



Lista de Exercícios 05 - Projetos de Circuitos Combinacionais

1. Projete um circuito para controlar o sistema de intercomunicação do prédio da Reitoria da UEM (Universidade Estadual de Maringá). O sistema deve obedecer a uma ordem de prioridades:
 - o 1º Reitor
 - o 2º Vice-Reitor
 - o 3º Assessor de Gabinete
 - o 4º Secretária

No caso de ocorrerem duas ou mais chamadas simultaneamente, somente uma chamada será atendida, a de maior prioridade.

2. Projete um circuito combinacional que ativa o alarme do carro sempre que os assentos do motorista e/ou do passageiro estão ocupados, mas o cinto de segurança não está colocado quando o carro é ligado. Os sinais DRIV e PASS são ativos em ALTO e indicam a presença do motorista e do passageiro, respectivamente. Estes sinais são fornecidos por chaves ativadas por pressão colocadas nos assentos. O sinal IGN é ativo em ALTO quando a chave de ignição está ligada. O sinal BELTD' é ativo em BAIXO e indica que o cinto de segurança do motorista não está colocado. O sinal BELTP' é o sinal referente ao cinto do passageiro. O alarme será ativado (BAIXO) sempre que o carro for ligado e um dos bancos dianteiros estiver ocupado e seu cinto não estiver colocado.
3. Projete um circuito comparador de magnitude para comparar dois dados binários. Os dados têm tamanho de 2 bits cada. Você deve comparar os bits do dado A (a_1a_0) com os bits do dado B (b_1b_0). O circuito deve indicar nas saídas se $A=B$, $A>B$ e $A<B$. Para cada situação verdadeira a respectiva saída deve indicar "1", caso contrário deve indicar "0".

4. Para revitalizar o Romi-Isetta, sua equipe deve desenvolver o projeto a seguir para atender itens de segurança para o carro. O projeto é descrito a seguir:

O carro tem um assento ejetável. Para algumas combinações de situações de pânico o circuito controlador aciona a ejeção do banco. As situações de pânico são informadas ao circuito por meio de sensores que indicam: inundação do veículo (A), fogo (B), colisão (C), falha dos freios (D), botão de sequestro (E) acionado pelo motorista. Essas variáveis em nível lógico 1 indicam que o respectivo sensor foi acionado. Qualquer uma das seguintes combinações de sensores ativados faz o circuito acionar a ejeção:

- o falha dos freios;
- o colisão e botão de sequestro;
- o colisão e inundação;
- o colisão e fogo;
- o inundação e botão de sequestro.

Projete o circuito de controle de acionamento da ejeção usando os nomes das variáveis do texto do exercício.

5. Projete um decodificador para, a partir de um código binário, escrever a sequência mostrada a seguir em um display de 7 segmentos (Figura 1).

caso	000	001	010	011	100	101	110	111
caracter								

Figura 1 - Caracteres do código

6. Projete um decodificador para efetuar a conversão do código BCD para decimal. O circuito deve ativar (nível 1) a única saída correspondente a cada combinação binária de entrada conforme mostrado na Figura 2.

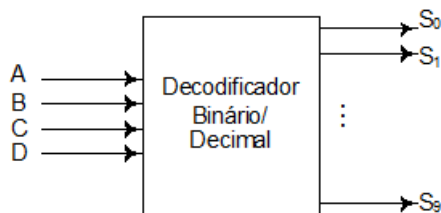
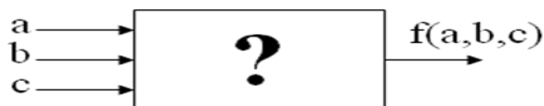


Figura 2 – Decodificador de BCD para decimal

7. Projete um decodificador 3 X 8 ativo em nível alto usando portas lógicas básicas.
8. A partir de 2 circuitos multiplexadores de 4 X 1 e 1 circuito multiplexador 2 X 1 elabore um sistema multiplexador de 8 X 1.
9. Projete um circuito demultiplexador de 1 X 4.
10. Projete um demultiplexador 1 X 8 usando apenas demultiplexadores 1 X 2.
11. Suponha que deseja-se comparar duas palavras de quatro bits - $a = a_1a_0$; $b = b_1b_0$ - no caso onde $a_n \neq b_n$, para $n = 0,1$, isto é, nenhum dos bits, nas mesmas posições, são iguais. Projete um circuito lógico que tenha uma saída $G=1$ quando essa condição for satisfeita.
12. Considere as duas palavras de quatro bits - $a = a_1a_0$; $b = b_1b_0$. Projete um componente lógico que produza uma saída $h=1$ quando as duas condições $a_1 = b_1$ e $a_0 \neq b_0$ são simultaneamente verdadeiras.
13. Considere as duas palavras de quatro bits - $a = a_1a_0$; $b = b_1b_0$. Projete um componente lógico que produza uma saída $G=1$ quando as duas condições $\bar{a}_1 = b_0$ e $a_0 = b_0 \oplus b_1$ são simultaneamente verdadeiras.
14. Projete um circuito somador de 1 bit, considerando as entradas A, B e C_{in} , e as saídas S e C_{out} .
15. Projete um circuito subtrator de 1 bit, considerando as entradas A, B e C_{in} , e as saídas S e C_{out} .
16. Implemente um código em VHDL para o circuito representado pela Tabela Verdade da Figura 3.



a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figura 3



17. Implemente um código em VHDL para o circuito representado pela Tabela Verdade da Figura 4.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Figura 4

