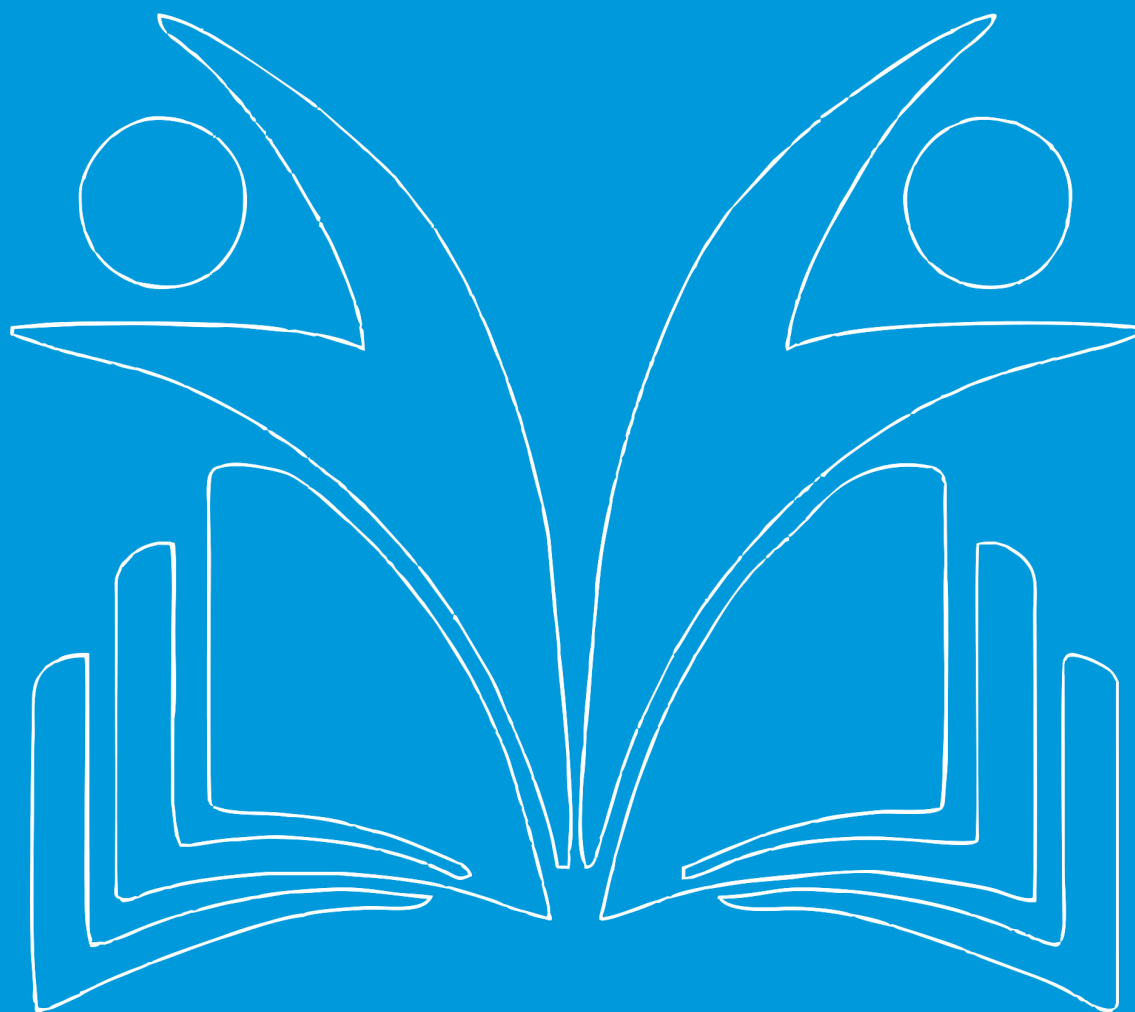
Dr. Rodrigo Couto Santos

Dra. Milena Gaion Malosso

PHD Marcos Pereira Dos Santos



10.37423/2025.edcl1078



Sumário

1. SISTEMAS NUMÉRICOS	5
1.1. INTRODUÇÃO	5
1.2. SISTEMA DECIMAL DE NUMERAÇÃO	5
1.3. SISTEMA BINÁRIO DE NUMERAÇÃO	6
1.4. SISTEMA HEXADECIMAL DE NUMERAÇÃO	7
1.5. SISTEMAS DE NUMERAÇÃO – TABELA DE CONVERSÃO	9
1.6. CONVERSÃO ENTRE BASES	10
1.6.1. BASE QUALQUER PARA A BASE 10	10
1.6.2. BASE 10 PARA BASE QUALQUER	10
1.6.3. CONVERSÃO ENTRE AS BASES 2 E 16	12
1.6.4. PARA CONVERTER UM NÚMERO BINÁRIO PARA HEXADECIMAL, SEGUIR OS SEGUINTE PASSOS:	13
1.6.5. PARA CONVERTER UM NÚMERO HEXADECIMAL PARA BINÁRIO, SEGUIR ESSES PASSOS:	14
1.7. MAPA MENTAL – CONVERSÃO ENTRE OS SISTEMAS DE NUMERAÇÃO	14
LISTA 1 – SISTEMAS DE NUMERAÇÃO	14
LISTA 3 – SISTEMAS DE NUMERAÇÃO - REVISÃO	16
2. PORTAS LÓGICAS	17
2.1. ÁLGEBRA BOOLEANA	17
2.2. VARIÁVEIS LÓGICAS	18
2.3. TABELA VERDADE	19
2.4. PORTAS LÓGICAS	20
2.4.1. PORTA E (AND):	20
2.4.2. PORTA OU (OR):	23
2.4.3. PORTA NÃO (NOT):	25
2.4.4. PORTA OU EXCLUSIVO - XOR	27
2.4.5. PORTA NOU EXCLUSIVO – XNOR	28
2.4.6. EQUAÇÕES BÁSICAS BOOLEANAS	29

3. CIRCUITOS COMBINACIONAIS.....	29
3.1. DEFINIÇÃO DE CIRCUITOS COMBINACIONAIS	29
3.2. IMPLEMENTAÇÃO DE CIRCUITOS COMBINACIONAIS.	30
3.3. OBTER EXPRESSÕES BOOLEANAS A PARTIR DE CIRCUITOS.....	31
3.4. OBTER O CIRCUITO LÓGICO A PARTIR DE EXPRESSÕES BOOLEANAS.	31
3.5. DETERMINAÇÃO DA TABELA-VERDADE A PARTIR DA EXPRESSÃO BOOLEANA.....	32
3.6. EXPRESSÕES BOOLEANAS OBTIDAS A PARTIR DA TABELA VERDADE.....	33
3.6.1. OBTENÇÃO DA EQUAÇÃO A PARTIR DE SOMA DE PRODUTOS	34
LISTA 4 – EXPRESSÃO BOOLEANA	35
LISTA 5 – EXPRESSÃO BOOLEANA – TABELA VERDADE - CIRCUITO.....	37
LISTA 6 – EXPRESSÃO BOOLEANA – TABELA VERDADE.....	39
LISTA 7 – EXPRESSÃO BOOLEANA – TABELA VERDADE – PROVE QUE	39
LISTA 8 – SOMA DE PRODUTOS.....	41
4. LABORATÓRIO	42
4.1. PORTA E.....	42
4.2. PORTA OU.....	44
REFERÊNCIAS	53

1. SISTEMAS NUMÉRICOS

1.1. INTRODUÇÃO

Sistemas de numeração utilizados em eletrônica digital são fundamentais para o correto funcionamento e comunicação entre os dispositivos eletrônicos. Esses sistemas são a base para a representação de informações e dados através de valores binários, e são amplamente utilizados em computadores, smartphones, circuitos integrados e outros dispositivos eletrônicos.

O sistema de numeração mais comumente utilizado na eletrônica digital é o sistema binário, que utiliza apenas dois dígitos – 0 e 1. Esse sistema é baseado no sistema decimal, que utiliza dez dígitos – de 0 a 9. No sistema binário, cada dígito representa um bit, que é a menor unidade de informação em eletrônica digital. A combinação dos bits permite representar qualquer informação ou dado em formato digital.

Além do sistema binário, existem outros sistemas de numeração usados em eletrônica digital, como o sistema octal e o sistema hexadecimal. O sistema octal utiliza a base 8, enquanto o sistema hexadecimal utiliza a base 16. Esses sistemas são utilizados para facilitar a manipulação e representação de dados binários de forma mais compacta e legível.

Na eletrônica digital, os sistemas de numeração são essenciais para o processamento e transmissão de informações entre diferentes componentes eletrônicos. Os circuitos digitais operam com base em sinais binários, que são representados por níveis de tensão correspondentes aos dígitos binários 0 e 1. A combinação desses sinais binários permite realizar operações lógicas e aritméticas, essenciais para o funcionamento de computadores e outros dispositivos eletrônicos.

Além disso, os sistemas de numeração são amplamente utilizados na representação de endereços de memória, contadores, registradores e outros componentes de sistemas digitais. Por exemplo, um endereço de memória pode ser representado em binário, octal ou hexadecimal, e essa representação determina a quantidade de bits necessários para representar o endereço completo.

1.2. SISTEMA DECIMAL DE NUMERAÇÃO

O sistema decimal de numeração é um sistema numérico que utiliza a base 10, ou seja, conta de 0 a 9. Nesse sistema, cada algarismo tem um valor posicional específico dentro de um número.

No sistema decimal, os números são escritos utilizando dez símbolos diferentes: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. A posição de cada algarismo determina o seu valor. Por exemplo, o número "345" representa 3 centenas, 4 dezenas e 5 unidades.

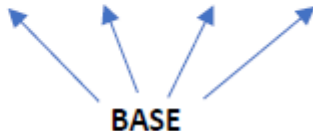
O valor de cada posição é sempre um múltiplo de dez. A primeira posição à direita representa 1, a próxima posição representa 10, depois 100, e assim por diante. A posição mais à esquerda representa a maior potência de dez.

Por exemplo, no número "345", o "5" está na posição das unidades, e representa 5 vezes o valor de um. O "4" está na posição das dezenas, e representa 4 vezes o valor de dez. E o "3" está na posição das centenas, e representa 3 vezes o valor de cem.

A formação de números maiores é feita pela combinação dos algarismos, seguindo as mesmas regras de posicionamento. A cada algarismo adicionado à esquerda, seu valor é multiplicado por 10. Por exemplo, o número "745" representa 7 centenas, mais 4 dezenas e mais 5 unidades.

O sistema decimal é amplamente utilizado na vida cotidiana, em diversas situações, como em transações financeiras, medidas, contagem, etc. Ele é considerado como o sistema numérico padrão e é a base para o estudo de matemática e outras ciências.

LEI DE FORMAÇÃO

$$\text{Número} = a_n.b^n + a_{n-1}.b^{n-1} + a_{n-2}.b^{n-2} + \dots + a_0.b^0$$
$$6753 = 6*1000 + 7*100 + 5*10 + 3*1 = 6*10^3 + 7*10^2 + 5*10^1 + 3*10^0$$


BASE

1.3. SISTEMA BINÁRIO DE NUMERAÇÃO

O sistema binário de numeração é um sistema de numeração que utiliza apenas dois dígitos: 0 e 1. Ao contrário do sistema decimal, que utiliza dez dígitos diferentes (0 a 9), o sistema binário é composto apenas por dois dígitos.

Cada dígito binário representa um valor específico na representação de um número. O dígito "0" representa o valor zero, enquanto o dígito "1" representa o valor um. Esses dígitos são combinados para formar diferentes números binários.

No sistema binário, o valor de cada dígito é determinado pela sua posição em relação aos outros dígitos na sequência. A posição do dígito mais à direita é a posição 0, seguida pela posição 1, 2, 3 e assim por diante.

Cada posição corresponde a uma potência de base 2. Por exemplo, na posição 0, o valor do dígito é multiplicado por 2 elevado a 0, que é igual a 1. Na posição 1, o valor do dígito é multiplicado por 2 elevado a 1, que é igual a 2. Na posição 2, o valor do dígito é multiplicado por 2 elevado a 2, que é igual a 4, e assim por diante.

Para representar um número no sistema binário, os dígitos 0 e 1 são usados e organizados de acordo com essa posição e valor. Por exemplo, o número binário "101" seria interpretado como $1 * 2^2 + 0 * 2^1 + 1 * 2^0$, que é igual a 5.

O sistema binário é amplamente utilizado em computadores e sistemas digitais porque esses dispositivos eletrônicos só entendem e processam informações nesse formato de dois dígitos.

0 equivale a 1bit

1 equivale a 1 bit

Exemplos:

(1011)₂ – tem 4 bits

(1011011)₂ – tem 7 bits

(11110101)₂ - 1 byte

1.4. SISTEMA HEXADECIMAL DE NUMERAÇÃO

O sistema hexadecimal de numeração é um sistema de base 16, o que significa que utiliza 16 símbolos diferentes para representar números. Esses símbolos são os dígitos 0 a 9 e as letras A a F.

O sistema hexadecimal é amplamente utilizado em computação e programação, pois é uma forma conveniente de representar valores binários de maneira mais compacta e legível. Isso ocorre porque cada dígito hexadecimal representa 4 bits de informação, enquanto cada dígito decimal representa apenas 1 bit.

Os valores hexadecimais podem ser usados para representar números inteiros, assim como no sistema decimal, mas também podem ser usados para representar endereços de memória, cores em formatos de imagem e outros dados em computação.

No sistema hexadecimal, os dígitos são ordenados da direita para a esquerda, assim como no sistema decimal, e cada posição tem um valor diferente. A posição mais à direita representa 16 elevado a 0, a próxima posição representa 16 elevado a 1, a seguinte representa 16 elevado a 2 e assim por diante.

Por exemplo, o número hexadecimal "3F" pode ser convertido para o sistema decimal da seguinte forma:

$$\mathbf{3F \text{ em decimal} = (3 * 16^1) + (F * 16^0) = (3 * 16) + (15 * 1) = 48 + 15 = 63}$$

Dessa forma, podemos converter números entre os sistemas decimal e hexadecimal facilmente, utilizando os valores das posições e os dígitos hexadecimais correspondentes.

Exemplos:

(1011)₁₆

(B0A)₁₆

(FACA)₁₆

1.5. SISTEMAS DE NUMERAÇÃO – TABELA DE CONVERSÃO

DECIMAL	BINÁRIO	HEXADECIMAL
$10^1 10^0$	$2^3 2^2 2^1 2^0$	$16^1 16^0$
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16		10
17		11

Utilizando a Tabela de Conversão:

$$(12)_{10} = (1100)_2 = (C)_{16}$$

1.6. CONVERSÃO ENTRE BASES

1.6.1. BASE QUALQUER PARA A BASE 10

Cada dígito que compõe o número na base informada é multiplicado pelo seu valor na base 10 elevado ao expoente correspondente à sua posição no número. Esta soma é equivalente ao número na base 10.

