

# Características Eléctricas das Famílias Lógicas

(particularizando para TTL)

1. Considere a seguinte tabela, que resume algumas das características eléctricas de um chip '04 da família TTL (*Standard, Low-Power Schottky e Schottky*):

Grandeza	7404	74LS04	74S04	Un.
$V_{OL}$	0.4	0.5	0.5	V
$V_{OH}$	2.4	2.7	2.7	V
$V_{IL}$	0.8	0.8	0.8	V
$V_{IH}$	2	2	2	V
$I_{OL}$	16	8	20	mA
$I_{OH}$	- 0.4	- 0.4	- 1	mA
$I_{IL}$	- 1.6	- 0.4	- 2	mA
$I_{IH}$	40	20	50	$\mu$ A
$I_{CCL}$	33	6.6	54	mA
$I_{CCH}$	12	2.4	24	mA

Para cada um dos chips em questão, determine o seu *fan-out*, às suas **margens de ruído** e a **potência média dissipada**.

**NOTA:** Para  $I_{CCL}$  e  $I_{CCH}$  escolheu-se os **valores típicos**, fazendo-se então a **média** dos dois. Por vezes as folhas de dados apresentam apenas os valores de  $I_{CC}$ ; neste caso, considerar o **valor típico**.

# Álgebra de Boole

1. Demonstre o teorema da **idempotência** ( $b.b = b$ ;  $b+b = b$ ).
2. Demonstre o teorema do **elemento absorvente do produto** ( $b.0 = 0$ ).
3. Prove o **teorema da absorção**:  $x + xy = x$ .
4. Prove o **teorema da adjacência**:  $xy + x\bar{y} = x$ .
5. Prove o **teorema do consenso** ( $xy + \bar{x}z + yz = xy + \bar{x}z$ ): “Dado um par de termos em que a variável  $x$  aparece num termo e complementada no outro, o termo de consenso é formado pelo produto das variáveis que acompanham  $x$  num termo e o seu complemento no outro.”
6. Prove que  $x + \bar{x}y = x + y$  (**teorema da simplificação**).
7. Demonstre o teorema da **unicidade do complemento**.
8. Prove (por **indução**) que a função **ou-exclusivo** é **associativa**.
9. Prove que  $xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz = x + yz$ .
10. Simplifique a seguinte função:  $f = \bar{x}yz + xy\bar{z} + xyz$ .
11. Simplifique a seguinte função:  $f = xyz + xy\bar{z} + x\bar{y}\bar{z} + \bar{x}yz$ .
12. Simplifique a seguinte função:  $y = x_1\bar{x}_2 + x_3 + \bar{x}_1\bar{x}_3x_4 + x_2\bar{x}_3x_4$ .
13. Considere a função apresentada na questão anterior (depois de simplificada).
  - a) Escreva a função  $y$  só com funções **NAND**. Faça o **diagrama lógico**.
  - b) Idem, mas com **NORs**.
14. Prove por **indução** os **teoremas de De Morgan**:
  - a)  $\overline{x + y} = \bar{x}.\bar{y}$
  - b)  $\overline{x.y} = \bar{x} + \bar{y}$

## Bases de Numeração e Aritmética

1. Qual a representação **binária** dos seguintes números?  
a)  $1234567_8$                                       b)  $DACE_{16}$                                       c)  $110100_3$   
d)  $110100_8$                                       e)  $110100_{10}$
2. Qual a representação **decimal** dos seguintes números?  
a)  $1234.567_8$                                       b)  $CF.BA_{16}$                                       c)  $110.100_3$   
d)  $110.100_8$                                       e)  $110.100_2$
3. Escreva os seguintes **números decimais** nas suas **representações binária, octal e hexadecimal**:  
a) 1987                                      b) 7777                                      c) 3.141  
d) 205.109375
4. Converta **directamente** para **binário** e **octal** os seguintes números **hexadecimais**:  
a) ADA.DE                                      b) D.ADA                                      c) EA03.5C
5. Um microcomputador trabalha com **palavras de 8 bits** (byte) e representa a informação numérica inteira na notação de **complemento para 2**.  
a) Qual o **maior** e o **menor número inteiro** que podem ser representados?  
b) Indique a **representação binária** dos seguintes números decimais: +93, -93, 0, -1.
6. Um microcomputador trabalha com palavras de **16 bits**. Qual a **representação binária** de  $-77_{10}$  usando:  
a) Notação **sinal e módulo** (magnitude).  
b) Notação em **complemento para 2**.  
c) Notação em **complemento para 1**.
7. Faça as seguintes **operações em binário (complemento para 2)**; os operandos apresentados estão na sua representação **decimal**.  
a)  $21 + 55$                                       b)  $21 \times 55$                                       c)  $48 - 21$   
d)  $-48 - 21$                                       e)  $21 - 48$                                       f)  $120 + 10$   
g)  $-120 - 10$

8. Repita as alíneas anteriores para binário em **complemento para 1**.
9. Execute as seguintes **operações em binário** (os operandos estão apresentados na base decimal). Admita que trabalha com palavras de comprimento fixo e igual a **10 bits**. Use a representação em **complemento para 2**.
- |                   |                |                    |
|-------------------|----------------|--------------------|
| a) $33 + 127$     | b) $453 - 322$ | c) $453 + 322$     |
| d) $-453 + 322$   | e) $-512 + 35$ | f) $-453 + (-322)$ |
| g) $322 + (-322)$ |                |                    |
10. Repita as alíneas anteriores para binário em **complemento para 1**.

