

**Aula de Problemas nº 4**  
**Circuitos Combinatórios Típicos**  
**Circuitos Aritméticos**

**Problema 1**

Implementar a função booleana  $f(A, B, C) = \bar{A}B + \bar{A}\bar{C} + \bar{B}C$  usando:

- a) Um decodificador 3:8 e uma porta lógica adicional;
- b) Multiplexers com 3 entradas de seleção;
- c) Multiplexers com 2 entradas de seleção.

R: A função indicada tem a seguinte tabela de verdade:

ABC	f
000	1
001	1
010	1
011	1
100	0
101	1
110	0
111	0

Logo pode ser representada pela seguinte soma de mintermos:  $f = \sum_m(0,1,2,3,5)$

a) Utiliza-se um decodificador com 3 variáveis de seleção às quais se ligam 'A', 'B' e 'C' ('A' é ligado à entrada de maior peso). Utiliza-se uma porta OR com 5 entradas que são ligadas às saídas 0, 1, 2, 3 e 5 do decodificador. Se o decodificador tiver saídas ativas a Low (i.e., saídas invertidas), então a porta a utilizar é uma NAND. Vejam os slides das aulas teóricas.

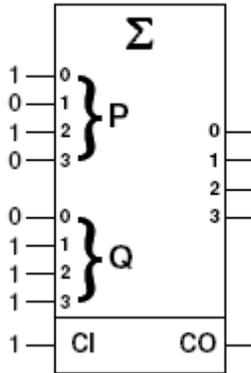
b) Ligam-se 'A', 'B' e 'C' às entradas de seleção do MUX ('A' liga-se à entrada de maior peso). A cada uma das 8 entradas de dados do MUX liga-se um '1' ou um '0' de acordo com a tabela de verdade, i.e., às entradas D0, D1, D2, D3 e D5 liga-se '1'. Às entradas restantes (D4, D6, D7 e D8) liga-se '0'. Vejam os slides das aulas teóricas.

c) Neste caso o MUX só tem 2 entradas de seleção e 4 entradas de dados. A forma mais direta de resolver consiste em ligar as variáveis de maior peso ('A' e 'B') às entradas de seleção, e ligam-se as entradas de dados D0 D1 D2 e D3 de acordo com a seguinte tabela: (Vejam os slides das aulas teóricas).

ABC	f	Entrada de dados
000	1	D0 = 1
001	1	
010	1	D1 = 1
011	1	
100	0	D2 = C
101	1	
110	0	D3 = 0
111	0	

**Problema 2**

Considere o circuito somador da figura abaixo. Sabendo que os números nas suas entradas estão representados em complemento para 2:



a) Qual o valor decimal dos números apresentados nas entradas?

R: 0101 = +5; 1110 = -2 (como o bit mais à esquerda é '1', estamos perante um número negativo, logo encontra-se o seu complemento para 2 para saber qual o seu valor)

b) Qual o resultado da operação realizada e quais os valores lógicos nas saídas do circuito?

R: É realizada a soma 0101+1110+1 (pois o Carry In é 1), logo o valor nas saídas é 10100 (em que o bit de maior peso é o CO - Carry Out). Como se diz que os números estão em complemento para 2 de 4 bits, então só se consideram os 4 bits menos significativos do resultado (i.e., ignora-se o CO). Sendo assim o resultado obtido é 00100 = +4

