Sistemas de Numeração

Sistema Decimal de Numeração

É o sistema comumente utilizado por nós no dia-a-dia, possui dez símbolos aos quais utilizamos diariamente mesmo sem perceber que o fazemos, pois estamos tão acostumados a utilizá-lo que seus mecanismos se tornaram automáticos para nós.

D = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

Contagem:

Ao efetuarmos a contagem, percebemos que a cada base completada, temos que o elemento mais imediatamente a esquerda é adicionado, até que sua própria base se complete e o ciclo se repete.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...,19, 20, 21, ..., 99, 100, 101, ..., 109, 110, ...

Formação dos números:

$$\begin{array}{llll} 3\ 8\ 2\ 7 & & & \\ & 7 * 10^0 = 7 * 1 & = & 7 \\ & 2 * 10^1 = 2 * 10 & = & 20 \\ & 8 * 10^2 = 8 * 100 & = & 800 \\ & 3 * 10^3 = 3 * 1000 & = & 3000 \\ & & & 3827 \end{array}$$

Soma decimal:

$$\begin{array}{llllll} & 1 & 1 & 1 & 1 & \\ \text{Ex 1)} & 3 & 8 & 7 & 1 & \\ + & 2 & 4 & 1 & 5 & \\ \hline & 6 & 2 & 4 & 9 & \\ \hline 1 & 2 & 5 & 3 & 5 & \end{array}$$

$$\begin{array}{llllll} & 1 & 1 & 1 & 1 & \\ & 1 & 1 & 1 & 1 & \\ \text{Ex 2)} & 9 & 8 & 4 & 6 & \\ & 1 & 0 & & & \\ & 8 & 7 & 9 & 8 & \\ + & & & 6 & 4 & \\ \hline & 5 & 6 & 7 & 9 & \\ & 4 & 3 & 3 & 3 & \\ \hline & 5 & 6 & 2 & 8 & \\ \hline 2 & 9 & 9 & 5 & 1 & \end{array}$$

Sistema Binário de Numeração:

O sistema binário é um sistema bastante antigo, porém suas aplicações eram restritas à sistemas matemáticos. Com o aperfeiçoamento da eletrônica, percebeu-se a necessidade de se ter um sistema mais simples, pois o sistema decimal era impraticável, devido a grande dificuldade de se gerar e reconhecer dez sinais diferentes. Assim o sistema binário se tornou ponto chave, já que apenas dois símbolos, que o compõe, tornara-se extremamente simples de representar, tal como "existe sinal elétrico" (representando zero) e "não existe sinal elétrico" (representando um).

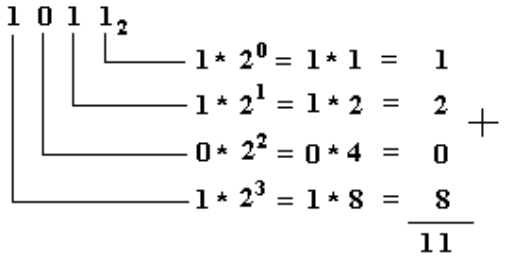
$$B = \{ 0, 1\}$$

Contagem:

O método de contagem é idêntico em qualquer sistema, o que deve-se observar apenas é que o número de elementos que compõe a cada base é diferenciado.

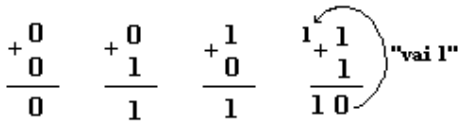
0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111...

Formação dos números:



Soma binária:

A soma de números binários é bastante simples de ser efetuada, basta seguir as seguintes regras:



Então podemos somar sem problema algum, qualquer número binário.

Ex 1)

```
  1   1
1 1 1 1 1 1
    1 0 1 1 1
    0 0 0 0
    1 1 1 0 1
+               0
    1 0 1 1 1
-----------------
1 0 0 1 0 1 1
```

Ex 2)

```
    1
  1 1 1 1 1
1 1 1 1 1 1
    1 1 0 1 1

    1 0 1 1 1
+   0     0 0
    1 0 0 0 1

    1 1 1 0 1
-------------------
1 1 0 0 0 0 0
```

Sistema Hexadecimal de Numeração:

O sistema hexadecimal de numeração compreende 16 símbolos e sua utilização se tornou necessária devido a termos neste sistema um múltiplo do sistema binário, ou seja, cada base hexadecimal compreende 4 bases binárias. Assim, quando precisamos representar um número binário de muitos dígitos, ou mesmo para facilitar a inserção de um código de programação de máquina, utilizamos o código hexadecimal.

H = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Contagem:

Segue o método tradicional.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, ...18, 19, 1A, 1B, ..., 1F, 20...
29, 2A, ..., 2F, 30, ...99, 9A, 9B, 9C, 9D, 9E, 9F, A0, A1, A2, ..., A9, AA, AB, ...AF, B0...

Formação dos números:

$C\ 1\ B\ 4_{16}$

$4 * 16^0 = 4 * 1 = 4$

$B * 16^1 = 11 * 16 = 176$

$1 * 16^2 = 1 * 256 = 256$ +

$C * 16^3 = 12 * 4096 = 49152$

49588

Soma Hexadecimal:

A soma hexadecimal tem uma complexidade maior, já que não temos o hábito de utilizar tal sistema no dia-a-dia, sendo assim, devemos ter em mente a tabelinha abaixo, a fim de podermos efetuar as somas de uma maneira mais simples.

A = 10	B = 11	C = 12	D = 13	E = 14	F = 15

Ex 1)

```
   1   1
     B 2 C 9
   + A 1 2 3
   ---------
     1 3 5 2
   ---------
   1 6 7 3 E
```

Ex 2)

```
       1
     1 1 1 1
     1 1 1 1
   B A B A C A
     D D E
     F A C A
   + C 7 A 4
   ---------
     C A F E
         9 2
       B 3
   ---------
   B C 8 1 4 5
```

Conversão entre bases:

Conversão Decimal – Binário:

É uma conversão bastante simples de ser efetuada, basta efetuar a divisão inteira do número decimal, sucessivas vezes pelo número 2, deixando o resto aparente. O número binário será formado pelo último quociente (que será sempre "1"), seguido pelo último resto, penúltimo resto, ante-penúltimo resto, até o primeiro resto.

Ex 1) $21_{10} = ?_2$

21 | 2				
1	10 | 2			
	0	5 | 2		
		1	2 | 2	
			0	1

$\therefore 21_{10} = 10101_2$

Ex 2) $75_{10} = ?_2$

75 | 2					
1	37 | 2				
	1	18 | 2			
		0	9 | 2		
			1	4 | 2	
				0	2 | 2
					0 1

$\therefore 75_{10} = 1001011_2$

Conversão Binário – Decimal

Normalmente as conversões de uma base qualquer para decimal são efetuadas pela fórmula geral de conversão de bases para decimal.

$$E_D = A_N X^N + A_{(N-1)} X^{(N-1)} + A_{(N-2)} X^{(N-2)} + \ldots + A_3 X^3 + A_2 X^2 + A_1 X + A_0$$

Onde:

E_D = Equivalente decimal
A = Dígito binário
N = Índice do dígito binário (deve-se começar sempre pelo zero)
X = Base em questão

Ex 1) $1101_2 = ?_{10}$

$E_D = A_3X^3 + A_2X^2 + A_1X + A_0$

$E_D = 1.2^3 + 1.2^2 + 0.2 + 1$

$E_D = 1.8 + 1.4 + 0.2 + 1$

$E_D = 8 + 4 + 0 + 1$

$E_D = 13$

$\therefore 1101_2 = 13_{10}$

Ex 2) $10101_2 = ?_{10}$

$E_D = A_4X^4 + A_3X^3 + A_2X^2 + A_1X + A_0$

$E_D = 1.2^4 + 0.2^3 + 1.2^2 + 0.2 + 1$

$E_D = 1.16 + 0.8 + 1.4 + 0.2 + 1$

$E_D = 16 + 0 + 4 + 0 + 1$

$E_D = 21$

$\therefore 10101_2 = 21_{10}$

Conversão Decimal – Hexadecimal

Aqui também se emprega o método de efetuar divisões sucessivas, porém deve-se, agora utilizar o número 16 como divisor.