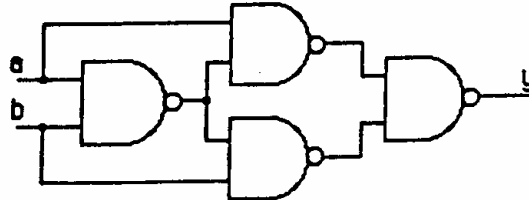
# Códigos Binários

1. a) Escreva o código **BCD 8421**.  
b) Qual a sua **redundância**? Qual a sua **distância mínima**? Que **outras propriedades** tem?  
c) Escreva o número **1905<sub>10</sub>** usando este código. Compare com a sua representação em binário.
2. Repita a pergunta anterior para o código **BCD 84-2-1**.
3. Construa um código BCD **ponderado** e **autocomplementar** com os pesos **87-4-2**.
4. a) Escreva o código **XS3** (ou de **Stibitz**).  
b) Que **propriedades** tem?  
c) Prove que este código **não é** um código **ponderado**.
5. a) Escreva o código **Aiken**.  
b) Que **propriedades** tem?  
c) Determine se este código **é ou não ponderado**.
6. a) Escreva um código BCD **autocomplementar** com os pesos **642-3**, e que distinga as **palavras menores que 5** das que são **maiores ou iguais a 5**.  
b) Modifique o código determinado em a) de forma a transformá-lo num **código detector de um erro**, pelo método do **bit de paridade**; pretende-se **paridade ímpar** para a palavra global.
7. Prove que para um código BCD ser **ponderado** e **autocomplementar**, a **soma dos seus pesos deve ser igual a 9**.
8. Pretende-se construir um sistema de transmissão de dados que seja capaz de **detectar erros** por cada envio de **um byte de informação**. Qual o **comprimento global mínimo** de cada palavra a transmitir, no caso de se pretender a **deteção de um único erro**?
9. a) Construa um código BCD de **5 bits**, sendo 4 bits iguais aos do código **BCD 8421** e o 5º **bit de paridade**; as palavras completas devem ter **paridade ímpar**.  
b) Qual a **redundância** e qual a **distância mínima** deste código?

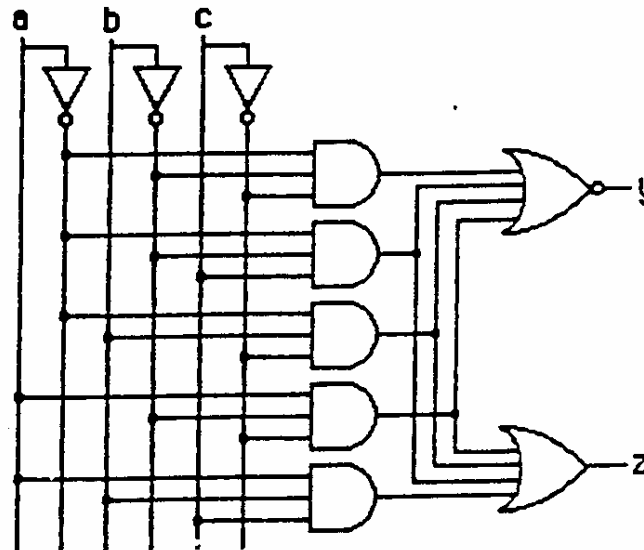
- c) Demonstre se este novo código é ou não **ponderado**.
10. a) Escreva o código **biquinário**.  
 b) Qual a sua **distância mínima**?  
 c) O que pode concluir quanto à sua **paridade**?  
 d) Descreva algumas estratégias de **detecção de erros**.
11. Considere o Código de **Hamming**.  
 a) Deseja-se transmitir 6 bits de dados (sejam **001101**,  $D = 6$ ). Determine quantos bits ( $T$ ) serão efectivamente enviados e calcule-os.  
 b) Determine os dados originais ( $D$  bits), correspondentes às seguintes palavras recebidas, sabendo que, **no máximo**, ocorreu **um erro** em cada palavra recebida:  
**111.011.110, 001.111.111, 010.011.001, 110.011.100, 101.100.000.**
- Nota:** Sendo  $D$  o número de bits de dados propriamente ditos e  $C$  o número de bits de controlo acrescentados pelo Código de Hamming, o número total de bits a transmitir será  $T = D + C$ , em que  $T$  é o menor valor inteiro que verifica a relação:  $2^D \cdot (T+1) \leq 2^T$ .
12. a) Escreva o **código Gray** com palavras de **4 bits**.  
 b) Que **propriedades** tem?  
 c) A partir deste código, construa um **código BCD** de **distância unitária, não cíclico**.  
 d) Idem, mas **cíclico**.  
 e) Seria possível construir um código de **distância unitária** com um **número ímpar de palavras**? Se sim, como. Se não, porquê?
13. Projecte um **codificador axial** (*optical shaft encoder*) com uma **resolução de 30°**, de **distância unitária e cíclico**.
14. a) Escreva o código decimal binário para um **display de 7 segmentos**, de acordo com as configurações apresentadas ao lado.
- 
- 
- b) Calcule a sua **redundância**.  
 c) Qual a sua **distância mínima**?  
 d) Conclua quanto à adequação deste código a uma aplicação de **detecção de erros**.
15. Considere agora também os valores de 0 a F (**hexadecimal**).
-

