

SD- práticas

António Teixeira

Regras básicas

Postulados de Huntington

Operador OR	Operador AND	Operador NOT
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\bar{0} = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\bar{1} = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

* P.1 Ambas as operações fechadas em B

$$\forall b_1, b_2 \in B \quad \begin{cases} b_1 + b_2 \in B \\ b_1 \cdot b_2 \in B \end{cases}$$

* P.2 Comutatividade

$$\forall b_1, b_2 \in B \quad \begin{cases} b_1 + b_2 = b_2 + b_1 \\ b_1 \cdot b_2 = b_2 \cdot b_1 \end{cases}$$

* P.3 Elementos Neutros

$$\exists b_0 \forall b \in B \quad b + b_0 = b \\ \exists b_1 \forall b \in B \quad b \cdot b_1 = b$$

* P.4 Distributividade

$$\begin{aligned} \forall b_1, b_2, b_3 \in B \quad & b_1 + (b_2 \cdot b_3) = (b_1 + b_2) \cdot (b_1 + b_3) \\ \forall b_1, b_2, b_3 \in B \quad & b_1 \cdot (b_2 + b_3) = (b_1 \cdot b_2) + (b_1 \cdot b_3) \end{aligned}$$

* P.5 Complementação

$$\forall b \exists \bar{b} \in B \quad \begin{cases} b + \bar{b} = b_1 \\ b \cdot \bar{b} = b_0 \end{cases}$$

* P.6 #B = 2

$$\forall b_1 \in B \quad \exists b_2 \in B \quad b_1 \neq b_2 \Leftrightarrow \#B \geq 2$$

T1 - Unicidade do elemento neutro

T2 - Idempotência

$$\forall b \in B \quad b + b = b; b \cdot b = b$$

T3 - Elemento absorvente $\forall b \in B \quad b + b_1 = b_1; b \cdot b_0 = b_0$

T4 - Absorção $\forall x, y \in B \quad x + xy = x$

T5 - Dualidade $b + \sim b = b_1, b \cdot \sim b = b_0$

T6 - Unicidade do complemento

T7 - Involução $(\bar{\bar{x}}) = x$

T8 - Associatividade $(x + y) + z = x + (y + z)$
 $(xy)z = x(yz)$

T9 - Leis de De Morgan $\begin{aligned} (\bar{x} + \bar{y}) &= \bar{x} \cdot \bar{y} \\ (\bar{x} \cdot \bar{y}) &= \bar{x} + \bar{y} \end{aligned}$

$$\sum_{i=1}^{n-1} x_i = \prod_{i=1}^n \bar{x}_i$$

$$\begin{aligned} \bar{f}(x_1, x_2, \dots, x_n, 0, 1, \dots) &= \\ f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n, 1, 0, \dots) &= \end{aligned}$$

$$T10 \quad x + \bar{x}y = x + y$$

$$T11 - \text{Consenso} \quad xy + \bar{x}z + yz = xy + \bar{x}z$$

(Termo mínimo de ordem i, m_i)

(Termo máximo de ordem i, M_i)

Teorema fundamental das funções booleanas:

$$Y = (x_1 x_2 \dots x_n) \quad f_i = f(Y \equiv i)$$

$$Y = f(X) = \sum_{i=0}^{2^n-1} f_i m_i$$

$$Y = f(X) = \prod_{i=0}^{2^n-1} (f_i + M_i)$$

$$Y = f(X) = \overline{\prod_{i=0}^{2^n-1} (f_i + M_i)}$$

$$Y = f(X) = \overline{\sum_{i=0}^{2^n-1} f_i m_i}$$

Teorema de Shannon

1ª forma canónica ou forma disjuntiva normal

$$Y = f(X) = \sum_{i=0}^{2^n-1} f_i m_i$$

com $X = (x_1 x_2 \dots x_n) \quad f_i = f(Y \equiv i)$

1 Demonstre os seguintes teoremas recorrendo aos postulados de Huntington:

a) Idempotência: $\forall b \in B \quad b \cdot b = b$

- Huntington postulates for Boolean algebra: defined on a set B with binary operators $+$ & \cdot , and the equivalence relation = (Edward Huntington, 1904):
- \neg (a) Closure with respect to $+$. (b) Closure with respect to \cdot .
- - (a) Identity element 0 with respect to $+$. (b) Identity element 1 with respect to \cdot .
- @ (a) Commutative with respect to $+$. (b) Commutative with respect to \cdot .
- \neg (a) is distributive over $+$. (b) \cdot is distributive over $+$.
- $\circ 8 \times 2 B; 9 \times 0 2 B$ (called the complement of x) such that (a) $x+x0 = 1$
- and (b) $x \cdot x0 = 0$.
- \pm There are at least 2 distinct elements in B .
- $b \cdot b = (b \cdot b) + b0 \text{ P3}$
- $b \cdot b = b \cdot b + b \cdot \neg b \text{ P5}$
- $b \cdot b = b \cdot (b + \neg b) \text{ P4}$
- $b \cdot b = b \cdot b1 \text{ P5}$
- $b \cdot b = b \text{ P3}$
- $b + b = b$
- $b + b = (b + b) \cdot b1 \text{ P3}$
- $b + b = (b + b) \cdot (b + \neg b) \text{ P5}$
- $b + b = b + (b \cdot \neg b) \text{ P4}$
- $b + b = b \text{ P5+P3}$

b) Elemento absorvente: $\forall b \in B \quad b \cdot b_0 = b_0, \quad b + b_1 = b_1$

- $b \cdot b_0 = b_0$
- $b \cdot b_0 = b \cdot b_0 + b_0 \text{ P3}$
- $b \cdot b_0 = b \cdot b_0 + b \cdot \neg b \text{ P5}$
- $b \cdot b_0 = b \cdot (b_0 + \neg b) \text{ P4}$
- $b \cdot b_0 = b \cdot \neg b \text{ P3}$
- $b \cdot b_0 = b_0 \text{ P5}$
- $b + b_1 = (b + b_1) \cdot b_1 \text{ P3}$
- $b + b_1 = (b + b_1) \cdot (b + \neg b) \text{ P5}$
- $b + b_1 = b + (b \cdot \neg b_1) \text{ P4}$
- $b + b_1 = b + \neg b \text{ P3}$
- $b + b_1 = b_1 \text{ P5}$

c) Unicidade do complemento

- $\sim bx = \sim bx.b1$ **P3**
- $= \sim bx.(b + \sim by)$ **P5**
- $= \sim bx.b + \sim bx.\sim by$ **P4**
- $= b0 + \sim bx.\sim by$ **P5**
- $= \sim bx.\sim by$ **P3**
- $\sim by = \sim by.b1$
- $= \sim by.(b + \sim bx)$
- $= \sim by.b + \sim by.\sim bx$
- $= b0 + \sim by.\sim bx$
- $= \sim bx.\sim by$
- Logo : $bx = by$

d) Unicidade do elemento neutro

- $b0a + b0b = b0a$ (P3)
- $b0b + b0a = b0b$ (P3)+(P2)
- Logo $b0a = b0b$

e) Absorção:

$$x + xy = x$$

- $b+b.c=b$
- $= b.1+b.c \text{ (P3)}$
- $= b.(1+c) \text{ (P4)}$
- $= b.1=b \text{ (P3)}$
- $b.(b+c)=(b+0).(b+c) \text{ (p3)}$
- $=b+0.c \text{ (P4)}$
- $=b+0=b \text{ (P3)}$

f) Simplificação:

$$x + \bar{x}y = x + y$$

- $a+\sim a.b=a+b$
- $=a.(1+b)+\sim a.b \text{ P3}$
- $=a+a.b+\sim a.b \text{ P4}$
- $=a+(a+\sim a).b \text{ P4 P5 P3}$
- $=a+b$

g) Consenso:

$$xy + \bar{x}z + zy = xy + \bar{x}z$$

- $a.b + \sim a.c + c.b = a.b + \sim a.b$
- $= a.b + \sim a.b + c.b.(a + \sim a)$
- $= a.b + \sim a.b + c.b.a + b.c.\sim a$
- $= a.b.(1+c) + \sim a.c(1+b)$
- $= a.b + c.\sim a$

h) Leis de DeMorgan:

$$\overline{x+y} = \bar{x} \cdot \bar{y}$$

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$

- | | |
|--|--|
| <ul style="list-style-type: none">• $\sim(a+b) = \sim a \cdot \sim b$• Sendo iguais,• $\sim(\sim(a+b)) \cdot (\sim a \cdot \sim b) = 0$• $=(a+b) \cdot (\sim a \cdot \sim b)$• $= a \cdot \sim a \cdot b + b \cdot \sim a \cdot \sim b$• $= 0 + 0 = 0$ | <ul style="list-style-type: none">• $(\sim a + \sim b) = \sim(a.b)$• Sendo iguais,• $= \sim(\sim(a.b)) + (\sim a \cdot \sim b) = 0$• $=(a.b) \cdot (\sim a + \sim b)$• $= a \cdot \sim a \cdot b + b \cdot \sim a \cdot \sim b$• $= 0 + 0 = 0$ |
|--|--|

2 Por indução directa (verificação exaustiva da tabela de verdade) demonstre

a) Teorema da involução: $\bar{\bar{x}} = x$

x	$\sim x$	$\sim \sim x$	x
0	1	0	0
1	0	1	1

b) Propriedade associativa: $x + (y + z) = (x + y) + z$

x	y	z	$x+y$	$y+z$	$(x+y)+z$	$x+(y+z)$
1	1	1	1	1	1	1
1	1	0	1	1	1	1
1	0	1	1	1	1	1
1	0	0	1	0	1	1
0	1	1	1	1	1	1
0	1	0	1	1	1	1
0	0	1	0	1	1	1
0	0	0	0	0	0	0

- 3 Mostre que o operador “ou” exclusivo (*XOR*), definido por $x \oplus y = \bar{x}y + \bar{y}x$, é associativo:

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z$$

- $x(+y) = x \cdot \sim y + \sim x \cdot y$
- **Associatividade:**
- $x(+)(y(+z))$
- $= x \cdot (\sim y \cdot \sim z + z \cdot \sim y) + \sim x \cdot (y \cdot \sim z + z \cdot \sim y)$
- $= x \cdot ((\sim y + z) \cdot (\sim z + y)) + \sim x \cdot y \cdot \sim z + \sim x \cdot z \cdot \sim y$
- $= x \cdot (\sim y \cdot \sim z + \sim y \cdot y + z \cdot \sim z + z \cdot y)$
- $+ \sim x \cdot y \cdot \sim z + \sim x \cdot z \cdot \sim y$
- $= x \cdot \sim y \cdot \sim z + x \cdot z \cdot y + \sim x \cdot y \cdot \sim z + \sim x \cdot z \cdot \sim y$

- $x(+y) = x \cdot \sim y + \sim x \cdot y$
- **Associatividade:**
- $(x(+y))(+z)$
- $= z \cdot (\sim x \cdot \sim y + x \cdot \sim y) + \sim z \cdot (x \cdot \sim y + y \cdot \sim x)$
- $= z \cdot ((\sim x + y) \cdot (\sim y + x)) + \sim z \cdot x \cdot \sim y + \sim z \cdot y \cdot \sim x$
- $= z \cdot (\sim x \cdot \sim y + \sim x \cdot x + y \cdot \sim y + y \cdot x) + \sim z \cdot x \cdot \sim y + \sim z \cdot y \cdot \sim x$
- $= z \cdot \sim x \cdot \sim y + x \cdot z \cdot y + \sim z \cdot x \cdot \sim y + \sim z \cdot y \cdot \sim x$
- $= x \cdot \sim y \cdot \sim z + x \cdot y \cdot z + \sim x \cdot y \cdot \sim z + \sim x \cdot \sim y \cdot z$

- 4 Recorrendo ao teorema da dualidade, determine o operador dual do “ou” exclusivo \oplus . Compare as tabelas de verdade.

- Dualidade obvia por observação dos postulados de Huntington: P1..P6
- $x(+y) = x \cdot \sim y + \sim x \cdot y$
- $x(\text{dual } (+))y$
- $= (\sim x + y) \cdot (x + \sim y)$
- $= \sim x \cdot x + \sim x \cdot \sim y + y \cdot x + y \cdot \sim y$
- $= x \cdot y + \sim x \cdot \sim y$
- $= x \Leftrightarrow y$
- Coincide com $\sim(x(+y))$

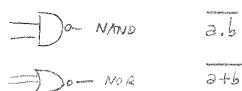
x	y	$\sim x$	$\sim y$	$x \cdot \sim y$	$\sim x \cdot y$	$x \cdot y$	$\sim x \cdot \sim y$	$x(+y)$	$x \Leftrightarrow y$
0	0	1	1	0	0	0	1	0	1
0	1	1	0	0	1	0	0	1	0
1	0	0	1	1	0	0	0	1	0
1	1	0	0	0	0	1	0	0	1

5 Mostre que $xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz = x + yz$

- $x.y.z + x.y.\sim z + x.\sim y.z + x.\sim y.\sim z + \sim x.y.z$
- $= x.y.(z + \sim z) + x.\sim y.(z + \sim z) + \sim x.y.z$
- $= x.y + x.\sim y + \sim x.y.z$
- $= x + \sim x.y.z$
- $= x.(1 + y.z) + \sim x.y.z$
- $= x + x.y.z + \sim x.y.z$
- $= x + (x + \sim x).y.z$
- $= x + y.z$

6 Mostre que os operadores $NAND$ ($\overline{x.y} = \bar{x} + \bar{y}$) e NOR ($\overline{x+y} = \bar{x}.\bar{y}$) são completos.

6)



NAND:

$$a \rightarrow \overline{D_o \text{ NAND } \overline{a.b}} \quad ; \text{ INVERT} \quad \overline{\overline{a.b}} = \overline{a} \quad (\text{Idempotência})$$

$$\begin{array}{c} \overline{a} \\ \overline{b} \end{array} \rightarrow \overline{D_o \text{ NAND } \overline{a.b}} \quad ; \text{ AND} \quad (\overline{a.b})(\overline{a.b}) = \overline{\overline{a.b}} = \overline{a.b} \quad (\text{Idempotência e Inversão})$$

$$\begin{array}{c} a \\ b \end{array} \rightarrow \overline{D_o \text{ NAND } \overline{a.b}} \quad ; \text{ OR} \quad \overline{\overline{a.b}} = a+b \quad (\text{De Morgan e Inversão})$$

NOR:

$$a \rightarrow \overline{D_o \text{ NOR } \overline{a+b}} \quad ; \text{ INVERT} \quad \overline{\overline{a+b}} = \overline{a+b} \quad (\text{Idempotência})$$

$$\begin{array}{c} \overline{a} \\ \overline{b} \end{array} \rightarrow \overline{D_o \text{ NOR } \overline{a+b}} \quad ; \text{ OR} \quad (\text{Idempotência e Inversão})$$

$$\begin{array}{c} a \\ b \end{array} \rightarrow \overline{D_o \text{ NOR } \overline{a+b}} \quad ; \text{ AND} \quad (\text{De Morgan e Inversão})$$

- Função completa permite geração de qualquer função.

7 Uma função booleana é uma regra (correspondência) que associa um elemento do conjunto $B = \{0,1\}$ a cada uma das 2^n combinações que as n variáveis independentes podem assumir. Recorrendo a uma notação vectorial para o caso geral do sistema digital com n entradas e m saídas, temos:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \longrightarrow \boxed{\mathbf{Y} = f(\mathbf{X})} \longrightarrow \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

- a) Quantas funções diferentes existem?
b) Concretize para $n = 4$ e $m = 1$.

- a) n variáveis $\Rightarrow 2^n$ combinações
- Logo: $2^{(2^n)}$ funções
- Com m saídas: $2^{(m \cdot 2^n)}$ funções
- b) $n=4, m=1 \rightarrow 2^{16}=65536$ funções

8 Considere a seguinte função booleana:

$$y = \bar{x}_1 \bar{x}_2 \bar{x}_4 + \bar{x}_1 x_3 x_4 + \bar{x}_1 \bar{x}_3 x_4 + x_1 x_2 x_4 + x_1 x_2 \bar{x}_4 + x_1 \bar{x}_2 x_3 + x_1 x_2$$

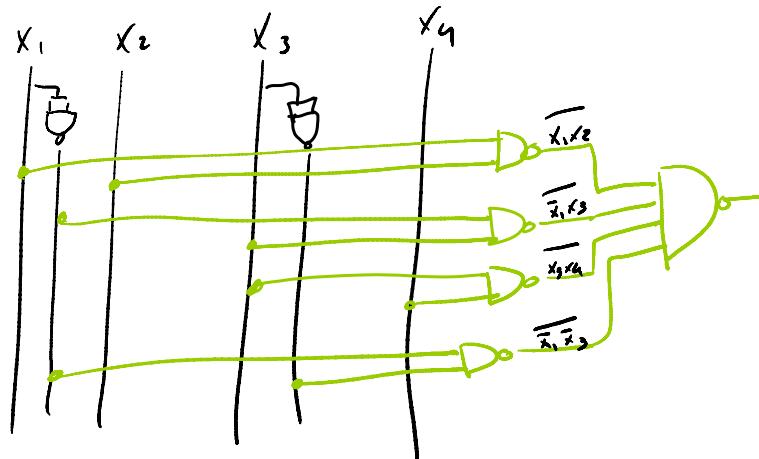
a) Simplifique-a

- $= \sim x_1 \sim x_3 (\sim x_4 + x_4) + x_1 x_2 (\sim x_4 + \sim x_4) + \sim x_1 x_3 x_4 + x_1 \sim x_2 x_3 + x_1 x_3 x_4$
- $= \sim x_1 \sim x_3 + x_1 x_2 + x_1 x_2 + \sim x_1 x_3 x_4 + x_1 \sim x_2 x_3$
- $= \sim x_1 \sim x_3 + x_1 (x_2 + \sim x_2 x_3) + \sim x_1 x_3 x_4$
- $= \sim x_1 \sim x_3 + x_1 (x_2 + x_3) + \sim x_1 x_3 x_4$
- $= \sim x_1 \sim x_3 + x_1 x_2 + x_3 (x_1 + \sim x_1 x_4)$
- $= \sim x_1 \sim x_3 + x_1 x_2 + x_3 (x_1 + x_4)$

$x_3 x_4$ $x_1 x_2$	00	01	11	10
00	1 $\cancel{x_1} \cancel{x_3}$	1	1 $x_2 x_4$	
01	1	1	1	
11	1 $x_1 x_2$	1	1 $x_1 x_3$	1
10			1	1

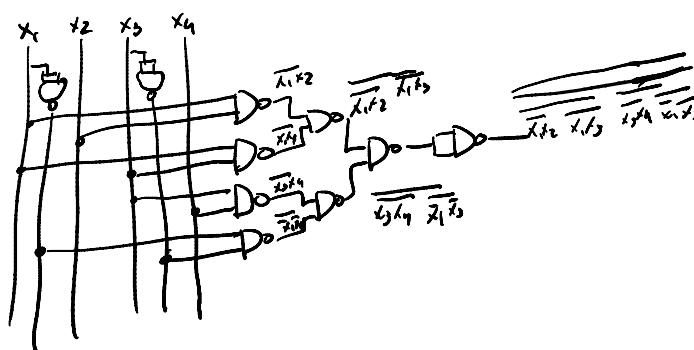
b) Reescreva y apenas com operadores *NAND*

- $=\sim x_1 \cdot \sim x_3 + x_1 \cdot x_2 + x_3 \cdot (x_1 + x_4)$
- $=\sim (\sim (\sim x_1 \cdot \sim x_3) \cdot \sim (x_1 \cdot x_2) \cdot \sim (x_3 \cdot x_1) \cdot \sim (x_3 \cdot x_4))$



c) Reescreva y apenas com operadores *NAND* que não poderão ter mais de 2 argumentos (entradas)

- $=\sim x_1 \cdot \sim x_3 + x_1 \cdot x_2 + x_3 \cdot (x_1 + x_4)$
- $=\sim (\sim (\sim x_1 \cdot \sim x_3) \cdot \sim (x_1 \cdot x_2) \cdot \sim (x_3 \cdot x_1) \cdot \sim (x_3 \cdot x_4))$
- $=\sim (\sim (\sim (\sim x_1 \cdot \sim x_3) \cdot \sim (x_1 \cdot x_2))) \cdot \sim (\sim (\sim (x_3 \cdot x_1) \cdot \sim (x_3 \cdot x_4)))$



d) Desenhe os diagramas lógicos correspondentes a b) e c) e proceda a uma análise de custos em termos de número e variedade de operadores envolvidos

- Sol1= 6 nand 2+1 nand 4
- Sol2= 10 nand 2
- As nand 2 custam:4/ 0.32€
- As nand 4 custam:2/ 0.42
- Sol1= 2*4nand2+1*2Nand2=1.06€
- Sol2= 3*4nand2=0.96€

9 A função “Maioria”, $M(x,y,z)$, é igual a 1 sempre que pelo menos dois dos seus três argumentos são iguais a 1:

$$M(x,y,z) = (x + y)(x + z)(y + z) = xy + xz + yz$$

Mostre que $M(x,y,z)$, juntamente com a operação de complementação e a constante “0”, forma um conjunto de operações funcionalmente completo.

- $M(x,y,z)$, $\sim x$, 0 -> operação funcionalmente completa
- Def: A partir da qual se consegue representar qualquer outra função booleana
- $M(x,y,0)=x.y$
- $\sim M(x,y,0)=\text{NAND}(x,y)$, que é uma operação funcionalmente completa.

