

Problema 2.7

Apartat A Escrivim la taula de veritat de Y_0 :

$x_3x_2x_1x_0$	Y_0
0000	1
0001	1
0010	1
0011	1
0100	0
0101	0
0110	1
0111	1
1000	0
1001	1
1010	1
1011	1
1100	0
1101	1
1110	0
1111	1

La forma canònica en PdS es troba fent producte dels 5 maxterms que apareixen a la taula de veritat:

$$\begin{aligned}
 Y_0(x_3, x_2, x_1, x_0) = & (x_3 + \overline{x_2} + x_1 + x_0) \\
 & \cdot (x_3 + \overline{x_2} + x_1 + \overline{x_0}) \\
 & \cdot (\overline{x_3} + x_2 + x_1 + x_0) \\
 & \cdot (\overline{x_3} + \overline{x_2} + x_1 + x_0) \\
 & \cdot (\overline{x_3} + \overline{x_2} + \overline{x_1} + x_0)
 \end{aligned}$$

Apartat B Construïm el mapa de Karnaugh:

$\begin{array}{c} \diagdown \\ x_1x_0 \end{array}$	00	01	11	10
$x_3x_2 \begin{array}{c} \diagdown \end{array}$				
00	1	1	1	1
01	0	0	1	1
11	0	1	1	0
10	0	1	1	1

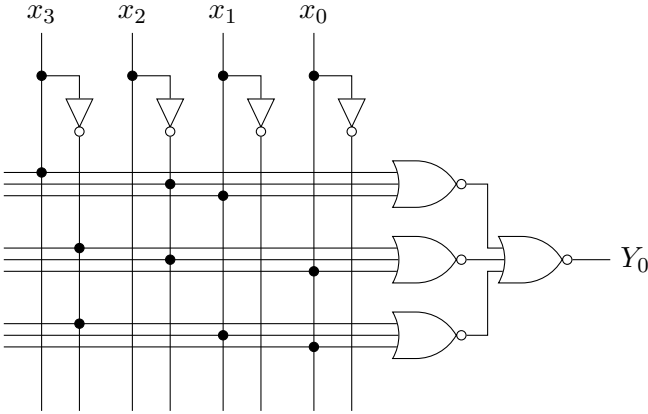
Escrivim l'expressió mínima en PdS:

$$Y_0(x_3, x_2, x_1, x_0) = (x_3 + \overline{x_2} + x_1) (\overline{x_3} + \overline{x_2} + x_0) (\overline{x_3} + x_1 + x_0)$$

Apliquem Morgan al producte per obtenir l'expressió en NOR de NORs:

$$Y_0(x_3, x_2, x_1, x_0) = \overline{\overline{x_3 + \overline{x_2} + x_1} + \overline{\overline{x_3} + \overline{x_2} + x_0} + \overline{\overline{x_3} + x_1 + x_0}}$$

El logigrama quedaria així:



Apartat C Escrivim la taula de veritat de Y_1 , junt amb la de Y_0 que hem escrit a dalt:

$x_3x_2x_1x_0$	Y_0	Y_1
0000	1	1
0001	1	0
0010	1	1
0011	1	0
0100	0	0
0101	0	0
0110	1	0
0111	1	1
1000	0	1
1001	1	1
1010	1	1
1011	1	1
1100	0	0
1101	1	1
1110	0	0
1111	1	1

D'acord amb això, les entrades $(0,0,0,1)$, $(0,0,1,1)$, $(0,1,1,0)$ i $(1,0,0,0)$ han de ser inespecificacions ja que tenen sortides diferents en les implementacions Y_0 i Y_1 .

Apartat D Fem el mapa de Karnaugh basant-nos en la taula de veritat de l'apartat B:

$\begin{array}{c} \diagdown \\ x_1x_0 \end{array}$	00	01	11	10
$x_3x_2 \begin{array}{c} \diagdown \end{array}$				
00	1	0	0	1
01	0	0	1	0
11	0	1	1	0
10	1	1	1	1

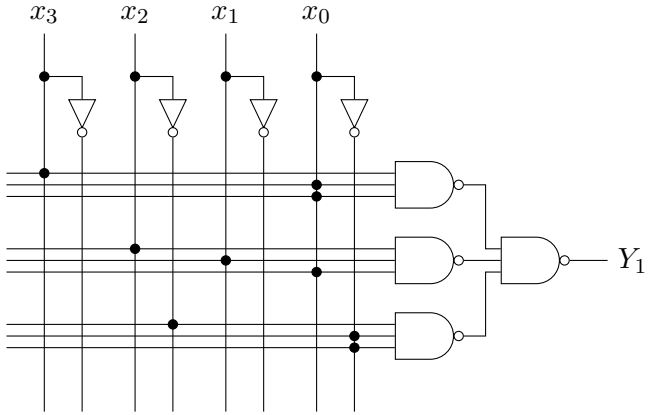
Escrivim una expressió simplificada en SdP:

$$Y_1(x_3, x_2, x_1, x_0) = x_3x_0 + x_2x_1x_0 + \overline{x_2x_0}$$

Apliquem Morgan a la suma per obtenir l'expressió en NAND de NANDs, i repetim factors en alguns productes per tal de tenir sempre 3 entrades:

$$Y_1(x_3, x_2, x_1, x_0) = \overline{\overline{x_3 \cdot x_0 \cdot x_0} + \overline{x_2 \cdot x_1 \cdot x_0} + \overline{x_2 \cdot x_0 \cdot x_0}}$$