Ex 1) $42_{10} = ?_{16}$

42	16
A ← ~~10~~	2

$\therefore 42_{10} = 2A_{16}$

Ex 2) $2292_{10} = ?_{16}$

2292	16
4	143

143	16
F ← ~~15~~	8

$\therefore 2292_{10} = 8F4_{16}$

Conversão Hexadecimal - Decimal

Deve-se utilizar a fórmula geral de conversão de bases para decimal.Onde A base em questão (X), agora deve ser o número 16.

$$E_D = A_NX^N + A_{(N-1)}X^{(N-1)} + A_{(N-2)}X^{(N-2)} + \ldots + A_3X^3 + A_2X^2 + A_1X + A_0$$

Ex 1) $58C_{16} = ?_{10}$

$E_D = A_2 X^2 + A_1 X + A_0$

$E_D = 5.16^2 + 8.16 + C$

$E_D = 5.256 + 8.16 + 12$

$E_D = 1280 + 128 + 12$

$E_D = 1420$

$\therefore 58C_{16} = 1420_{10}$

Ex 2) $1B3AF_{16} = ?_{10}$

$E_D = A_4 X^4 + A_3 X^3 + A_2 X^2 + A_1 X + A_0$

$E_D = 1.16^4 + B.16^3 + 3.16^2 + A.16 + F$

$E_D = 1.65536 + 11.4096 + 3.256 + 10.16 + 15$

$E_D = 65536 + 45056 + 768 + 160 + 15$

$E_D = 111535$

$\therefore 1B3AF_{16} = 111535_{10}$

Conversão Hexadecimal – Binário

O sistema de conversão hexadecimal – binário é o método mais simples de todos. O método se baseia na premissa de que cada base hexadecimal equivale a 4 bases binárias, ou seja, temos que $16^1 = 2^4$. Assim fazemos a conversão de cada dígito hexadecimal em seu equivalente de 4 bits.

Ex 1) $C41_{16} = ?_2$

C	4	1
1100	0100	0001

$\therefore C41_{16} = 110001000001_2$

Ex 2) $1D20E_{16} = ?_2$

1	D	2	0	E
0001	1101	0010	0000	1110

→ **Obs: - Zeros a esquerda devem ser suprimidos!**

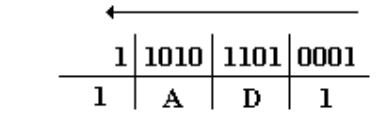
$\therefore 1D20E_{16} = 11101001000001110_2$

Conversão Binário – Hexadecimal

A conversão binário – hexadecimal também é bastante simples, pois se faz da maneira oposta à conversão hexadecimal – binário porém, deve-se tomar o cuidado de separar os dígitos binários em grupos de 4 dígitos a partir da direita para a esquerda e não o contrário. Assim, cada grupo de 4 dígitos binários deverá ser convertido em seu equivalente hexadecimal de 1 dígito.

Ex 1) $1101011010001_2 = ?_{16}$

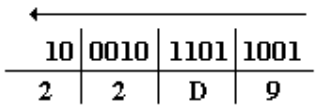
Obs: - Sempre da direita para a esquerda!



1	1010	1101	0001
1	A	D	1

$\therefore 1101011010001_2 = 1AD1_{16}$

Ex 2) $10001011011001_2 = ?_{16}$



10	0010	1101	1001
2	2	D	9

$\therefore 10001011011001_2 = 22D9_{16}$

Operações Lógicas

As operações lógicas estão sempre relacionadas com o sistema de numeração binário e seu resultado sempre será binário, ou seja, **Verdadeiro ou Falso**, **Sim ou Não**, **0 ou 1**.

Para melhor entendimento, devemos considerar os conceitos abaixo que complementam a teoria das operações lógicas.

Portas Lógicas : São dispositivos eletrônicos que realizam as operações lógicas.

Tabelas Verdade : É uma tabela que contém todos os valores possíveis de uma operação ou de uma expressão lógica.

Expressão Lógica : É a sentença matemática que representa uma determinada tabela verdade ou circuito lógico.

Circuito Lógico : É a representação gráfica de uma determinada tabela verdade ou expressão lógica através de portas lógicas.

As operações lógicas se dividem em 3 grupos :

- Operações Lógicas **Fundamentais**
- Operações Lógicas **Complementares** (ou Combinadas)
- Operações Lógicas **Exclusivas**

Operações Lógicas Fundamentais

As operações lógicas fundamentais são aquelas operações mais simples, das quais surgiram as outras operações. São em número de 3, a saber: **not** (“não lógico”), **and** (“e lógico”) e **or** (“ou lógico”).

