

## Definição do Exercício T15

30/Março/2017

Pode ser realizado em até quartetos  
Deve ser iniciado na aula do dia 30 de março

Quem não estiver em aula no momento do início do exercício deverá realizar INDIVIDUAL  
Entregar uma versão por grupo no moodle

1) Simplifique usando regras as seguintes expressões booleanas:

$$a) S1 = \overline{A}.\overline{B}.C.D + A.\overline{B}.C.\overline{D} + \overline{A}.\overline{B} + A.C + A.D$$

$$b) S2 = (\overline{A+B}) + \overline{A}.\overline{B} + \overline{A}.B.C$$

$$c) S3 = A.(\overline{A.B} + \overline{A} + \overline{B} + A.B)$$

2) Gere a tabela verdade das versões simplificadas de S1, S2 e S3

3) Monte os diagramas de tempo para as saídas S1, S2 e S3, considerando esta amostragem

A= 01011100110100...

B= 11111000100110...

C= 00010010101011...

D= 11001100101110...

(ignore entradas que não sejam usadas, por exemplo, para S3 considere apenas A e B)

4) Uma transmissão de dados antigamente usava apenas 7 bits para representar caracteres. Era o ASCII puro (na tabela ASCII todas as letras e símbolos estão dentro de sete bits). Em um tempo onde muitos erros de transmissão podiam ocorrer, o oitavo bit, mais significativo, era usado para indicar paridade. Se a quantidade de bits em UM nos 7 bits de dado fosse PAR, então este oitavo bit fica em 0, indicando ser PAR. Caso contrário, este BIT ficaria em 1.

Veja alguns exemplos: letras 'A' e 'C' (65 e 67 da tabela ASCII).

- 65 em sete bits fica: 1000001
- 67 em sete bits fica: 1000011

B7	B6	B5	B4	B3	B2	B1	B0	
0	1	0	0	0	0	0	1	65 ('A')
1	1	0	0	0	0	1	1	67 ('C')

Observe o bit 7 sendo 0 para o 65 (65 tem dois bits em 1, logo é PAR) e sendo 1 para o 67 (que tem 3 bits em 1, sendo IMPAR).

Faça um circuito combinacional que receba estes 8 bits e gere uma saída chamada de ERRO. O circuito deve verificar se o bit 7 está correto e acender (colocar em 1) a saída ERRO se houve erro de paridade. Monte no logisim.

5) Implemente a questão B do T14 usando decodificadores.