Por exemplo, para converter o número 101 da base 2 para a base 10, a lei de formação seria: $1 \cdot (2^2) + 0 \cdot (2^1) + 1 \cdot (2^0) = 4$.

Exemplos:

$$a) (1100)_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = (12)_{10}$$

$$b) (2BF)_{16} = 2 \cdot 16^2 + B \cdot 16^1 + F \cdot 16^0 = (703)_{10}$$

1.6.2. BASE 10 PARA BASE QUALQUER

O método das divisões sucessivas é uma técnica utilizada para converter um número da base 10 para qualquer outra base. Basicamente, consiste em realizar divisões sucessivas do número decimal pelo valor da base desejada, até que não seja mais possível dividir. O resultado dessa divisão é então utilizado como dígito na nova base, e o processo de divisões sucessivas é repetido com o resto não nulo da divisão anterior, até que o restando seja zero.

Para converter um número da base 10 para a base 2, dividir sucessivamente o número por 2 e registrar o resto de cada divisão, de baixo para cima.

Exemplo:

236 na base 10 para a base 2.

Passo a passo:

1. Divida o número 236 por 2: $236 \div 2 = 118$, resto 0
2. Divida o quociente obtido (118) por 2: $118 \div 2 = 59$, resto 0
3. Divida o quociente obtido (59) por 2: $59 \div 2 = 29$, resto 1
4. Divida o quociente obtido (29) por 2: $29 \div 2 = 14$, resto 1
5. Divida o quociente obtido (14) por 2: $14 \div 2 = 7$, resto 0

6. Divida o quociente obtido (7) por 2: $7 \div 2 = 3$, resto 1

7. Divida o quociente obtido (3) por 2: $3 \div 2 = 1$, resto 1

8. Divida o quociente obtido (1) por 2: $1 \div 2 = 0$, resto 1

Anotar os restos de baixo para cima, que representam os dígitos do número na base 2:

236 na base 10 = 11101100 na base 2.

Exemplos:

A) Converter o número 342 na base 10, para a base 2 e 16.

Para converter o número 342 da base 10 para a base 2 (binária), método da divisão sucessiva por 2.

Passo a passo:

Passo 1: Dividir 342 por 2, mantendo a parte inteira da divisão.

$342 \div 2 = 171$ (resto 0)

Passo 2: Dividir o resultado da divisão anterior por 2 novamente.

$171 \div 2 = 85$ (resto 1)

Passo 3: Repetir o processo com o novo resultado.

$85 \div 2 = 42$ (resto 1)

Passo 4: Continuar dividindo por 2.

$42 \div 2 = 21$ (resto 0)

Passo 5: Dividir por 2 novamente.

$21 \div 2 = 10$ (resto 1)

Passo 6: Continuar o processo.

$10 \div 2 = 5$ (resto 0)

Passo 7: Dividir por 2 novamente.

$5 \div 2 = 2$ (resto 1)

Passo 8: Continuar dividindo.

$2 \div 2 = 1$ (resto 0)

Passo 9: Agora, anotar o último quociente e os restos de baixo para cima para obter o número na base 2:

342 na base 10 = 101010110 na base 2.

Para converter o número 342 da base 10 para a base 16 (hexadecimal), método de divisão sucessiva por 16. Passo a passo:

Passo 1: Dividir 342 por 16, mantendo a parte inteira da divisão.

$$342 \div 16 = 21 \text{ (resto 6)}$$

Passo 2: Dividir o resultado da divisão anterior por 16 novamente.

$$21 \div 16 = 1 \text{ (resto 5)}$$

Passo 3: Agora, anotar o último quociente e os restos de baixo para cima e, caso algum deles seja maior que 9, substituir pela respectiva letra:

342 na base 10 = 156 na base 16 (hexadecimal).

Portanto:

$$(342)_{10} = (101010110)_2 = (156)_{16}$$

Converter o número 539 na base 10, para a 16.

Para converter o número 539 na base 10 para a base 16, seguir os seguintes passos:

Passo 1: Divida 539 por 16 e escreva o quociente e o resto. O resto é o algarismo menos significativo na base 16.

$$539 \div 16 = 33 \text{ restante } 11$$

Portanto, o algarismo menos significativo é B.

Passo 2: Divida o quociente do passo anterior (33) por 16 e escreva o novo quociente e resto.

$$33 \div 16 = 2 \text{ restante } 1$$

Portanto, o próximo algarismo é 1.

Passo 3: Os algarismos obtidos a partir das divisões sucessivas, começando pelo quociente e resto mais recente até o primeiro quociente menor que 16, formam o número na base 16.

O número 539 na base 10 é igual a 21B na base 16.

1.6.3. CONVERSÃO ENTRE AS BASES 2 E 16

A conversão entre as bases 2 e 16 envolve a conversão de números binários (base 2) para números hexadecimais (base 16) e vice-versa. A base 2 utiliza apenas dois símbolos (0 e 1) para representar números, enquanto a base 16 utiliza os dígitos de 0 a 9 e as letras de A a F.

1.6.4. PARA CONVERTER UM NÚMERO BINÁRIO PARA HEXADECIMAL, SEGUIR OS SEGUINTE PASSOS:

Passo 1: Dividir o número binário em grupos de 4 dígitos começando da direita para a esquerda. Se o número de dígitos não for múltiplo de 4, adicionar zeros à esquerda para completar o último grupo.

Passo 2: Converter cada grupo de 4 dígitos para um único dígito hexadecimal. Usar a seguinte tabela de conversão:

Binário: 0000 - Hexadecimal: 0

Binário: 0001 - Hexadecimal: 1

Binário: 0010 - Hexadecimal: 2

Binário: 0011 - Hexadecimal: 3

Binário: 0100 - Hexadecimal: 4

Binário: 0101 - Hexadecimal: 5

Binário: 0110 - Hexadecimal: 6

Binário: 0111 - Hexadecimal: 7

Binário: 1000 - Hexadecimal: 8

Binário: 1001 - Hexadecimal: 9

Binário: 1010 - Hexadecimal: A

Binário: 1011 - Hexadecimal: B

Binário: 1100 - Hexadecimal: C

Binário: 1101 - Hexadecimal: D

Binário: 1110 - Hexadecimal: E

Binário: 1111 - Hexadecimal: F

Passo 3: Juntar os dígitos hexadecimais para obter o número hexadecimal resultante.

Exemplo:

Convertendo o número binário 110110101 para hexadecimal.

Passo 1: O número binário possui 9 dígitos, então possui os grupos: (1), (1011) e (0101).

Passo 2: Convertendo cada grupo: (1)=1, (1011)=B e (0101)=5

Passo 3: Juntando os dígitos hexadecimais, obtemos o número hexadecimal resultante: 1B5.

Portanto, o número binário 110110101 é equivalente ao número hexadecimal 1B5.

1.6.5. PARA CONVERTER UM NÚMERO HEXADECIMAL PARA BINÁRIO, SEGUIR ESSES PASSOS:

Passo 1: Converta cada dígito hexadecimal para um grupo de 4 dígitos binários usando a tabela de conversão acima.

Passo 2: Junte os grupos de 4 dígitos binários para obter o número binário resultante.

Exemplo:

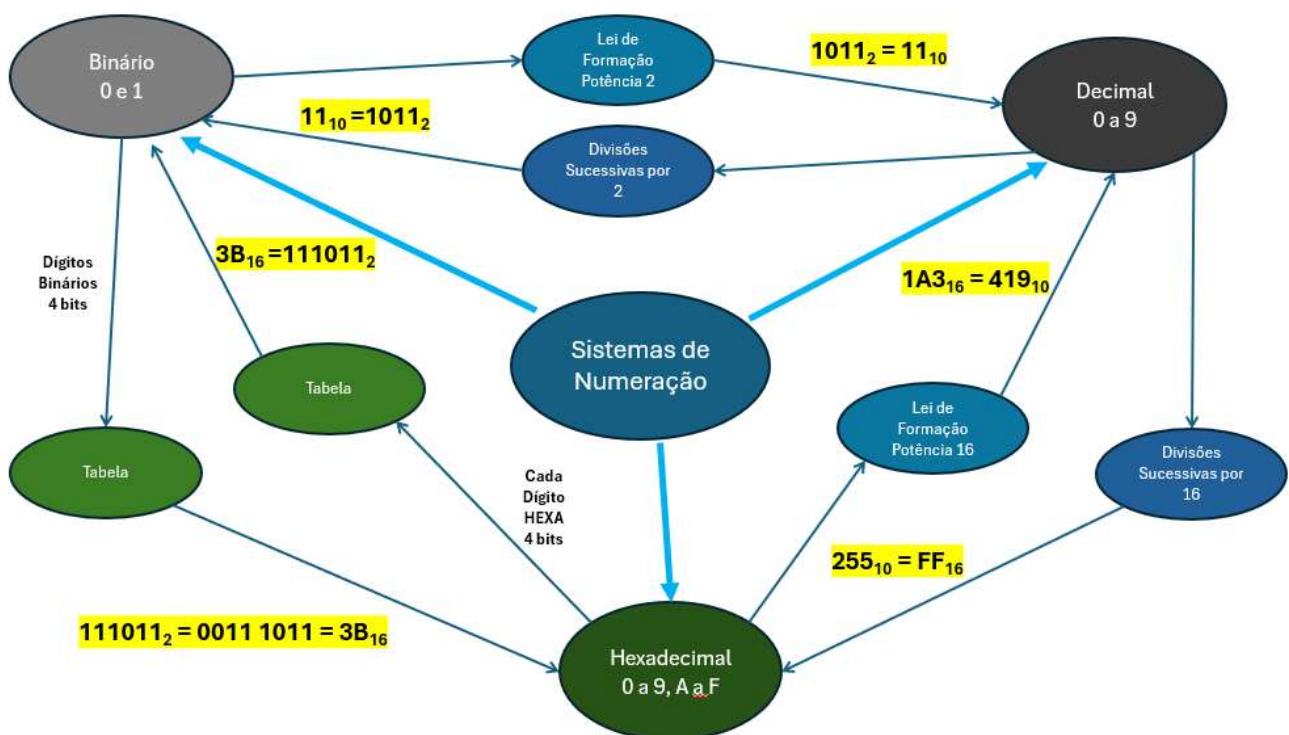
Convertendo o número hexadecimal C8 para binário.

Passo 1: Convertendo o dígito C = 1100 e o dígito 8 = 1000.

Passo 2: Juntar os grupos de 4 dígitos binários, o número binário resultante: 11001000.

Portanto, o número hexadecimal C8 é equivalente ao número binário 11001000.

1.7. MAPA MENTAL – CONVERSÃO ENTRE OS SISTEMAS DE NUMERAÇÃO



LISTA 1 – SISTEMAS DE NUMERAÇÃO

1) Converta para o sistema decimal.

- $(111101)_2 = (\quad)_{10}$
- $(11FA)_{16} = (\quad)_{10}$
- $(101001)_2 = (\quad)_{10}$

d) $(011101)_2 = (\quad)_{10}$

e) $(AE01)_{16} = (\quad)_{10}$

f) $(1FC8)_{16} = (\quad)_{10}$

2) Converta para binário.

a) $(227)_{10} = (\quad)_2$

b) $(10)_{10} = (\quad)_2$

c) $(28)_{10} = (\quad)_2$

3) Converta para hexadecimal.

a) $(385)_{10} = (\quad)_{16}$

b) $(135)_{10} = (\quad)_{16}$

c) $(1001)_{10} = (\quad)_{16}$

d) $(487)_{10} = (\quad)_{16}$

4) Converta de hexa para binário.

a) $(2ED)_{16} = (\quad)_2$

b) $(ABF)_{16} = (\quad)_2$

c) $(86)_{16} = (\quad)_2$

d) $(8F)_{16} = (\quad)_2$

5) Converta de binário para hexa.

a) $(10001000)_2 = (\quad)_{16}$

b) $(1110011)_2 = (\quad)_{16}$

c) $(111100111100)_2 = (\quad)_{16}$

6)

Converta para o sistema indicado.

a) $133)_{10} = (\quad)_2$

b) $6677)_{10} = (\quad)_{16}$

c) $c)(11110001)_2 = (\quad)_{16}$

d) $d)(4F)_{16} = (\quad)_{10}$

e) $e)(6F)_{16} = (\quad)_2$

LISTA 2 – SISTEMAS DE NUMERAÇÃO – Sistema Octal

1) Elabore um sistema numérico de base 8(octal). Apresente os algarismos e a sequência de números equivalentes ao sistema decimal até $(21)_{10}$.

2) Transforme os números abaixo, para o sistema numérico criado no exercício anterior.

a) $(205)_{10} = (\quad)_8$

b) $(10011011)_2 = (\quad)_8$

c) $(DF)_{16} = (\quad)_8$

d) $(B2F)_{16} = (\quad)_8$

3) Contar em octal de 6668 a 7108.

LISTA 3 – SISTEMAS DE NUMERAÇÃO - REVISÃO

Converta para o sistema indicado.

$(28)_{10} = (\quad)_2$

$(77)_{10} = (\quad)_{16}$

$(17)_{16} = (\quad)_{10}$

$(100011)_2 = (\quad)_{10}$

$(F39)_{16} = (\quad)_2$

$(101010111111)_2 = (\quad)_{16}$

$(FOCA)_{16} = (\quad)_2$

$(FOCA)_{16} = (\quad)_{10}$

$(61)_{10} = (\quad)_2$

$(1000)_{16} = (\quad)_{10}$

2. PORTAS LÓGICAS

2.1. ÁLGEBRA BOOLEANA

A Álgebra Booleana é um ramo da matemática e da lógica que lida com expressões e operações lógicas baseadas nos valores binários (verdadeiro/falso, 1/0) e nas operações fundamentais da lógica, como AND, OR e NOT. Ela foi desenvolvida por George Boole no século XIX e é amplamente aplicada na eletrônica digital, sistemas de computadores e design de circuitos.

Os princípios da Álgebra Booleana são baseados em três operações fundamentais: AND (E), OR (OU) e NOT (NÃO). Essas operações são usadas para manipular variáveis lógicas que podem assumir os valores verdadeiro (1) ou falso (0). Além disso, a Álgebra Booleana possui propriedades e regras que permitem simplificar e analisar expressões lógicas de maneira sistemática.