Aula 3

Antonio Teixeira

Exercícios

1 Determine as formas canónicas da seguinte função booleana

$$f(x, y, z) = \bar{x}y + \bar{z} + \bar{x}\bar{y}z$$

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

- 1^a forma
- Sum(m0,2,3,4,5,6)
- 2^o forma
- Prod(M1,7)
- 3^o forma
- ~Prod(~m0,2,3,4,5,6)
- 4^o forma
- ~Sum(~M1,7)

- 2 Relativamente às variáveis independentes x , y , e z , determine as formas canónicas das funções booleanas f , g , h e w expressas na seguinte tabela de verdade

x	y	z	f	g	h	w
0	0	0	0	1	0	1
0	0	1	1	0	1	0
0	1	0	1	1	1	0
0	1	1	0	0	1	0
1	0	0	1	1	1	1
1	0	1	0	0	1	1
1	1	0	0	1	1	0
1	1	1	1	0	0	1

- $f = \text{sum}(m_1, m_2, m_4, m_7)$
- $g = \text{prod}(M_1, M_3, M_5, M_7)$
- $h = \text{prod}(M_0, M_7)$
- $w = \text{sum}(m_0, m_4, m_5, m_7)$

Representação

2-var map	
m_0	m_1
m_2	m_3

x	y	0	1
0	$x' y'$	$x' y$	
1	$x y'$	$x y$	

3-var map	
m_0	m_1
m_4	m_5
m_6	m_7

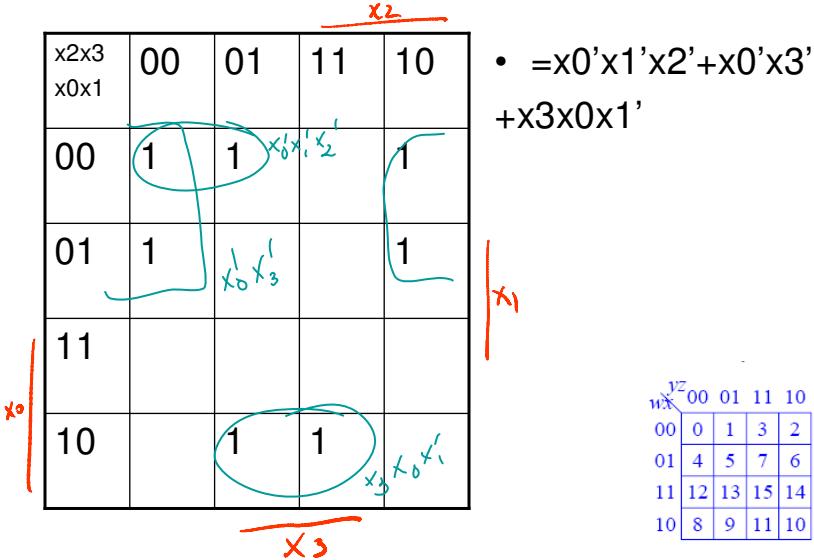
x	y	z	00	01	11	10
0	$x' y' z'$	$x' y' z$				
1	$x y' z'$	$x y' z$				

w	x	y	z
00	0	1	3
01	4	5	7
11	12	13	15
10	8	9	11

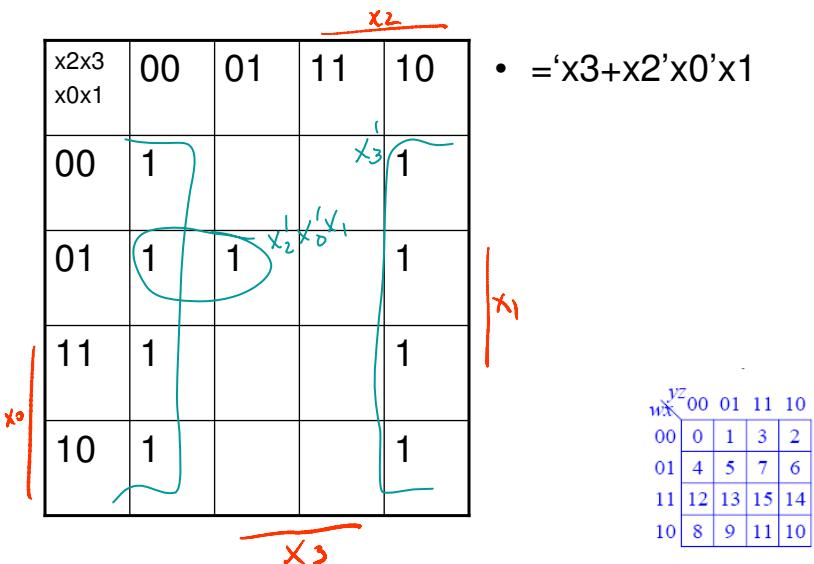
$A = 0$				$A = 1$				
bc^{DE}	00	01	11	10	bc^{DE}	00	01	11
00	0	1	3	2	00	16	17	18
01	4	5	7	6	01	20	21	22
11	12	13	15	14	11	28	29	31
10	8	9	11	10	10	24	25	27

3 Determine as formas mais simples (em soma de produtos) das seguintes funções:

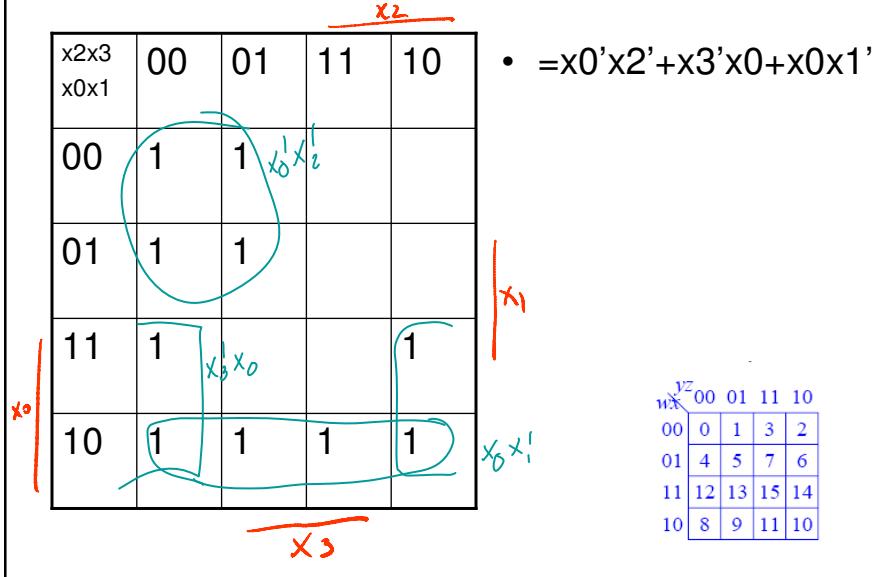
a) $f(x_0, x_1, x_2, x_3) = \sum m(0, 1, 2, 4, 6, 9, 11)$



b) $f(x_0, x_1, x_2, x_3) = \sum m(0, 2, 4, 5, 6, 8, 10, 12, 14)$

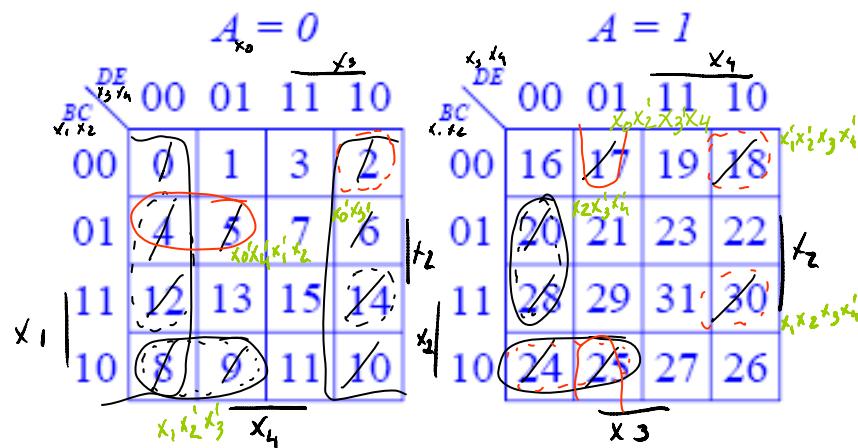


c) $f(x_0, x_1, x_2, x_3) = \sum m(0, 1, 4, 5, 8, 9, 10, 11, 12, 14)$



d) $f(x_0, x_1, x_2, x_3, x_4) = \sum m(0, 2, 4, 5, 6, 8, 9, 10, 12, 14, 17, 18, 20, 24, 25, 28, 30)$

$$= x_0 x_4' + x_1 x_2' x_3' + x_0' x_4' x_1' x_2 + x_0 x_2' x_3' x_4 + x_2 x_3' x_4' + x_1 x_2 x_3 x_4' + x_1' x_2' x_3 x_4'$$



d) $f(x_0, x_1, x_2, x_3, x_4) = \sum m(0, 2, 4, 5, 6, 8, 9, 10, 12, 14, 17, 18, 20, 24, 25, 28, 30)$

$x_0=1$

x_1x_2	00	01	11	10	10	11	01	00
x_3x_4	$x_0'x_4'$						$x_2x_3'x_4'$	
00	0	5	12	8	24	28	20	16
01	1	5	13	9	25	29	$x_1x_2'x_3'$	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

Handwritten annotations in red circles highlight specific groups of minterms:
 - Top row: $x_0'x_4'$ (m0, m4), $x_2x_3'x_4'$ (m28, m30)
 - Second row: $x_0'x_1'x_2'x_3'$ (m5, m13, m25, m29)
 - Third row: $x_1'x_2'x_3'x_4'$ (m11, m27, m31)
 - Bottom row: $x_1'x_2x_3x_4'$ (m10, m26, m30)
 - Middle row: $x_0x_1x_2x_3x_4$ (m2, m6, m14, m20, m22, m18)

Recapitulando

☞ Each maxterm is the complement of its corresponding minterm: $m_i' = M_i$.

	x	y	z	Minterms	Notation	Maxterms	Notation
0	0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
1	0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
2	0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
3	0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
4	1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
5	1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
6	1	1	0	xyz'	m_6	$x' + y' + z$	M_6
7	1	1	1	xyz	m_7	$x' + y' + z'$	M_7

- 4 Determine as formas mais simples (em soma de produtos) das duas funções seguintes.
Compare os resultados obtidos e comente.

a) $f(x_0, x_1, x_2, x_3) = \sum m(0, 1, 4, 5, 12, 13)$

b) $f(x_0, x_1, x_2, x_3) = \prod M(2, 3, 6, 7, 8, 9, 10, 11, 14, 15)$

x_0, x_1	w, x	y, z	00	01	11	10
00	0	1	3	2		
01	4	5	7	6		
11	12	13	15	14		
10	8	9	11	10		

x_0, x_1	w, x	y, z	00	01	11	10
00	0	1	0	0		
01	4	5	0	0		
11	12	13	05	04		
10	0	0	01	00		

$$=x_0'x_2' + x_1x_2'$$

-As funções são iguais

- 5 Por vezes, a saída que corresponde a uma dada combinação das variáveis de entrada não se conhece ou é irrelevante. Esta circunstância pode ajudar na simplificação da função booleana, pois dá liberdade para se criarem novas adjacências no mapa de Karnaugh. Simplifique então as seguintes funções:

a) $f(x_0, x_1, x_2, x_3) = \sum m(4, 5, 6, 8, 9, 10, 13) + \sum d(0, 7, 15)$

b) $f(x_0, x_1, x_2, x_3) = \sum m(1, 3, 5, 7, 9) + \sum d(6, 12, 13)$

c) $f(x_0, x_1, x_2, x_3) = \prod M(1, 2, 3, 11, 12, 14) + \prod D(0, 7, 15)$

x_0, x_1	w, x	y, z	00	01	11	10
00	0	1	3	2		
01	4	5	X	X		
11	12	13	X5	14		
10	X	9	11	10		

$$=x_0'x_2'x_3' + x_0'x_1 \\ + x_1x_3 + x_0x_1'x_3'$$

x_0, x_1	w, x	y, z	00	01	11	10
00	0	1	3	2		
01	4	5	7	X		
11	X	X	15	14		
10	8	9	11	10		

$$=x_2'x_3 + x_0'x_3'$$

x_0, x_1	w, x	y, z	00	01	11	10
00	X	0	0	0		
01	4	5	X	6		
11	Q	13	X5	14		
10	8	9	11	10		

$$=x_0'x_1 + x_0x_2'x_3 \\ + x_0x_1'x_2 + x_0x_1'x_3'$$

- 6 Mostre, através de um exemplo com 4 variáveis independentes, que uma função booleana pode admitir mais que uma forma mínima.

- $= w'x + xy'z + wx'y'$

w\ x	yz	00	01	11	10
00	0	1	3	2	
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

- $= w'x + wy'z + wx'y'$

- Mesma função.

- 7 Preencha mapas de Karnaugh de 4 variáveis de forma a encontrar funções que obejam os seguintes critérios:

- A soma de produtos mínima e o produto de somas mínimo têm o mesmo nº de termos e variáveis.
- A soma de produtos mínima tem menos termos e variáveis que o produto de somas mínima.
- O produto de somas mínima tem menos termos e variáveis que a soma de produtos mínima.

w\ x	yz	00	01	11	10
00	0	1	3	2	
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

$$=wy' + w'y$$

$$=(w+y) \cdot (w+y')$$

w\ x	yz	00	01	11	10
00	0	1	3	2	
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

$$=wy'$$

$$=y+w'$$

w\ x	yz	00	01	11	10
00	0	1	3	2	
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

$$=wy' + x + zw'$$

$$=(w'+x+y) \cdot (w+x+y)$$

- 7 Preencha mapas de Karnaugh de 4 variáveis de forma a encontrar funções que obedeçam aos seguintes critérios:

- A soma de produtos mínima e o produto de somas mínima têm o mesmo nº de termos e variáveis.
- A soma de produtos mínima tem menos termos e variáveis que o produto de somas mínima.
- O produto de somas mínima tem menos termos e variáveis que a soma de produtos mínima.

	w'	x'	y'	z'
wx	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$$\begin{aligned}f_1 &= w'y' + w'y \\f'_1 &= w'y' + yw' \\f''_1 &= (w'+y)(y'+w)\end{aligned}$$

	w'	x'	y'	z'
wx	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$$\begin{aligned}f_2 &= y'z + w'z + x' \Sigma \\f'_2 &= z' + w'xy \\f''_2 &= z \cdot (w+x'+y')\end{aligned}$$

	w'	x'	y'	z'
wx	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$$\begin{aligned}f_3 &= w'y + wxz \\f'_3 &= yw' + y'z' + wz' + \\&\quad z'w \\f''_3 &= (y'+w)(y+z)(z+w)\end{aligned}$$

- 8 Considere uma função booleana de 4 variáveis que assume o valor lógico 1 sempre que um nº ímpar de variáveis independentes vale 1.

- Construa a correspondente tabela de verdade
- Exprime a função nas 1ª e 2ª formas canônicas
- Determine uma expressão mínima na forma de soma de produtos.
- Tente exprimir a função com base na operação "ou exclusivo" (símbolo ' \oplus).

- 1^a
 - Sum(m1,2,4,7,8,11,13,14)
- 2^a
 - Prod(M0,3,5,6,9,10,12,15)

	w'	x'	y'	z'
wx	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$$\begin{aligned}&= w'x'y'z + w'x'yz' + w'xy'z' \\&+ w'xyz + wxy'z + wxyz' + \\&+ wxy'z' + wx'yz \\&= (w+x+y+z) \cdot (w+x+y+z') \cdot (w+x'+y+z) \\&\cdot (w+x'+y+z') \cdot (w'+x+y+z) \\&\cdot (w'+x+y+z') \cdot (w'+x+y'+z) \\&\cdot (w'+x+y+z) \cdot (w'+x+y'+z') \\&\cdot (w'+x+y'+z') \\&= w'x'(y'z + yz') + w'x(y'z + yz) + wx(y'z + yz') + wx'(y'z + yz) \\&= w'x'(y'z + yz') + w'x(y'z + yz) + wx(y'z + yz') + wx'(y'z + yz) \\&= w'x'(y(+z) + w'x(y(+z)) + wx(y(+z) + wx'(y(+z)) \\&= (w(+x)y(+z) + (w(+x)y(+z))' \\&= (w(+x)y(+z) + (w(+x)y(+z))' \\&= (w(+x)y(+z) + (w(+x)y(+z))' \\&= (w(+x)y(+z) + (w(+x)y(+z))'\end{aligned}$$

Aula 4

António Teixeira

- 1 O circuito da figura 1 representa uma implementação possível dum operador lógico conhecido. Determine analiticamente o operador em causa e desenhe o diagrama lógico correspondente à implementação directa (i.e. com portas lógicas *AND*, *OR* e *NOT*). Quais as vantagens e/ou desvantagens desta relativamente ao circuito da figura 1?

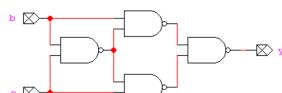
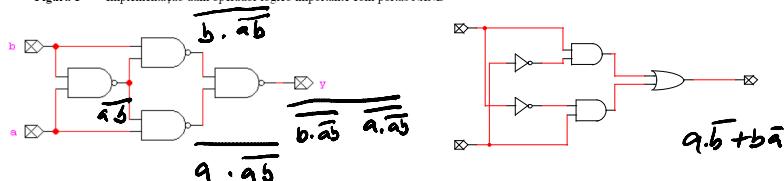


Figura 1 Implementação dum operador lógico importante com portas *NAND*



-Vantagens do NAND

$$\begin{aligned} &= (b \cdot \overline{a\bar{b}}) + (a \cdot \overline{b\bar{a}}) = b \cdot \overline{a\bar{b}} + b \cdot \overline{b\bar{a}} + a \cdot \overline{a\bar{b}} + a \cdot \overline{b\bar{a}} \\ &= b^2 + a\bar{b} = a \oplus b \end{aligned}$$

-1 menor numero de portas

-2 < numero de tipos de portas

-Menor numero de Chip types

-Maior velocidade (Nand e NOR são tipicamente mais rápidas)

- 2 Usando o programa *LogicWorks* construa um subcircuito do tipo *AOI* (*And-Or-Invert*) de acordo com as indicações da figura 2. A seguir, implemente o circuito da figura 1 recorrendo a blocos *AOI* e a inversores adicionais. Simule o seu funcionamento.

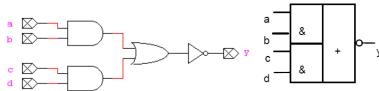
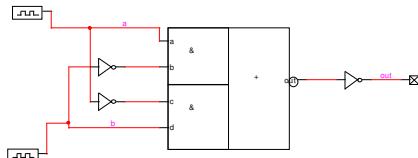
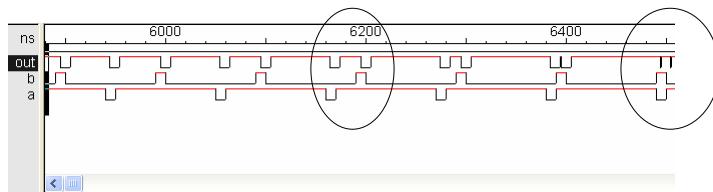


Figura 2 Bloco AOI: Esquema lógico e símbolo alternativo (notação IEEE)



Glitches



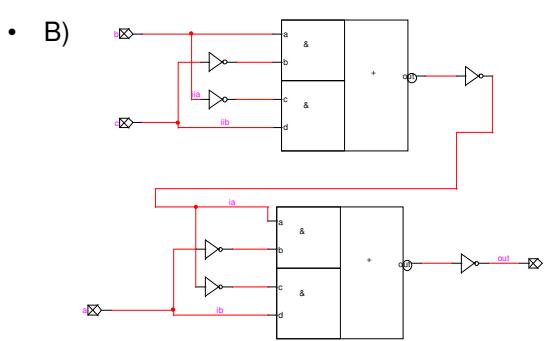
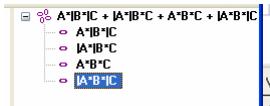
- 3 Uma sala tem 3 portas de acesso, *A*, *B* e *C*. Junto de cada porta está um interruptor capaz de ligar ou desligar a luz. Quando qualquer um dos interruptores é actuado muda o estado da iluminação da sala. Admite-se que a luz está apagada quando todos os interruptores estão desactivados.

- a) Escreva a equação lógica do estado da iluminação, *L* (*L*=1 para luz acesa)
- b) Implemente o circuito com blocos *AOI* e inversores adicionais. Simule o circuito.
- c) Sugira uma nova arquitectura interna para os blocos *AOI* que resulte num circuito mais rápido (i.e. com menor atraso de propagação).

- A)
$$\begin{aligned} & ab'c' + a'b'c + abc + a'bc' \\ & = a(b'c' + bc) + a'(b'c + bc') \\ & = a(b(+c))' + a'(b(+c)) \\ & = a(+)(b(+c)) \end{aligned}$$

Truth table					
Dec	Hex	A	B	C	F _n
0	0	0	0	0	0
1	1	0	0	1	1
2	2	0	1	0	1
3	3	0	1	1	0
4	4	1	0	0	1
5	5	1	0	1	0
6	6	1	1	0	0
7	7	1	1	1	1

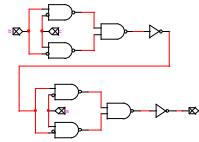
A \ B \ C	00	01	11	10
0	0	1	0	1
1	1	0	1	0



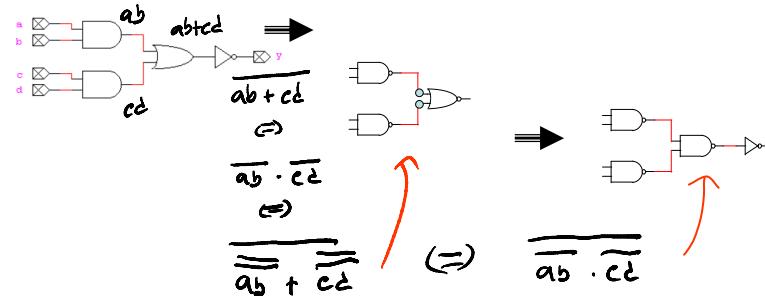
- 3 Uma sala tem 3 portas de acesso, A , B e C . Junto de cada porta está um interruptor capaz de ligar ou desligar a luz. Quando qualquer um dos interruptores é actuado muda o estado da iluminação da sala. Admite-se que a luz está apagada quando todos os interruptores estão desactivados.

- Escreva a equação lógica do estado da iluminação, L ($L=1$ para luz acesa)
- Implemente o circuito com blocos AOI e inversores adicionais. Simule o circuito.
- Sugira uma nova arquitectura interna para os blocos AOI que resulte num circuito mais rápido (i.e. com menor atraso de propagação).

Sol. simplificada



• C) AOI

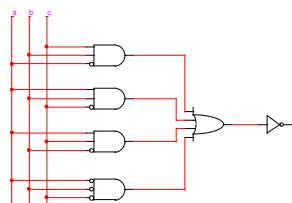


c) Circuito com menos atraso

- $a(+)(b(+))c$
 $= ab'c' + a'b'c + abc + a'bc'$
 $= (a+b'+c') * (a'+b'+c) * (a+b+c) * (a'+b+c')$
 $= (a'bc + abc' + a'b'c' + ab'c)'$

Karnaugh Map

A \ B \ C	00	01	11	10
0	0	1	0	1
1	1	0	1	0



- 4 O circuito da figura 3a tem apenas uma entrada, a , que está a “1” quando o interruptor está aberto e a “0” quando o interruptor está fechado. Considere que todas as portas têm um atraso de propagação de 10 unidades de tempo.

- Admita que em $t=0$ o interruptor foi fechado. Procure prever o comportamento do circuito a partir desse instante, tendo em conta a evolução do sinal a indicada no diagrama temporal da figura 3b; complete esse diagrama para os restantes sinais.
- Construa o circuito no *LogicWorks*, não se esquecendo de especificar o atraso de propagação das portas. Execute a simulação e confronte os resultados com a sua previsão.

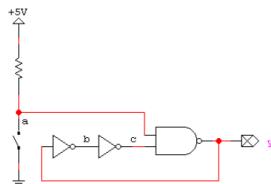
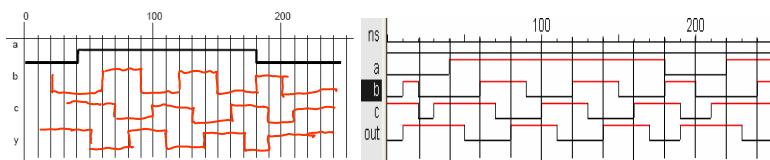


Figura 3a Pulse Shaper



Simulação

5 Considere a seguinte função booleana
 $f(A,B,C,D) = \sum m(1,3,5,7,8,9,12,13)$

- Determine uma implementação mínima a 2 níveis.
- Recorra ao *LogicWorks* para demonstrar, através de um exemplo, que existem *hazards* estáticos (ou seja, condições para a ocorrência de ‘glitches’).
- Determine uma implementação que evite *hazards* estáticos.