Operação Lógica NOT (“não lógico”)

A operação lógica not sempre atua sobre uma única entrada, estabelecendo que **“a saída sempre será o inverso da entrada”**.

Símbolo: " ¬ "

Expressão: $S = \overline{A}$

Lê-se: “ – S é igual a A barra” ou
“ – S é igual a A barrado” ou
“ – S é igual ao inverso de A” ou
“ – S é igual a not A” ou
“ – S é igual a não A”

Porta Lógica:

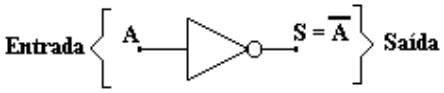


Tabela Verdade:

Entrada	Saída
A	$S = \overline{A}$
F	V
V	F

Ou

Entrada	Saída
A	$S = \overline{A}$
0	1
1	0

Operação Lógica AND (“e lógico”)

A operação lógica and atua sobre duas ou mais entradas, estabelecendo que “**a saída somente será verdadeira se todas as entradas forem verdadeiras**”.

Símbolo: " · "

Expressão: $S = A.B$

Lê-se: “ – S é igual a A and B” ou
“ – S é igual a and de A e B” ou
“ – S é igual a A e B”

Porta Lógica:

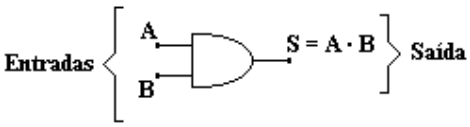


Tabela Verdade:

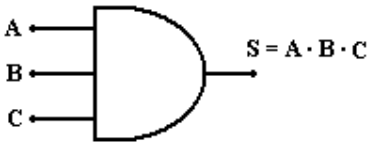
Entradas		Saída
A	B	S = A.B
F	F	F
F	V	F
V	F	F
V	V	V

Ou

Entradas		Saída
A	B	S = A.B
0	0	0
0	1	0
1	0	0
1	1	1

Exercício: Complete a tabela verdade para uma porta and de 3 entradas.

Entradas			Saída
A	B	C	S = A · B · C
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Operação Lógica OR (“ou lógico”)

A operação lógica or atua sobre duas ou mais entradas, estabelecendo que “**a saída somente será falsa se todas as suas entradas forem falsas”**.
Símbolo: " + "

Expressão: **S = A+B**

Lê-se: “-S é igual a A or B” ou
“-S é igual a or de A e B” ou
“-S é igual a A ou B”

Porta Lógica:

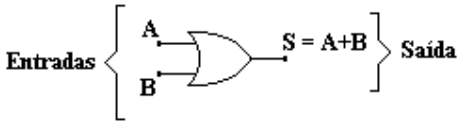


Tabela Verdade:

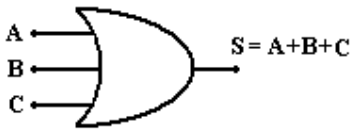
Entradas		Saída
A	B	S = A+B
F	F	F
F	V	V
V	F	V
V	V	V

Ou

Entradas		Saída
A	B	S = A+B
0	0	0
0	1	1
1	0	1
1	1	1

Exercício: Complete a tabela verdade para uma porta or de 3 entradas.

Entradas			Saída
A	B	C	S = A+B+C
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Operações Lógicas Complementares

As operações lógicas complementares também são conhecidas como operações auxiliares ou combinadas. São concebidas a partir da união da operação lógica not com as operações lógicas and e or, dando origem às operações complementares nand (not and) e nor (not or) respectivamente.

Operação Lógica NAND ("não e lógico")

A operação lógica nand atua sobre duas ou mais entradas, estabelecendo que **"que a saída somente será falsa se todas as suas entradas forem verdadeiras".**

Símbolo: " ⎯. "

Expressão: $S = \overline{A \cdot B}$

Lê-se: "-S é igual a A nand B" ou

"-S é igual a nand de A e B" ou
"-S é igual a A não e B"

Porta Lógica:

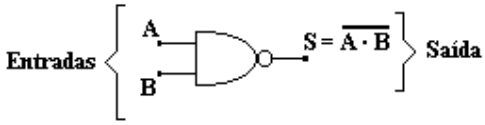


Tabela Verdade:

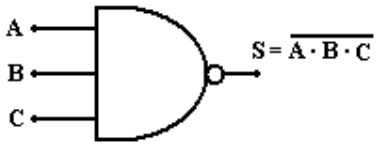
Entradas		Saída
A	B	S = $\overline{A.B}$
F	F	V
F	V	V
V	F	V
V	V	F

Ou

Entradas		Saída
A	B	S = $\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

Exercício: Complete a tabela verdade para uma porta nand de 3 entradas.