Operação AND (E):

A operação AND é representada pelo símbolo de multiplicação ou ponto ($\cdot$). Ela retorna verdadeiro (1) somente quando todas as variáveis de entrada são verdadeiras. Caso contrário, retorna falso (0).

Operação OR (OU):

A operação OR é representada pelo símbolo de adição (+) ou pelo símbolo de pipe (|). Ela retorna verdadeiro (1) se pelo menos uma das variáveis de entrada for verdadeira. Retorna falso (0) apenas se todas as variáveis de entrada forem falsas.

Operação NOT (NÃO):

A operação NOT é representada pelo símbolo de sobrelinha ($\neg$ BARRADO) ou pela exclamação (!). Ela inverte o valor de uma variável de entrada, retornando o oposto do valor original. Se a entrada for verdadeira, retorna falso, e se a entrada for falsa, retorna verdadeiro.

Além dessas operações, a Álgebra Booleana também inclui outras propriedades importantes, como a identidade, a lei de De Morgan e a lei da distribuição, que ajudam a simplificar e transformar expressões lógicas complexas.

A Álgebra Booleana é amplamente utilizada em circuitos digitais, projetos de computadores, programação lógica e em muitas outras aplicações relacionadas ao processamento de informações digitais. Ela fornece as bases teóricas para a manipulação e análise de sistemas lógicos e ajuda a compreender a lógica subjacente aos sistemas digitais modernos.

Conjunto de valores:

{Falso, Verdadeiro} - raciocínio humano

{Desligado, Ligado} - circuitos de chaveamento

{0, 1} - sistema binário

{0V, +5V} - eletrônica digital

Conjunto de Operações:

complementação

multiplicação lógica

adição lógica

2.2. VARIÁVEIS LÓGICAS

As variáveis lógicas digitais são elementos fundamentais na representação e manipulação de informações em sistemas digitais. Elas são usadas para armazenar e representar estados lógicos, que podem ser interpretados como verdadeiros (1) ou falsos (0). Essas variáveis são amplamente utilizadas na eletrônica digital, programação e processamento de dados.

Existem duas principais formas de variáveis lógicas digitais: variáveis booleanas e bits.

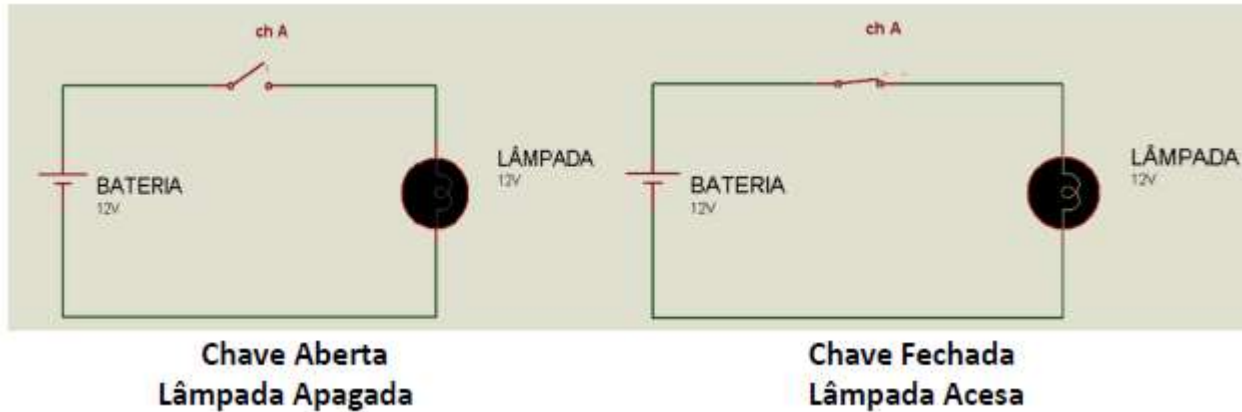
Variáveis Booleanas:

As variáveis booleanas são usadas para representar estados lógicos e são nomeadas em referência à Álgebra Booleana. Elas podem ter dois valores possíveis: verdadeiro (true) e falso (false). Na representação binária, o valor verdadeiro é associado a 1 e o valor falso é associado a 0. Geralmente, as variáveis booleanas são denotadas por letras maiúsculas, como A, B, C, etc.

As variáveis booleanas são utilizadas em expressões lógicas para realizar operações lógicas, como AND, OR e NOT. Elas também podem ser combinadas e manipuladas para controlar o fluxo de execução de programas, tomar decisões e representar condições em algoritmos e sistemas de controle.

Exemplo:

Se tivermos uma variável booleana A que representa o estado "interruptor ligado", seu valor será verdadeiro (1) quando o interruptor estiver ligado e falso (0) quando estiver desligado. Essa variável pode ser usada em expressões lógicas para verificar se o interruptor está ligado ou desligado.



Este circuito pode ser representado por variáveis lógicas da seguinte forma:

Chave A → variável C

Lâmpada → variável L

A partir daí, tem-se:

A chave está aberta → $C = 0$

A chave está fechada → $C = 1$

A lâmpada está apagada → $L = 0$

A lâmpada está acesa → $L = 1$

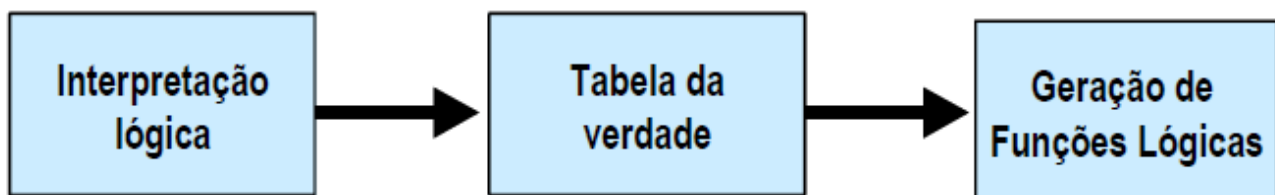
Matematicamente:

$$L = C$$

Conhecida como
função lógica

2.3. TABELA VERDADE

A tabela verdade é uma representação tabular que mostra todas as possíveis combinações de valores de entrada e os correspondentes valores de saída para uma função ou expressão lógica em um sistema digital. Ela é usada para descrever o comportamento de uma função lógica e permite analisar e entender o funcionamento de circuitos digitais.



Em uma tabela verdade, as colunas representam as variáveis de entrada e a última coluna representa a saída correspondente. Cada linha da tabela mostra uma combinação específica de valores de entrada e o resultado da função lógica para esses valores.

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

A tabela verdade é uma ferramenta essencial para analisar, projetar e verificar o comportamento de circuitos e funções lógicas. Ela permite identificar casos especiais, determinar a corretude da função lógica e auxiliar na simplificação de expressões lógicas complexas.

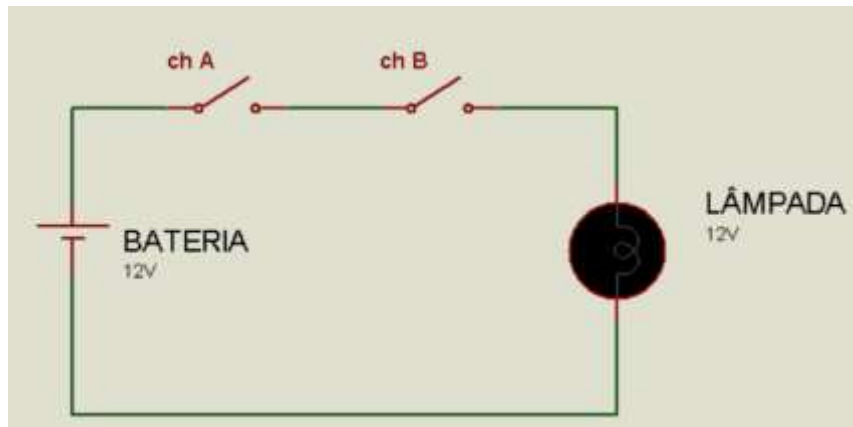
2.4. PORTAS LÓGICAS

As portas lógicas digitais mais básicas: a porta E (AND), a porta OU (OR) e a porta NÃO (NOT). Essas portas são componentes fundamentais na eletrônica digital e são utilizadas para realizar operações lógicas em sinais digitais.

Essas portas lógicas são a base para a construção de circuitos digitais mais complexos, permitindo realizar operações lógicas e manipular dados digitais de diversas maneiras.

2.4.1. PORTA E (AND):

A porta E recebe dois sinais de entrada e produz um sinal de saída somente se ambos os sinais de entrada forem verdadeiros (ou seja, se ambos os sinais estiverem em nível alto). Caso contrário, a saída será falsa (nível baixo).



Exemplo:

Suponha que temos duas entradas A e B. Se A representar "chave ligada" e B representar "botão pressionado", uma porta E será usada para verificar se a chave está ligada e o botão foi pressionado ao mesmo tempo. Se ambos os sinais forem verdadeiros, a saída será verdadeira (nível alto), indicando que ambas as condições foram atendidas.

- A **PORTA E** é aquela que assume o valor 1 quando todas as variáveis de entrada forem iguais a 1; e assume o valor 0 quando pelo menos uma das variáveis de entrada for igual a 0.
- Expressão Booleana: **$S = A \bullet B$**

Tabela da verdade para a porta E:

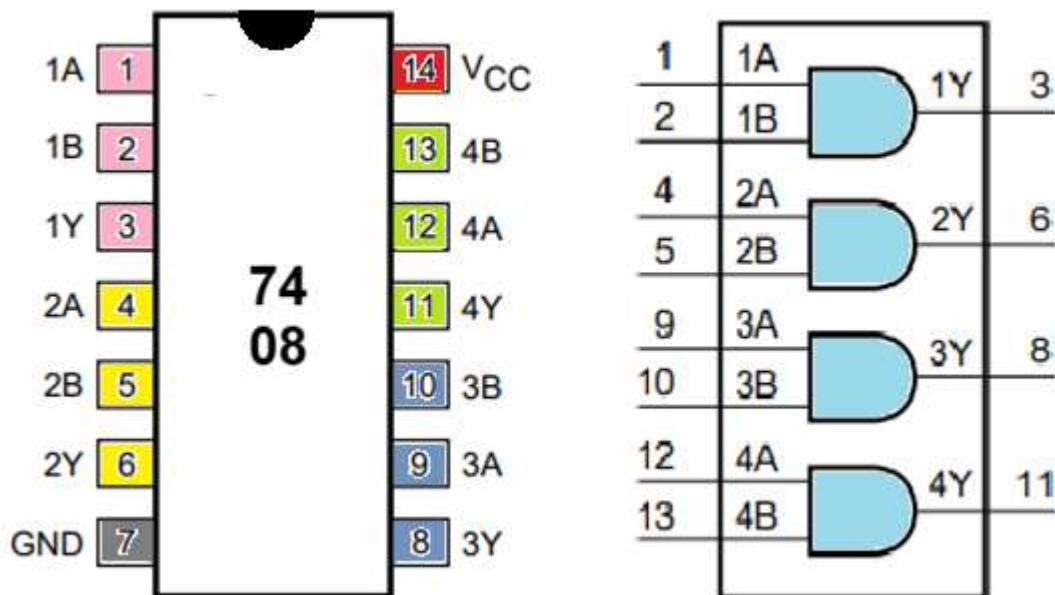
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Símbolo:



Circuito Integrado: 7408

Esses são os pinos básicos de um CI 7408. Cada porta lógica AND dentro do chip tem duas entradas e uma saída.

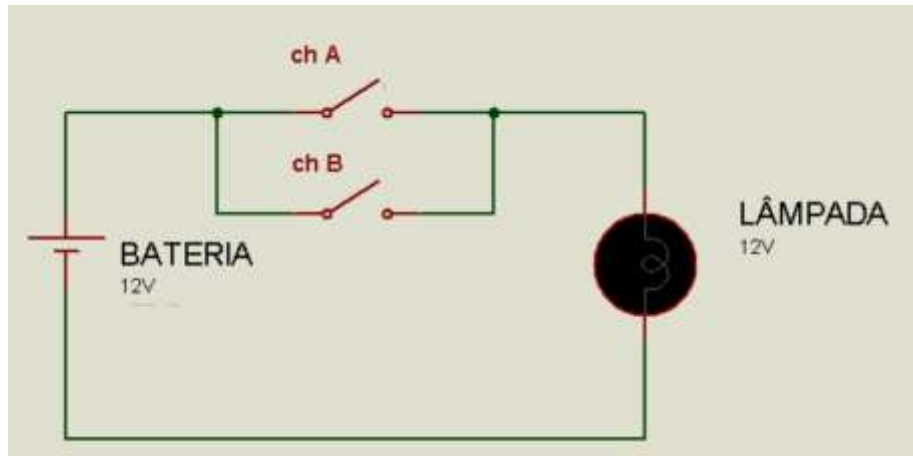


O CI 7408 é um dos chips da família TTL (Transistor-Transistor Logic), que consiste em portas lógicas AND de 4 entradas. Cada pino tem uma função específica. Segue a descrição dos pinos do CI 7408:

1. Entrada A1 (1A): É uma das entradas da primeira porta lógica AND.
2. Entrada B1 (1B): É a outra entrada da primeira porta lógica AND.
- 3. Saída 1 (1Y): É a saída da primeira porta lógica AND.**
4. Entrada A2 (2A): É uma das entradas da segunda porta lógica AND.
5. Entrada B2 (2B): É a outra entrada da segunda porta lógica AND.
- 6. Saída 2 (2Y): É a saída da segunda porta lógica AND.**
- 7. GND (terra):** Este é o pino de referência de terra. Deve ser conectado ao terra do circuito.
- 8. Saída 3 (3Y):** É a saída da terceira porta lógica AND.**
9. Entrada A3 (3A): É uma das entradas da terceira porta lógica AND.
10. Entrada B3 (3B): É a outra entrada da terceira porta lógica AND.
- 11. Saída 4 (4Y): É a saída da quarta porta lógica AND.**
12. Entrada A4 (4A): É uma das entradas da quarta porta lógica AND.
13. Entrada B4 (4B): É a outra entrada da quarta porta lógica AND.
- 14. VCC:** Este é o pino de alimentação. Deve ser conectado à fonte de alimentação positiva (+5V em geral).

2.4.2. PORTA OU (OR):

A porta OU recebe dois sinais de entrada e produz um sinal de saída verdadeiro se pelo menos um dos sinais de entrada for verdadeiro (ou seja, se pelo menos um dos sinais estiver em nível alto). A saída será falsa apenas se ambos os sinais de entrada forem falsos (níveis baixos).



Exemplo:

Suponha que temos duas entradas A e B. Se A representar "luz acesa" e B representar "alarme ativado", uma porta OU será usada para verificar se a luz está acesa ou o alarme está ativado. Se pelo menos um dos sinais de entrada for verdadeiro, a saída será verdadeira (nível alto), indicando que uma das condições foi atendida.