Karnaugh Map

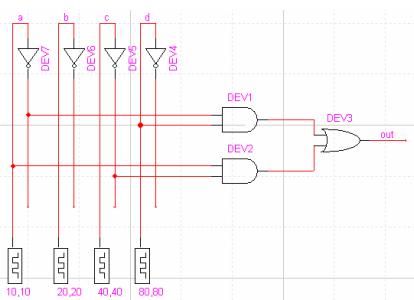
A B \ C D	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	1	0	0
10	1	1	0	0

Minimization result

As is:
 $|A^*D + A^*|C$

Optimized:
 $(|A^*D + A^*|C)$

a)



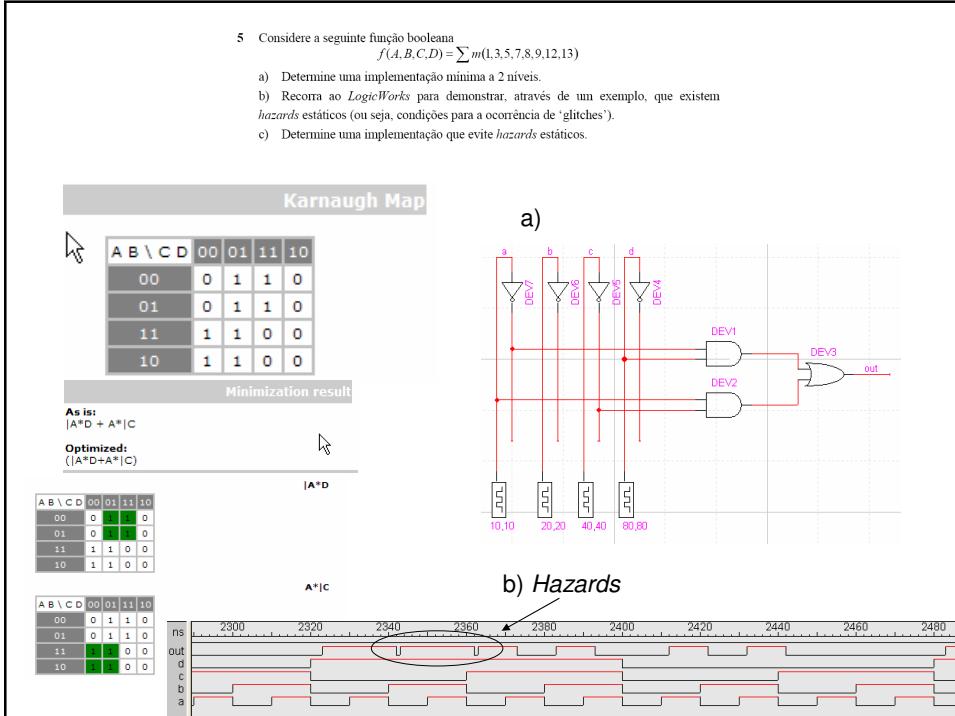
b) Hazards

A B \ C D	00	01	11	10
00	0	1	0	
01	0	1	0	
11	1	1	0	0
10	1	1	0	0

$|A^*D$

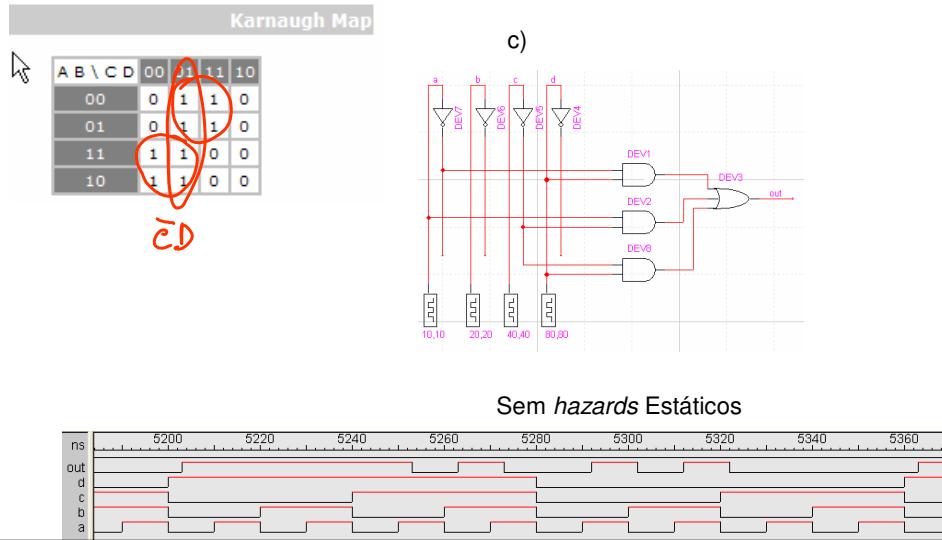
A B \ C D	00	01	11	10
00	0	1	0	
01	0	1	0	
11	1	1	0	0
10	1	1	0	0

$|A^*|C$



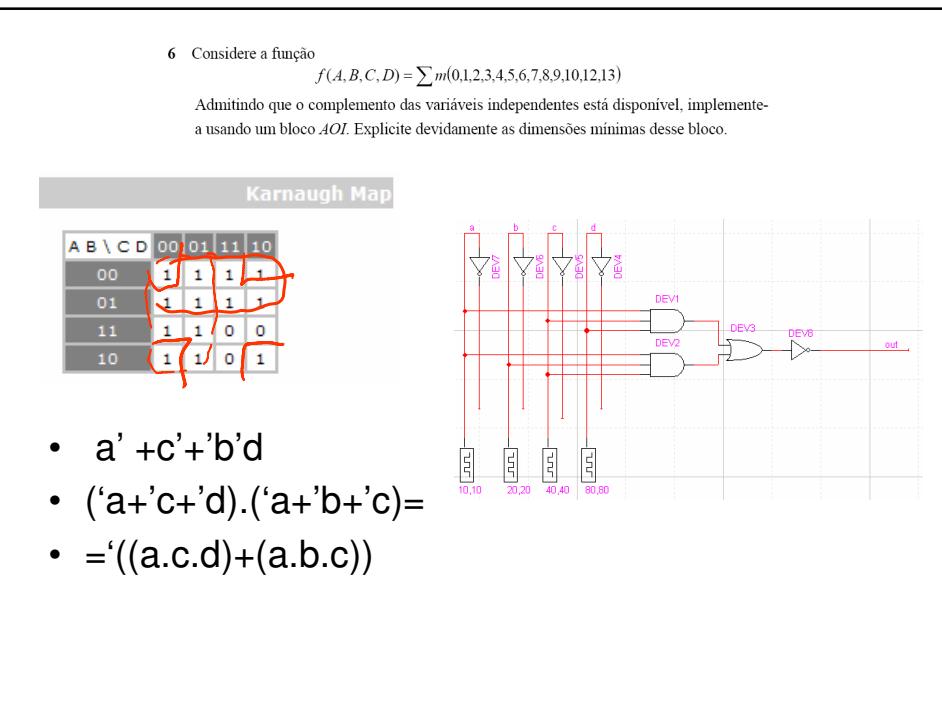
5 Considere a seguinte função booleana
 $f(A,B,C,D) = \sum m(1,3,5,7,8,9,12,13)$

- Determine uma implementação mínima a 2 níveis.
- Recorra ao *LogicWorks* para demonstrar, através de um exemplo, que existem *hazards* estáticos (ou seja, condições para a ocorrência de "glitches").
- Determine uma implementação que evite *hazards* estáticos.



6 Considere a função
 $f(A,B,C,D) = \sum m(0,1,2,3,4,5,6,7,8,9,10,12,13)$

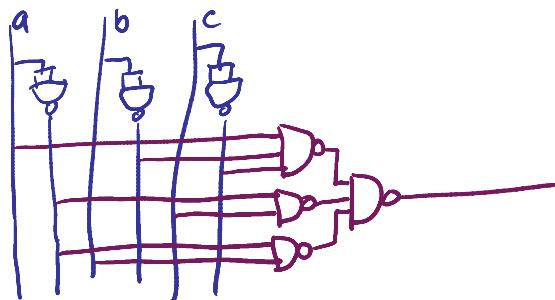
Admitindo que o complemento das variáveis independentes está disponível, implemente-a usando um bloco *AOI*. Explicite devidamente as dimensões mínimas desse bloco.



7 Desenhe o esquema lógico das seguintes expressões com base apenas em portas *NAND*.
 [Assuma que todas as variáveis estão disponíveis também na forma complementada].

a) $y = A\bar{B}\bar{C} + \bar{A}C + \bar{A}B$

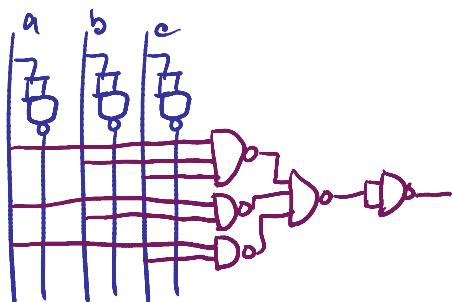
- $\bar{a}\bar{b}\bar{c} + \bar{a}c + \bar{a}b = \overline{\overline{\bar{a}\bar{b}\bar{c}} + \bar{a}c + \bar{a}b}$
- $= \overline{\overline{\bar{a}\bar{b}\bar{c}}} \cdot \overline{\bar{a}c} \cdot \overline{\bar{a}b}$



b) $y = (\bar{A} + \bar{B} + \bar{C})(\bar{A} + \bar{B})(\bar{A} + \bar{C})$

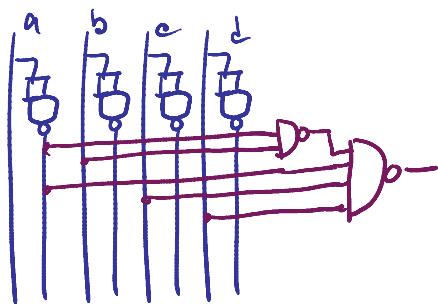
$$(\bar{a} + \bar{b} + \bar{c})(\bar{a} + \bar{b})(\bar{a} + \bar{c}) =$$

$$\overline{(\overline{a} \overline{b} \overline{c})} (\overline{\bar{a} \bar{b}}) (\overline{\bar{a} \bar{c}}) =$$



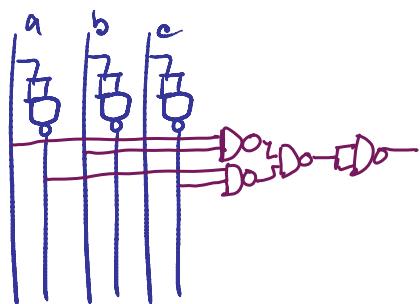
c) $y = \bar{A}B + A + \bar{C} + \bar{D}$

$\bar{a}\bar{b} \cdot \bar{a} \cdot c \cdot b$



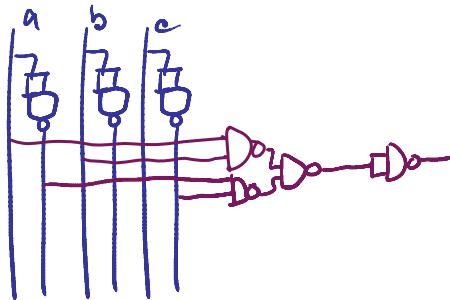
d) $y = (\bar{A}\bar{B})(\bar{A}\bar{C})$

$(\bar{a}\bar{b})(\bar{a}\bar{c})$



e) $y = \overline{AB} + \overline{AC}$

$$\overline{\overline{(ab)}(\bar{a}\bar{c})}$$



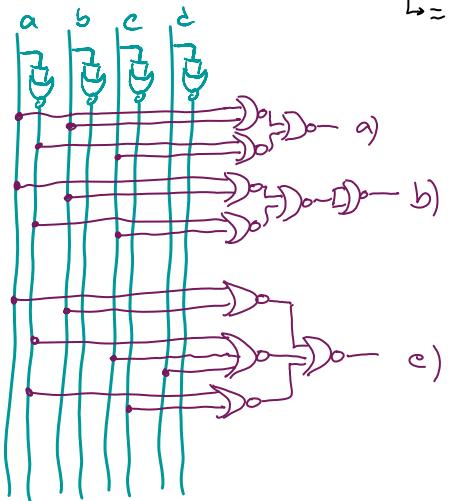
8 Desenhe o esquema lógico das seguintes expressões com base apenas em portas NOR.
[Assuma que todas as variáveis estão disponíveis também na forma complementada].

a) $y = (A+B)(\bar{A}+C) = (\overline{a+b}) + (\bar{a}+\bar{c}) = \overline{(a+b)} + \overline{(\bar{a}+\bar{c})}$

b) $y = (A+B)(\bar{A}+C) = \overline{(a+b)} + \overline{(\bar{a}+\bar{c})}$

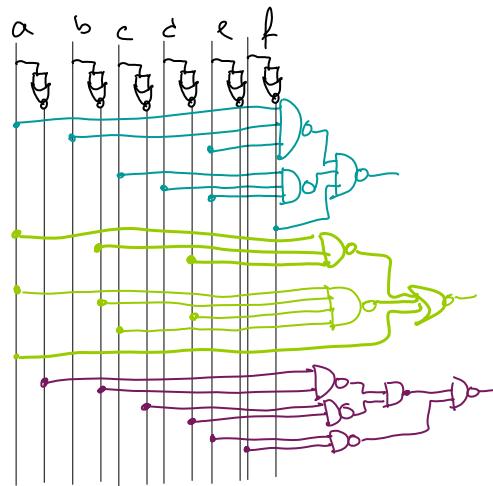
c) $y = (A+B)(\bar{A}+C+D)(\bar{A}+\bar{C})$

$$= \overline{(a+b)} + \overline{(\bar{a}+\bar{c}+\bar{d})} + \overline{(\bar{a}+\bar{c})}$$



- 9 Desenhe o esquema lógico das seguintes expressões usando portas NOR e/ou NAND.
 [Assuma que todas as variáveis estão disponíveis também na forma complementada].

a) $y = (AB + CD)E + F$
 b) $y = A\bar{B}(\bar{B} + C)\bar{D} + \bar{A}$
 c) $y = (A + B)(C + D) + EF$



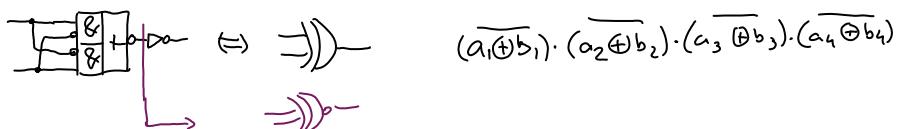
a) $\overline{\overline{abe} + \overline{cd}\bar{d} + \bar{f}}$
 $= \overline{\overline{abe}} \cdot \overline{\overline{cd}\bar{d}} \cdot \bar{f}$

b) $\overline{ab} + (\overline{b} + c)\overline{d} \cdot \bar{a}$

c) $(\overline{a} + \overline{b}) + (\overline{c} + \overline{d}) \cdot \overline{ef}$
 ou
 $\overline{(a \cdot b)} \cdot \overline{(c \cdot d)} \cdot \overline{ef}$

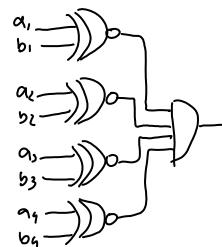
$\overline{ab\bar{d}\bar{d} + ab\bar{d}c + \bar{a}}$
 $\overline{a\bar{b}\bar{d} + a\bar{b}\bar{c}c + \bar{a}}$
 $= \overline{\overline{ab\bar{d}}} \cdot \overline{\overline{a\bar{b}\bar{c}c}} \cdot \bar{a}$

- 10 Projecte um circuito para detectar a igualdade entre duas palavras A e B de 4 bits cada. A saída, Z , será "1" quando $A=B$ (igualdade bit a bit). Implemente o circuito utilizando para tal portas NOR e blocos AOI semelhantes aos do problema 2.



a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4	f
0	0	0	0	0	0	0	0	1
.	-	-	-
0	1	0	0	0	1	0	0	1
0	1	0	0	0	1	0	1	0

~ ~ ~ ~ ~ com flexo !!!



- 11 Pretende-se construir um diagrama temporal que descreva o comportamento do circuito da figura 4. Admita que os atrasos dos inversores, da porta *NAND* e das portas *XOR* são, respectivamente, de 2, 5 e 8 unidades de tempo.

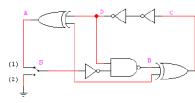
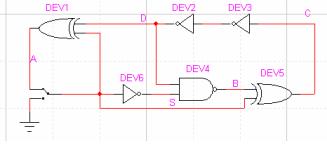
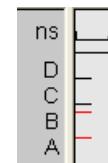
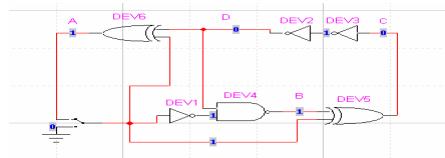


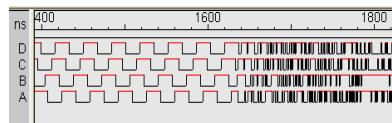
Figura 4



- A) Condições Iniciais não oscilante



- Condições oscilação



- 12 Para as seguintes funções booleanas, sugira implementações livres de *hazards*.

a) $F(A,B,C,D) = \sum m(0,4,5,6,7,9,11,13,14)$

b) $F(A,B,C,D) = (A + B)(\bar{B} + C)$

c) $F(A,B,C,D) = \prod M(0,1,3,5,7,8,9,13,15)$

- a)

Karnaugh Map

A \ B	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	1
10	0	1	1	1

Minimization result

As is:
 $|A^*B + B^*C^*|D + A^*|B^*D + B^*|C^*D + |A^*|C^*|D$

tacd

- b)

Karnaugh Map

A \ B	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	0	1	1
10	1	1	1	1

Minimization result

As is:
 $|B + |D * A + |D * B + C$

C+A

- c)

Karnaugh Map

A \ B	00	01	11	10
00	0	0	0	1
01	1	0	1	1
11	1	0	0	1
10	0	1	1	1

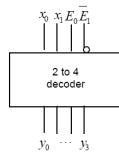
Minimization result

As is:
 $|B + |D * A + |D * B + C$

C+D

Folha 5

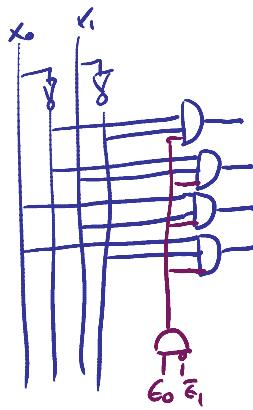
- 1 Projete um descodificador de 2 entradas para 4 saídas. O circuito deverá ter 2 entradas adicionais de validação (*enable*) uma activa a "1" e outra activa a "0".



- $Y_0 = x_0'x_1'$
- $Y_1 = x_0'x_1$
- $Y_2 = x_0x_1'$
- $Y_3 = x_0x_1$

Figura 1 Descodificador de 2 para 4

x0	x1	y0	y1	y2	y3
0	0	1			
0	1		1		
1	0			1	
1	1				1



- 2 Usando o programa *LogicWorks* crie um subcírcuito com o bloco da figura 1. Construa então, a partir desse bloco elementar, um descodificador de 4 entradas para 16 saídas. Justifique as suas opções e simule o funcionamento do circuito.

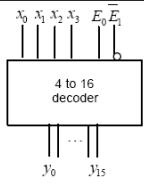
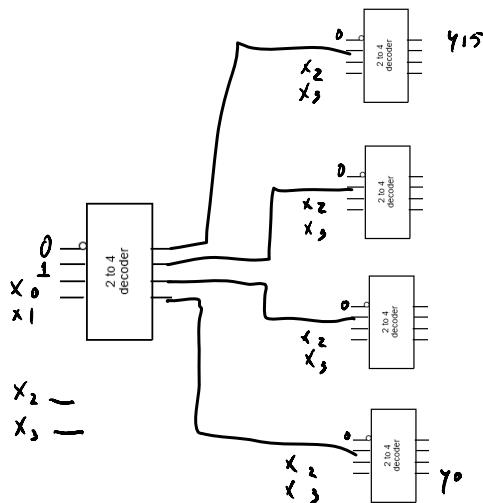


Figura 2 Descodificador de 4 para 16



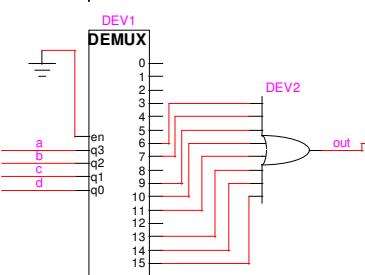
- 3 Considere a seguinte função booleana não necessariamente mínima. Sugira uma implementação baseada em descodificadores de 4 entradas para 16 saídas e portas *OR* adicionais.

$$f(A, B, C, D) = \bar{A}BC + AD + AC$$

$$\begin{aligned} &= \bar{A}bc(d + \bar{d}) + ad(c + \bar{c}) + ac(b + \bar{b}) \\ &= \bar{a}bcd + \bar{a}bcd + adc(b + \bar{b}) + ad\bar{c}(b + \bar{b}) + acb(d + \bar{d}) + ac\bar{c}(d + \bar{d}) \\ &= \bar{a}bcd + \bar{a}bcd + accb + adcb + ad\bar{b} + ad\bar{c}\bar{b} + acbd + adcb + \\ &\quad ac\bar{d}\bar{b} + ac\bar{c}\bar{b} \end{aligned}$$

	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

$\sum m(6, 7, 9-11, 13-15)$



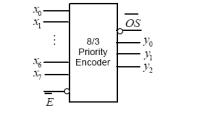
- 4 Projecte um codificador de prioridade para o código binário de 3 bits. Para além das entradas e saídas de dados, o circuito deverá ter uma linha de entrada de *enable*, *E*, e uma linha de saída, *OS*, indicando se a codificação é válida (ambas *active low*). Uma codificação é válida quando a entrada *E* está activa e é activada pelo menos uma das entradas de dados. Verifique por simulação o funcionamento do circuito.

$$y_0 = h_1 + h_3 + h_5 + h_7$$

$$y_1 = h_2 + h_3 + h_6 + h_7$$

$$y_2 = h_4 + h_5 + h_6 + h_7$$

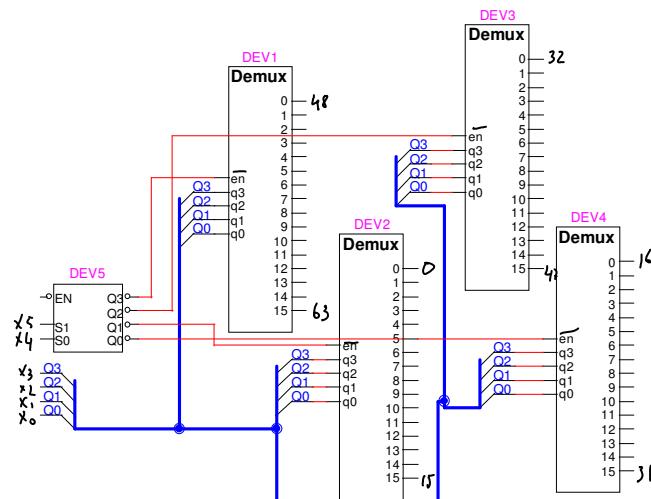
$$OS = \overline{(x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7)} \cdot \overline{E}$$



3 Codificador de prioridade de 8 para 3

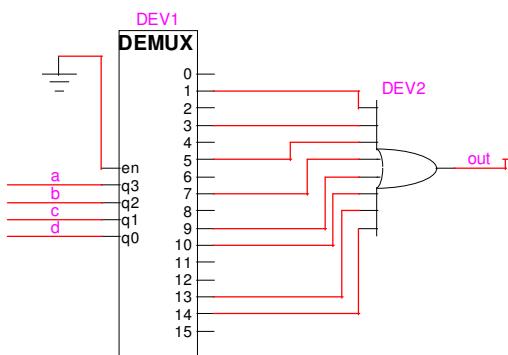
	y_0	y_1	y_2
h_0	0	0	0
h_1	1	0	0
h_2	0	1	0
h_3	1	1	0
h_4	0	0	1
h_5	1	0	1
h_6	0	1	1
h_7	1	1	1

- 5 Mostre como implementaria um descodificador de 6 entradas para 64 saídas usando descodificadores genéricos 2:4 e 4:16.