Entradas			Saída
A	B	C	S = $\overline{A \cdot B \cdot C}$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Operação Lógica NOR ("não ou lógico")

A operação lógica nor atua sobre duas ou mais entradas, estabelecendo que **"a saída somente será verdadeira se todas as suas entradas forem falsas"**.

Símbolo: " $\overline{\quad + \quad}$ "

Expressão: $S = \overline{A+B}$

Lê-se: "-S é igual a A nor B" ou

"-S é igual a nor de A e B" ou
"-S é igual a A não ou B"

Porta Lógica:

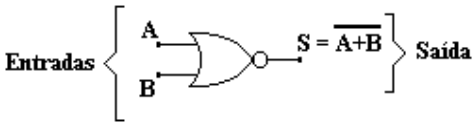


Tabela Verdade:

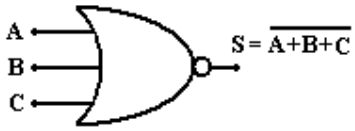
Entradas		Saída
A	B	S = $\overline{A+B}$
F	F	V
F	V	F
V	F	F
V	V	F

Ou

Entradas		Saída
A	B	S = $\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Exercício: Complete a tabela verdade para uma porta nor de 3 entradas.

Entradas			Saída
A	B	C	S = $\overline{A+B+C}$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Operações Lógicas Exclusivas

Pelo motivo de "aparecerem" sempre em expressões lógica, foram criadas as operações lógicas exclusivas, sendo em número de duas, a saber: Exclusive Or (xor) e Not Exclusive Or (nxor).

Operação Lógica XOR ("ou exclusivo")

A operação lógica xor estabelece que **"a saída somente será verdadeira se todas as suas entradas forem diferentes"**.

Símbolo: " ⊕ "

Expressão: $S = A \oplus B$

Lê-se: "-S é igual a A xor B" ou
"-S é igual a xor de A e B" ou
"-S é igual a A exclusive or B" ou
"-S é igual a A ou exclusivo B"

Porta Lógica:

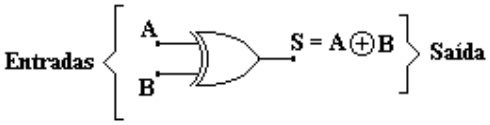


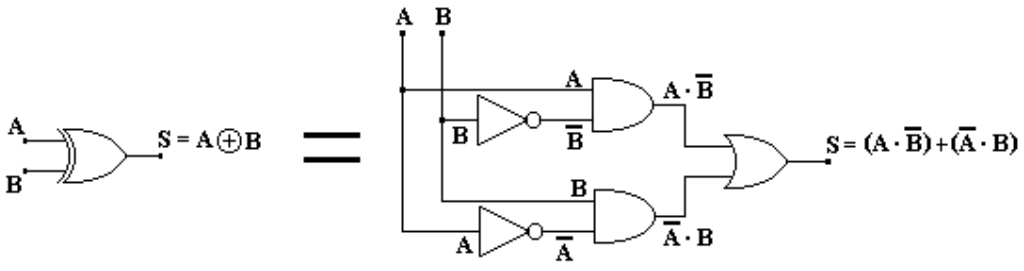
Tabela Verdade:

Entradas		Saída
A	B	S=A⊕B
F	F	F
F	V	V
V	F	V
V	V	F

Ou

Entradas		Saída
A	B	S=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Circuito Lógico Equivalente:



Operação Lógica NXOR ("não ou exclusivo")

A operação lógica nxor estabelece que **"a saída somente será verdadeira se todas as suas entradas forem iguais".** Por esta característica, esta operação também é conhecida com operação coincidência.