## Circuitos Combinatórios

1. O circuito da figura representa a implementação de uma função lógica conhecida. Qual é? Que **vantagens** tem sobre a implementação tradicional? Resolver **analiticamente** e por **indução** (tabelas de verdade).

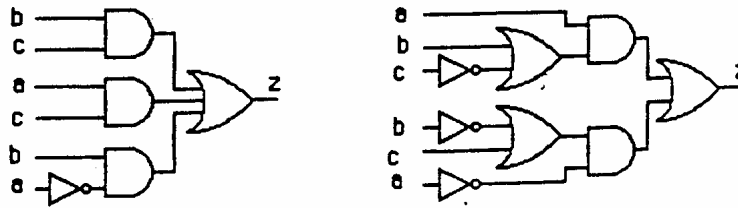


2. Quais as funções lógicas do diagrama?



3. Escreva as seguintes funções em termos de **somas de mintermos** e em termos de **produtos de maxtermos**.
- a) 1001 0110 ( $96_{16}$ )      b) 0101 0101 ( $55_{16}$ )      c) 1000 0001 ( $81_{16}$ )  
 d) 0111 1110 ( $7E_{16}$ )      e) 1011 0001 0110 0101 ( $B165_{16}$ )
4. Uma sala tem 3 portas de acesso (A, B e C). Junto de cada porta está um interruptor capaz de ligar ou desligar a luz. Quando qualquer um dos interruptores é actuado, o estado da luz muda (acende se estiver apagada, apaga se estiver acesa). Admite-se que a luz está apagada quando todos os interruptores estão para baixo (zero).  
**Escreva a equação lógica do estado de iluminação** ( $L = 1$ : luz acesa) em função do estado dos interruptores A, B e C.

5. Simplifique os seguintes circuitos usando o **método de Karnaugh**.



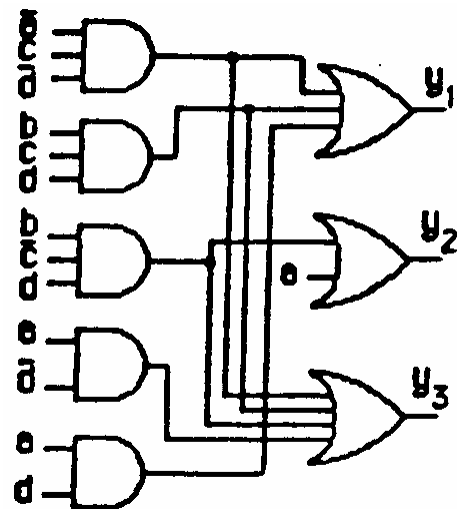
6. Seja  $F = 1111\ 1011\ 0111\ 0110$  ( $FB76_{16}$ )

- Escreva a função na mais simples **soma de produtos** e na mais simples **produto de somas**.
- Escrever a função com **NORs** e **NANDs**.

7. Usando mapas de **Karnaugh**, simplifique as seguintes funções booleanas.

- $y_1 = \sum m(1,2,3,6)$
- $y_2 = \sum(0,2,3,4)$
- $y_3 = \prod M(2,4,6,7)$
- $z_1 = \sum m(1,6,9,11,13,14,15)$
- $z_2 = \sum m(1,3,5,7,9,10,11,13,15)$
- $z_3 = \prod(0,2,8,9,12,13)$
- $z_4 = \sum m(0,1,4,6,7,8,9,10,14)$
- $z_5 = \sum m(0,1,2,5,6,8,9,12,15)$
- $w_1 = \sum m(0,1,7,8,9,10,13,14) + d(2,4,5)$
- $w_2 = \sum m(1,5,7,12) + d(4,6,8,13)$
- $w_3 = \prod M(0,1,2,4,7,9,14,15) + d(8,10,13)$

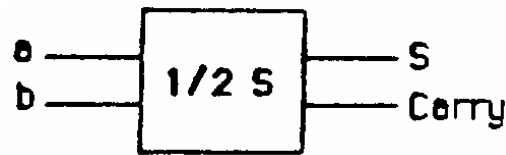
- Dado o circuito da figura ao lado, determinar as **tabelas de verdade** das funções  $y_1$ ,  $y_2$  e  $y_3$ .
- Comparar** com as tabelas de verdade do problema anterior.
- Comparar** as **expressões algébricas** de cada uma das funções, contar as **gates** e concluir.