**Problema 3**

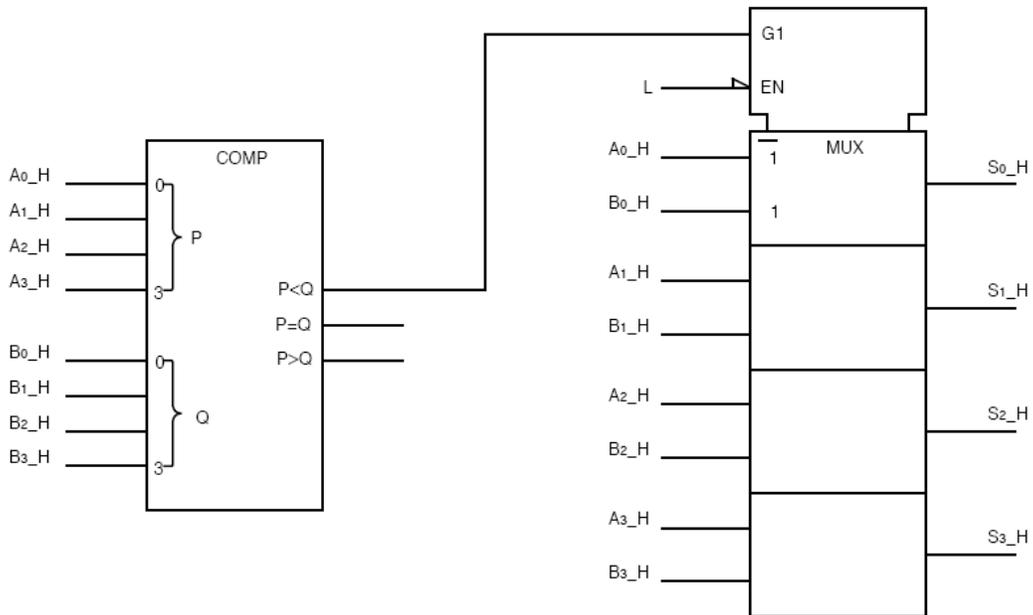
Implemente uma função que some 2 números de 2 bits utilizando 3 Mux de 8:1. Sugestão: comece por fazer as tabelas de verdade das 3 funções de saída.

R: A soma de 2 números de 2 bits ( $A_1A_0, B_1B_0$ ), cabe em 3 bits ( $S_2S_1S_0$ ). Começa-se por fazer uma tabela de verdade contendo as variáveis  $A_1A_0B_1B_0$ , e a saída da soma algébrica de A e B ( $S_2S_1S_0$ ). Depois temos que implementar as funções  $S_2, S_1$  e  $S_0$  (cada uma de 4 bits) utilizando MUXs com 3 var de seleção. É só utilizar o método aplicado no problema 1c) para cada uma das três funções.

**Problema 4**

Projete um circuito que receba dois números de 4 bits,  $A_3A_2A_1A_0$  e  $B_3B_2B_1B_0$ , e apresente na saída o maior desses 2 números também em 4 bits ( $S_3, S_2, S_1, S_0$ ).

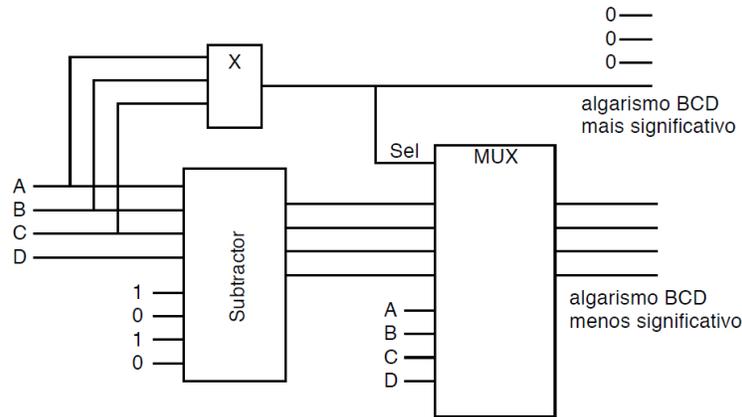
R: Utiliza-se um circuito comparador para se saber qual o maior dos números, e 4 mux 2:1 para seleccionar qual dos números de 4 bits se apresenta na saída (cada mux seleciona um dos bits). No logograma abaixo usa-se o circuito integrado 74LS157, um Quad MUX 2:1, estudado nas aulas teóricas



**Problema 5**

Projete um circuito que transforme um número binário de 4 bits num número representado em BCD.

R: Há muitas soluções possíveis. Uma delas é:



Os números binários de 4 bits vão de 0 a 15 ( $0000_2$  a  $1111_2$ ).

A correspondência entre as representações pedidas é, portanto:

Número binário	Representação em decimal	Número em BCD
0 0 0 0	0	0 0 0 0 0 0 0 0
0 0 0 1	1	0 0 0 0 0 0 0 1
0 0 1 0	2	0 0 0 0 0 0 1 0
0 0 1 1	3	0 0 0 0 0 0 1 1
0 1 0 0	4	0 0 0 0 0 1 0 0
0 1 0 1	5	0 0 0 0 0 1 0 1
0 1 1 0	6	0 0 0 0 0 1 1 0
0 1 1 1	7	0 0 0 0 0 1 1 1
1 0 0 0	8	0 0 0 0 1 0 0 0
1 0 0 1	9	0 0 0 0 1 0 0 1
1 0 1 0	10	0 0 0 1 0 0 0 0
1 0 1 1	11	0 0 0 1 0 0 0 1
1 1 0 0	12	0 0 0 1 0 0 1 0
1 1 0 1	13	0 0 0 1 0 0 1 1
1 1 1 0	14	0 0 0 1 0 1 0 0
1 1 1 1	15	0 0 0 1 0 1 0 1

Por análise da tabela verifica-se que quando o número binário é maior que 9, os 4 bits menos significativos do número em BCD se obtêm subtraindo 10.