18. Implemente um código em VHDL para o circuito representado pela Figura 5.

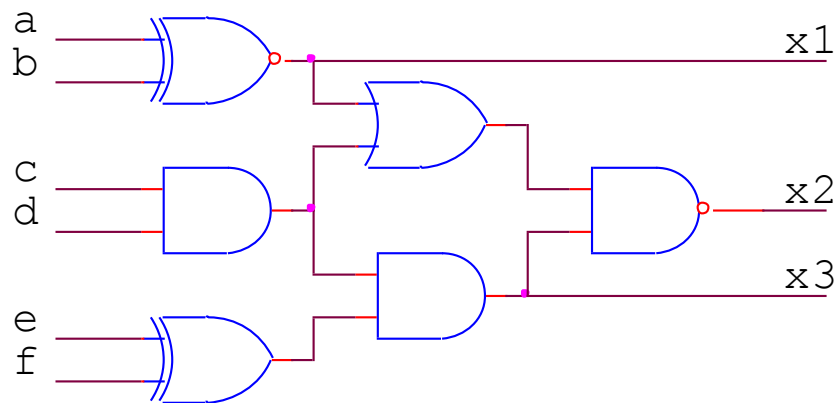


Figura 5

19. A partir da Tabela Verdade (Figura 6), obtenha a expressão e projete em VHDL.

A	B	C	X	Y	Z
0	0	0	1	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	0	0

Figura 6

20. Implemente um código em VHDL para o circuito representado pela Tabela Verdade da Figura 7.



a(1)	a(0)	C
0	0	1
0	1	0
1	0	1
1	1	1

Figura 7

21. Implemente um código em VHDL para o circuito representado pela Tabela Verdade da Figura 8.

X3	X2	X1	X0	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0



1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Figura 8

22. Projete um *buffer* de saída de três estados (*buffer tri-state*) usado no controle de barramentos de dados. O circuito que controla a transferência de dados em um barramento é denominado *buffer tri-state* (*driver* de barramento). Quando habilitado (entrada de *Enable* nível alto), o *buffer tri-state* passa o sinal da entrada A para a saída S. Quando desabilitado (entrada de *Enable* nível baixo), ele bloqueia a entrada A e a sua saída igual a Z é vista pelo barramento como um estado de alta impedância. A Figura 9 mostra um circuito *buffer tri-state* e a sua tabela Verdade.

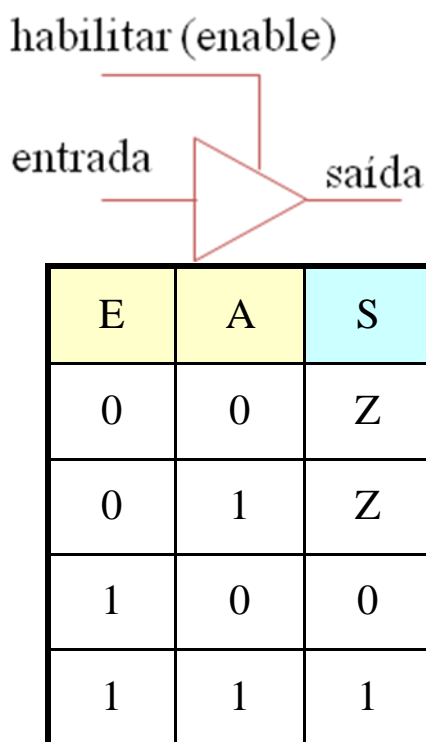


Figura 9

23. Com base no exercício 23, projete um *buffer* de saída de três estados (*buffer tri-state*) para controlar um barramento de dados de 8 bits.

24. Projete um somador paralelo de 4 bits (Figura 10).

NOTA: Além da observação importante no início desta lista, outras possíveis soluções deste exercício deve ser usando os comandos FOR LOOP e WHILE LOOP.

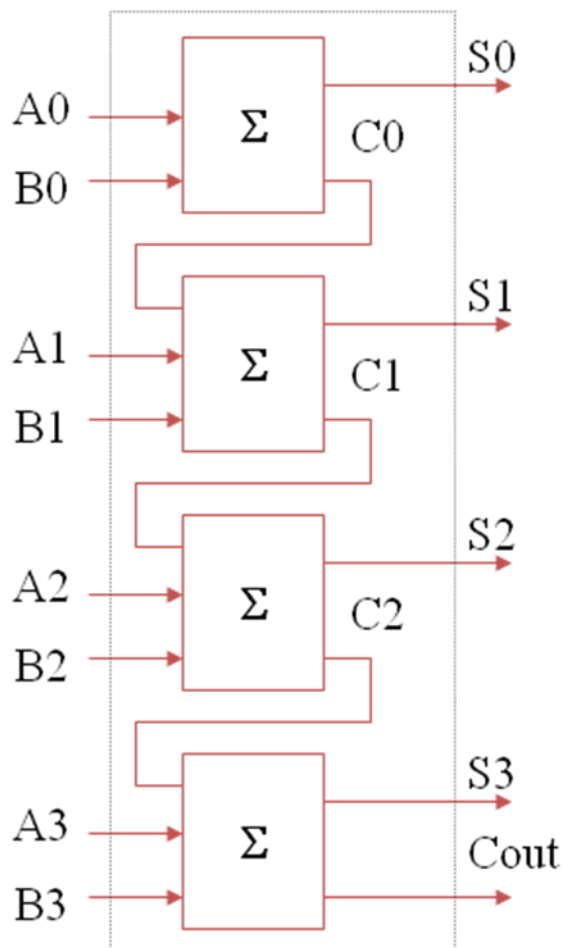


Figura 10