- A **PORTA OU** é aquela que assume valor 1 quando uma ou mais variáveis de entrada forem iguais a 1; e assume o valor 0 quando todas as variáveis de entrada forem iguais a 0.
- Expressão Booleana: $S = A + B$

Tabela da verdade para a porta OU:

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

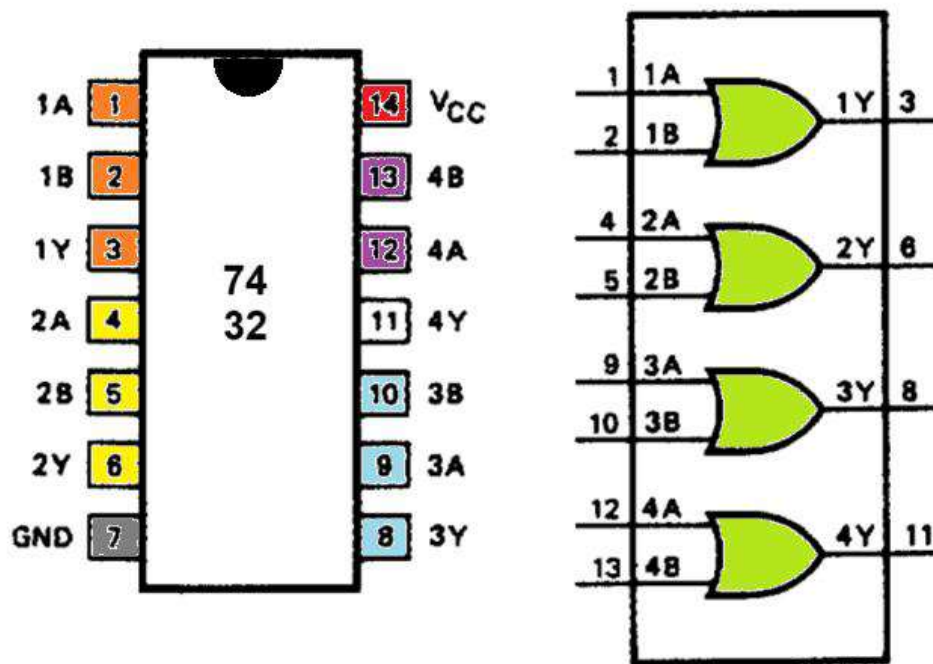
Símbolo:



Circuito Integrado: 7432

O CI 7432 é outro chip da família TTL, sendo composto por portas lógicas OR de 4 entradas. Segue a descrição dos pinos do CI 7432:

1. Entrada A1 (1A): É uma das entradas da primeira porta lógica OR.
 2. Entrada B1 (1B): É a outra entrada da primeira porta lógica OR.
 - 3. Saída 1 (1Y): É a saída da primeira porta lógica OR.**
 4. Entrada A2 (2A): É uma das entradas da segunda porta lógica OR.
 5. Entrada B2 (2B): É a outra entrada da segunda porta lógica OR.
 - 6. Saída 2 (2Y): É a saída da segunda porta lógica OR.**
 7. **GND** (terra): Este é o pino de referência de terra. Deve ser conectado ao terra do circuito.
 - 8. Saída 3 (3Y): É a saída da terceira porta lógica OR.**
 9. Entrada A3 (3A): É uma das entradas da terceira porta lógica OR.
 10. Entrada B3 (3B): É a outra entrada da terceira porta lógica OR.
 - 11. Saída 4 (4Y): É a saída da quarta porta lógica OR.**
 12. Entrada A4 (4A): É uma das entradas da quarta porta lógica OR.
 13. Entrada B4 (4B): É a outra entrada da quarta porta lógica OR.
 - 14. VCC:** Este é o pino de alimentação. Deve ser conectado à fonte de alimentação positiva (+5V em geral).
- Esses são os pinos básicos de um CI 7432. Cada porta lógica OR dentro do chip tem duas entradas e uma saída.



2.4.3. PORTA NÃO (NOT):

A porta NÃO (também conhecida como inversor) recebe um sinal de entrada e produz a negação desse sinal na saída. Ou seja, se a entrada for verdadeira (nível alto), a saída será falsa (nível baixo), e vice-versa.

Exemplo:

Suponha que temos uma entrada A que representa "interruptor ligado". Uma porta NÃO será usada para inverter o estado do interruptor. Se a entrada A for verdadeira (interruptor ligado), a saída será falsa (interruptor desligado), e vice-versa.

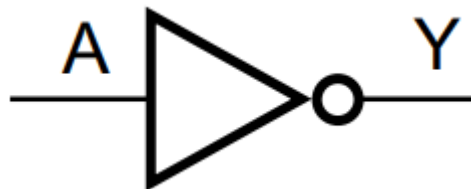
- A **PORTA NÃO**, ou função complemento, ou ainda, função inversora é a que inverte o estado da variável de entrada. Se a variável de entrada for 1, ela se tornará 0 na saída. Se a variável de entrada for 0, ela se tornará 1 na saída.

$$S = \bar{A}$$

Tabela da verdade para a porta NÃO:

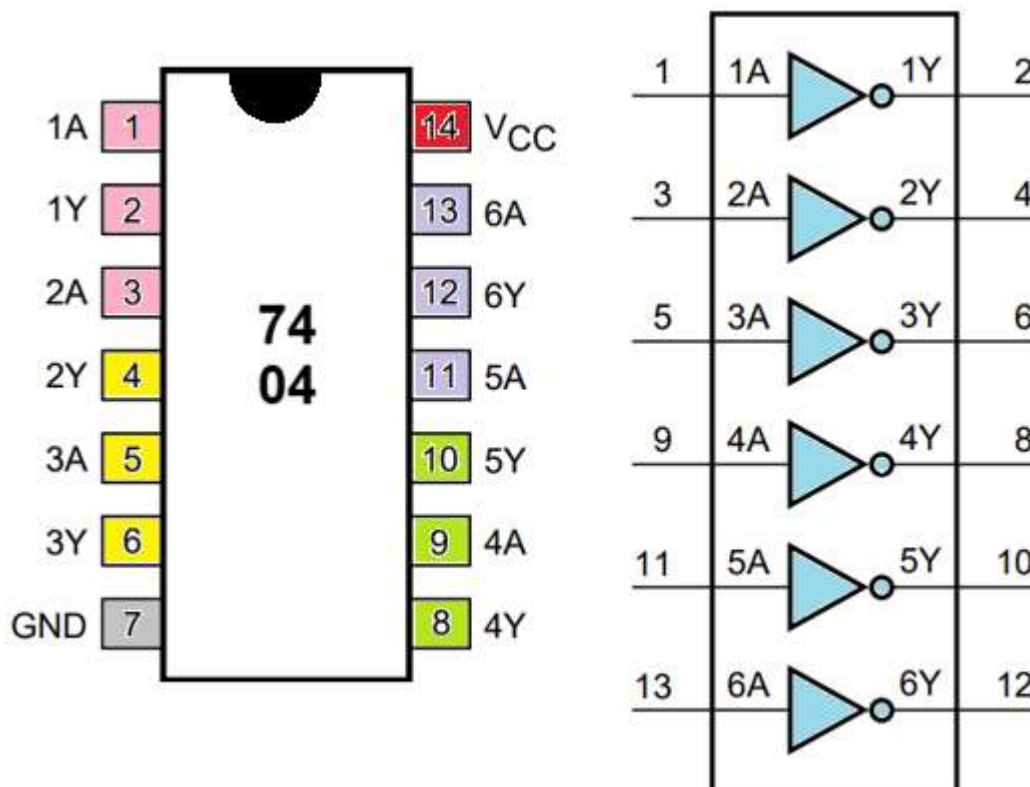
A	S
0	1
1	0

Símbolo:



Circuito Integrado: 7404

Esses são os pinos básicos de um CI 7404. Cada porta lógica NOT dentro do chip tem uma entrada e uma saída.



O CI 7404 é um chip da família TTL que consiste em seis inversores (NOT gates). Aqui está a descrição dos pinos do CI 7404:

1. Entrada A1 (1A): É a entrada do primeiro inversor.

2. Saída 1 (1Y): É a saída do primeiro inversor.
3. Entrada A2 (2A): É a entrada do segundo inversor.
4. Saída 2 (2Y): É a saída do segundo inversor.
5. Entrada A3 (3A): É a entrada do terceiro inversor.
6. Saída 3 (3Y): É a saída do terceiro inversor.
7. **GND** (terra): Este é o pino de referência de terra. Deve ser conectado ao terra do circuito.
8. Saída 4 (4Y): É a saída do quarto inversor.
9. Entrada A4 (4A): É a entrada do quarto inversor.
10. Saída 5 (5Y): É a saída do quinto inversor.
11. Entrada A5 (5A): É a entrada do quinto inversor.
12. Saída 6 (6Y): É a saída do sexto inversor.
13. Entrada A6 (6A): É a entrada do sexto inversor.
14. **VCC**: Este é o pino de alimentação. Deve ser conectado à fonte de alimentação positiva (+5V em geral).

2.4.4. PORTA OU EXCLUSIVO - XOR

A expressão que representa a porta XOR é:

$$S = \bar{A} \cdot B + A \cdot \bar{B}$$

Uma alternativa de representação da porta XOR é:

$$S = A \oplus B$$

Lê-se: A OU EXCLUSIVO B.

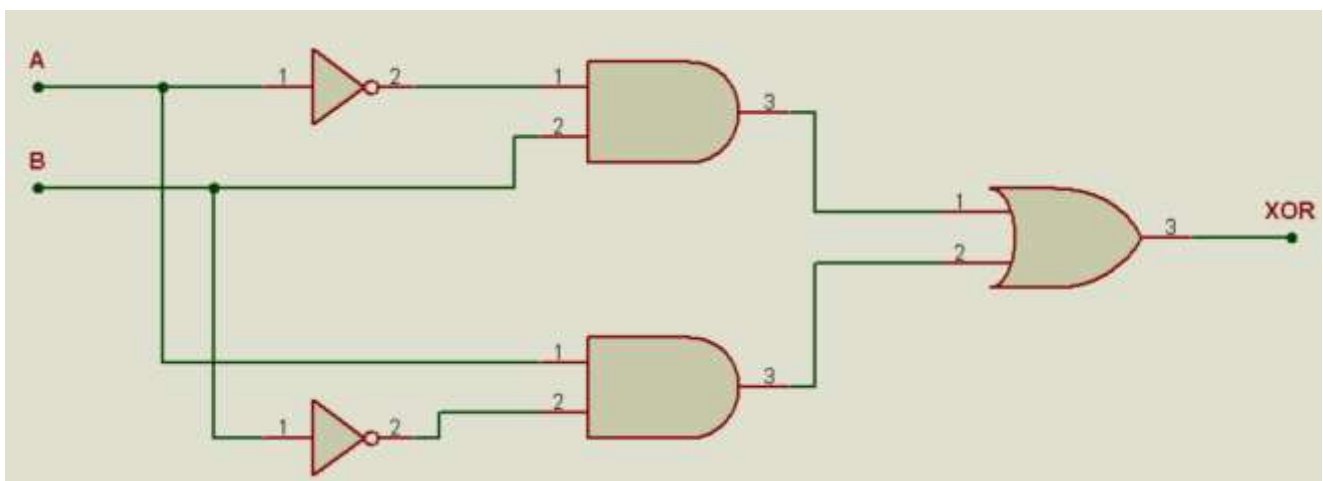
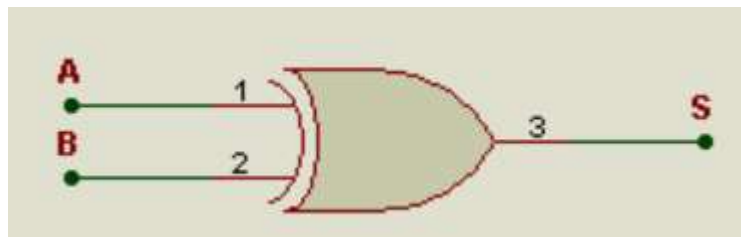


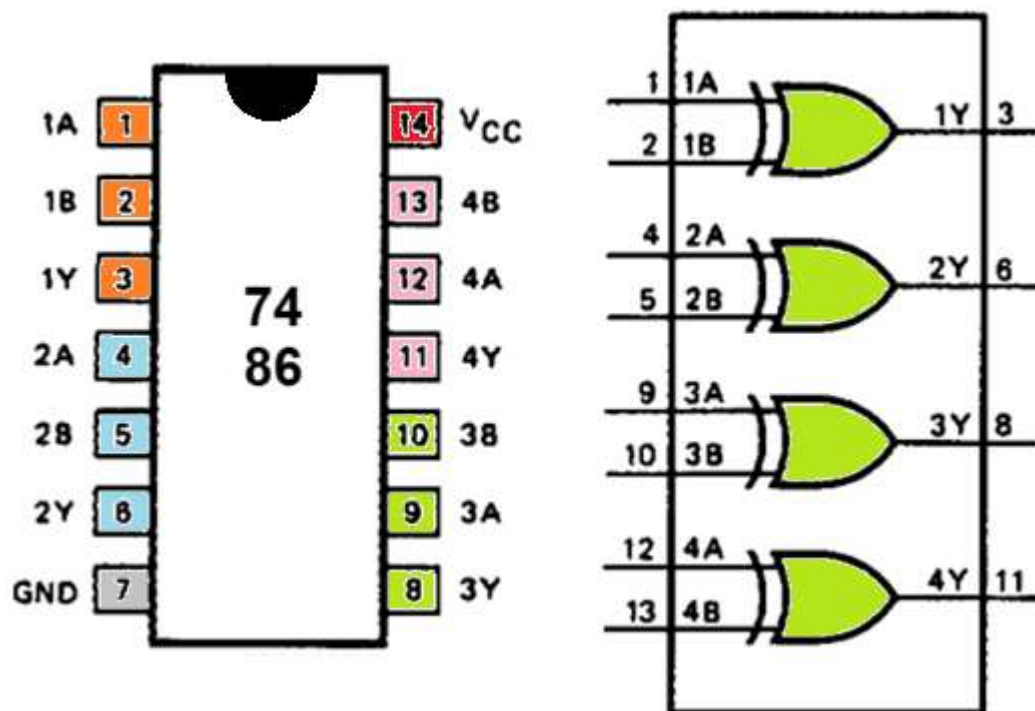
Tabela da verdade para a porta XOR:

XOR		
Entradas		Saída
A	B	$S = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Símbolo:



Circuito Integrado: 7486



2.4.5. PORTA NOU EXCLUSIVO – XNOR

A expressão que representa a porta XNOR é:

$$S = A \cdot B + \bar{A} \cdot \bar{B}$$

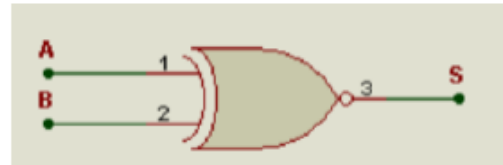
Uma alternativa de representação da porta XNOR é:

$$S = \overline{A \oplus B} = A \odot B$$

Lê-se: A COINCIDÊNCIA B.