Assim, é necessário começar por determinar se o número à entrada é ou não maior que 9.

Isso é fácil. Vamos supor que os bits do número são as variáveis  $A$ ,  $B$ ,  $C$  e  $D$ , sendo  $A$  a mais significativa. O número é maior que 9 se a função  $X=A(B+C)$  for 1.

Portanto  $X$  é o bit menos significativo do algarismo BCD de maior peso.

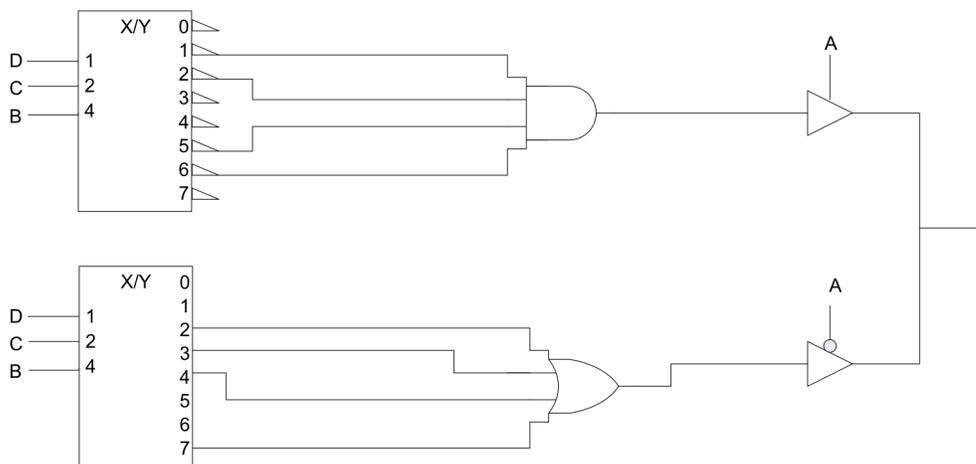
De seguida utiliza-se um circuito subtrator para subtrair 6 quando o número é maior que 9.

Finalmente utiliza-se um MUX para selecionar se pretendemos os 4 bits originais ou o resultado da subtração. O MUX utilizado é um Quad MUX 2:1, 4 mux com uma variável de selecção que seleciona uma de duas entradas (ou seja temos um circuito que seleciona uma de duas entradas de quatro bits cada). Ver a simbologia normalizada nos slides das aulas teóricas (74LS157)

Há muitas outras soluções. Por exemplo: Utilizar um somador com 6 em vez do subtrator (como foi feito na aula); Utilizar um somador em que na entrada Q se soma 0 ou 6 consoante o número é maior que 9 ou não; Utilizar um comparador para indicar se o número na entrada é maior que 9; Utilizar só muxs, etc.

### Problema 6

Qual a expressão lógica da função implementada pelo seguinte circuito:



R: Os circuitos na saída das portas AND e OR são buffers Tri-State, controlados respetivamente pelas variáveis A e !A. Quando a variável de controlo está ativa, o buffer tri-state deixa passar o sinal que está na sua entrada. Quando a variável de controlo está inativa, o circuito está num estado de alta impedância e não deixa passar nenhuma corrente (logo é como se não existisse nenhum fio a ligar o buffer até à saída. A junção da saída dos 2 buffers implementa um OR lógico. Assim:

- Na saída do AND temos a função  $g = \overline{m1 + m2 + m5 + m6} = \overline{\overline{B}CD + \overline{B}C\overline{D} + B\overline{C}D + BCD}$
- Na saída do OR temos a função  $h = m2 + m3 + m4 + m7 = \overline{B}C\overline{D} + \overline{B}CD + B\overline{C}\overline{D} + BCD$
- A função implementada pelo circuito é  $f = A.g + \overline{A}.h$