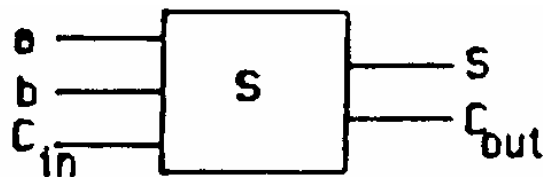
- Conceber um circuito com 3 entradas cuja saída seja '1' quando aparecer na entrada um **número ímpar de uns**.
- Conceber um circuito com 4 entradas que **detecte as combinações de entrada** com 2 ou 3 uns.



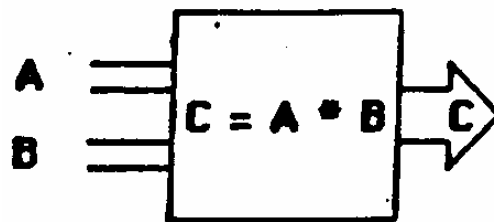
11. Projecte um circuito **meio-somador** (*half adder*). Este circuito gera o resultado da soma ( $s$ ) de dois bits ( $a$  e  $b$ ), bem como o transporte (*carry*).



12. Projecte um **somador completo**. Tem 3 entradas: além dos 2 bits a somar ( $a$  e  $b$ ), há ainda o transporte da soma anterior ( $c_{in}$ ).



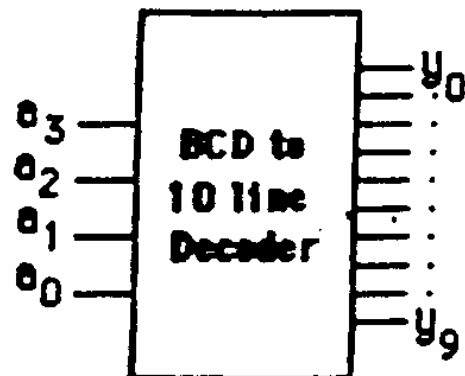
13. Projecte um circuito que **multiplique** duas palavras de dois bits cada.



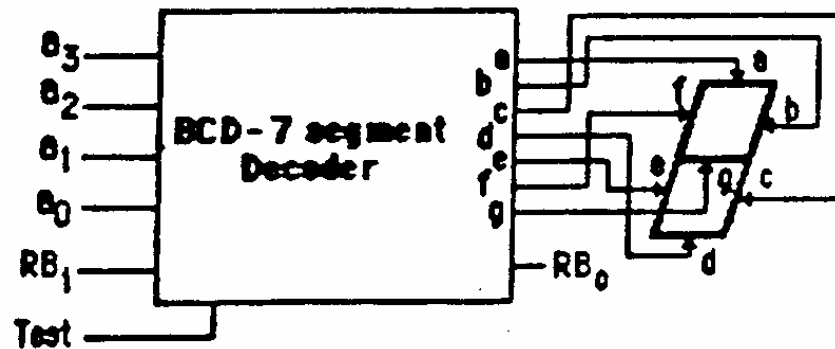
14. Projectar um circuito **descodificador do código**

**BCD8421**. Considere as seguintes hipóteses:

- Circuito **mais simples**.
- Inexistência de erros caso surjam **palavras que não pertencem ao código**.



15. a) Projectar um conversor de código **BCD8421** para *display* de 7 segmentos (ver página seguinte). Projecte o circuito **o mais simples possível**, usando **somas de produtos**.
- b) Qual o **aspecto visual** relativo às **entradas não pertencentes ao código BCD**?
- c) Modifique o circuito para permitir **mais uma entrada e mais uma saída para eliminar os zeros à esquerda** (*ripple blanking*).
- d) Proceda ainda a uma alteração mais, de forma a que todos os segmentos fiquem acesos quando uma **entrada de teste** estiver activa.