XOR		
Entradas		Saída
A	B	$S = A \odot B$
0	0	0
0	1	1
1	0	1
1	1	0

Símbolo:



2.4.6. EQUAÇÕES BÁSICAS BOOLEANAS

$$S = A + 1 = 1$$

$$S = A + 0 = A$$

$$S = A * 1 = A$$

$$S = A * 0 = 0$$

$$S = A + A = A$$

$$S = A * A = A$$

$$S = A + A' = 1$$

$$S = A * A' = 0$$

3. CIRCUITOS COMBINACIONAIS

3.1. DEFINIÇÃO DE CIRCUITOS COMBINACIONAIS

Circuitos combinacionais são tipos de circuitos digitais em que a saída depende exclusivamente das entradas atuais, sem levar em consideração o histórico anterior das entradas. Esses circuitos realizam uma operação lógica direta, sem armazenar informações.

Em outras palavras, circuitos combinacionais executam uma função lógica definida pelas entradas para produzir uma saída correspondente. Eles são projetados para lidar com operações instantâneas e não possuem elementos de memória.

Exemplos de circuitos combinacionais incluem:

1. Porta lógica AND: Essa porta recebe duas entradas e fornece uma saída alta (1) somente quando ambas as entradas estão em nível alto (1). Caso contrário, a saída é baixa (0).
2. Porta lógica OR: Assim como a porta AND, a porta OR recebe duas entradas, mas fornece uma saída alta (1) quando qualquer uma das entradas está em nível alto (1).
3. Decodificador: É um circuito combinacional usado para converter códigos binários em uma combinação de saídas, onde cada saída representa um valor único.
4. Somador binário: É um circuito que combina duas entradas binárias e produz uma soma, além de um bit de transporte (carry).

Esses são apenas alguns exemplos de circuitos combinacionais. Existem muitos outros circuitos que realizam funções lógicas complexas, incluindo multiplexadores, demultiplexadores, comparadores, entre outros.

3.2. IMPLEMENTAÇÃO DE CIRCUITOS COMBINACIONAIS.

A implementação de circuitos combinacionais envolve várias etapas, desde a definição do problema até a construção física do circuito. Principais etapas envolvidas:

1. Definição de requisitos: Determinar os requisitos do circuito combinacional, como o número de entradas e saídas, as funções lógicas desejadas e as restrições de tempo ou recursos.
2. Projeto lógico: Projetar o circuito em nível lógico, usando diagramas lógicos, tabelas-verdade ou expressões booleanas. Identifique as portas lógicas necessárias e suas interconexões.
3. Simplificação: Simplificar as funções lógicas usando técnicas como mapa de Karnaugh, álgebra booleana ou minimização de expressões booleanas. Isso ajuda a reduzir o número de portas lógicas necessárias e a melhorar a eficiência do circuito.
4. Seleção de componentes: Escolher os componentes eletrônicos necessários para implementar as portas lógicas do circuito.
5. Esquemático: Desenhar o esquema do circuito combinacional utilizando símbolos padrão para as portas lógicas e componentes selecionados. Certifique-se de que as conexões entre eles estejam corretas.
6. Layout: Converter o desenho esquemático em um layout físico, posicionando os componentes e traçando as trilhas de cobre em uma placa de circuito impresso (PCB) ou projetando um circuito integrado.
7. Fabricação: Produzir o circuito físico de acordo com o layout, seguindo os processos de fabricação adequados. Isso pode incluir a fabricação de PCBs, a montagem manual ou robótica dos componentes, a soldagem e outros processos de fabricação eletrônica.
8. Teste e validação: Verificar se o circuito está funcionando corretamente, aplicando estímulos adequados às entradas e verificando se as saídas estão de acordo com as expectativas. Faça as correções necessárias, se houver problemas.

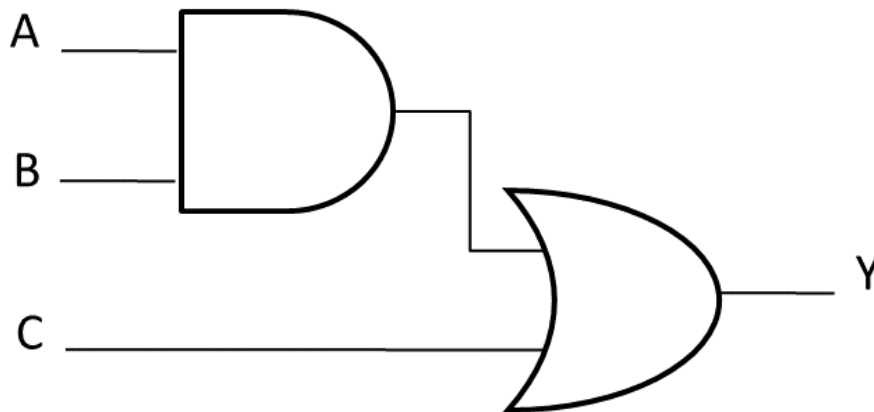
É importante observar que a implementação de circuitos combinacionais pode variar dependendo do nível de complexidade e dos recursos disponíveis. Para circuitos mais simples, pode-se usar uma

abordagem mais manual, enquanto para circuitos mais complexos, pode-se utilizar software de design específico ou até mesmo ferramentas CAD (Computer-Aided Design) para auxiliar na implementação.

3.3. OBTER EXPRESSÕES BOOLEANAS A PARTIR DE CIRCUITOS.

Escrever a expressão booleana que é executada por qualquer circuito lógico.

Por exemplo, escrever a expressão booleana que o circuito a seguir executa.



Divida o circuito em duas partes. A porta AND fornece a saída $S = A.B$, que será uma entrada na porta OR. A outra entrada da porta OR, será a variável C. Portanto, a expressão booleana final será: $Y = A.B + C$

3.4. OBTER O CIRCUITO LÓGICO A PARTIR DE EXPRESSÕES BOOLEANAS.

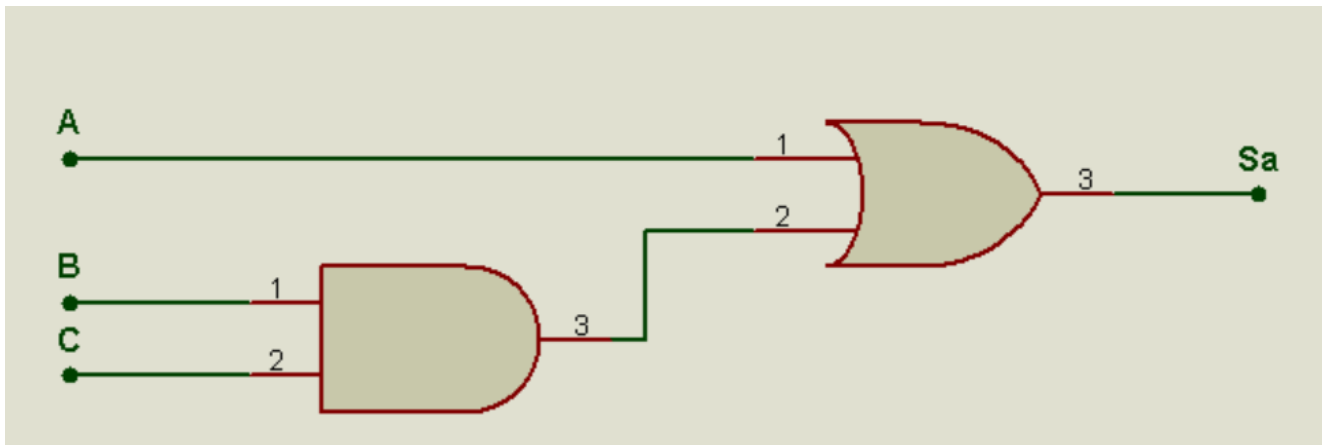
Para obter o circuito lógico a partir de expressões booleanas, seguir os seguintes passos:

1. Simplificar as expressões booleanas, se necessário, utilizando as propriedades da álgebra booleana, como a lei de Morgan e as leis de De Morgan.
2. Identificar as portas lógicas necessárias para realizar as operações desejadas. As portas lógicas mais comuns são: portas AND, OR, NOT e XOR.
3. Esboçar o diagrama do circuito lógico, colocando as portas lógicas necessárias e conectando as entradas e saídas corretamente.
4. Verificar se as expressões booleanas são implementadas corretamente no circuito lógico, analisando as saídas esperadas para todas as combinações possíveis de entradas.
5. Caso necessário, fazer os ajustes no circuito lógico para garantir que as expressões booleanas sejam corretamente implementadas.

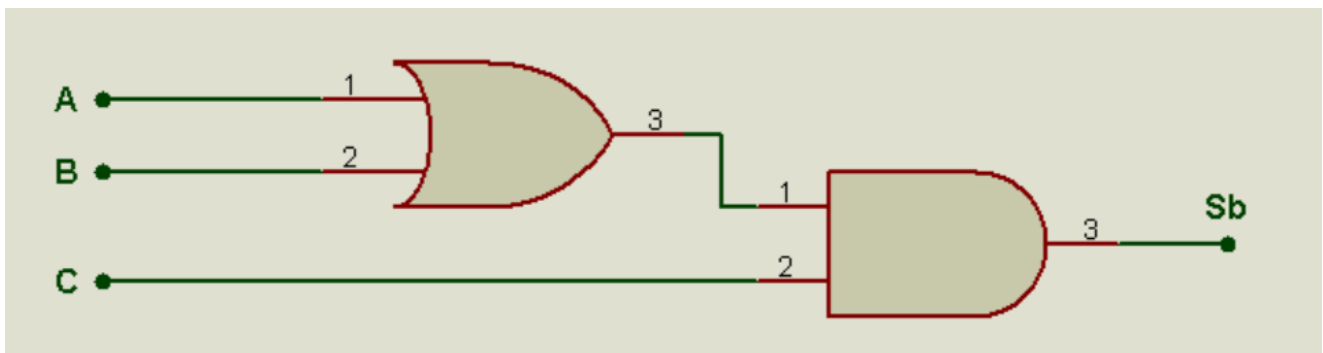
É importante lembrar que a simplificação das expressões booleanas pode ser fundamental para reduzir o número de portas lógicas necessárias e tornar o circuito mais eficiente.

Exemplo:

Dada a expressão booleana: $Sa = A + B.C$. Esboce o diagrama do circuito lógico.



B) Dada a expressão booleana: $Sb = (A+B).C$. Esboce o diagrama do circuito lógico.



Importante lembrar sobre as precedências:

1 - ()

2 - NOT

3 - AND

4 - OR

3.5. DETERMINAÇÃO DA TABELA-VERDADE A PARTIR DA EXPRESSÃO BOOLEANA

Problema: Dado uma expressão booleana, determinar a tabela-verdade que a represente.

A solução para este tipo de problema pode ser realizada por meio dos seguintes passos.

1. Montar uma tabela com todas as possibilidades.

2. Definir tantas colunas quantos forem os membros da expressão e preenchê-las com os resultados lógicos.
3. Acrescentar uma coluna para o resultado da expressão e preencher com os resultados lógicos.

O número de combinações possíveis entre as variáveis de entrada é dado por:

$$N = 2^n$$

Sendo N o número de combinações possíveis e n o número de variáveis lógicas de entrada.

Assim, para um sistema com três variáveis de entrada, teremos 2^3 , ou seja, oito combinações possíveis.

Por exemplo, dada a seguinte expressão lógica: $S = \overline{(A \cdot B)} \cdot C$

entradas					saída
A	B	C	A.B	$\overline{(A \cdot B)}$	$S = \overline{(A \cdot B)} \cdot C$
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	0	0

3.6. EXPRESSÕES BOOLEANAS OBTIDAS A PARTIR DA TABELA VERDADE

Supor que um circuito lógico de duas entradas A e B deva proporcionar na saída S os estados lógicos dados na tabela verdade abaixo:

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

Basicamente dois métodos através dos quais é possível obter diretamente a expressão da saída (S) na sua forma geral ou canônica.

São elas:

SOMA DE PRODUTOS (somente 1)

PRODUTO DE SOMAS (somente 0)

3.6.1. OBTENÇÃO DA EQUAÇÃO A PARTIR DE SOMA DE PRODUTOS

Procedimento:

a) Para cada condição em que a coluna de saída da tabela verdade for "1", faz-se o produto das variáveis de entrada, que devem ser negadas sempre que corresponderem ao estado zero.

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

$$\overline{A} \bullet \overline{B}$$

$$\overline{A} \bullet B$$

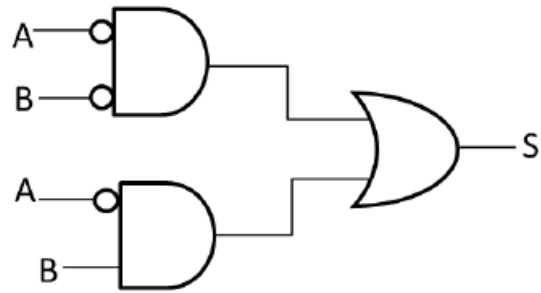
b) Soma-se os produtos assim obtidos igualando-se tudo a S.

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

c) Portanto: De posse da expressão característica da tabela verdade pode-se montar o circuito lógico correspondente.