Símbolo: " $\odot$ "

Expressão: $S = A \odot B$

Lê-se: "-S é igual a A nxor B" ou
"-S é igual a nxor de A e B" ou
"-S é igual a A exclusive or B" ou
"-S é igual a A ou exclusivo B" ou
"-S é igual a A coincidência B"

Porta Lógica:

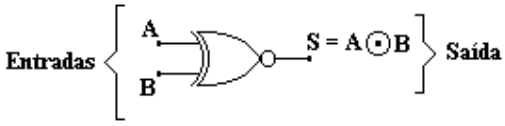


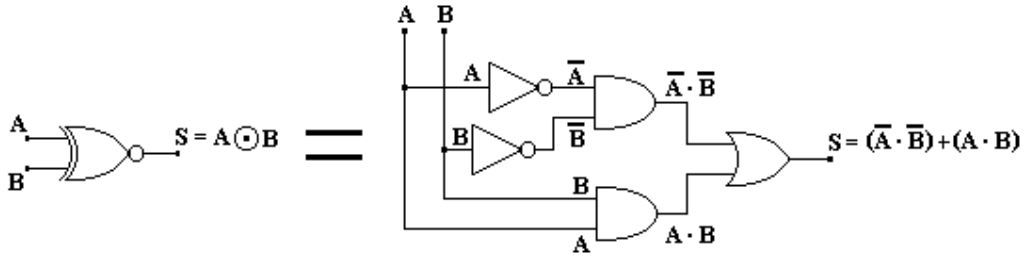
Tabela Verdade:

Entradas		Saída
A	B	S=A⊙B
F	F	V
F	V	F
V	F	F
V	V	V

Ou

Entradas		Saída
A	B	S=A⊙B
0	0	1
0	1	0
1	0	0
1	1	1

Circuito Lógico Equivalente:



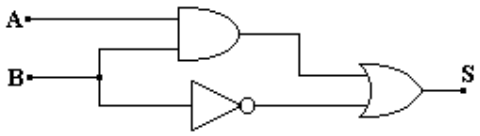
Circuitos Lógicos:

Com o decorrer do tempo iremos notar que os circuitos lógicos, tabelas verdade e expressões lógicas sempre andam junto e de qualquer um, podemos chegar aos outros. Para tanto existem técnicas das quais faremos uso para atingir nosso intuito.

Então temos três caminhos a seguir, ou seja, a partir do circuito lógico, a partir da expressão lógica e por fim, a partir da tabela verdade. Começaremos pelo mais fácil, ou seja, partindo de um circuito já pronto.

Através de uma associação de portas lógicas foram montados os circuitos lógicos abaixo. Com os conhecimentos adquiridos até agora, desenvolva a expressão lógica e a tabela verdade para cada um.

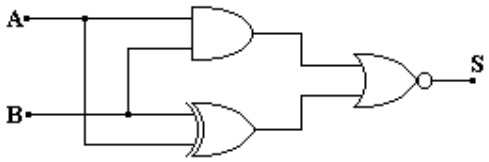
1)



S = _____

Entradas		Saída
A	B	S

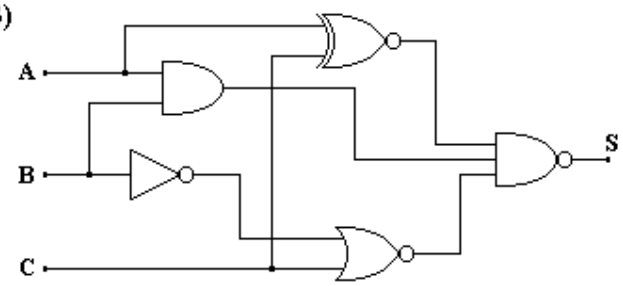
2)



S = _____

Entradas		Saída
A	B	S

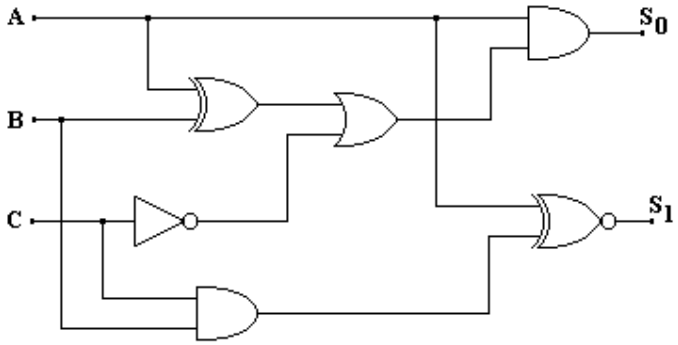
3)



S = _____

Entradas			Saída
A	B	C	S

4)



Entradas			Saídas	
A	B	C	S_0	S_1

S_0 = _____

S_1 = _____