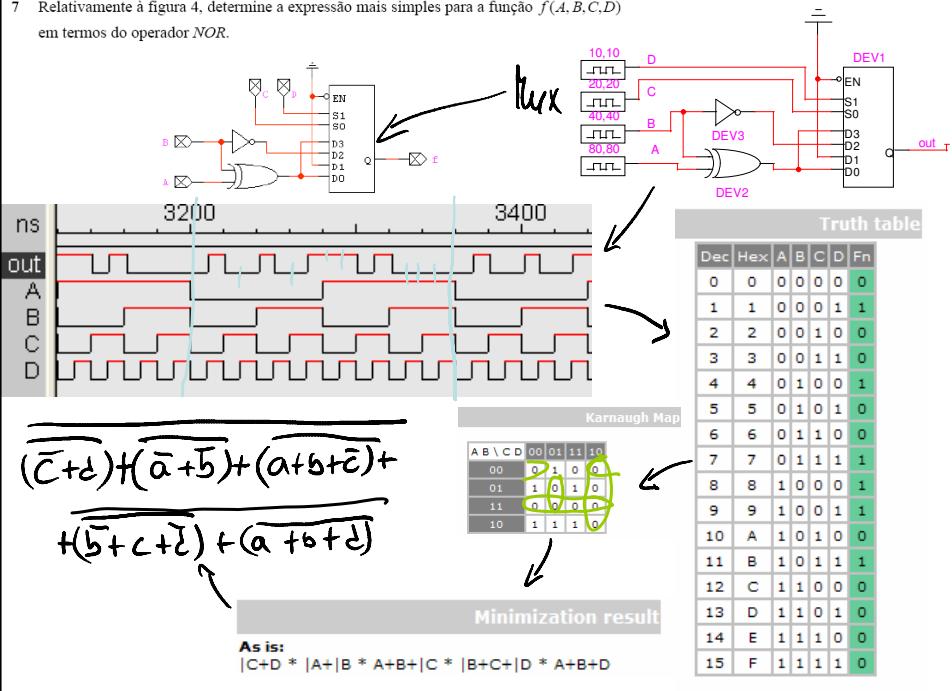


- 6 Implemente a função $f(A, B, C, D) = \overline{A}\overline{B}D + \overline{A}BD + A\overline{C}\overline{D} + AC\overline{D}$ recorrendo a um descodificador 4:16 e a uma porta OR com fan in arbitrário.

	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0



- 7 Relativamente à figura 4, determine a expressão mais simples para a função $f(A, B, C, D)$ em termos do operador NOR.



Folha 6

1 Projete um sistema de multiplexagem de modo a poder selecionar uma de 8 palavras de 4 bits cada. Utilize como blocos elementares os circuitos da Figura 1.

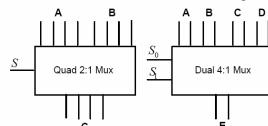
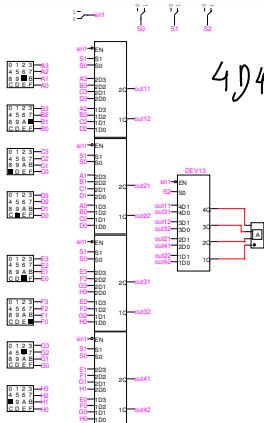
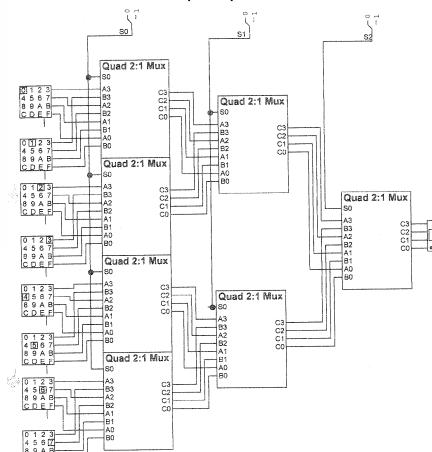


Figura 1 Multiplexores elementares

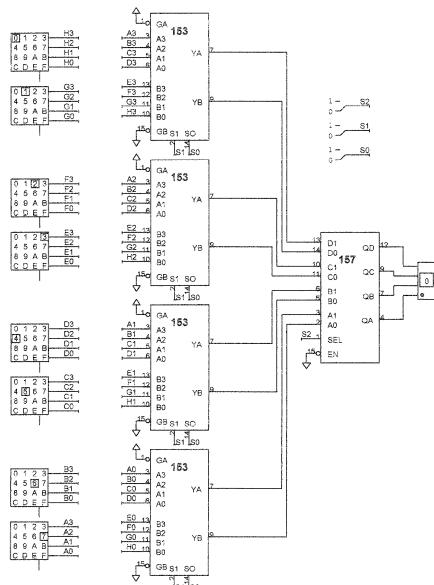
Justifique as suas opções e simule o circuito recorrendo aos componentes correspondentes que integram a biblioteca "7400" no ambiente *LogicWorks*. Alternativamente, pode criar os blocos elementares da figura 4 sob a forma de subcircuitos.



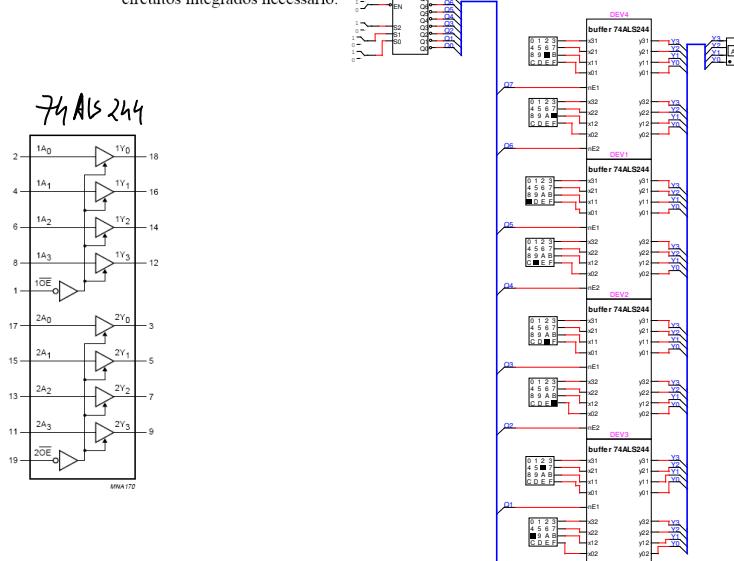
74151 + 74152



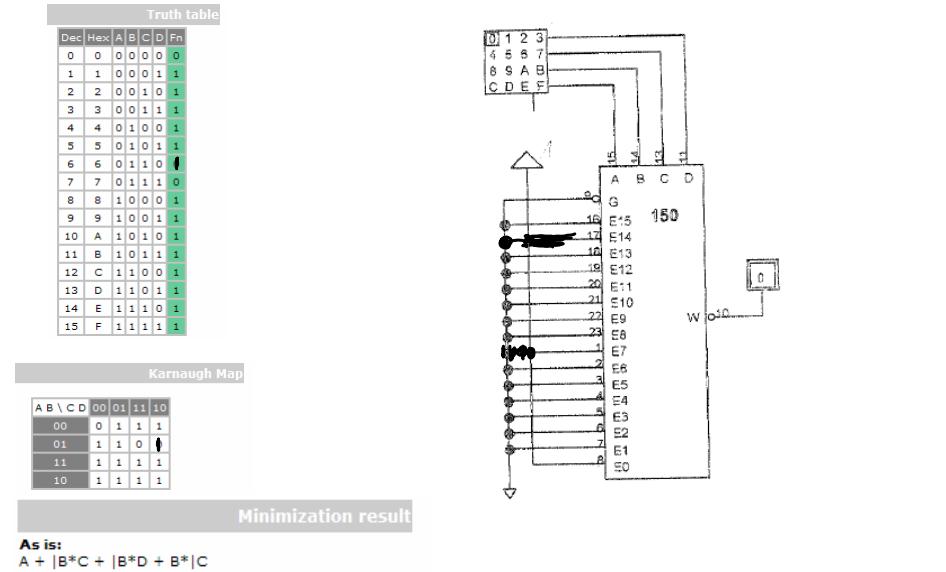
Família ALS



- 2 Repita o problema anterior mas agora considere uma implementação baseada em *buffers* de 3 estados do tipo 74244 e ainda lógica adicional que julgue pertinente utilizar. Compare esta solução com a do problema anterior sob a perspectiva do número de circuitos integrados necessário.



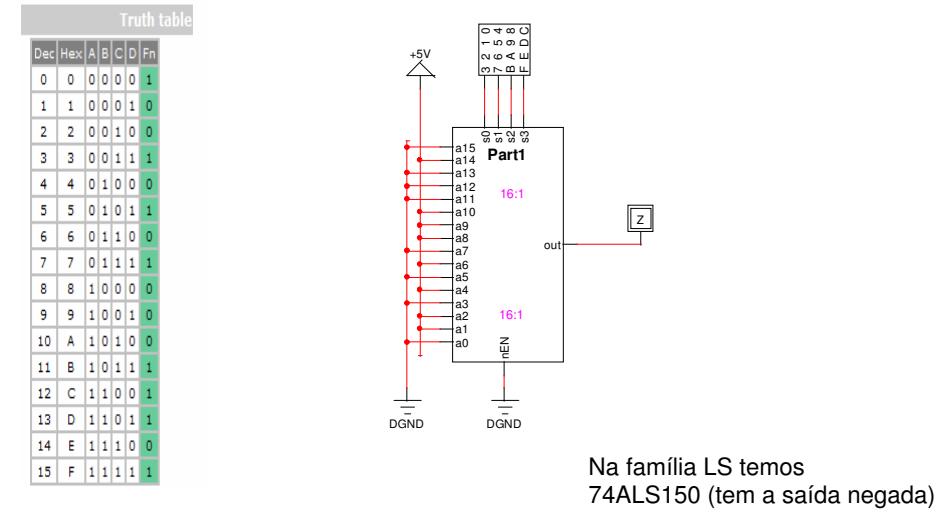
- 3 Implemente a função $F(A, B, C, D) = A + \overline{C}D + B\overline{D} + \overline{B}D + \overline{B}C$, usando apenas um *multiplexer*, as variáveis independentes (não complementadas) e as constantes “0” e “1”.



- 4 Sugira implementações da função $F(A, B, C, D) = \sum m(0, 3, 5, 7, 11, 12, 13, 15)$ baseadas em

- Multiplexer 16:1*
- Multiplexer 8:1*
- Multiplexer 4:1* e lógica elementar adicional.

[Nota: nas alíneas b) e c), admite que dispõe das variáveis também na forma complementada]



Na família LS temos
74ALS151 (tem a saída negada)

4b

Truth table

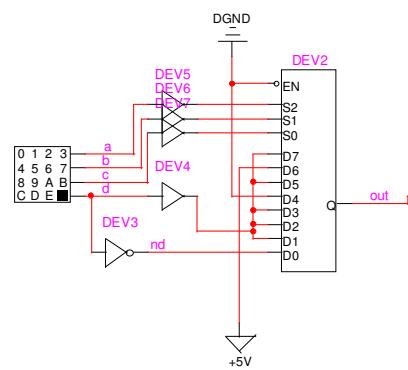
Dec	Hex	A	B	C	D	Fn
0	0	0	0	0	0	1
1	1	0	0	0	1	0
2	2	0	0	1	0	0
3	3	0	0	1	1	1
4	4	0	1	0	0	0
5	5	0	1	0	1	0
6	6	0	1	1	0	0
7	7	0	1	1	1	1
8	8	1	0	0	0	0
9	9	1	0	0	1	0
10	A	1	0	1	0	0
11	B	1	0	1	1	1
12	C	1	1	0	0	1
13	D	1	1	0	1	0
14	E	1	1	1	0	0
15	F	1	1	1	1	1

Truth table

Dec	Hex	A	B	C	Fn
0	0	0	0	0	1
1	1	0	0	1	0
2	2	0	1	0	0
3	3	0	1	1	0
4	4	1	0	0	0
5	5	1	0	1	0
6	6	1	1	0	1
7	7	1	1	1	0

Karnaugh Map

A \ B C	00	01	11	10
0	0'	0	0	0
1	0	0	0	1



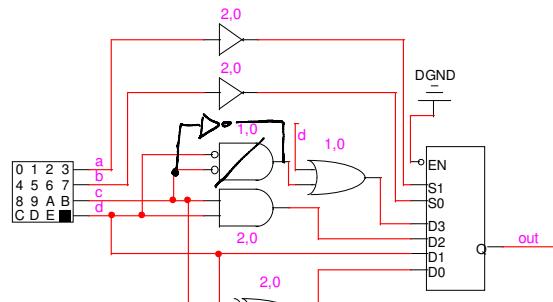
$$D' \rightarrow A'B'C' \quad D \rightarrow C + A'B \quad 0 \rightarrow AB'C' \quad 1 \rightarrow ABC'$$

Na família LS temos
74ALS153 (tem a saída negada)
E é dual

4c

Truth table

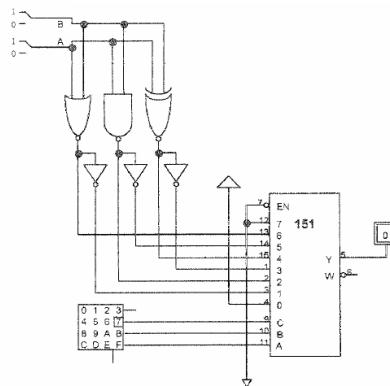
Dec	Hex	A	B	C	D	Fn
0	0	0	0	0	1	-
1	1	0	0	1	0	$C \oplus D$
2	2	0	0	1	0	-
3	3	0	0	1	1	-
4	4	0	1	0	0	-
5	5	0	1	0	1	-
6	6	0	1	1	0	D
7	7	0	1	1	1	-
8	8	1	0	0	0	-
9	9	1	0	0	1	$C \cdot D$
10	A	1	0	1	0	-
11	B	1	0	1	1	-
12	C	1	1	0	1	-
13	D	1	1	0	1	$D + C'D' \Leftrightarrow C' + D$
14	E	1	1	1	0	-
15	F	1	1	1	1	-



- 5 Pretende-se construir um subsistema computacional com 2 entradas de dados, A e B , e 3 entradas de controlo, C_2 , C_1 e C_0 . A saída do circuito, F , obedece à seguinte tabela de verdade:

C_2	C_1	C_0	F
0	0	0	1
0	0	1	$A + B$
0	1	0	$\overline{A} \bullet B$
0	1	1	$A \oplus B$
1	0	0	$\overline{A} \oplus B$
1	0	1	$A \bullet B$
1	1	0	$\overline{A} + B$
1	1	1	0

Implemente o circuito com base num *multiplexer* e lógica elementar adicional.



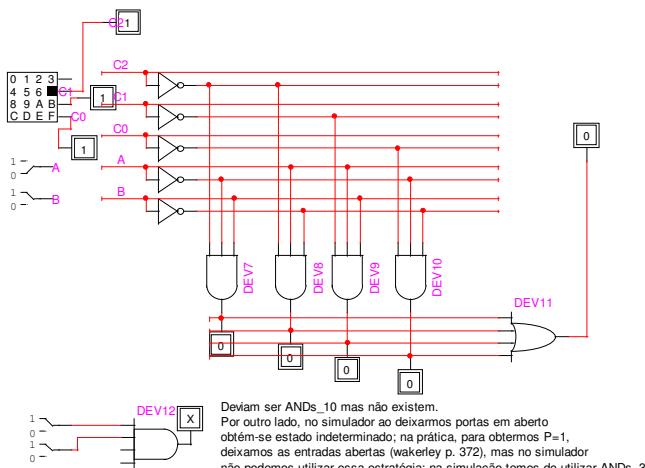
Folha 7

António Teixeira

1)

- a) Pretende-se implementar o circuito com uma *PLA* (*Programmable Logic Array*). Indique as respectivas dimensões mínimas e desenhe uma arquitectura para o dispositivo incluindo o mapa das ligações a efectuar nos planos dos produtos e das somas.

- Fazer a tabela de 5 variaveis... simplificar
- 4 produtos
- PLA- NxM com P product terms
- N entradas, M saídas, P produtos
 - As And têm 2^n inputs
- 4 and 10, 1 or de 4



1)

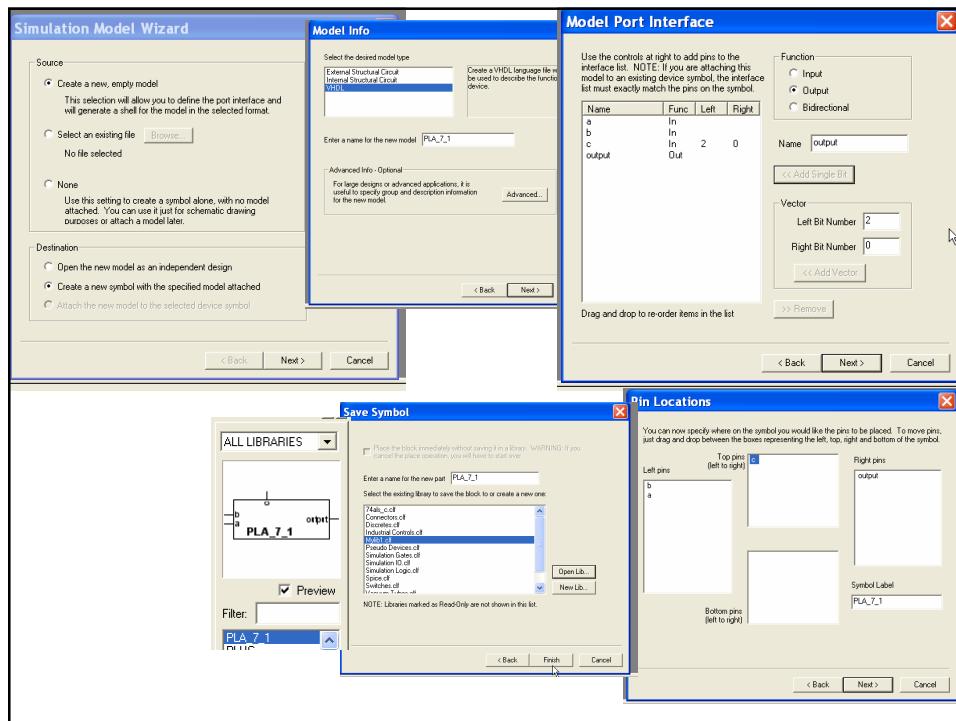
b) Descreva a *PLA* em linguagem *VHDL*. Simule o comportamento do circuito.

- Descrevendo dig circ:

- Schematics
- HDL (hardware desc. Langauge)
 - PALASM, CUPL, ABEL, VHDL
- Usados para descrever equações lógicas implementadas em PLD

- Utilizando VHDL:

- File-> new-> Model Wizard
 - Simulation model wizard
 - Source- create a new, empty model
 - Destination – create a new symbol with the specific model attached
 - Model info
 - Select the desired model type
 - » VHDL
 - Enter a name for the new model
 - » PLA_7_1



Código

```
• library IEEE;
• use IEEE.std_logic_1164.all;

• entity PLA_7_1 is
• port(
•         a      : in      std_logic;
•         b      : in      std_logic;
•         output : out    std_logic;
•         c      : in      std_logic_vector(2 downto 0)
• );
• end PLA_7_1;

• architecture arch1 of PLA_7_1 is
• signal Prod1,Prod2,Prod3,Prod4 : std_logic;
• begin
•     -- Your VHDL code defining the model goes here
•     Prod1 <= not C2 and not A and B;
•     Prod2 <= not C2 and A and not B;
•     Prod3 <= not C1 and A and B;
•     Prod4 <= not C0 and not A and not B;
•     F <= Prod1 or Prod2 or Prod3 or Prod4;
• end arch1;
```

- 2 Pretende-se construir um circuito combinatório que funcione como um transcodificador entre os códigos *AIKEN* e *BCD*. Uma entrada *x* indicará o sentido da codificação. Se for 0, a conversão será de *BCD* para *AIKEN*; se for 1, será em sentido contrário, naturalmente. Para além do código de saída, o circuito deverá contemplar uma saída *z* que detecte a presença à entrada de uma palavra não pertencente ao código. [Nota: o código *AIKEN* é um código decimal binário, ponderado e autocomplementar com pesos 2421].
- a) Implemente e simule o circuito com uma *ROM* (*Read-Only Memory*).

b) Implemente e simule o circuito com uma *PLA*.

- 3 Descreva em *VHDL* um circuito combinatório que faça o deslocamento à direita duma palavra de 8 bits. O número de bits deslocado é dado pelas variáveis N_2, N_1, N_0 , podendo pois variar de 0 a 7. Considere uma entrada adicional CS para definir o tipo de deslocamento: $CS=0$ para deslocamento não circular e $CS=1$ para deslocamento circular.

