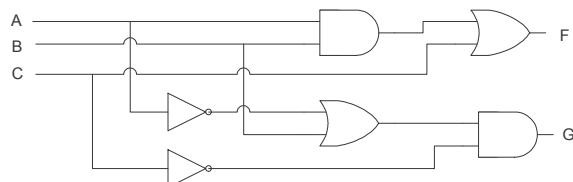


### 1ª Lista de Exercícios

#### Gararito

1. Converta  $(109,125)_{10}$  para:
  - a. base 2:  $(1101101,001)_2$
  - b. base 8;  $(155,1)_8$
  - c. base 16.  $(6D,2)_{16}$
  
2. Represente  $(+13)_{10}$ ,  $(-13)_{10}$ ,  $(0)_{10}$  usando o menor número de bits possível, em:
  - a. Sinal magnitude;  $01101; 11101; 00000$
  - b. Complemento a 1;  $01101; 10010; 11111$  ou  $00000$
  - c. Complemento a 2;  $01101; 10011; 00000$
  - d. Excesso de 16.  $11101; 00011; 10000$
  
3. Realize as seguintes operações em complemento a 2 de 5 bits. Diga se o resultado está correto ou não.
  - a.  $-110_2 - 1010_2 = 11010 + 10110 = 10000 = (-16)_{10}$ ; correto
  - b.  $-110_2 + 1010_2 = 11010 + 01010 = 00100 = (4)_{10}$ ; correto
  - c.  $110_2 - 1010_2 = 00110 + 10110 = 11100 = (-4)_{10}$ ; correto
  - d.  $110_2 + 1010_2 = 00110 + 01010 = 10000 = (-16)_{10}$ ; incorreto, pois bit de sinal passou de ++ para -
  
4. A representação em ponto flutuante curta da IBM usa base 16, um bit de sinal, um expoente de sete bits em excesso de 64 e uma fração normalizada de 24 bits na base 16.
  - a. Qual número é representado pelo seguinte padrão de bits?  
 $1\ 0111111\ 01110000\ 00101000\ 00000001$   
 Mostre a resposta em decimal.  
 $-(0,70\ 28\ 01)_{16} \times 16^{-1} = 459520,0625$
  - b. Represente os seguintes números nesta notação:
 

i. $1,0 = (0,1)_{16} \times 16^1$	0	1000001	00010000	00000000	00000000
ii. $0,5 = (0,8)_{16} \times 16^0$	0	1000000	10000000	00000000	00000000
iii. $1/64 = (0,4)_{16} \times 16^{-1}$	0	0111111	01000000	00000000	00000000
iv. $0,0$	0	0000000	00000000	00000000	00000000
  
5. Dado o circuito lógico abaixo, construa uma tabela-verdade que descreva seu comportamento.



A	B	C	F	G
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	0

6. Projete uma rede mínima, com soma de produtos, que implemente a função “maioria” de cinco entradas, ou seja, a saída será 1 sempre que três ou mais entradas forem iguais a 1. Faça uma versão do sistema utilizando um MUX 16 para 1.

A	B	C	D	E	S
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

Soma de produtos (por Mapa de Karnaugh):

$$S = ABD + ABC + BCD + ACD + ECD + EAB + EBC + EBD + EAC + EAD$$

E=0					E=1				
AB\CD	00	01	11	10	AB\CD	00	01	11	10
00	0	0	0	0	00	0	0	1	0
01	0	0	1	0	01	0	1	1	1
11	0	1	1	1	11	1	1	1	1
10	0	0	1	0	10	0	1	1	1

Mux: Utilizar 4 entradas nas linhas de endereço do MUX, a 5ª entrada nas linhas de entrada (as linhas de entrada recebem “0”, “1”, E ou E’).

7. Projete uma rede NAND de dois níveis mínima que calcule o produto de dois números inteiros, com valores de entrada de 0 a 3.