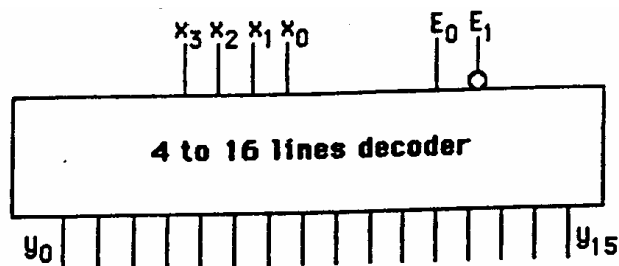
16. Projectar um conversor de código **BCD8421** para *display* de 7 segmentos, de forma a que sejam **apagados todos os segmentos** quando apareça uma palavra na **entrada** que **não pertença ao código BCD**.
17. Projecte um conversor de código **hexadecimal** para *display* de 7 segmentos.
18. Os modernos gravadores digitais de banda magnética, quando usados no modo *streaming*, usam um código conhecido por GCR (*Group Code Recording*). Com este código são gravadas palavras de **5 bits por cada 4 bits de informação**, de acordo com a tabela dada.

B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	G <sub>4</sub>	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>
0	0	0	0	1	1	0	0	1
0	0	0	1	1	1	0	1	1
0	0	1	0	1	0	0	1	0
0	0	1	1	1	0	0	1	1
0	1	0	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1
0	1	1	0	1	0	1	1	0
0	1	1	1	1	0	1	1	1
1	0	0	0	1	1	0	1	0
1	0	0	1	0	1	0	0	1
1	0	1	0	0	1	0	1	0
1	0	1	1	0	1	0	1	1
1	1	0	0	1	1	1	1	0
1	1	0	1	0	1	1	0	1
1	1	1	0	0	1	1	1	0
1	1	1	1	0	1	1	1	1

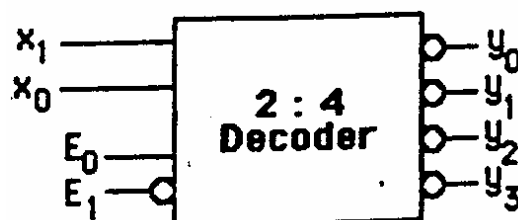
- a) Projecte um **conversor de código** para ser usado na **fase de gravação**.
- b) Projecte um **conversor de código** para ser usado na **fase de leitura**.
19. Projecte o circuito que converta palavras no **código de O'Brien** nas correspondentes palavras do **código de Libaw-Craig** (ver tabela deste códigos na página seguinte). Projecte também o circuito que procede à conversão inversa.

Dec.	Código de O'Brien				Código de Libaw–Craig				
0	0	0	0	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0	1
2	0	0	1	0	0	0	0	1	1
3	0	1	1	0	0	0	1	1	1
4	0	1	0	0	0	1	1	1	1
5	1	1	0	0	1	1	1	1	1
6	1	1	1	0	1	1	1	1	0
7	1	0	1	0	1	1	1	0	0
8	1	0	1	1	1	1	0	0	0
9	1	0	0	1	1	0	0	0	0

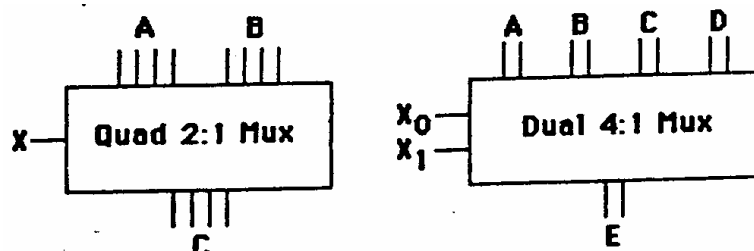
20. Projecte um **descodificador de 4 para 16 linhas** (hexadecimal), conforme indicado na figura. Deverá ter **duas entradas de enable**, uma activa no nível lógico 1 e outra no nível lógico 0.



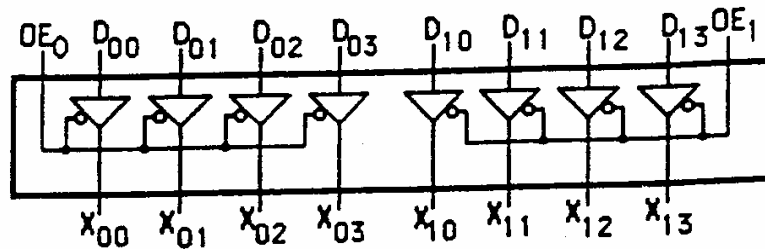
21. Projecte um **descodificador de 4 para 16 linhas** (hexadecimal), usando como bloco elementar o seguinte circuito **descodificador de 2 para 4 linhas**:



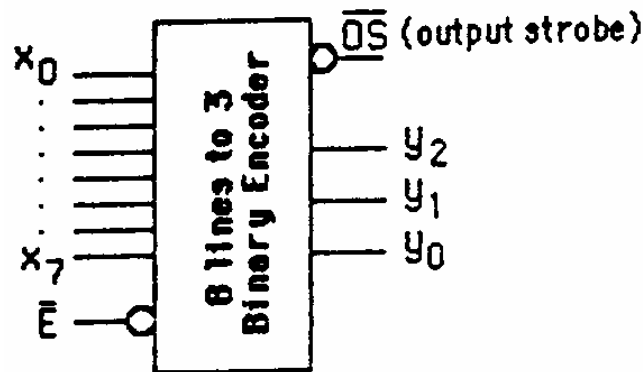
22. Projecte um circuito **multiplexer** de modo a poder seleccionar uma de 8 palavras de 4 bits cada. Use os seguintes circuitos como blocos elementares:



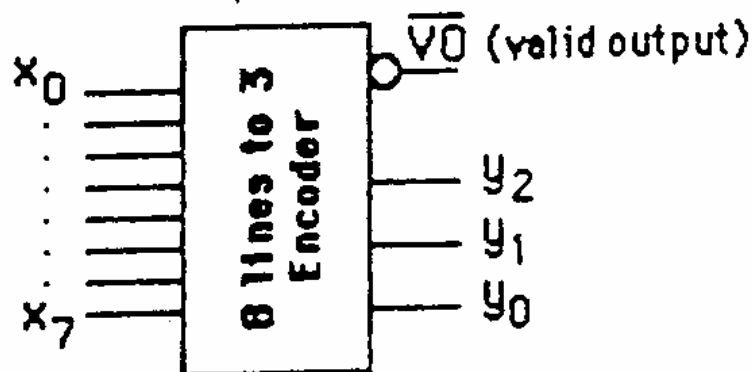
23. Projecte um circuito que escolha uma de 8 palavras de 4 bits cada, usando o **buffer tri-state 74LS244**:



24. Projecte um **codificador com prioridades** para o código binário de **3 bits**. Deverá ter uma linha de saída indicando **codificação válida** e uma entrada de **enable** (ambas activas no nível lógico baixo).



25. a) Projecte um circuito **codificador** de 8 entradas para o **código Gray**. Deverá ter uma linha de saída indicando entrada activa.  
b) Faça as alterações necessárias de modo a transformar o circuito de a) num **codificador com prioridades**.

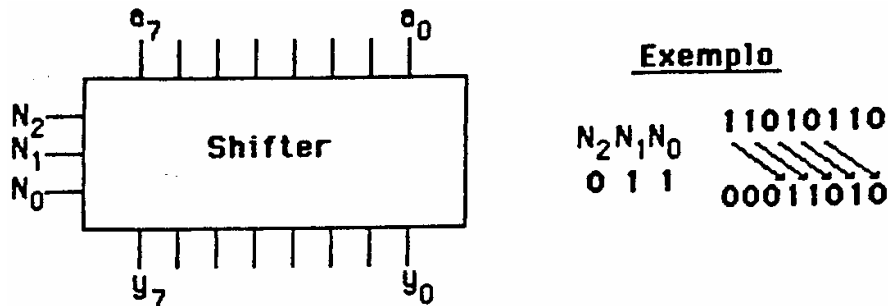


26. Pretende-se construir um circuito combinatório que faça o **deslocamento à direita de uma palavra de 8 bits**. O número de bits deslocados é dado pelas variáveis  $N_2N_1N_0$ , podendo variar de 0 a 7.

Indicar quais as **características** que uma **PLA** deverá ter para implementar este circuito.

Considere as seguintes opções:

- Deslocamento **não circular**.
- Deslocamento **circular**.

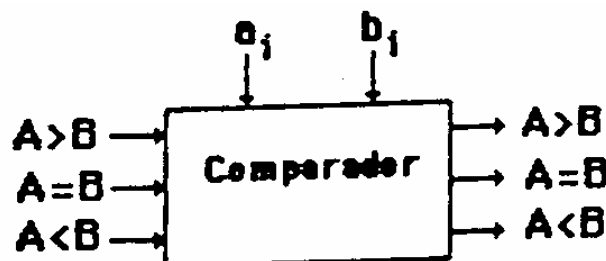


27. Numa fábrica há uma conduta terminada por uma válvula, por onde passa um fluido aquecido. Na conduta encontram-se sensores de temperatura que fazem as seguintes indicações de temperatura:  $\geq 10^\circ\text{C}$ ,  $\geq 100^\circ\text{C}$ ,  $\geq 200^\circ\text{C}$ . Pretende-se **controlar a válvula** de modo a que esteja aberta se a temperatura for inferior a  $10^\circ\text{C}$  ou estiver compreendida entre  $100^\circ\text{C}$  e  $200^\circ\text{C}$ . A válvula estará fechada se a temperatura estiver entre  $10^\circ\text{C}$  e  $100^\circ\text{C}$  ou se for superior a  $200^\circ\text{C}$ . Projecte um circuito lógico que **controle a abertura da válvula** e **accione um sinal de alarme** sempre que a temperatura exceder os  $200^\circ\text{C}$ . Considere as seguintes situações:

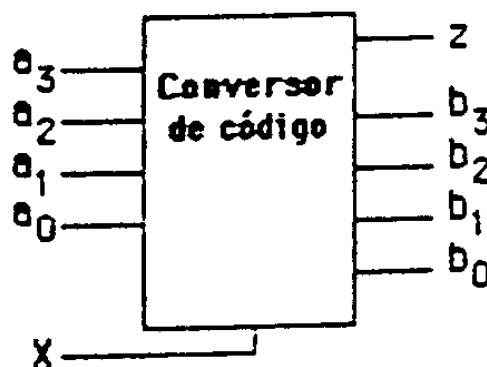
- Impossibilidade de ocorrência de erros** nas leituras dos sensores.
- Possibilidade de ocorrência de erros** nas leituras dos sensores — sempre que isso acontecer, **fechar a válvula** e **acionar um segundo alarme** (que indicará a existência do erro).

28. Projecte um **circuito iterativo** que compare duas palavras **A** e **B** de **n** bits cada. Cada bloco elementar terá, além dos dois bits  $a_i$  e  $b_i$  a comparar, três **entradas**  $A > B$ ,  $A = B$  e  $A < B$ , e três **saídas**  $A > B$ ,  $A = B$  e  $A < B$ .

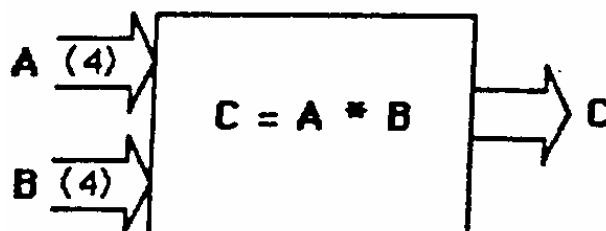
**Sugestão:** Pretende-se que a comparação seja feita a partir dos bits mais significativos.



29. Repita a problema anterior, realizando o elemento iterativo com uma **ROM**. **Dimensione-a** e escreva o seu **conteúdo**.
30. Pretende-se construir um circuito combinatório que sirva de **conversor de código BCD 8421 para Aiken** (se a entrada  $X = 0$ ), e de **Aiken para BCD 8421** (para  $X = 1$ ). O circuito deverá ter uma **saída Z** que indique se à entrada está presente uma palavra não pertencente ao código em questão.
- Escreva as **equações booleanas** do circuito.
  - Quais as **características mínimas** de uma **PLA** capaz de o implementar?
  - Qual a **dimensão mínima** e o **conteúdo** de uma **ROM** capaz de o implementar?



31. Pretende-se construir um circuito que aceite como entradas duas palavras A e B de 4 bits cada, e que mostre a maior dessas palavras num **display de 7 segmentos** e a menor noutra **display**.
32. Construa um **circuito multiplicador** de duas palavras de 4 bits cada, usando:
- Circuitos **multiplicadores** de duas palavras de 2 bits (Problema 13).
  - Somadores** binários de 4 bits.



# Circuitos Sequenciais

1. Apresente as **tabelas de verdade**, **mapas K**, **expressões simplificadas** e **circuito lógico** dos seguintes circuitos sequenciais:

- Um *flip-flop* JK a funcionar como SR.
- Um *flip-flop* JK a funcionar como T.
- Um *flip-flop* JK a funcionar como D.
- Um *flip-flop* SR a funcionar como JK.
- Um *flip-flop* SR a funcionar como T.
- Um *flip-flop* SR a funcionar como D.
- Um *flip-flop* T a funcionar como JK.
- Um *flip-flop* T a funcionar como SR.
- Um *flip-flop* T a funcionar como D.
- Um *flip-flop* D a funcionar como JK.
- Um *flip-flop* D a funcionar como SR.
- Um *flip-flop* D a funcionar como T.

2. Projecte um contador síncrono de **módulo 4**.

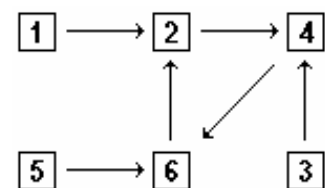
- Utilizando *flip-flops* JK.
- Utilizando *flip-flops* SR.
- Utilizando *flip-flops* T.
- Utilizando *flip-flops* D.

3. Repita o exercício anterior para um contador de **módulo 6**.

4. Repita a alínea anterior para uma máquina com **os seguintes estados**:

- ...4...5...6...7...4...
- ...1...0...4...6...5...1...
- ...2...3...4...5...2...

5. Projecte uma máquina que efectue as seguintes **transições de estados**:



6. Projecte um **contador bidireccional de módulo 8**, controlado por uma entrada de direcção *DIR*. Utilize *flip-flops* tipo JK.