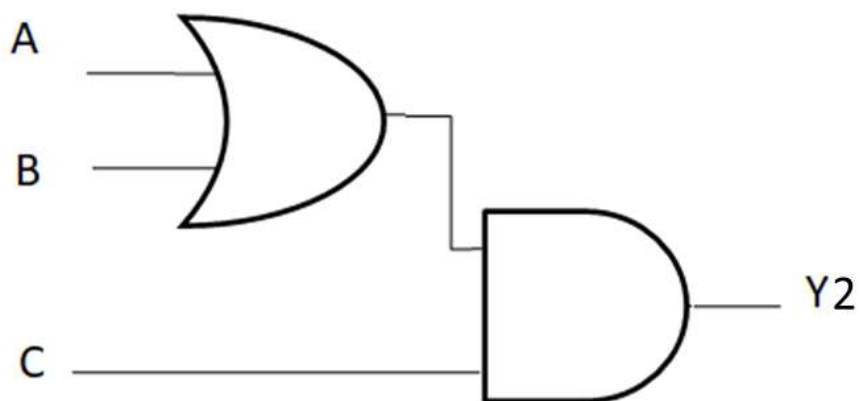
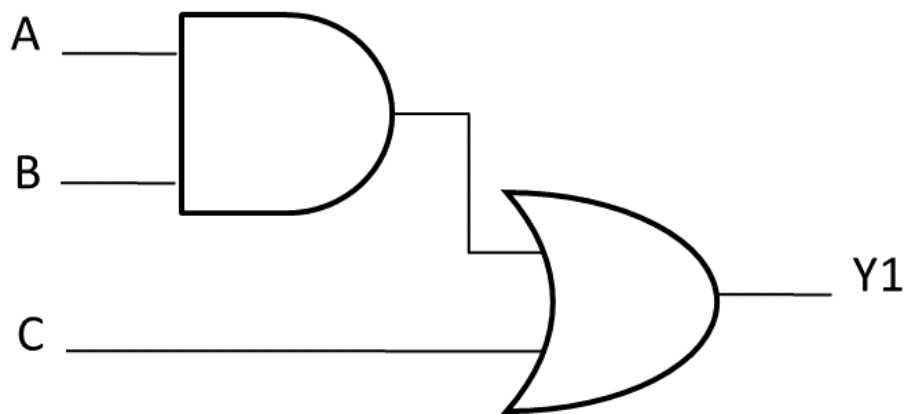
$$S = \overline{A} \bullet \overline{B} + \overline{A} \bullet B$$

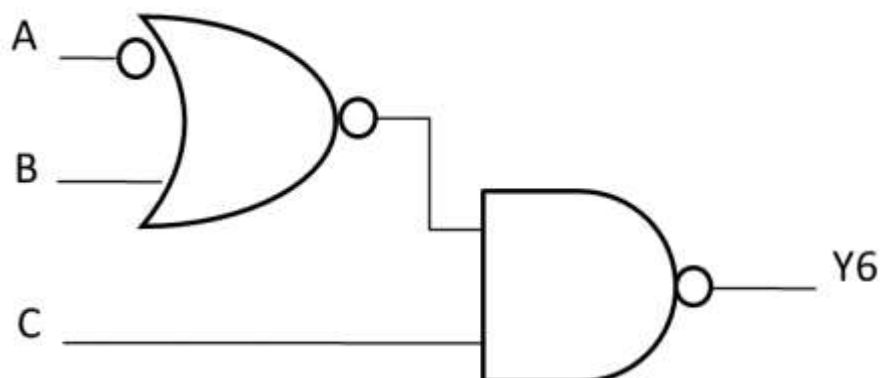
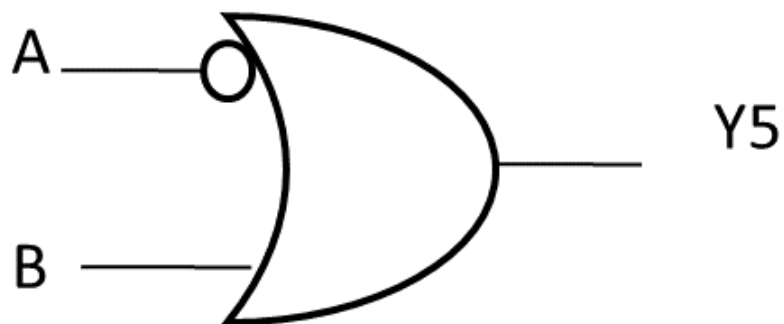
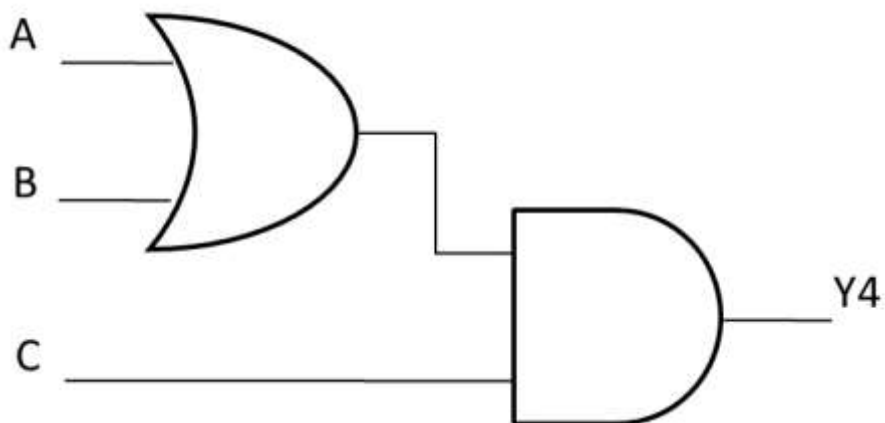
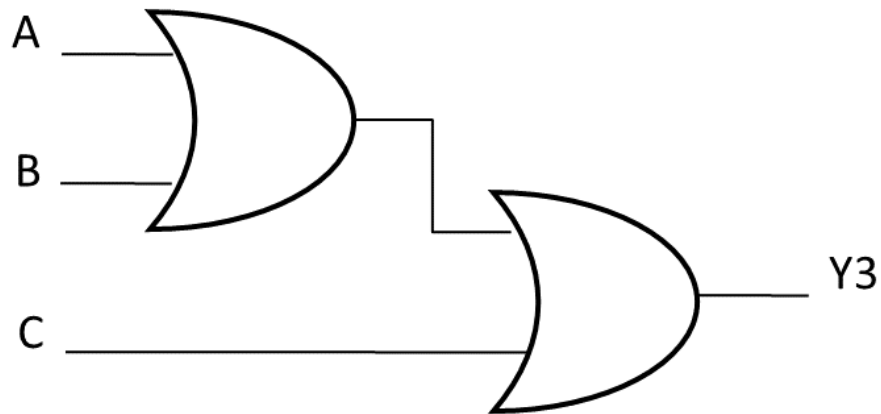
A	B	S
0	0	1
0	1	1
1	0	0
1	1	0



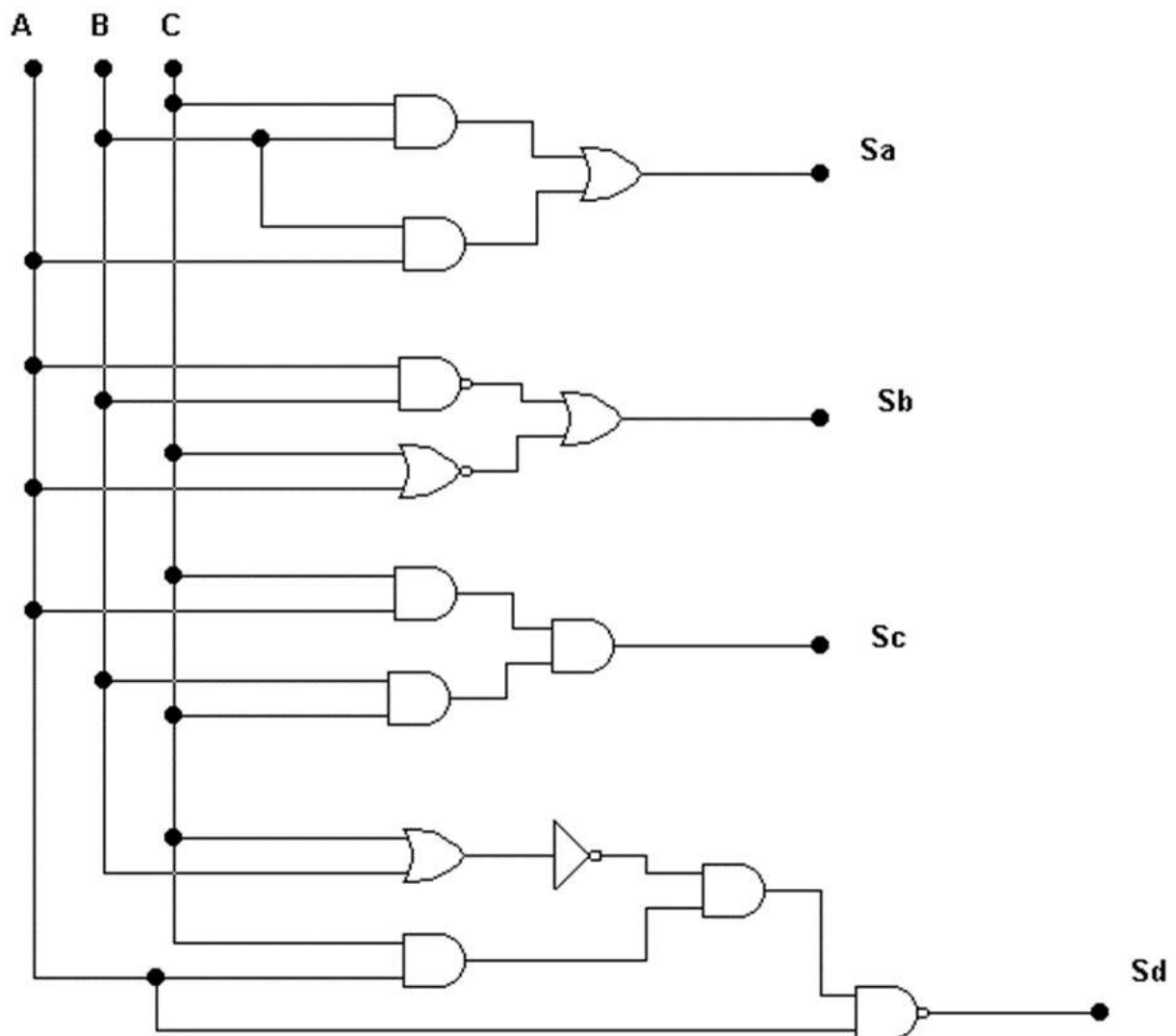
LISTA 4 – EXPRESSÃO BOOLEANA

1) Escrever a expressão booleana, dos circuitos abaixo:





2) Obtenha a expressão booleana para cada um dos circuitos abaixo:

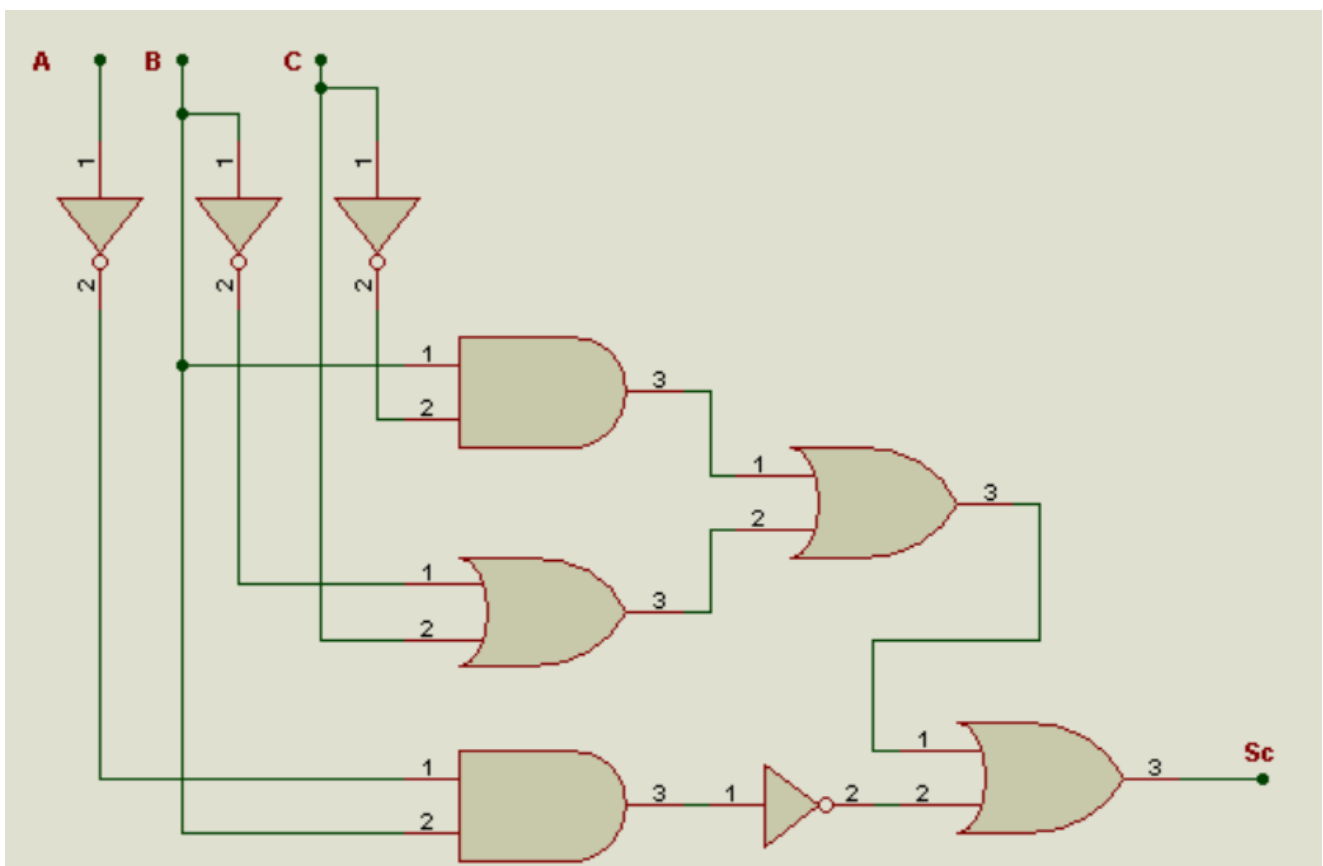
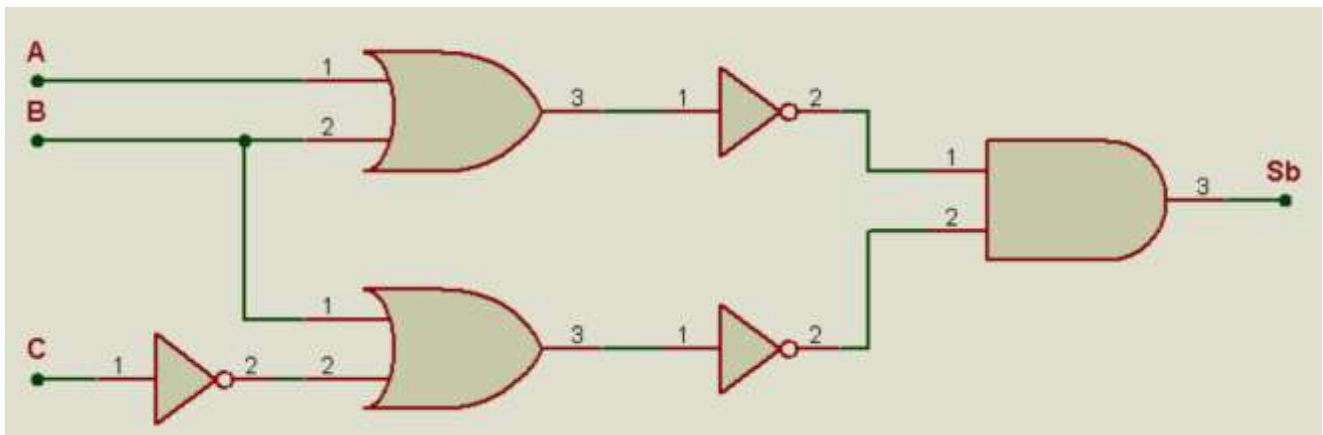
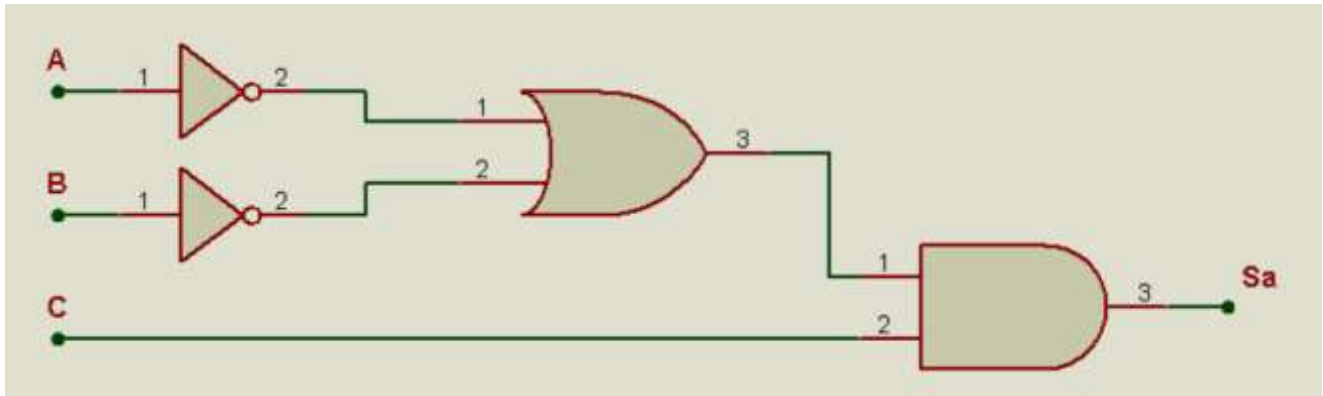