Entradas: A1A0, B1B0 (ambas podem ir de 0 (00)<sub>2</sub> a 3(11)<sub>2</sub>), saída: S3S2S1S0 (pode ir de 0 (0<sub>2</sub>) a 9 (1001<sub>2</sub>))

A1	A0	B1	B0	S3	S2	S1	S0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Karnaugh:

S3					S2				
$A_1A_0 \setminus B_1B_0$	00	01	11	10	$A_1A_0 \setminus B_1B_0$	00	01	11	10
00	0	0	0	0	00	0	0	0	0
01	0	0	0	0	01	0	0	0	0
11	0	0	1	0	11	0	0	0	1
10	0	0	0	0	10	0	0	1	1
$S3 = A_1A_0B_1B_0$					$S2 = A_1B_1B_0' + A_1A_0'B_1$				
S1					S0				
$A_1A_0 \setminus B_1B_0$	00	01	11	10	$A_1A_0 \setminus B_1B_0$	00	01	11	10
00	0	0	0	0	00	0	0	0	0
01	0	0	1	1	01	0	1	1	0
11	0	1	0	1	11	0	1	1	0
10	0	1	1	0	10	0	0	0	0
$S1 = A_1B_1'B_0 + A_0B_1B_0' + A_1A_0'B_0 + A_1'A_0B_1$					$S0 = A_0B_0$				

NAND de 2 níveis: a partir da soma de produtos obtida acima, basta substituir OR's e AND's por NAND's. Matematicamente, barre duas vezes as expressões e utilize DeMorgan.

8. Usando uma PLA, implemente um sistema que produza o quadrado de um dígito BCD. A saída deve estar na representação binária.

Resposta semelhante à anterior: 4 entradas ( $A_3A_2A_1A_0$ ), que possuem valores de 0000 a 1001, 6 linhas de saída (valores possíveis de 000000 a 101001). A PLA possui as duas matrizes, de AND's e de OR's, programáveis. Basta então utilizar as somas de produtos obtidas a partir dos mapas de Karnaugh.

9. Projetar um conversor do código A para o código B:

$A_3$	$A_2$	$A_1$	$A_0$	$B_3$	$B_2$	$B_1$	$B_0$
0	1	1	1	1	1	0	0
1	0	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	0	0	1	1	0	0	1
1	1	0	1	1	0	0	0
1	1	1	1	0	1	1	1
0	1	0	1	0	1	1	0
0	0	0	0	0	1	0	1
0	0	1	0	0	0	1	1
3, 8, A, C, E				X	X	X	X

- Implementar com uma PROM 4x4.
  - Implementar com dispositivo PAL 4x4 com 4 produtos por saída.
- a. A PROM tem uma rede ORs programáveis e uma rede de ANDs fixos, nesta rede aparecem todos os min-termos: de  $A_3A_2A_1A_0$  a  $A_3A_2A_1A_0$ . Portanto, não é necessário fazer minimização, basta pegar os 1's das saídas tabela-verdade e escolher os ANDs correspondentes da PROM.

- b. Neste caso, é necessária minimização, pois a PAL tem 4 ORs de saída, cada qual pré-conectado a quatro ANDs. Ou seja, cada saída é um OR de 4 ANDs, cada um dos ANDs é programável.

Karnaugh:

<b>B3</b>					<b>B2</b>				
$A_3A_2 \setminus A_1A_0$	00	01	11	10	$A_3A_2 \setminus A_1A_0$	00	01	11	10
00	0	1	X	0	00	1	0	X	0
01	X	0	1	1	01	X	1	1	0
11	X	1	0	X	11	X	0	1	X
10	X	1	X	X	10	X	0	X	X
$B3 = A2'A0 + A3A1' + A3'A2A1$					$B2 = A1'A0' + A3'A2A0 + A1A0$				
<b>B1</b>					<b>B0</b>				
$A_3A_2 \setminus A_1A_0$	00	01	11	10	$A_3A_2 \setminus A_1A_0$	00	01	11	10
00	0	0	X	1	00	1	1	X	1
01	X	1	0	1	01	X	0	0	0
11	X	0	1	X	11	X	0	1	X
10	X	1	X	X	10	X	1	X	X
$B1 = A3'A2A1' + A3A2' + A3A2 + A1A0'$					$B0 = A2' + A3A1$				

Como resultado da minimização, não houve nenhuma soma com mais de 4 produtos, então a PAL proposta é suficiente. Basta programar os ANDs com os produtos que devem entrar em cada um dos 4 ORs para gerar as 4 saídas.