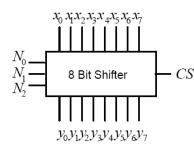


Figura 2 *Shifter* de 8 bits

- 4** Os modernos gravadores digitais de banda magnética, quando usados em modo de *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 a seguinte tabela. Projecte um sistema de transcodificação para ser usado tanto na fase da gravação como na fase da leitura. Uma entrada de controlo W/\bar{R} indica o modo de funcionamento (gravação ou leitura). Baseie a sua solução de acordo com as seguintes condições:

a) Uma só *PLA*

- b) Duas *PLA* e “buffers tri-state” do tipo 74244

B ₃ B ₂ B ₁ B	G ₄ G ₃ G ₂ G ₁
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

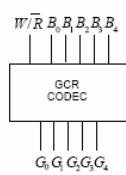
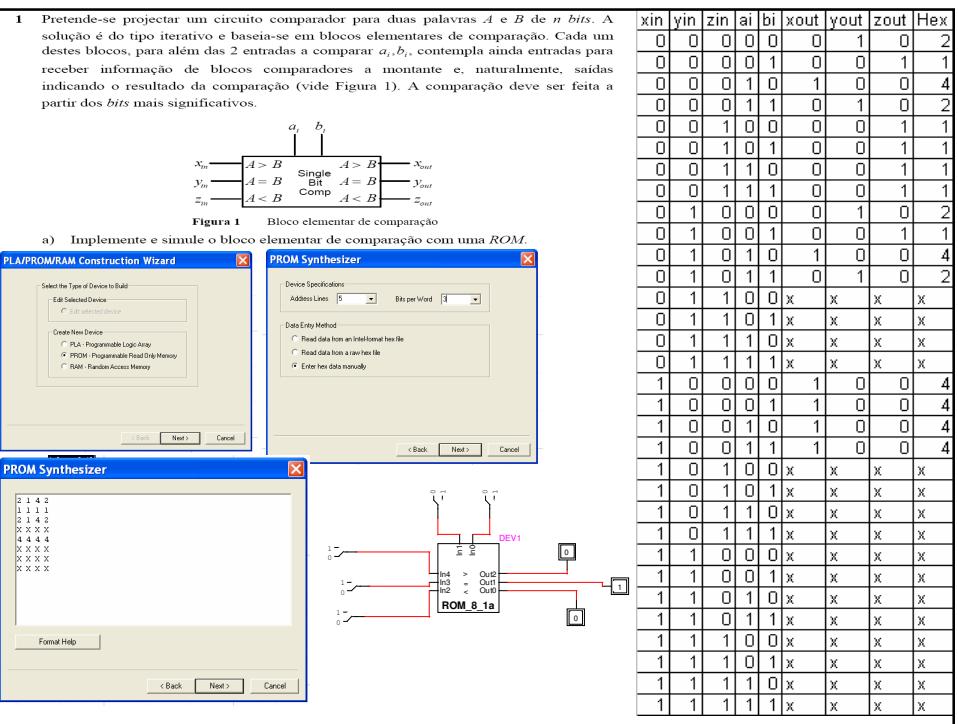


Figura 3 *Group Code Recording Codec*



b) Implemente e simule o bloco elementar de comparação com uma PLA.

• X_{out}

$$= x_{in} + \bar{z}_{in} a_i \bar{b}_i$$

• Y_{out}

$$\bar{x}_{in} \cdot \bar{z}_{in} (\bar{a}_i \bar{b}_i + a_i b_i)$$

• Z_{out}

$$z_{in} + \bar{x}_{in} \bar{a}_i b_i$$

Conclui-se que não depende de y_{in} , como era de prever sendo por isso uma entrada que poderemos não considerar.

AB \ CD	00	01	11	10
00				1
01				
11	x	x	x	x
10				1

AB \ CD	00	01	11	10
00	1	1	1	1
01	x	x	x	x
11	x	x	x	x
10	x	x	x	x

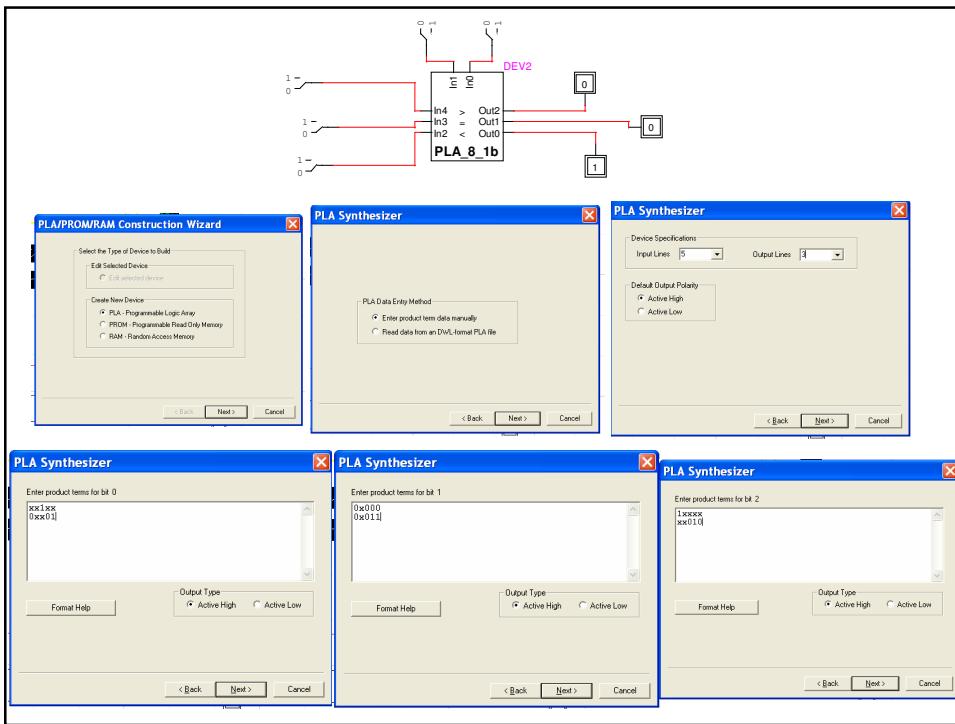
AB \ CD	00	01	11	10
00	1	1	1	1
01	x	x	x	x
11	x	x	x	x
10	x	x	x	x

AB \ CD	00	01	11	10
00				1
01	x	x	x	x
11	x	x	x	x
10	x	x	x	x

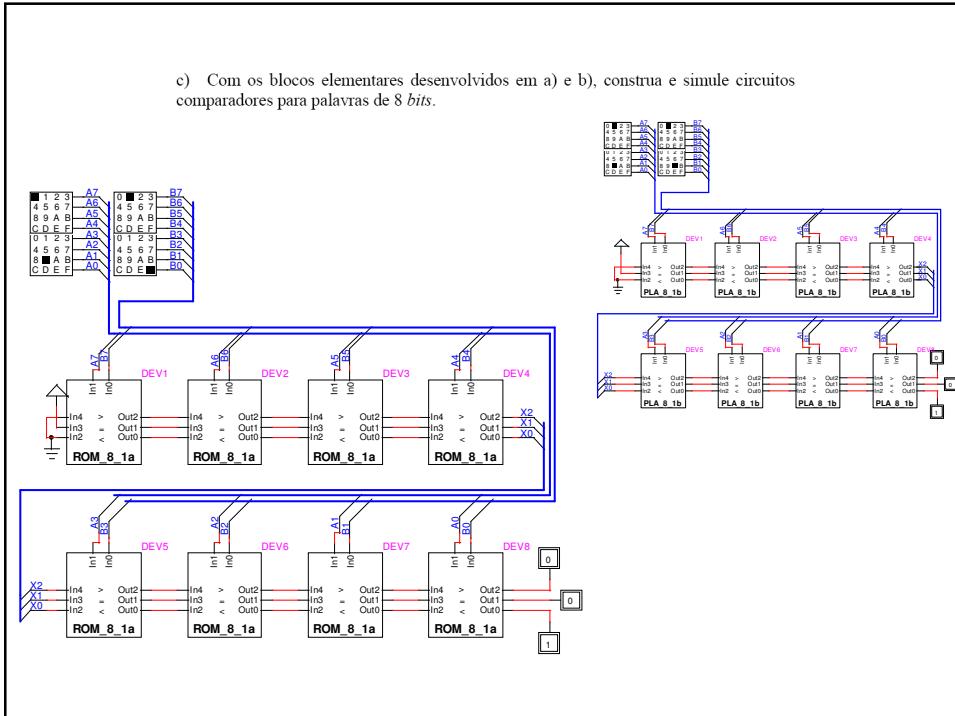
AB \ CD	00	01	11	10
00				1
01	x	x	x	x
11	x	x	x	x
10	x	x	x	x

AB \ CD	00	01	11	10
00				1
01	x	x	x	x
11	x	x	x	x
10	x	x	x	x

$$\begin{aligned} A &= y_{in} \\ B &= z_{in} \\ C &= a_i \\ D &= b_i \end{aligned}$$



c) Com os blocos elementares desenvolvidos em a) e b), construa e simule circuitos comparadores para palavras de 8 bits.



- 2 Projete um circuito somador/subtrator construído a partir de somadores completos de 1 bit interligados em cascata.

- a) Implemente e simule o somador completo de 1 bit com portas lógicas elementares.

$S = x \oplus y \oplus c_{in}$

$C_{out} = x \oplus y \oplus c_{in}$

Initial function

Type:	SOP
Number of variables:	3
Karnaugh Map	
A \ B \ C	00 01 11 10 0 1 0 1 1 2 2 0 1 0 1 3 3 0 1 1 0 4 4 1 0 0 1 5 5 1 0 1 0 6 6 1 1 0 0 7 7 1 1 1 1
$A = x$	$B = y$
$C = c_{in}$	

Truth table

Dec	Hex	A	B	C	Fn
0	0	0	0	0	0
1	1	0	0	1	1
2	2	0	1	0	1
3	3	0	1	1	0
4	4	1	0	0	1
5	5	1	0	1	0
6	6	1	1	0	0
7	7	1	1	1	1

Set's form

One(1): 1,2,4,7
Zero(0): 0,3,5,6
Don't care(*):

Minimization result

As is: $A^x B^y C + A^x B^z C + A^y B^z C + A^x B^y C$
Optimized: $B^x(A @ C) + B^y(A @ C)$

$S = a \oplus b \oplus c = x \oplus y \oplus c_{in}$

Initial function

Type:	SOP
Number of variables:	3
Karnaugh Map	
A \ B \ C	00 01 11 10 0 0 0 1 0 1 0 0 1 1 2 2 0 1 0 3 3 0 1 1 1 4 4 1 0 0 0 5 5 1 0 1 1 6 6 1 1 0 0 7 7 1 1 1 1
$A = x$	$B = y$
$C = c_{out}$	

Truth table

Dec	Hex	A	B	C	Fn
0	0	0	0	0	0
1	1	0	0	1	0
2	2	0	1	0	0
3	3	0	1	1	1
4	4	1	0	0	0
5	5	1	0	1	1
6	6	1	1	0	1
7	7	1	1	1	1

Set's form

One(1): 3,5,7
Zero(0): 0,2,4
Don't care(*):

Minimization result

As is: $A^x B^y C + B^x C + A^y C + A^x C$
Optimized: $B^x(A + C) + A^y C$

$C_{out} = x \oplus y \oplus c_{in}$

Diagram

$D = b_{out} + b_{in}$

$D_{out} = b_{out} + b_{in}$

Initial function

Type:	SOP
Number of variables:	3
Karnaugh Map	
A \ B \ C	00 01 11 10 0 0 0 1 0 1 0 0 1 1 2 2 0 1 0 3 3 0 1 1 0 4 4 1 0 0 1 5 5 1 0 1 0 6 6 1 1 0 0 7 7 1 1 1 1
$A = x$	$B = y$
$C = b_{in}$	

Truth table

Dec	Hex	A	B	C	Fn
0	0	0	0	0	0
1	1	0	0	1	1
2	2	0	1	0	1
3	3	0	1	1	0
4	4	1	0	0	1
5	5	1	0	1	0
6	6	1	1	0	0
7	7	1	1	1	1

Set's form

One(1): 1,2,4,7
Zero(0): 0,3,5,6
Don't care(*):

Minimization result

As is: $A^x B^y C + |A^x|B^y C + A^x B^z C + |A^x|B^z C$
Optimized: $|B^x(A @ C) + B^y(A @ C)$

$D = a \oplus b \oplus c = x \oplus y \oplus b_{in}$

$D_{out} = x \oplus y \oplus b_{in}$

Initial function

Type:	SOP
Number of variables:	3
Karnaugh Map	
A \ B \ C	00 01 11 10 0 0 1 0 0 1 0 0 1 1 2 2 0 1 0 3 3 0 1 1 1 4 4 1 0 0 0 5 5 1 0 1 0 6 6 1 1 0 0 7 7 1 1 1 1
$A = x$	$B = y$
$C = b_{out}$	

Truth table

Dec	Hex	A	B	C	Fn
0	0	0	0	0	0
1	1	0	0	1	1
2	2	0	1	0	1
3	3	0	1	1	1
4	4	1	0	0	0
5	5	1	0	1	0
6	6	1	1	0	0
7	7	1	1	1	1

Set's form

One(1): 1-3,7
Zero(0): 0,4-6
Don't care(*):

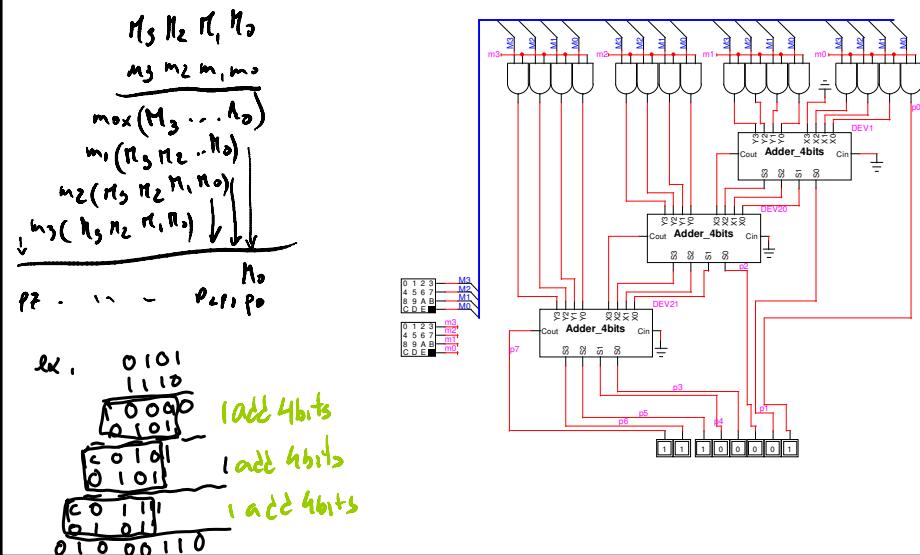
Minimization result

As is: $B^x C + |A^x| B + |A^x| C$
Optimized: $B^x(C + |A|) + A^x C$

$D_{out} = x \bar{y} + \bar{x} b_{in} + y b_{in}$

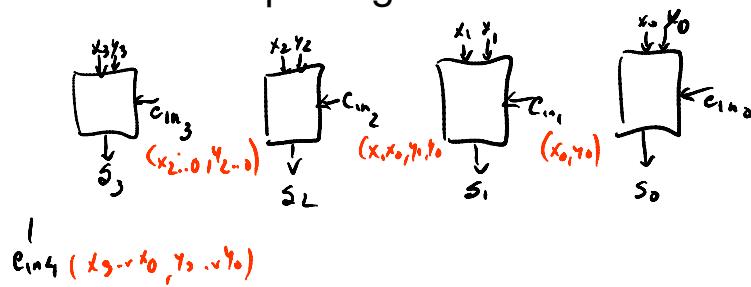
Diagram

3 Construa um circuito multiplicador para palavras de 4 bits baseado em somadores completos de 1 bit e portas lógicas AND.

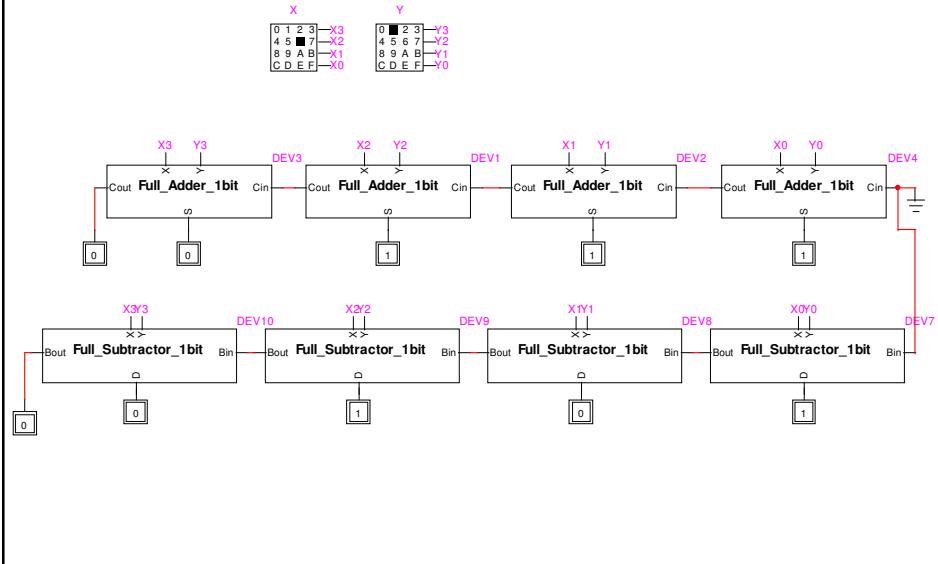


4 Descreva em VHDL e simule um circuito somador de 4 bits que incorpore a lógica carry lookahead.

- 4 Adders, mas os Carry out são calculados por logica externa.

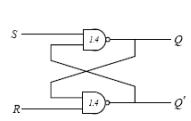


b) Para o bloco elementar desenvolvido em a) construa e simule um circuito somador/subtrator para palavras de 4 bits.



Folha 9

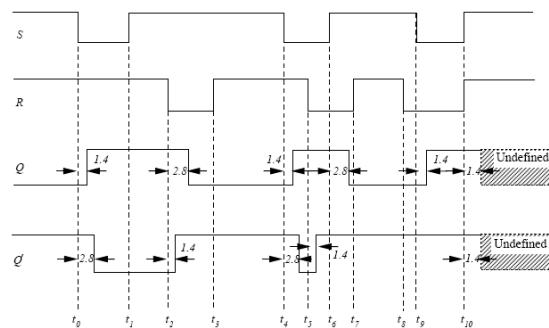
Nand type latch



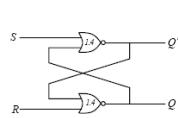
(a) Logic schematic

S	R	Q	$Q \text{ (next)}$	$Q' \text{ (next)}$
0	0	X	1	1
0	1	X	1	0
1	0	X	0	1
1	1	0	0	1
1	1	1	1	0

(c) Truth table



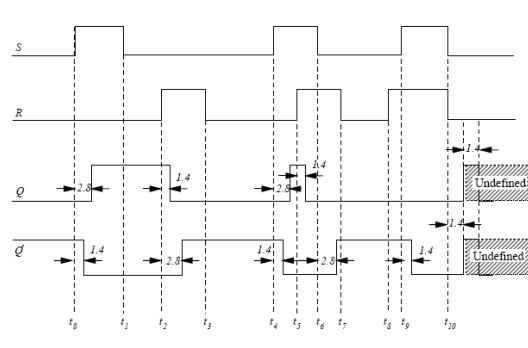
Nor type latch



(a) Logic schematic

S	R	Q	$Q \text{ (next)}$	$Q' \text{ (next)}$
0	0	0	0	1
0	0	1	1	0
0	1	X	0	1
1	0	X	1	0
1	1	X	0	0

(c) Truth table



(b) Timing diagram

SR latch with CLK

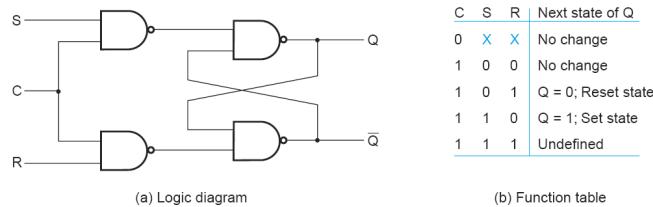


Fig. 4-7 SR Latch with Control Input

SR Latch, gated Not Type

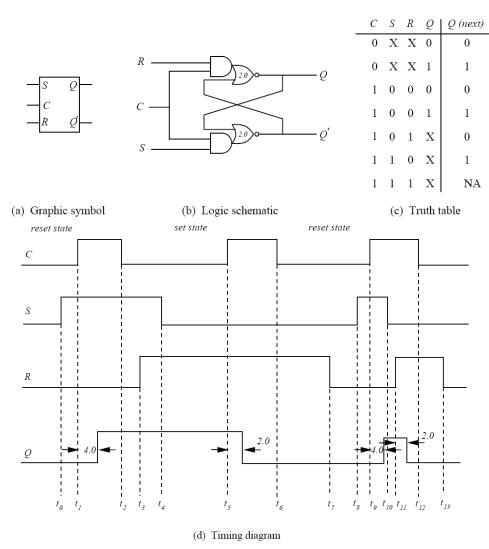
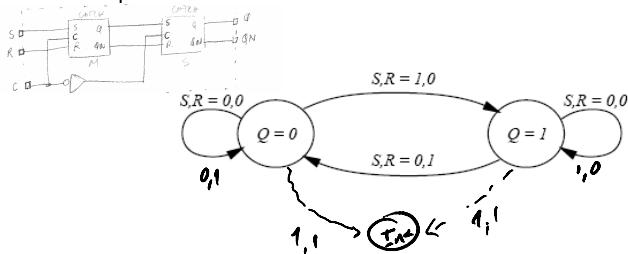


Figure 5: Gated SR latch (NOR-type) [Gajski].

1 Elabore as tabelas de estados, de transições e de excitação para os flip-flops dos tipos SR, JK, D e T. Desenhe os respectivos diagramas de estados.

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table																																			
SR		<table border="1"> <thead> <tr> <th>S</th> <th>R</th> <th>$Q_{(next)}$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>NA</td> </tr> </tbody> </table>	S	R	$Q_{(next)}$	0	0	Q	0	1	0	1	0	1	1	1	NA	$Q_{(next)} = S + R'Q$ $SR = 0$	<table border="1"> <thead> <tr> <th>Q</th> <th>$Q_{(next)}$</th> <th>S</th> <th>R</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>X</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>0</td> </tr> </tbody> </table>	Q	$Q_{(next)}$	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0
S	R	$Q_{(next)}$																																					
0	0	Q																																					
0	1	0																																					
1	0	1																																					
1	1	NA																																					
Q	$Q_{(next)}$	S	R																																				
0	0	0	X																																				
0	1	1	0																																				
1	0	0	1																																				
1	1	X	0																																				



Tab. Trans

Q	S	R	Q^*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Ind
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Ind

Tab. estados

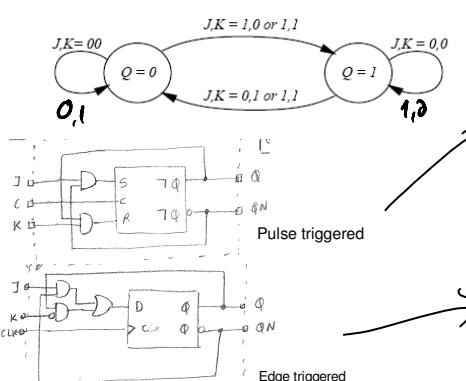
S	R	C	Q	Q_n
x	X	0	Qa	Qna
0	0		Qa	Qna
0	1		0	1
1	0		1	0
1	1		Ind	Ind

Tab. De estados: qual o efeito de cada combinação de entradas?