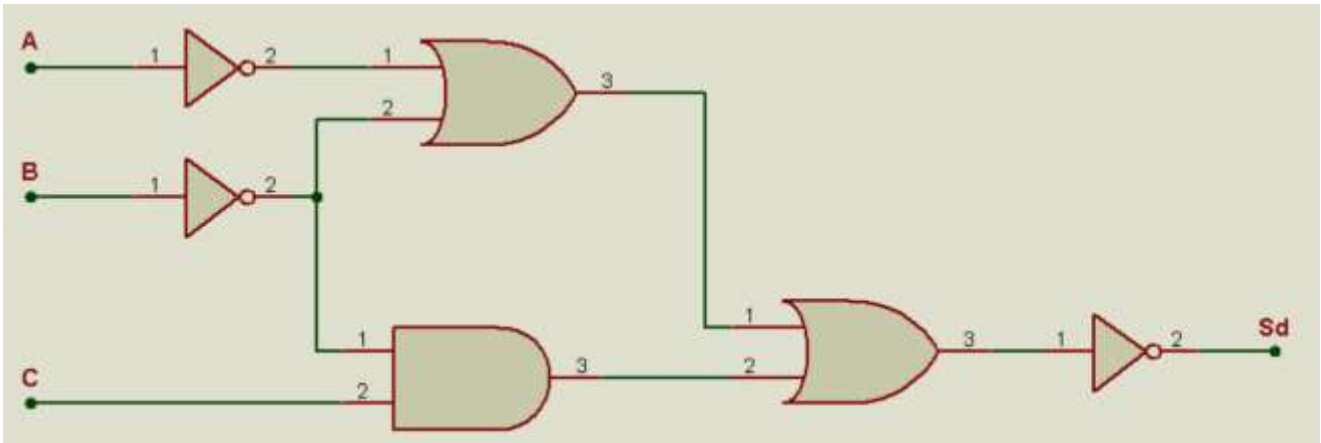
LISTA 5 – EXPRESSÃO BOOLEANA – TABELA VERDADE - CIRCUITO

Dado os circuitos abaixo:

Obter a expressão booleana do circuito;

Montar a tabela verdade.





LISTA 6 – EXPRESSÃO BOOLEANA – TABELA VERDADE

Seja a expressão booleana dada a seguir. Determinar a tabela-verdade que a represente.

$$S = A \cdot B$$

$$S = A \cdot B \cdot C$$

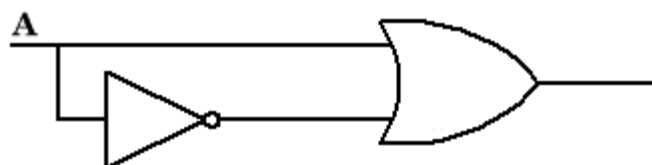
$$S = \bar{A} \cdot \bar{B} \cdot C$$

$$S = (A + \bar{B} + C) \cdot \bar{A}$$

LISTA 7 – EXPRESSÃO BOOLEANA – TABELA VERDADE – PROVE QUE

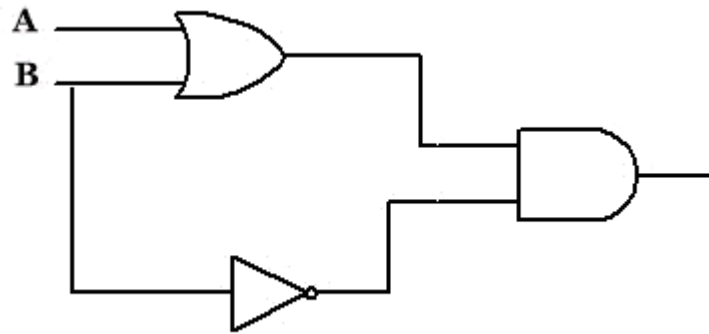
1) Dado o circuito ao lado: Escrever a expressão booleana;

Montar a tabela verdade.

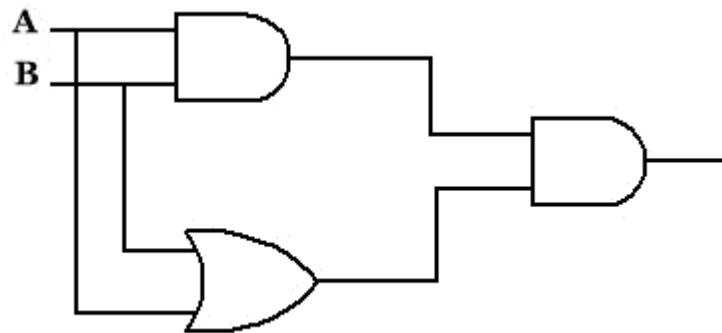


2) Dado o circuito ao lado: Escrever a expressão booleana;

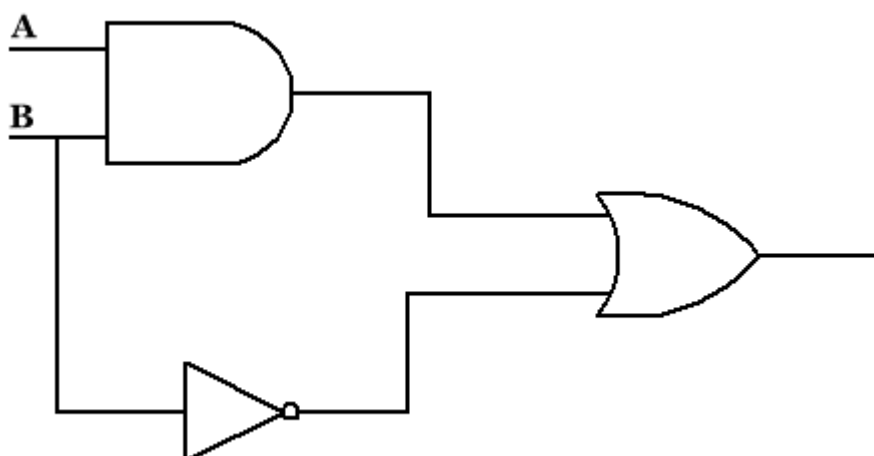
Montar a tabela verdade.



3) Dado o circuito ao lado: Escrever a expressão booleana;
Montar a tabela verdade.



4) Dado o circuito abaixo: Escrever a expressão booleana;
Montar a tabela verdade.



5) Dada as expressões abaixo: Monte a tabela verdade
Desenhe o circuito lógico

$$Sa = A.C + \overline{A.B.C}$$

$$Sb = A.\overline{B} + \overline{A} + A.B.\overline{C}$$

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

$$Sd = \overline{A.B} + \overline{A}.\overline{B}$$

6) prove que:

$$\left. \begin{array}{l} \text{a) } \overline{A \bullet B} = \overline{A} + \overline{B} \\ \text{b) } \overline{A + B} = \overline{A} \bullet \overline{B} \\ \text{c) } A \bullet B = \overline{\overline{A} + \overline{B}} \\ \text{d) } A + B = \overline{\overline{A} \bullet \overline{B}} \\ \text{e) } \overline{A} \bullet \overline{B} \neq \overline{A \bullet B} \\ \text{f) } \overline{A} + \overline{B} \neq \overline{A + B} \end{array} \right\} \text{De Morgan}$$

LISTA 8 – SOMA DE PRODUTOS

1) Monte as expressões booleana das saídas para a tabela verdade abaixo utilizando o método SOMA DOS PRODUTOS.

A	B	S1	S2	S3
0	0	1	1	1
0	1	0	1	1
1	0	1	1	0
1	1	1	0	1

2) Monte as expressões booleana das saídas para a tabela verdade abaixo utilizando o método SOMA DOS PRODUTOS.

A	B	C	S1	S2	S3
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	1	0	1	1

4. LABORATÓRIO

4.1. PORTA E

Resumo

A função E é definida como:

"Produce uma saída 1 somente quando todas as entradas forem 1"

A Figura 1 mostra a tabela verdade, o símbolo e a expressão booleana para a porta E.

ENTRADA		SAÍDA
A	B	$S = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

a)



b)

$$S = A \cdot B$$

c)

Figura 1 a) Tabela Verdade b) Símbolo c) Expressão Booleana

Objetivo

A familiarização com os GATES E, tabela verdade, conceito de entradas múltiplas.

EXPERIÊNCIA 1

a) Material Utilizado.

1 x CI74LS08

b) Lay-out do CI74LS08.

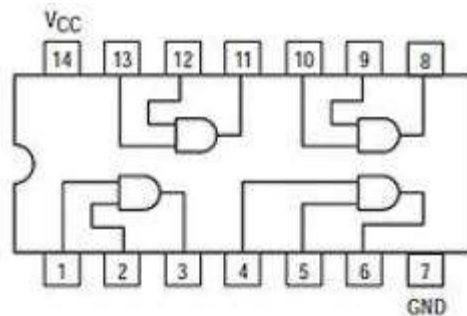


Figura 2 Lay-out CI74LS08

c) Monte o circuito da Figura 3.

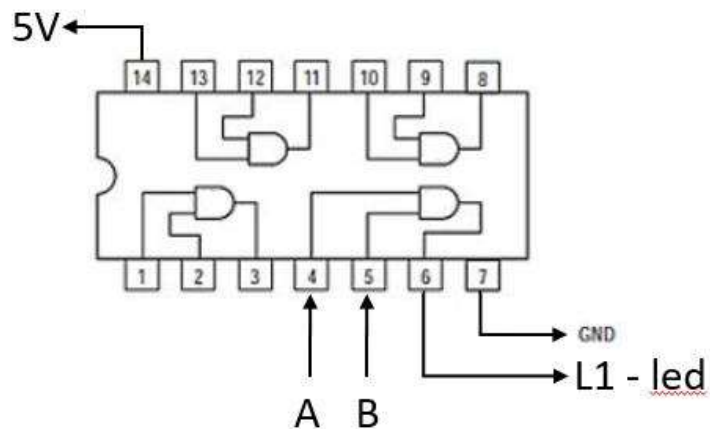


Figura 3 Circuito com Gate E

d) Complete a Tabela Verdade

ENTRADA		SAÍDA
A	B	$L0=A.B$
0	0	
0	1	
1	0	
1	1	

Figura 4 Tabela Verdade da Função E de duas entradas

EXPERIÊNCIA 2

Dada a expressão booleana: $S = (A.B).C$

Desenhe o circuito lógico da expressão booleana;

Monte o circuito da expressão booleana, utilizando o CI74LS08;

Crie e complete a tabela verdade do circuito do item (a).

EXPERIÊNCIA 3

Dada a expressão booleana: $S = (A.B).(B.C)$

Desenhe o circuito lógico da expressão booleana;

Monte o circuito da expressão booleana, utilizando o CI74LS08;

Crie e complete a tabela verdade do circuito do item (a).

EXPERIÊNCIA 4

Dada a expressão booleana: $S = ((A.B).(B.C)).A$

Desenhe o circuito lógico da expressão booleana;

Monte o circuito da expressão booleana, utilizando o CI74LS08;

Crie e complete a tabela verdade do circuito do item (a)

4.2. PORTA OU

Resumo

A função OU é definida como: “Se um ou mais entradas forem 1 a saída será 1”.

A Figura 5 mostra a tabela verdade, o símbolo e a expressão booleana para a porta OU:

ENTRADA		SAÍDA
A	B	$S = A+B$
0	0	0
0	1	1
1	0	1
1	1	1

a)



b)

$$S = A + B$$

c)

Figura 5 Porta OU a) Tabela Verdade b) Símbolo Lógico c) Expressão Booleana

Objetivo

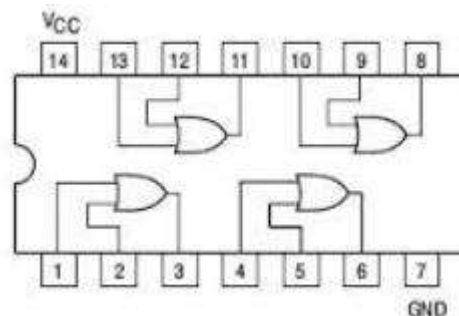
Neste ponto será examinado a tabela verdade da porta OU, o conceito de múltiplas entradas.