7. Projecte um **detector de sequências** que detecte as seguintes sequências na sua entrada X:

- a) 0110      b) 1111      c) 101      d) 01010

8. Apresente o diagrama de estados de uma máquina que apresente a **situação actual da nossa alma** de acordo com os seguintes critérios:

- Cada **boa acção** praticada redime-nos de um **pecado** cometido (antes ou depois);
- Se o nosso saldo for nulo, somos considerados **justos**;
- Se tivermos um saldo positivo (até 4) de boas acções, somos considerados **bons**;
- Se o saldo positivo for igual a 5, vamos para o paraíso (somos **eleitos**);
- Se tivermos um saldo negativo (excesso de pecados) menor ou igual a 3, somos **pecadores**;
- Com 4 pecados somos ainda **pecadores**, mas deve soar um **alarme de perigo** de perdição;
- Ao 5º pecado vamos para o inferno (somos **condenados**).

9. Projectar um **contador up-down síncrono de módulo 16**, que conte em binário. O contador deverá ter entradas para programação síncrona comandada pela linha  $\overline{PL}$  (**Parallel Load**). Deverá ter ainda uma linha  $\overline{MR}$  (**Master Reset**) que faz o *reset* assíncrono de todos os *flip-flops*. Usar *flip-flops* tipo D com entradas de *preset* ( $\overline{PRE}$ ) e *clear* ( $\overline{CLR}$ ).

10. a) Minimize a **máquina sequencial síncrona** representada pela

**tabela de estados** ao lado:

b) Arranje a **menor sequência de entradas** que faça a distinção entre os estados A e B.

	X = 0	X = 1
A	B, 1	H, 1
B	F, 1	D, 1
C	D, 0	E, 1
D	C, 0	F, 1
E	D, 1	C, 1
F	C, 1	C, 1
G	C, 1	D, 1
H	C, 0	A, 1

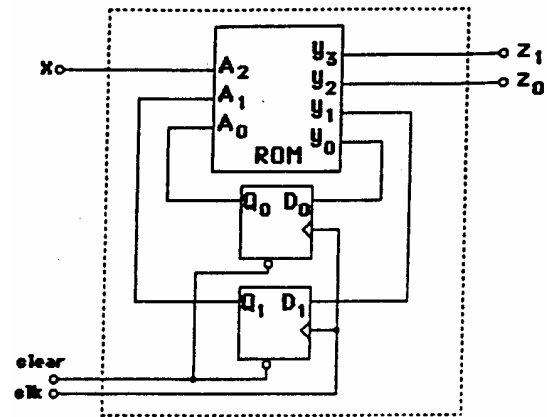
11. **Repetir o problema anterior** para as seguintes máquinas sequenciais síncronas:

	X = 0	X = 1
A	B, 0	E, 0
B	E, 0	D, 0
C	D, 1	A, 0
D	C, 1	E, 0
E	B, 0	D, 0

	X = 0	X = 1
A	D, 0	H, 1
B	F, 1	C, 1
C	D, 0	F, 1
D	C, 0	E, 1
E	C, 1	D, 1
F	D, 1	D, 1
G	D, 1	C, 1
H	B, 1	A, 1



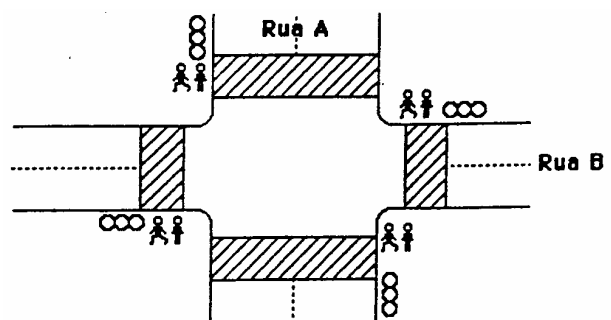
12. Projecte um **circuito sequencial síncrono** que realize o **complemento para dois** dum número que entra em série através dos seus bits menos significativos.
13. Desenhe o diagrama de estados de um *shift-register* de 4 bits.
14. Desenhe o **diagrama de estados** do seguinte circuito sequencial síncrono com uma entrada  $x$  e duas saídas  $z_0$  e  $z_1$ .



É o seguinte o **conteúdo da ROM**:

$A_2$	$A_1$	$A_0$	$y_3$	$y_2$	$y_1$	$y_0$
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	1	1	1	0
1	1	0	0	1	1	0
1	1	1	1	1	1	1

15. A figura representa um cruzamento de duas ruas. Dado o tipo de tráfego, há muito mais movimento na rua A que na rua B. Assim, pretende-se um sistema que controle os semáforos reguladores de trânsito. O verde deverá estar aceso durante 40s na rua A e durante 20s na



rua B. A passagem de verde para vermelho é sempre antecedida por um período de 5s em amarelo. Por questões de segurança, há um período de 5s que medeia entre o início de vermelho numa rua e o início de verde na outra rua. Projecte o sistema que controla os **semáforos**, bem como as luzes de passagem dos **peões**.