Tab. De transição: que transição de estado provoca cada comb de entradas?

Tab de excitação: como excitar o FF para obter cada transição possível?

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table																																			
JK		<table border="1"> <thead> <tr> <th>J</th> <th>K</th> <th>$Q_{(next)}$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Q'</td> </tr> </tbody> </table>	J	K	$Q_{(next)}$	0	0	Q	0	1	0	1	0	1	1	1	Q'	$Q_{(next)} = JQ' + K'Q$	<table border="1"> <thead> <tr> <th>Q</th> <th>$Q_{(next)}$</th> <th>J</th> <th>K</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>X</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>X</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>0</td> </tr> </tbody> </table>	Q	$Q_{(next)}$	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0
J	K	$Q_{(next)}$																																					
0	0	Q																																					
0	1	0																																					
1	0	1																																					
1	1	Q'																																					
Q	$Q_{(next)}$	J	K																																				
0	0	0	X																																				
0	1	1	X																																				
1	0	X	1																																				
1	1	X	0																																				



Tab. Estados

S	R	C	Q	Q_n
x	X	0	Qa	Qna
0	0		Qa	Qna
0	1		0	1
1	0		1	0
1	1		Qna	Qa

Tab. Trans

Q	S	R	Q^*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table
D		$\begin{array}{ c c } \hline D & Q \text{ (next)} \\ \hline 0 & 0 \\ 1 & 1 \\ \hline \end{array}$	$Q \text{ (next)} = D$	$\begin{array}{ c c c } \hline Q & Q \text{ (next)} & D \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ \hline \end{array}$

Tab. estados

Tab. Trans

D	Clk	Q	Qn
x	0,1	Qa	QNa
0		0	1
1		1	0

Q	D	Q*
0	0	0
0	1	1
0	0	0
1	1	1

reset state set state reset state

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table
T		$\begin{array}{ c c } \hline E \ T & Q \text{ (next)} \\ \hline 0 & Q \\ 1 & Q' \\ \hline \end{array}$	$E \bar{T} + \bar{E}' Q$ $Q \text{ (next)} = TQ' + \bar{T}Q$	$\begin{array}{ c c c } \hline Q & Q \text{ (next)} & T \bar{E} \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ \hline \end{array}$

Toggle w/o enable

Tab. Trans

Q	E T	Q*
0	0	0
0	1	1
1	0	1
1	1	0

Tab. estados

E T	Clk	Q	Qn
0		Qa	QNa
1		QNa	Qa

Tab. estdos

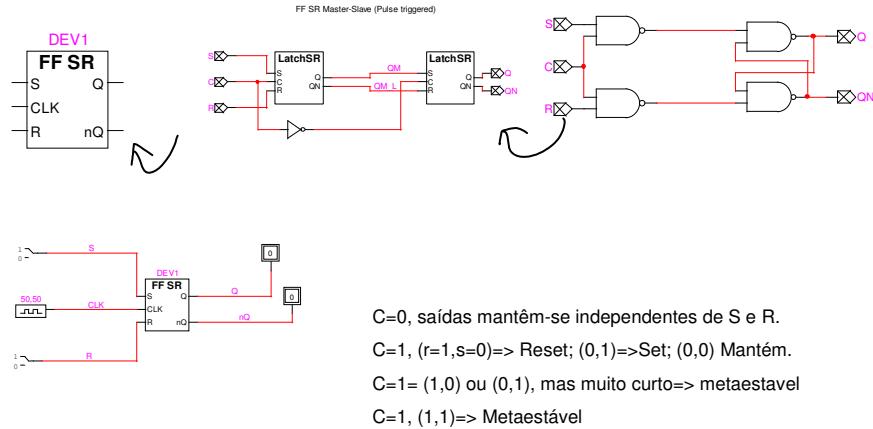
E T	Q	Qn
0,1	Qa	QNa
	QNa	Qa

$\Phi^* = \bar{Q}$ | *lógica de característica*

Tab. Trans

Q	Q*
0	1
1	0

2 Construa um flip-flop JK na estrutura Master-Slave. Use subcircuitos para evidenciar os blocos que a compõem. Teste o seu funcionamento.

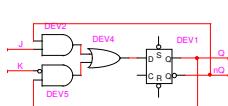


3 Realize um flip-flop JK a partir de:

a) um flip-flop do tipo D.

b) um flip-flop SR.

a) JK edge triggered



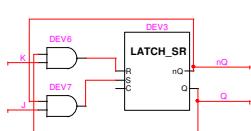
Q	J	K	D	Q*
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Q	Q(next)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$$Q(\text{next}) = JQ' + K'Q$$

$$= D$$

b) JK pulse triggered



Q	J	K	S	R	Q*
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	0	1	1
1	1	1	0	0	0

Q	Q(next)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$$S = JQ'$$

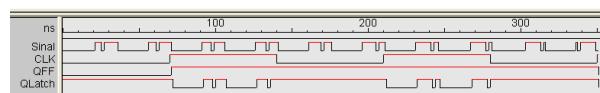
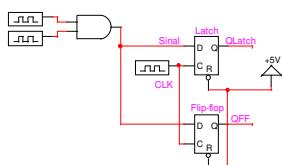
$$R = KQ$$

- 4 Recorra ao simulador *DesignWorks* para obter diagramas temporais que ilustrem as diferenças entre dispositivos de memória *clocked latch* e *edged triggered flip-flop*.

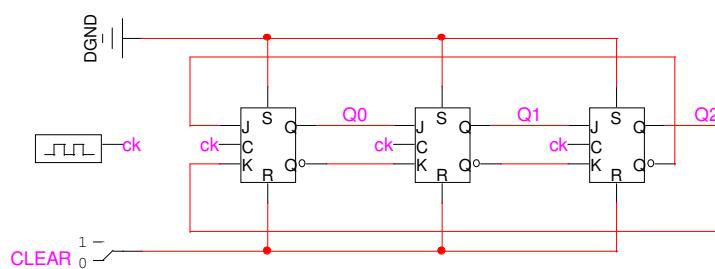
Um FF 'edge-triggered' só pode alterar o valor da sua saída nos instantes em que o sinal CLK sofre uma transição (flanco, ou 'edge').

A latch, se for habilitada por um sinal de relogio (*clocked latch*), fica aberta durante a fração do período de CLK em que este está activo; nesse intervalo, a saída segue a entrada e por isso pode mudar em qualquer instante (anteriormente).

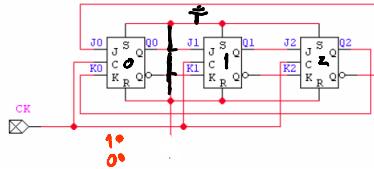
→ ver esquema de teste e correspondente diagrama temporal construídos no DesignWorks (página seguinte)



5- analise o seguinte circuito



5 Analise o circuito seguinte. Desenhe e verifique o seu diagrama de estados.



$$\begin{array}{l} J_0 = Q_2' \\ K_0 = Q_2 \\ J_1 = Q_0 \\ K_1 = Q_0' \\ J_2 = Q_1 \\ K_2 = Q_1' \end{array}$$

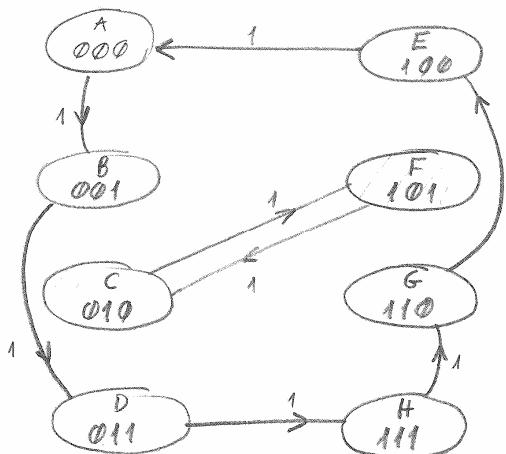
- 1 determinar as equações de excitação
- 2 inserir as eq de excitação na eq caract do FF, para obter as eq de transição
 - $Q^* = JQ' + K'Q$
 - Substituindo:
 - $Q_2^* = J_2 Q_2' + K_2' Q_2 = Q_1 Q_2' + Q_1' Q_2 = Q_1$
 - $Q_1^* = Q_0$
 - $Q_0^* = Q_2'$
- 3 Tabela de transições.

N	Q2	Q1	Q0	Q2*	Q1*	Q0*	N
A	0	0	0	0	0	1	B
B	0	0	1	0	1	1	D
C	0	1	0	1	0	1	F
D	0	1	1	1	1	1	H
E	1	0	0	0	0	0	A
F	1	0	1	0	1	0	C
G	1	1	0	1	0	0	E
H	1	1	1	1	1	0	G

Estados Combinações das eq de transições

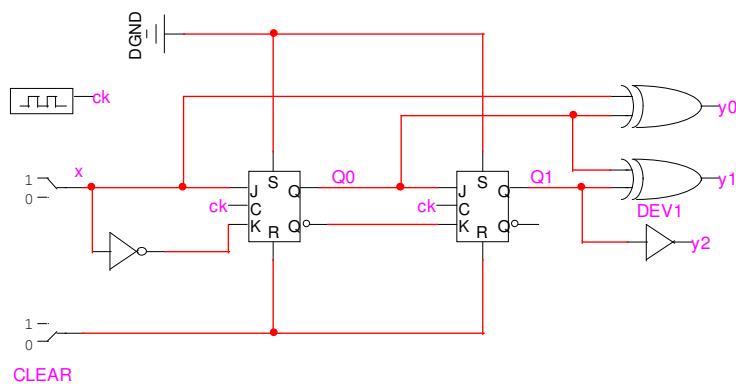
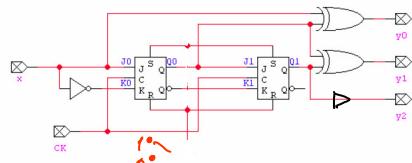
- 4 determinar as eq de saída.
 - Neste caso não temos saídas explicitamente. Consideramos que as saídas são as próprias variáveis de estado, Q2, Q1, Q0. não há eq a determinar
- 5 adicionar os valores das saídas à tabela de transições, para cada estado (m. Moore) ou estado/combinações das entradas (m. Mealy), para criar uma tabela de transiões/saídas. Não aplicável neste caso, pois as saídas são as próprias variáveis de estado
- 6 atribuir nomes aos estados e construir , a partir da tabela de transiões/saídas, uma tabela de estados/saídas, feito na própria tabela
- 7 Desenhar um diagrama de estados a partir da tabela de estados/saídas.

Diagrama de estados



Notas: o estado inicial é determinado pelas entradas S e R assíncronas. Se for A, então os estados C e F nunca serão visitados. Admitindo que a sequência pretendida é efectivamente ABDHGEA, se por ruido ou outro meio este circuito entrar no CF não vai sair destes de forma espontânea., sol. Esquema de Pre-set.

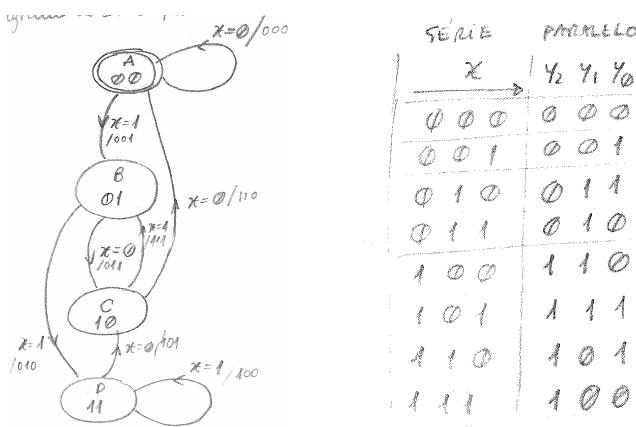
- 6 Estabeleça os diagramas de estados e de saídas do circuito seguinte. Mostre que se trata de um conversor série-paralelo entre códigos binário e de Gray para palavras de 3 bits.



- 1 eq excitação
 - $J_0 = x$
 - $K_0 = x'$
 - $J_1 = Q_0$
 - $K_1 = Q_0'$
- 2 eq transição.
 - $Q_1^* = J_1 Q_1' + K_1' Q_1 = Q_0$
 - $Q_0^* = J_0 Q_0' + K_0' Q_0 = x$
- 3 tabela de transições

x	Q1	Q0	N	Q1*	Q0*	N	Y2	Y1	Y0
0	0	0	A	0	0	A	0	0	0
0	0	1	B	1	0	C	0	1	1
0	1	0	C	0	0	A	1	1	0
0	1	1	D	1	0	C	1	0	1
1	0	0	A	0	1	B	0	0	1
1	0	1	B	1	1	D	0	1	0
1	1	0	C	0	1	B	1	1	1
1	1	1	D	1	1	D	1	0	0

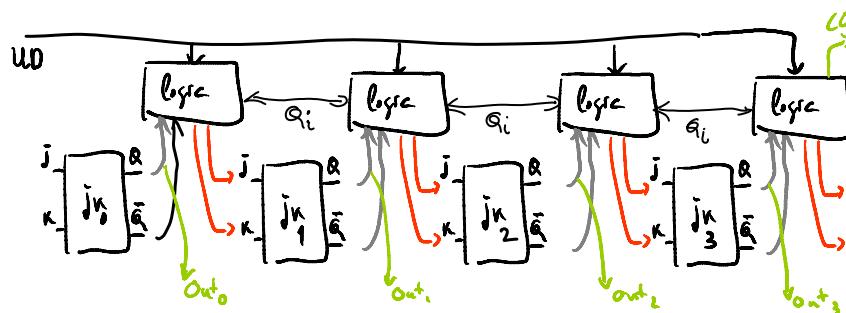
- 4 Eq de saída
 - $Y_0 = J_0 (+) Q_0 = x (+) Q_0$
 - $Y_1 = Q_1 (+) Q_0$
 - $Y_2 = Q_1$
 - Maquina de Mealy pois as saídas dependem explicitamente das entradas.
- 5 referido na tabela de transições/saídas
- 6 atribuídos na tabela de transições/estados/saídas
- 7 Diagrama de estados/saídas



Folha 10

Antonio Teixeira

1 Projecte um contador decimal do tipo *up-down*, sendo a contagem feita no código Aiken (2421). Use flip-flops JK. Considere contagem circular: '9' é seguido de '0' em contagem ascendente (modo *up*) e vice-versa em contagem descendente (modo *down*).



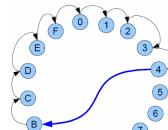
Variáveis: UD- up down (input)

Ji Ki – entradas dos FF JK i

Qi Qi' – saídas dos FF JK i

Qi* - próximo estado do FF JK i

CO- Carry out (output) para utilizar em outros contadores se necessário



Decimal	Aiken	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	1011	0101
6	1100	0110
7	1101	0111
8	1110	1000
9	1111	1001

UD=0

Q3	Q2	Q1	Q0	SN	Q* 3	Q* 2	Q* 1	Q* 0	SN *	C O
0	0	0	0	S0	1	1	1	1	S9	1
0	0	0	1	S1	0	0	0	0	S0	0
0	0	1	0	S2	0	0	0	1	S1	0
0	0	1	1	S3	0	0	1	0	S2	0
0	1	0	0	S4	0	0	1	1	S3	0
0	1	0	1	X1	x	x	x	x		0
0	1	1	0	X2	x	x	x	x		0
0	1	1	1	X3	x	x	x	x		0
1	0	0	0	X4	x	x	x	x		0
1	0	0	1	X5	x	x	x	x		0
1	0	1	0	X6	x	x	x	x		0
1	0	1	1	S5	0	1	0	0	S4	0
1	1	0	0	S6	1	0	1	1	S5	0
1	1	0	1	S7	1	1	0	0	S6	0
1	1	1	0	S8	1	1	0	1	S7	0
1	1	1	1	S9	1	1	1	0	S8	0

UD=1

Q3	Q2	Q1	Q0	SN	Q* 3	Q* 2	Q* 1	Q* 0	SN *	C O
0	0	0	0	S0	0	0	0	1	S1	0
0	0	0	1	S1	0	0	1	0	S2	0
0	0	1	0	S2	0	0	0	1	S3	0
0	0	1	1	S3	0	1	0	0	S4	0
0	1	0	0	S4	1	0	1	1	S5	0
0	1	0	1	X1	x	x	x	x		
0	1	1	0	X2	x	x	x	x		
0	1	1	1	X3	x	x	x	x		
1	0	0	0	X4	x	x	x	x		
1	0	0	1	X5	x	x	x	x		
1	0	1	0	X6	x	x	x	x		
1	0	1	1	S5	1	1	0	0	S6	0
1	1	0	0	S6	1	1	0	1	S7	0
1	1	0	1	S7	1	1	1	0	S8	0
1	1	1	0	S8	1	1	1	1	S9	0
1	1	1	1	S9	0	0	0	0	S0	1

For a JK FF

UD=0

SN	Q3	Q2	Q1	Q0	Q	J3	k3	Q2	j2	k2	Q1	j1	k1	Q0	J0	K0
S0	0	0	0	0	1	1	X	1	1	X	1	1	X	1	1	X
S1	0	0	0	1	0	0	X	0	0	X	0	0	X	0	X	1
S2	0	0	1	0	0	0	X	0	0	X	0	x	1	1	1	X
S3	0	0	1	1	0	0	X	0	0	X	1	x	0	0	X	1
S4	0	1	0	0	0	0	X	0	x	1	1	1	x	1	1	X
X1	0	1	0	1												
X2	0	1	1	0												
X3	0	1	1	1												
X4	1	0	0	0												
X5	1	0	0	1												
X6	1	0	1	0												
S5	1	0	1	1	0	x	1	1	1	X	0	x	1	0	X	1
S6	1	1	0	0	1	X	0	0	x	1	1	1	X	1	1	X
S7	1	1	0	1	1	X	0	1	x	0	0	0	X	0	X	1
S8	1	1	1	0	1	X	0	1	X	0	0	x	1	1	1	X
S9	1	1	1	1	1	X	0	1	x	0	1	x	0	0	X	1

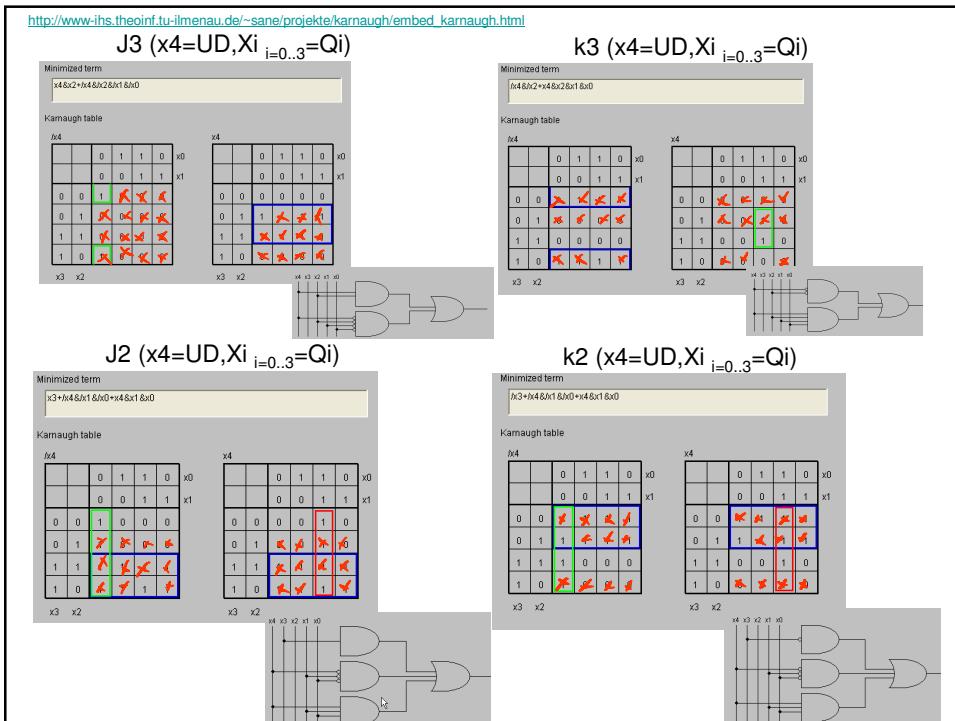
For a JK FF

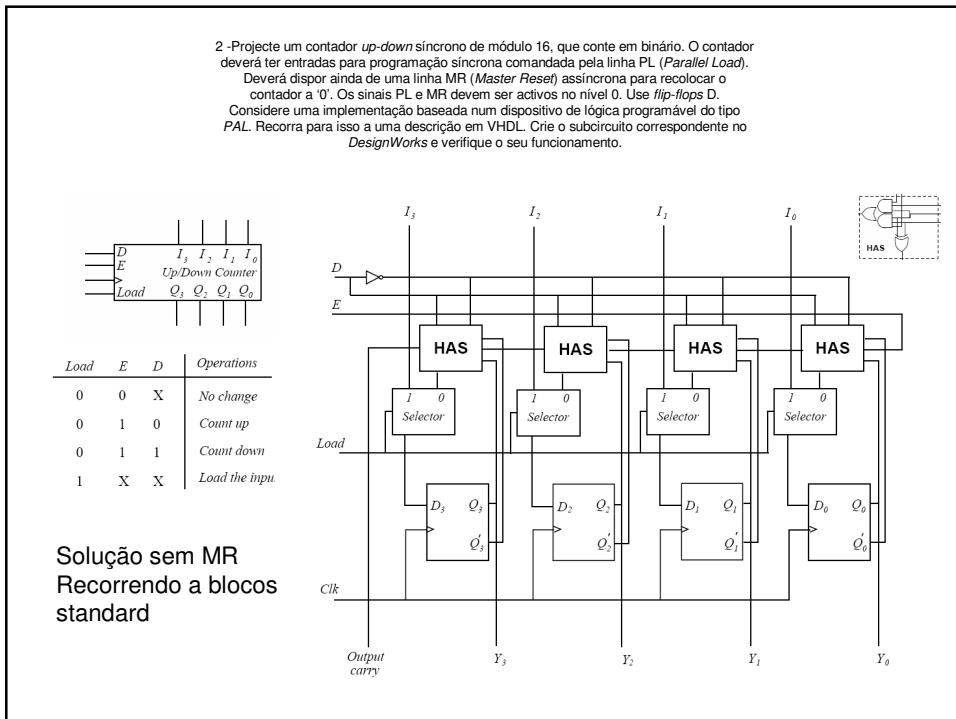
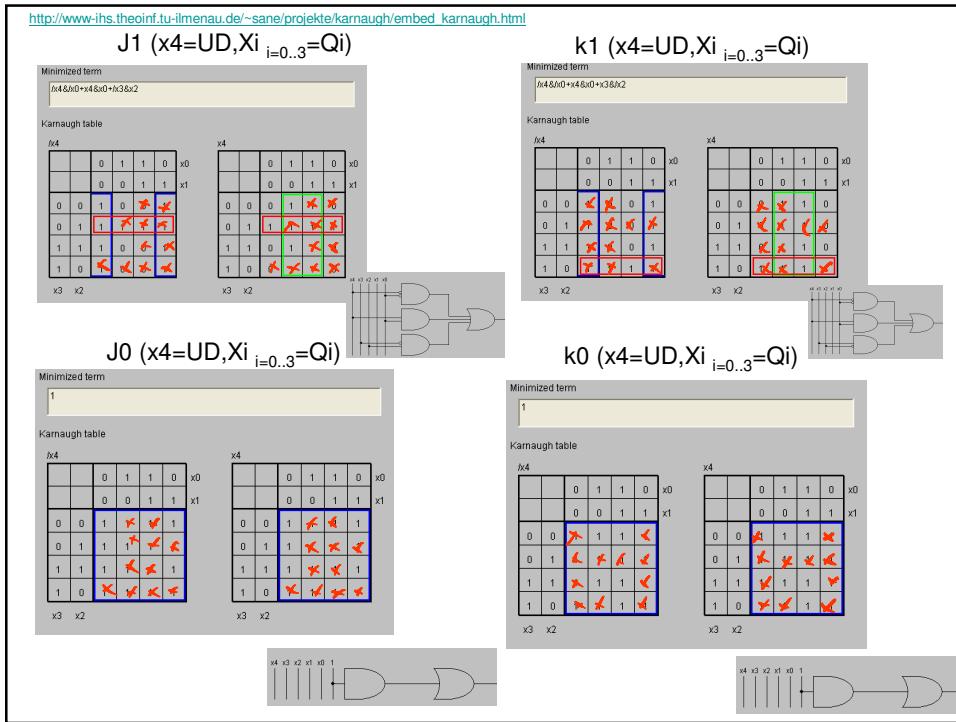
Q	$Q(next)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$$Q(next) = JQ' + KQ$$