EXPERIÊNCIA 1

Material utilizado.

1 x CI74LS32

Lay-out do CI74LS32.



c) Monte o circuito da Figura 6.

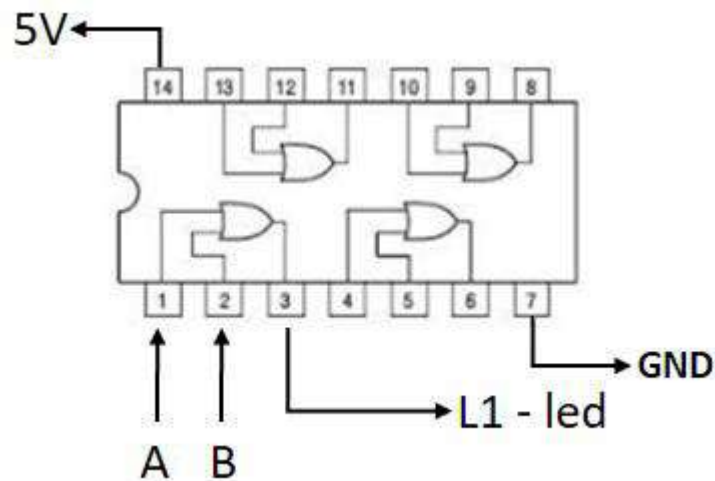


Figura 6 Circuito com a Porta OU

d) Complete a Tabela Verdade.

ENTRADA		SAÍDA
A	B	$L0 = A+B$
0	0	
0	1	
1	0	
1	1	

Figura 7 Tabela Verdade da porta OU

EXPERIÊNCIA 2

Material utilizado.

1 x CI74LS32

Monte o circuito da Figura 8.

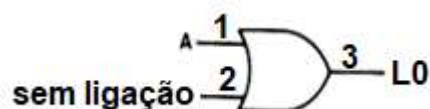


Figura 8 Porta OU com entrada flutuante

c) Completar a tabela da Figura 9.

ENTRADA	SAÍDA
A	L0
0	
1	

Figura 9 Tabela Verdade da Porta OU com entrada flutuante

EXPERIÊNCIA 3

Material utilizado.

1 x CI74LS32

Montar o circuito da Figura 10, ligando o pino 14 ao +5V e o pino 7 ao GND.

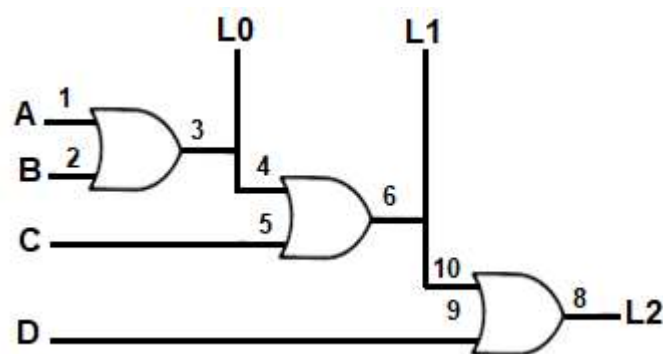


Figura 10 Circuito OU

c) Completar a tabela da Figura 11.

A	B	C	D	$L0=A+B$	$L1=(A+B)+C$	$L2=((A+B)+C)+D$
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Figura 11 Tabela Verdade

EXPERIÊNCIA 4

Dada a expressão booleana: $L0 = (A+B).(A.C)$

Desenhe o circuito lógico.

Monte o circuito da expressão booleana, utilizando o CI74LS08 e CI74LS32.

Crie e complete a tabela verdade do circuito do item (a).

EXPERIÊNCIA 5

Dada a expressão booleana: $L0=((B+C).A).C$

Desenhe o circuito lógico.

Monte o circuito da expressão booleana, utilizando os CI74LS08 e CI74LS32.

Crie e complete a tabela verdade do circuito do item (a).

Resumo

A função NÃO providencia na saída um nível lógico que é o inverso do nível lógico da entrada.

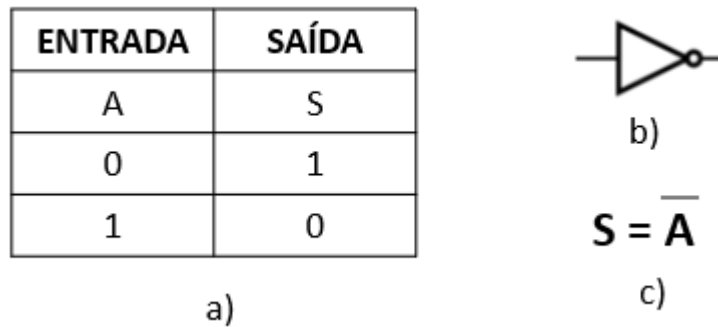


Figura 12 Porta INV a) Tabela Verdade b) Símbolo Lógico c) Expressão Booleana

OBJETIVO

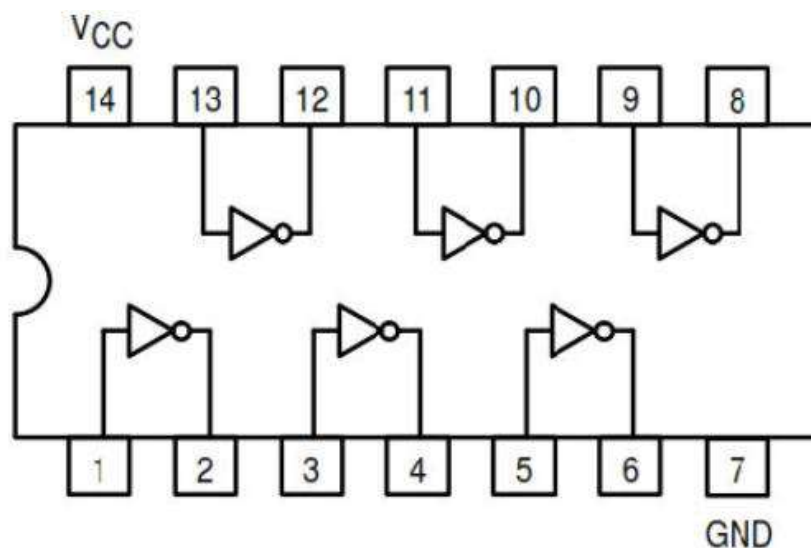
Será examinado a tabela verdade da porta INV e algumas aplicações da porta INV.

EXPERIÊNCIA 1

Material utilizado

1 x CI74LS04

Layout do CI74LS04



c) Monte o circuito da Figura 13.

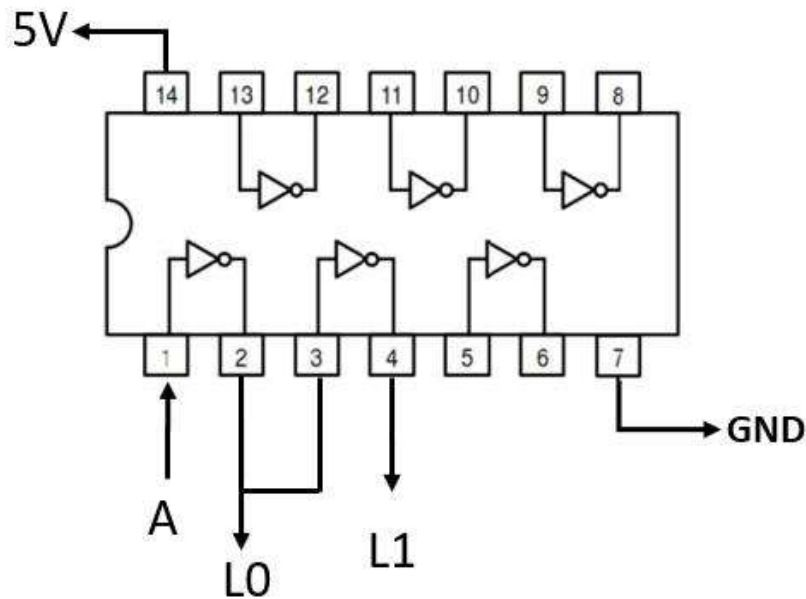


Figura 13 – Circuito com porta INV

d) Complete a tabela.

ENTRADA	SAÍDA	
A	L0	L1
0		
1		

Figura 14 Tabela Verdade da porta INV

e) Demora de Propagação

As portas INV podem ser usadas para introduzir demora de propagação em uma determinada linha. Cada porta INV introduz um atraso típico de 10ns, para TTL padrão.

f) Fan-Out (Capacidade de Cargas)

Um parâmetro importante dos circuitos integrados é a quantidade de outros gates (portas) do mesmo tipo, ou cargas, que a saída de um determinado gate poderá alimentar. Esta característica é chamada de "FAN-OUT" e nos circuitos TTL tem valor típico de 10.

Se o circuito exige que mais de 10 cargas devem ser controladas por uma determinada saída, podemos usar gates INV como BUFFER, de maneira a aumentar este número.

EXPERIÊNCIA 2

Dado o circuito da Figura 15:

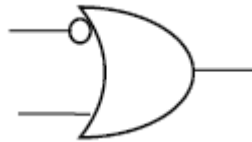


Figura 15 - Circuito Combinacional

a) Identificar as portas utilizadas no circuito, identificação a pinagem a ser utilizada de acordo com o layout dos CIs.

B) Montar o circuito da Figura 14 no módulo (kiT) fazendo todas as interligações.

c) Montar a tabela verdade do circuito.

EXPERIÊNCIA 3

Dado o circuito da Figura 16 :

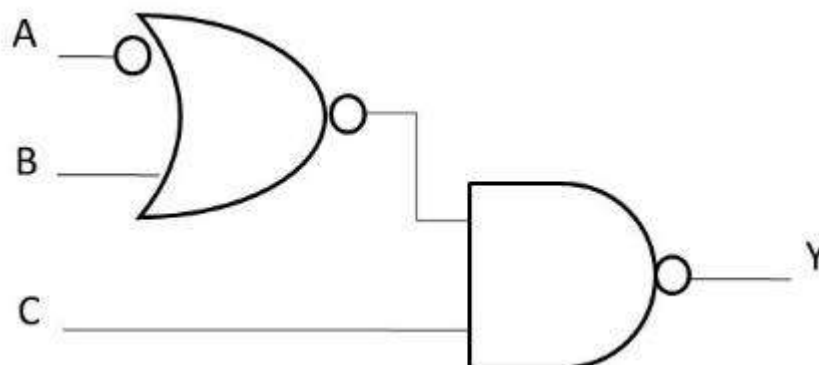


Figura 16 Circuito Combinacional

a) Identificar as portas utilizadas no circuito, identificação a pinagem a ser utilizada de acordo com o layout dos CIs.

b) Montar o circuito da Figura 15 no módulo (kiT) fazendo todas as interligações.

c) Montar a tabela verdade do circuito.

EXPERIÊNCIA 4

Dado o circuito da Figura 17:

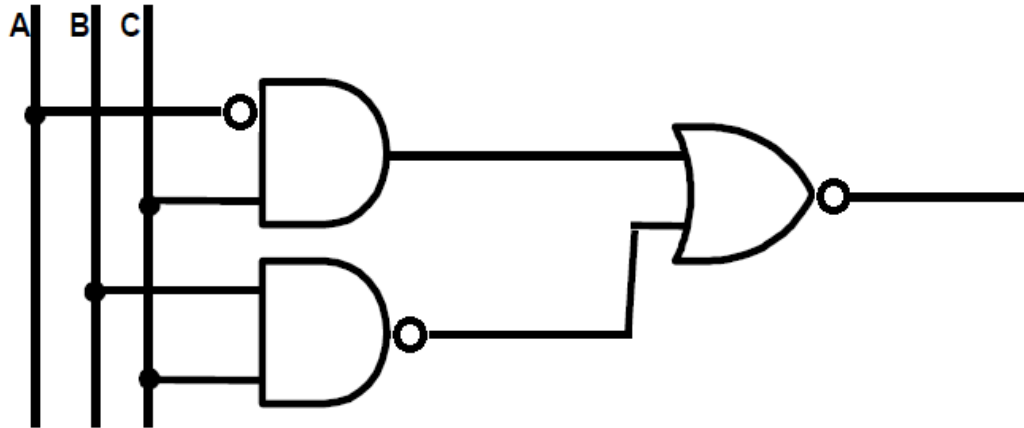


Figura 17 - Circuito Combinacional

- Identificar as portas utilizadas no circuito, identificação a pinagem a ser utilizada de acordo com o layout dos CIs.
- Montar o circuito da Figura 15 no módulo (kiT) fazendo todas as interligações.
- Montar a tabela verdade do circuito.

REFERÊNCIAS

BOYLESTAD, Robert L.. Introdução à Análise de Circuitos. 10ª ed. São Paulo: Pearson.

BOYLESTAD, Robert L.; NASHELSKY, Louis. Dispositivos Eletrônicos e Teoria de Circuitos. 8ª ed. São Paulo: Pearson.

CAPUANO, Francisco Gabriel. Elementos de eletrônica digital. 42. ed. rev. São Paulo: Érica, 2019. 440 p.; PDF. ISBN 978-85-365-3038-3 (e-book)

DACHI, Édison P.; HAUPT, Alexandre G. Eletrônica Digital [livro eletrônico]. São Paulo: Bluccher, 2018. 230p.; PDF. ISBN 978-85-212-1009-2 (e-book)

MALVINO, Albert Paul. Eletrônica - Vol. 1 e 2. 4ª ed. São Paulo: Pearson.

TOCCI, Ronald J., Neal S. Widmer e Gregory L. Moss. Sistemas Digitais: Princípios e Aplicações. Editora Pearson, 2012.

TOCCI, Ronald J.; WIDMER, Neal S; MOSS, Gregory L. Sistemas digitais: princípios e aplicações. 10. ed. São Paulo: Pearson Prentice Hall, 2007. 804 p. ISBN 9788576050957.

USATEGUI, José; MARTIN, Juan C. Eletrônica Digital y Microprogramable [livro eletrônico]. Madrid: Parainfo, 2010. PDF. ISBN: 978-84-9732-505-9 (e-book)

 conhecimentolivre.org/home

 contato@conhecimentolivre.org

 [editoraconhecimentolivre](https://www.instagram.com/editoraconhecimentolivre)

ELETRÔNICA DIGITAL 1

#FÁCILDEMAIS

Antonio Tadeu Pellison



EDITORA CONHECIMENTO LIVRE