UD=1

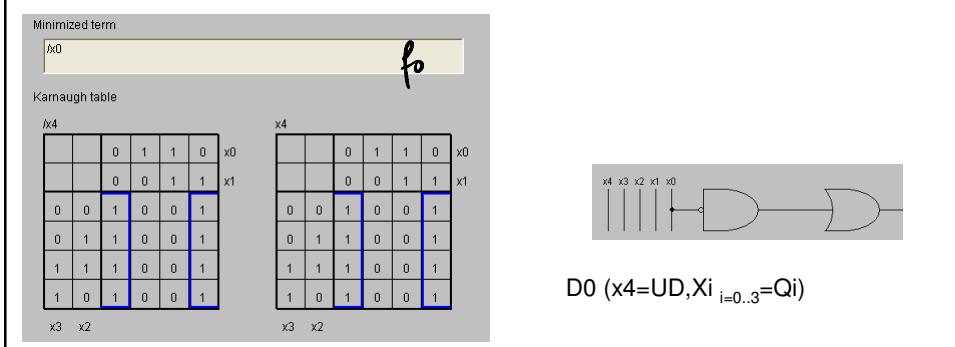
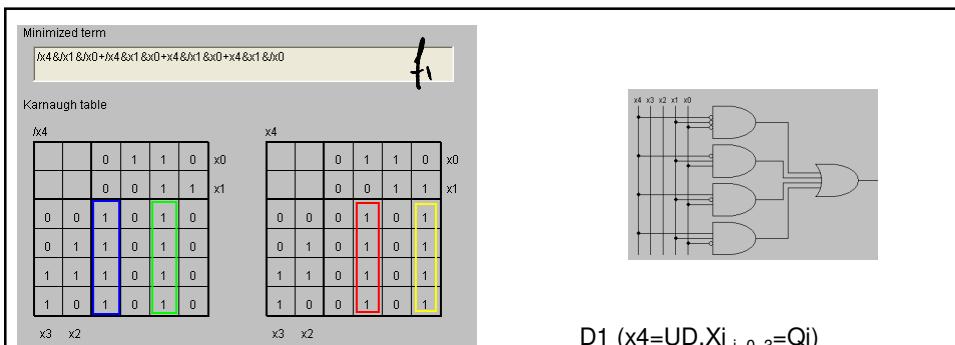
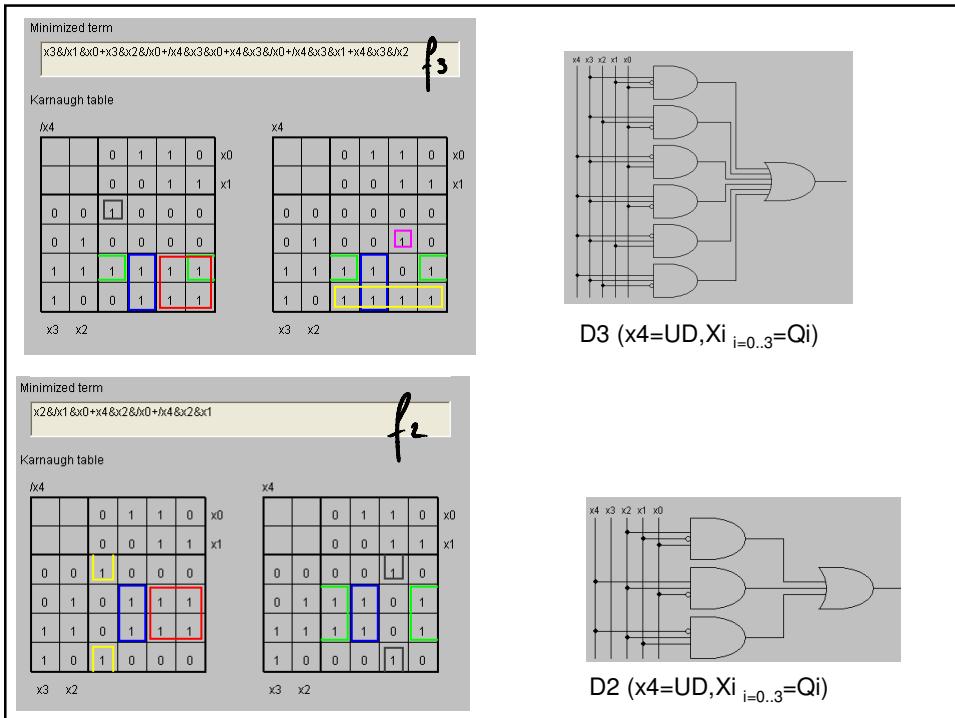
SN	Q3	Q2	Q1	Q0	Q'3*	J3	K3	Q2*	j2	K2	Q1*	J1	k1	Q0*	J0	K0
S0	0	0	0	0	0	0	X	0	0	X	0	0	X	1	1	X
S1	0	0	0	1	0	0	X	0	0	X	1	1	X	0	X	1
S2	0	0	1	0	0	0	X	0	0	X	1	0	X	1	1	X
S3	0	0	1	1	0	0	X	1	1	X	0	x	1	0	X	1
S4	0	1	0	0	1	1	X	0	X	1	1	1	X	1	1	X
X1	0	1	0	1												
X2	0	1	1	0												
X3	0	1	1	1												
X4	1	0	0	0												
X5	1	0	0	1												
X6	1	0	1	0												
S5	1	0	1	1	1	X	0	1	1	X	0	x	1	0	X	1
S6	1	1	0	0	1	X	0	1	0	X	0	0	X	1	1	X
S7	1	1	0	1	1	X	0	1	X	0	1	1	X	0	X	1
S8	1	1	1	0	1	X	0	1	X	0	1	x	0	1	1	X
S9	1	1	1	1	0	X	1	0	X	0	1	x	0	0	X	1





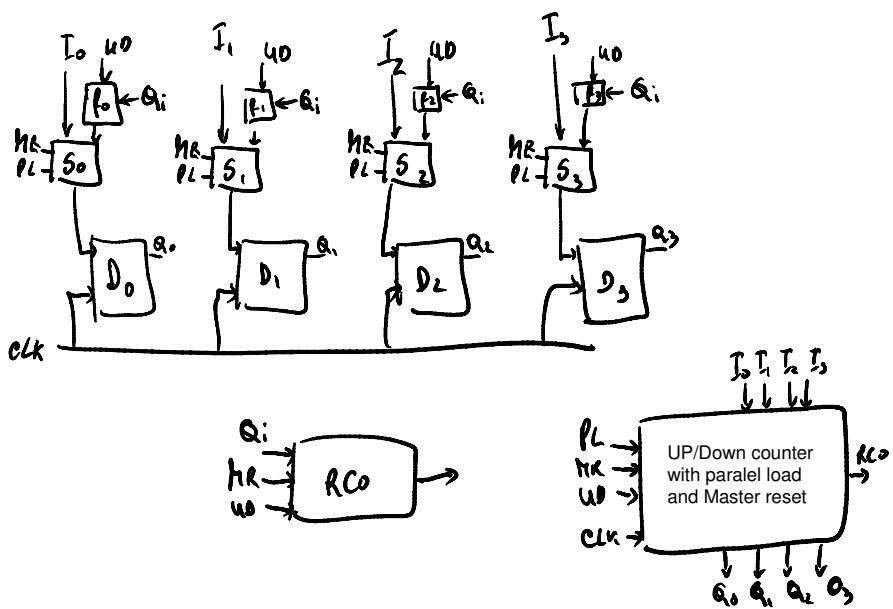
UD=0										UD=1									
Q3	Q2	Q1	Q0	SN	Q* 3	Q* 2	Q* 1	Q* 0	S N*	Q3	Q2	Q1	Q0	SN	Q* 3	Q* 2	Q* 1	Q* 0	SN
0	0	0	0	S0	1	1	1	1	S15	0	0	0	0	0	0	0	0	1	S1
0	0	0	1	S1	0	0	0	0	S0	0	0	0	1	0	0	0	1	0	S2
0	0	1	0	S2	0	0	0	0	S1	0	0	1	0	0	0	0	1	1	S3
0	0	1	1	S3	0	0	0	1	S2	0	0	1	1	0	0	1	0	0	S4
0	1	0	0	S4	0	0	1	1	S3	0	1	0	0	0	0	1	0	1	S5
0	1	0	1	S5	0	1	0	0	S4	0	1	0	1	1	0	0	1	0	S6
0	1	1	0	S6	0	1	0	1	S5	0	1	1	0	0	0	1	1	1	S7
0	1	1	1	S7	0	1	1	0	S6	0	1	1	1	0	0	0	0	0	S8
1	0	0	0	S8	0	1	1	1	S7	1	0	0	0	1	0	0	0	1	S9
1	0	0	1	S9	1	0	0	0	S8	1	0	0	1	0	0	1	0	0	S10
1	0	1	0	S10	1	0	0	1	S9	1	0	1	0	0	1	0	1	1	S11
1	0	1	1	S11	1	0	1	0	S10	1	0	1	1	0	0	1	1	0	S12
1	1	0	0	S12	1	0	1	1	S11	1	1	0	0	0	1	1	0	1	S13
1	1	0	1	S13	1	1	0	0	S12	1	1	0	1	1	0	1	1	0	S14
1	1	1	0	S14	1	1	0	1	S13	1	1	1	0	1	1	1	1	1	S15
1	1	1	1	S15	1	1	1	0	S14	0	0	0	0	0	0	0	0	0	S0

FF D										$UD = 1$										
SN	Q3	Q2	Q1	Q0	Q 3*	D3	Q2 *	D2	Q1 *	D1	Q0 *	D0	Q3 *	D3	Q2 *	D2	Q1 *	D1	Q0 *	D0
S0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1
S1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
S2	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1
S3	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0
S4	0	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1
S5	0	1	0	1	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0
S6	0	1	1	0	0	0	1	1	0	0	1	1	0	0	1	1	1	1	1	1
S7	0	1	1	1	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0
S8	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	1	1
S9	1	0	0	1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0
S10	1	0	1	0	1	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1
S11	1	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	0	0	0
S12	1	1	0	0	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1
S13	1	1	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0
S14	1	1	1	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
S15	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0



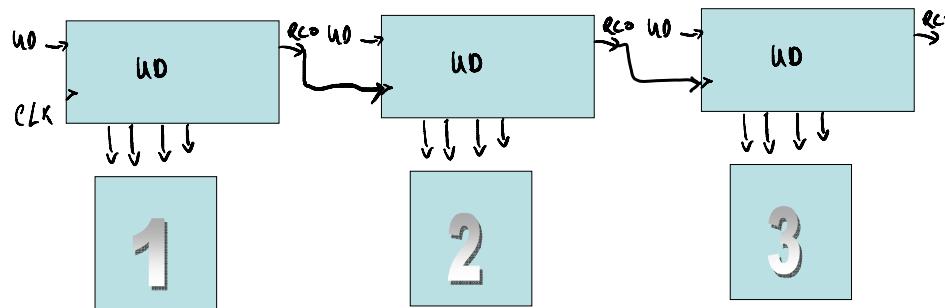
Acrescentando as linhas MR e PL ambas activas a “low”

- $D_3 = MR \cdot (PL' \cdot I_3 + PL \cdot f_3)$
- $D_2 = MR \cdot (PL' \cdot I_2 + PL \cdot f_2)$
- $D_1 = MR \cdot (PL' \cdot I_1 + PL \cdot f_1)$
- $D_0 = MR \cdot (PL' \cdot I_0 + PL \cdot f_0)$
- $RCO = MR \cdot (UD \cdot Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0 + \sim(UD + Q_3 + Q_2 + Q_1 + Q_0))$



3- Para aperfeiçoar o contador projectado no exercício 1, crie uma saída adicional (*RCO*) para assinalar o estado 0 no modo *down* e 9 no modo *up*. Esta saída permite montagens em cascata – como? Projecte agora um cronómetro rudimentar *up-down* com 3 dígitos.

- $RCO = UD \cdot Q3 \cdot Q2 \cdot Q1 \cdot Q0 + \sim(UD + Q3 + Q2 + Q1 + Q0)$



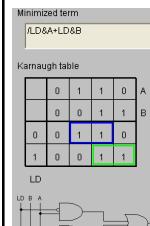
Folha 11

Antonio Teixeira

1 A figura A representa um *shift-register* de 4 bits, que faz deslocamento à esquerda, com inicialização síncrona.

- Desenhe o esquema interno deste bloco e explique o seu princípio de funcionamento.
- Na figura B temos um circuito baseado neste bloco. Desenhe o diagrama de estados deste circuito e explique que função realiza.

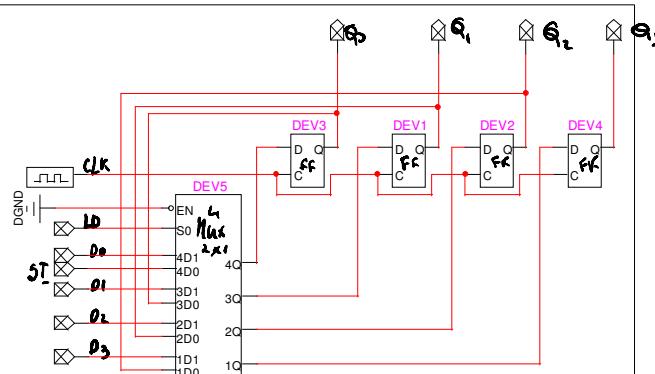
Mux



LD	Q0*	Q1*	Q2*	Q3*
0	D0	D1	D2	D3
1	SI	Q0	Q1	Q2



Figura A



$$\begin{aligned} F0 &= \bar{L}D \cdot D_0 + L \bar{D} \cdot S_I \\ F1 &= \bar{L}D \cdot D_1 + L \bar{D} \cdot Q_0 \\ F2 &= \bar{L}D \cdot D_2 + L \bar{D} \cdot Q_1 \\ F3 &= \bar{L}D \cdot D_3 + L \bar{D} \cdot Q_0 \end{aligned}$$

Eq característica
De um FF D

$$Q^* = D$$

- Na figura B temos um circuito baseado neste bloco. Desenhe o diagrama de estados deste circuito e explique que função realiza.

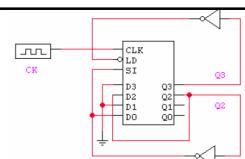


Figura B

SN	Q3	Q2	Q1	Q0	SI	LD	Q3*	Q2*	Q1*	Q0*	SN*
0	0	0	0	0	1	1	0	0	0	1	1
1	0	0	0	1	1	0	0	0	1	1	3
2	0	0	1	0	0	1	0	1	0	1	5
3	0	0	1	1	0	0	0	1	1	1	7
4	0	1	0	0	1	1	1	0	0	0	8
5	0	1	0	1	1	0	1	0	1	0	A
6	0	1	1	0	0	1	1	1	0	0	C
7	0	1	1	1	0	0	1	1	1	0	E
8	1	0	0	0	1	1	0	0	0	1	1
9	1	0	0	1	1	0	0	0	0	1	1
A	1	0	1	0	0	1	0	0	0	1	1
B	1	0	1	1	0	0	0	0	0	1	1
C	1	1	0	0	1	1	0	1	0	0	4
D	1	1	0	1	1	0	0	1	0	0	4
E	1	1	1	0	0	1	0	1	0	0	4
F	1	1	1	1	0	0	0	1	0	0	4

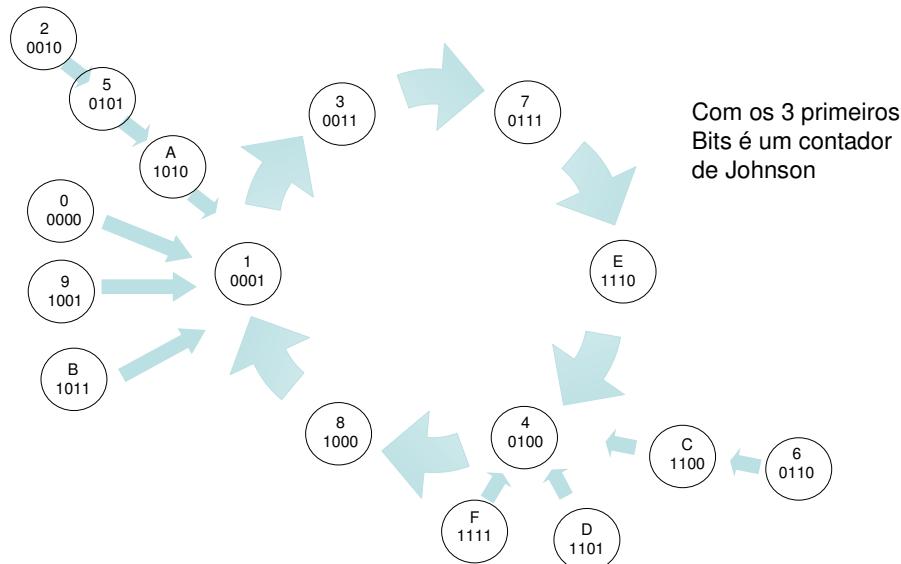
Eq. De excitação

$$\begin{aligned} S_I &= Q_2 \\ D_2 &= Q_2 \\ D_1 &= 0 \\ D_3 &= 0 \\ D_0 &= \bar{Q}_2 \\ L D &= Q_3 \end{aligned}$$

Eq. de Transição

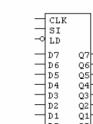
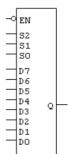
$$\begin{aligned} Q_0^* &= F_0 = \bar{L}D \cdot D_0 + L \bar{D} \cdot S_I \\ &= Q_3 / Q_2 + Q_3 / Q_2 = \bar{Q}_2 \\ Q_1^* &= F_1 = \bar{L}D \cdot D_1 + L \bar{D} \cdot Q_0 \\ &= Q_3 . 0 + Q_3 . Q_0 = \bar{Q}_3 . Q_0 \\ Q_2^* &= F_2 = \bar{L}D \cdot D_2 + L \bar{D} \cdot Q_1 \\ &= Q_3 . Q_2 + Q_3 . Q_1 \\ Q_3^* &= F_3 = \bar{L}D \cdot D_3 + L \bar{D} \cdot Q_0 \\ &= Q_3 . 0 + Q_3 . Q_2 = \bar{Q}_3 . Q_2 \end{aligned}$$

Diagrama de estados



2. Projete uma linha de atraso de comprimento N , variável entre 1 e 64. Este comprimento N será especificado por 6 linhas A5..0 (que conterão a representação binária de $N-1$). Para o seu projeto dispõe dos seguintes blocos elementares:

- i) Multiplexor 8:1 ii) Shift register de 8 bits com saídas em paralelo



Naturalmente, pretende-se minimizar o número de blocos necessário. Explique detalhadamente o raciocínio que o conduziu à solução escolhida. Teste-a.
[Sugestão: note que $N-1 = (A5..3) * 8 + (A2..0)$]

Mux de 64->1
Pode ser construído com

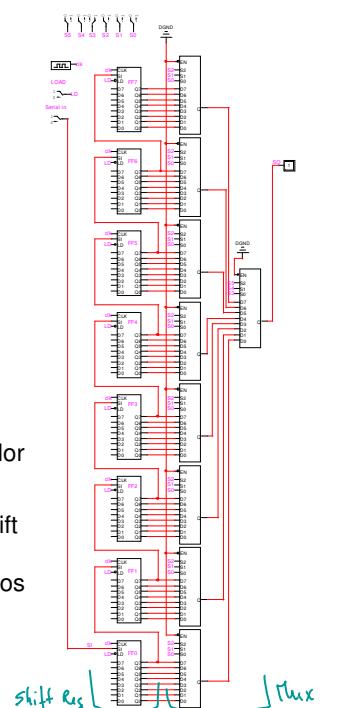
$$8 \times (8 \rightarrow 1) + 1 \times (8 \rightarrow 1)$$

Controlados por
A2..0 A5..3

Um shift register de 8 bits é um atraso de valor até 8.

Se encadearmos 8 Shift Reg. Através do SI (serial input) = Q7 temos

Um atraso máximo de
 8×8



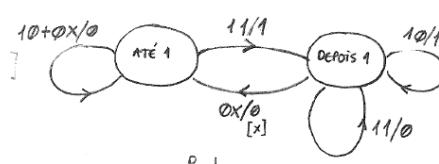
Folha 12

António Teixeira

1 Projecte um circuito sequencial síncrono que realize o complemento para dois dum número, de comprimento arbitrário, que entra no circuito começando pelo seu bit menos significativo. Admita que este circuito tem uma entrada de reset síncrona e activa a '0'.

- Complemento para 2.
 - É o complemento para 1 seguido da soma de 1.
 - Ou: mater a entrada até ao primeiro 1 seguido da inversão dos seguintes.
 - Eg.
 - **0011010**
 - Complemento para 1
 - **1100101**
 - **+1**
 - **1100110**
 - **Inverte Mantém**

- X- entrada sequencial
- R- Reset Síncrono (activa baixa)
- Y- saída

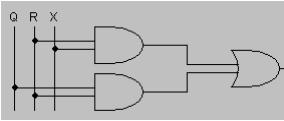
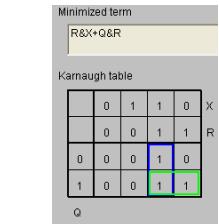


RESET	INPUT	ANTERIOR	SEGUINTE	OUT
R	X	Q	Q^*	Y
0	0	0 \bar{S}_0	S_0	0
0	0	1 \bar{S}_1	S_0	0
0	1	0 \bar{S}_0	S_0	0
0	1	1 \bar{S}_1	S_0	0
1	0	0 \bar{S}_0	S_0	0
1	0	1 \bar{S}_1	S_1	1
1	1	0 \bar{S}_0	S_1	1
1	1	1 \bar{S}_1	S_1	0

Resolução com FF D, $Q^*=D$

solução

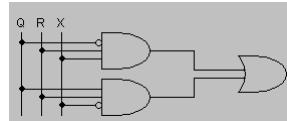
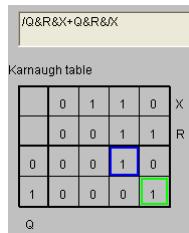
- Q^*



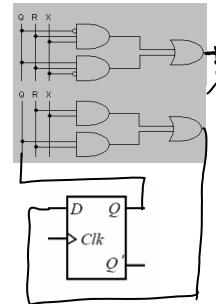
Optimized

$$(X+Q).R$$

- Y

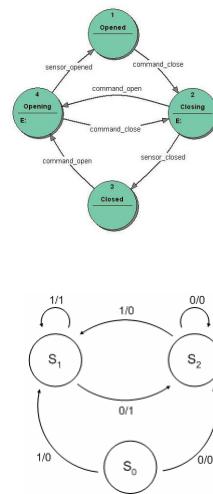


$$(X(+Q).R$$

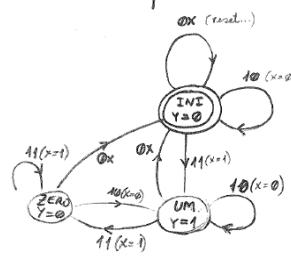


Maquinas

- In the [theory of computation](#), a **Moore machine** is a [finite state automaton](#) where the outputs are determined by the **current state alone** (and do not depend directly on the input). The [state diagram](#) for a Moore machine will include an [output signal for each state](#). Compare with a [Mealy machine](#), which maps [transitions](#) in the machine to outputs.
- In the [theory of computation](#), a **Mealy machine** is a [finite state machine](#) (and more accurately, a [finite state transducer](#)) that **generates an output based on its current state and an input**. This means that the [state diagram](#) will include both an input and output signal for each transition edge. In contrast, the output of a [Moore finite state machine](#) depends only on the machine's current state; transitions have no input attached. However, for each Mealy machine there is an equivalent Moore machine whose states are the [union](#) of the Mealy machine's states and the [Cartesian product](#) of the Mealy machine's states and the input alphabet.
- http://en.wikipedia.org/wiki/Finite_state_machine



Solução Moore



00	01	11	10	Y
INI	INI	UM	ZERO	
UM	INI	INI	ZERO	1
ZERO	INI	INI	ZERO	0

Q	00	01	11	10	Y
00	00	00	11	00	0
01	01	X	X	X	X
11	00	00	10	11	1
10	00	00	10	11	0

R_L, X

CODIFICAÇÃO:

INI : 00
UM : 11
ZERO : 10

$Q_1 \rightarrow \text{ñ usado}$

Não: seria interessante experimentar outra codificação.

$$Q_1^* = R_L$$

$$Q_0^* = Q_1 \oplus R_L$$

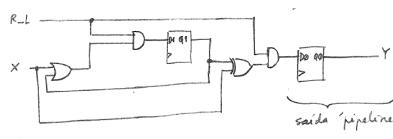
$$D = X \cdot R_L + Q_1 \cdot R_L$$

$$= (X + Q_1) \cdot R_L$$

$$Q_0^* = \bar{Q}_1 \cdot X \cdot R_L + Q_1 \cdot \bar{X} \cdot R_L$$

$$= (X \oplus Q_1) \cdot R_L$$

$$Y = Q_0$$



Nota caso: Modelo de Mealy = Modelo de Mealy + pipeline de saída

2. Projete um detector de sequências cuja saída, y , será '1' sempre que ocorrer a sequência de entradas 1101. São permitidas sequências sobrepostas.

Exemplo:
x 01011101101011
y 00000001001000

Realize o circuito com flip-flops JK. Verifique o seu funcionamento.

Q	Q(next)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$$Q(\text{next}) = JQ' + K'Q$$

$Q(\text{next}) = JQ' + K'Q$

$J = Q_0, X$

$K = \bar{Q}_0$

$J_1 = Q_0$

$K_1 = \bar{Q}_0$

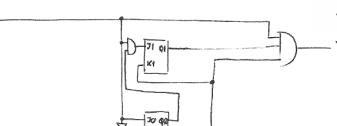
$J_0 = X$

$K_0 = \bar{X}$

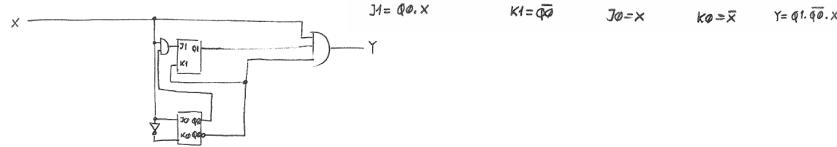
$Y = Q_1 \cdot Q_0 \cdot X$

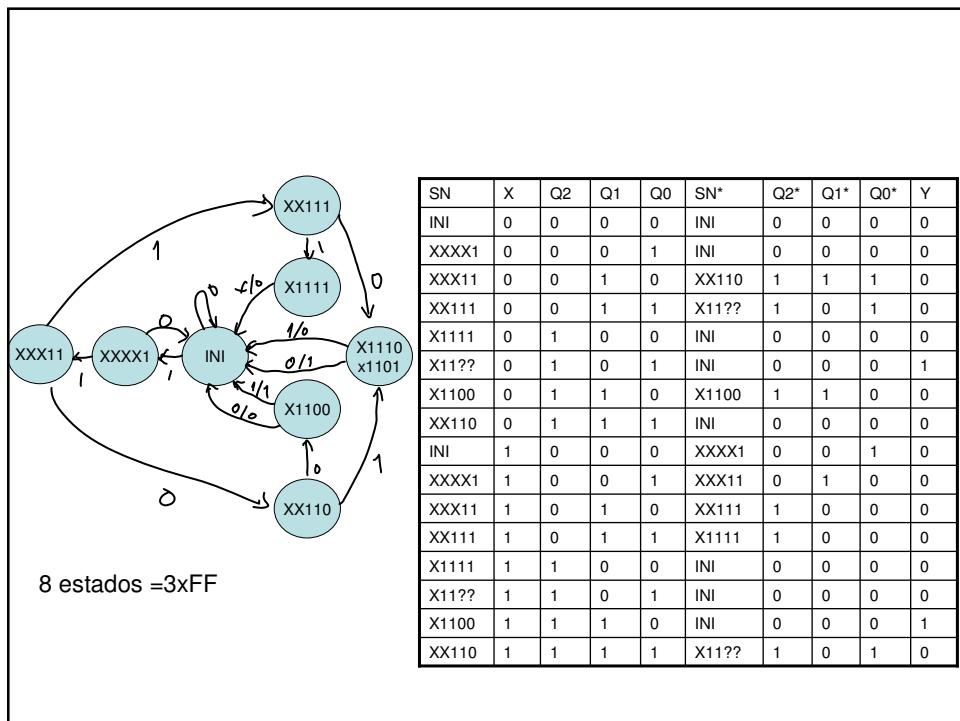
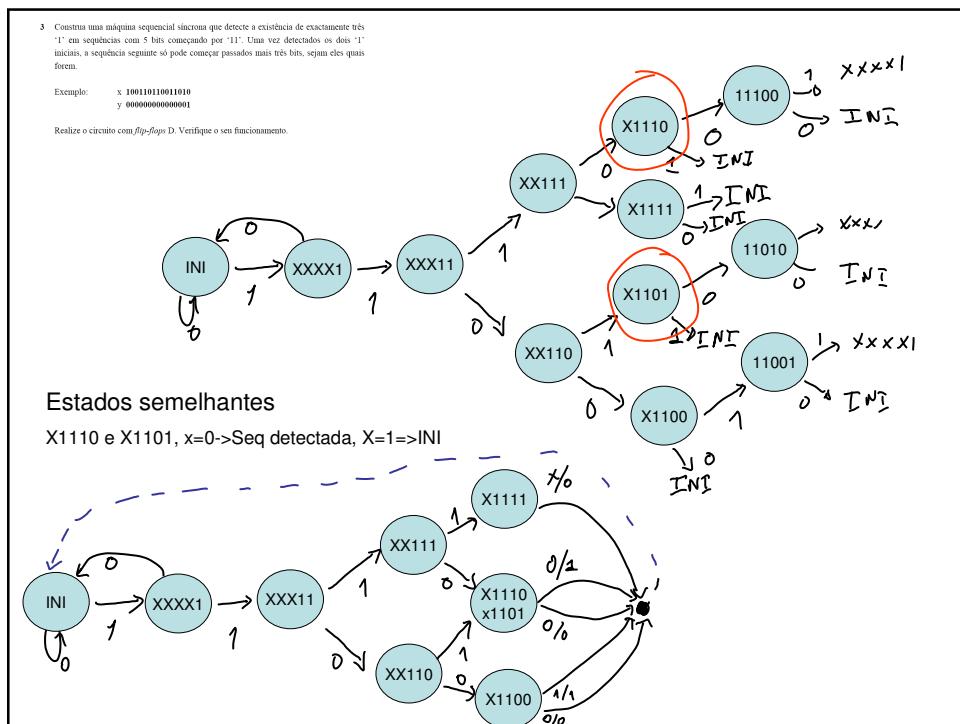
Sn	Q1	Q0	X	Sn*	Q1*	J1	K1	Q0*	J0	K0	Y
INI	0	0	0	INI	0	0	X	0	0	X	0
1	0	1	0	INI	0	0	X	0	X	1	0
11	1	1	0	110	1	X	0	0	X	1	0
110	1	0	0	INI	0	X	1	0	0	X	0
INI	0	0	1	1	0	0	X	1	1	X	0
1	0	1	1	11	1	1	X	0	X	0	0
11	1	1	1	11	1	X	0	0	X	0	0
110	1	0	1	1	0	X	1	1	1	X	1

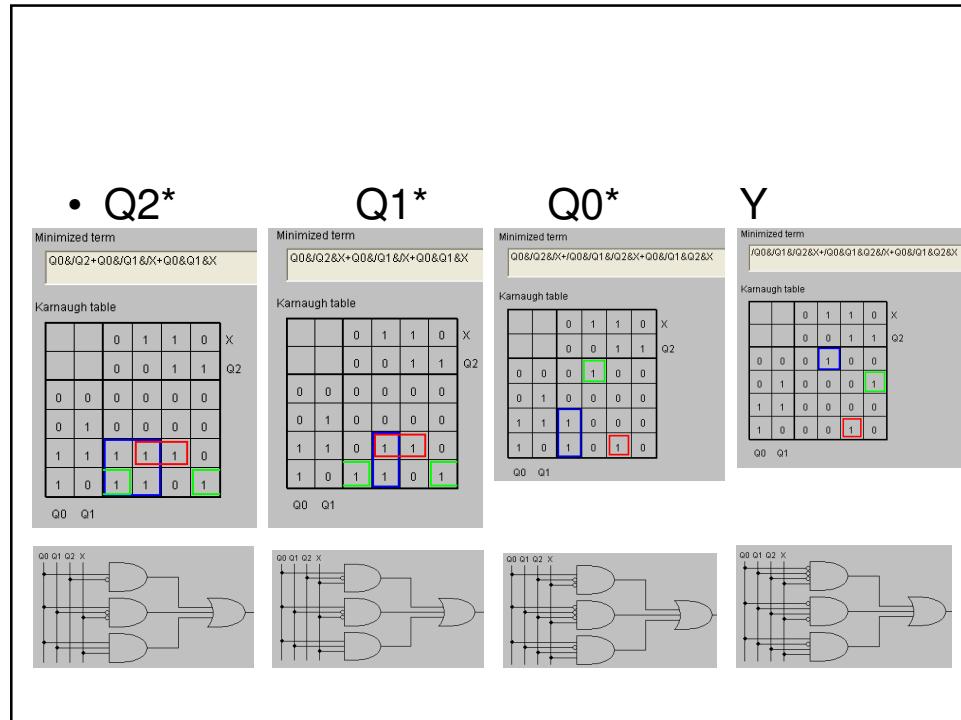
4 estados = 2×2



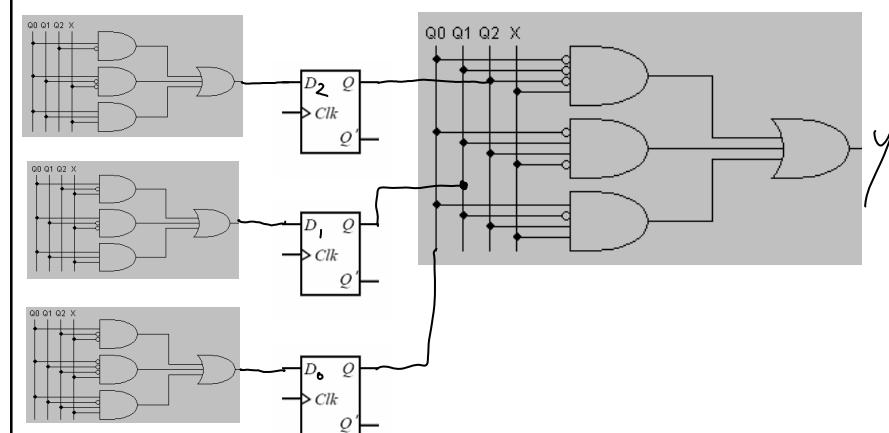
$\overline{J_1}$	$\overline{K_1}$	$\overline{J_0}$	$\overline{K_0}$
Q_1	\overline{Q}_1	Q_0	\overline{Q}_0
Q_1^*	\overline{Q}_1^*	Q_0^*	\overline{Q}_0^*
$J_1 = Q_0, X$	$K_1 = \bar{Q}_0$	$J_0 = X$	$K_0 = \bar{X}$
$Y = Q_1 \cdot Q_0 \cdot X$			







Circuito final



Folha 13

Antonio Teixeira

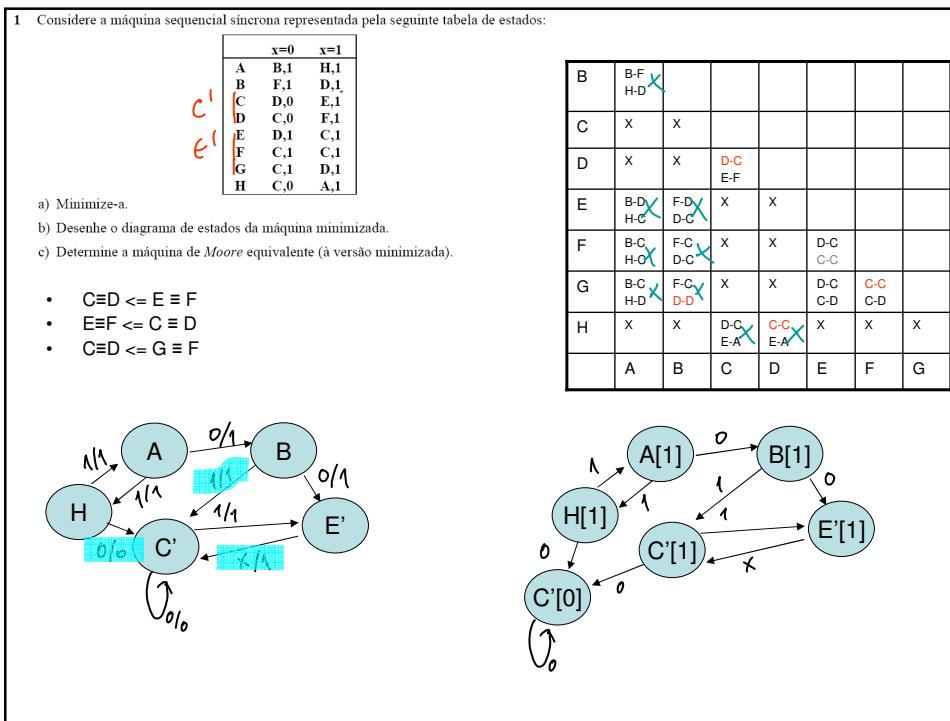
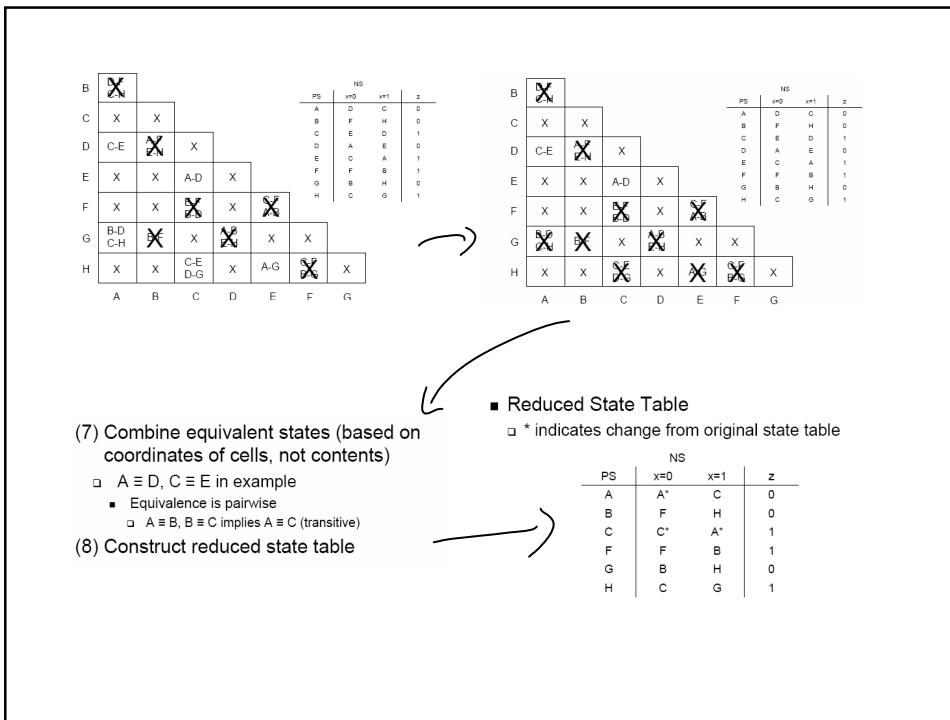
	B	C-H		PS	NS	x=0	x=1	z
B	X	X		A	D	C	0	
C				B	F	H	0	
D		X		C	E	D	1	
E	X	X	X	D	A	E	0	
F	X	X	X	E	C	A	1	
G				F	B	H	0	
H	X	X	X	G	C	G	1	

	B	C-H		PS	NS	x=0	x=1	z
B	X	X		A	D	C	0	
C				B	F	H	0	
D		X		C	E	D	1	
E	X	X	X	D	A	E	0	
F	X	X	X	E	C	A	1	
G				F	B	H	0	
H	X	X	X	G	C	G	1	

	B	C-H		PS	NS	x=0	x=1	z
B	X	X		A	D	C	0	
C				B	F	H	0	
D		X		C	E	D	1	
E	X	X	X	D	A	E	0	
F	X	X	X	E	C	A	1	
G				F	B	H	0	
H	X	X	X	G	C	G	1	

	B	C-H		PS	NS	x=0	x=1	z
B	X	X		A	D	C	0	
C				B	F	H	0	
D		X		C	E	D	1	
E	X	X	X	D	A	E	0	
F	X	X	X	E	C	A	1	
G				F	B	H	0	
H	X	X	X	G	C	G	1	

	B	C-H		PS	NS	x=0	x=1	z
B	X	X		A	D	C	0	
C				B	F	H	0	
D		X		C	E	D	1	
E	X	X	X	D	A	E	0	
F	X	X	X	E	C	A	1	
G				F	B	H	0	
H	X	X	X	G	C	G	1	

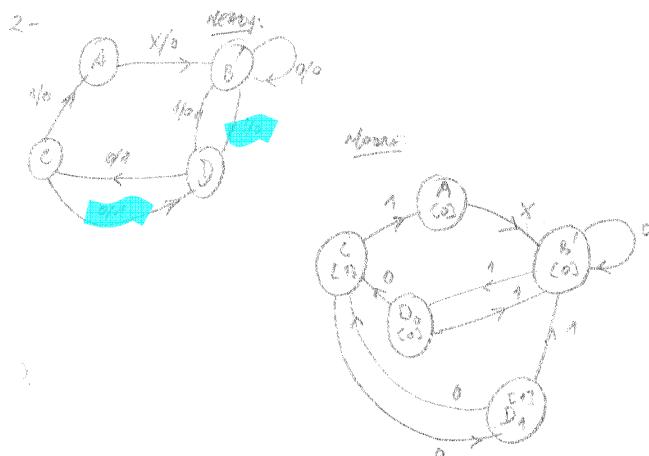


2 Repita o exercício anterior com as seguintes máquinas sequenciais síncronas:

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

B	B C X			
C	X	X		
D	X	X	D C X	
E	E D X	E B D	X	X
	A	B	C	D

$E \equiv B \Rightarrow E \equiv B;$

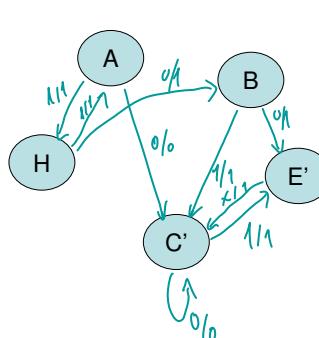


2 Repita o exercício anterior com as seguintes máquinas sequenciais síncronas:

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

B	X						
C	H F X	X					
D	H F X	X	D C				
E	X	F C X	X				
F	X	F C X	X	C D			
G	X	F C X	X	C D	D C		
H	X	F B C	X	C D	D A	D B X	
	A	B	C	D	E	F	G

$C \equiv D \Rightarrow F \equiv E; F \equiv E \Rightarrow C \equiv D;$
 $E \equiv G \Rightarrow C \equiv D; F \equiv G \Rightarrow C \equiv D$



- 3** Um pequeno sistema de computação em *hardware* digital tem como tarefa calcular sequencialmente a expressão $((A \times 2 + B) \oplus A)$, em que A e B são operandos de 4 bits sem sinal. O sistema engloba dois blocos fundamentais: o bloco de manipulação de dados (*datapath*) e o bloco de controlo. Este exercício foca-se apenas no *datapath*. Considere que tem à sua disposição **um** registo de armazenamento de 4 bits, **um** somador completo de 4 bits e multiplexers 2:1. Admita a existência do complemento dos operandos A e B .