El logigrama quedaria així:



Problema 2.8

Apartat A Mentalment transformem el logigrama a AND de ORs i escrivim les sortides del primer nivell de portes, que enumerarem de dalt a baix:

$$\begin{aligned}t_0 &= s + \overline{x_1} \\ t_1 &= x_1 + x_0 \\ t_2 &= s + x_1 \\ t_3 &= \overline{s} + \overline{x_1} + \overline{x_0} \\ t_4 &= \overline{x_1} + x_0\end{aligned}$$

Escrivim les expressions de z_2 , z_1 i z_0 en funció de les sortides de dalt:

$$\begin{aligned}z_2 &= t_0 t_1 t_2 \\ z_1 &= t_1 t_2 t_3 \\ z_0 &= t_1 t_4\end{aligned}$$

Escrivim la taula de veritat de z_2 , z_1 i z_0 :

$s x_1 x_0$	z_2	z_1	z_0
000	0	0	0
001	0	0	1
010	0	1	0
011	0	1	1
100	0	0	0
101	1	1	1
110	1	1	0
111	1	0	1

Escrivim els mapes de Karnaugh per a z_2 ...

$\begin{smallmatrix} x_1 x_0 \\ s \end{smallmatrix}$	00	01	11	10
0	0	0	0	0
1	0	1	1	1

Per a z_1 ...

$\begin{smallmatrix} x_1 x_0 \\ s \end{smallmatrix}$	00	01	11	10
0	0	0	1	1
1	0	1	0	1

Per a z_0 ...

$\begin{smallmatrix} x_1 x_0 \\ s \end{smallmatrix}$	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Escrivim les expressions simplificades en PdS:

$$\begin{aligned}z_2 &= (x_1 + x_0) s \\ z_1 &= (x_1 + x_0) (s + x_1) (\overline{s} + \overline{x_1} + \overline{x_0}) \\ z_0 &= x_0\end{aligned}$$

Apartat B Escrivim les expressions simplificades ara en SdP:

$$\begin{aligned}z_2 &= s \overline{x_1} x_0 + s x_1 \\ z_1 &= s \overline{x_1} x_0 + \overline{s} x_1 + x_1 \overline{x_0} \\ z_0 &= x_0\end{aligned}$$

Fixem-nos que, tot i que existeixen expressions més simples individualment per a z_2 , la que hem fet servir ens beneficia perquè reutilitza productes de z_1 , reduint el nombre de portes lògiques necessàries.

Ara, apliquem Morgan a les sumes per obtenir les expressions en NAND de NANDs (fent simplificacions trivials tipus $\overline{\overline{x}} = x$):

$$\begin{aligned}z_2 &= \overline{\overline{(s \overline{x_1} x_0)} \cdot \overline{(s x_1)}} \\ z_1 &= \overline{\overline{(s \overline{x_1} x_0)} \cdot \overline{(\overline{s} x_1)} \cdot \overline{(x_1 \overline{x_0})}} \\ z_0 &= x_0\end{aligned}$$

Ara, comptem que hi ha 4 productes negats únics en el primer nivell, i 2 en el nivell superior. El nombre de portes NAND mínimes per implementar-les, sense comptar la implementació dels negats, seria **6**.

Apartat C El circuit serveix per a canviar del codi d'entrada (signe – valor absolut) a Ca2 de 3 bits.

Problema 2.16 Suposant que l'entrada de la funció ha sigut (A, B) durant un temps suficientment llarg, es diu que hi ha hagut un espuri estàtic si, en aplicar una nova entrada (A', B') , s'observen *dos* canvis de sortida abans d'arribar a la sortida definitiva. Aplicant la definició en la nostra xarxa, hi ha hagut espuri estàtic si i només si:

$$\overline{AB + \overline{A + B}} \neq \overline{A'B' + \overline{A + B}} \quad \text{i} \quad \overline{A'B' + \overline{A + B}} \neq \overline{A'B' + \overline{A' + B'}}$$

Manipulem:

$$AB + \overline{A + B} \neq A'B' + \overline{A + B} \quad \text{i} \quad A'B' + \overline{A + B} \neq A'B' + \overline{A' + B'}$$

$$AB \neq A'B' \quad \text{i} \quad \overline{A + B} = 0 \quad \text{i} \quad A'B' = 0 \quad \text{i} \quad \overline{A + B} \neq \overline{A' + B'}$$

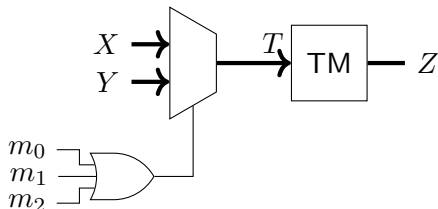
$$AB \neq 0 \quad \text{i} \quad A + B = 1 \quad \text{i} \quad A'B' = 0 \quad \text{i} \quad 0 \neq \overline{A' + B'}$$

$$AB = 1 \quad \text{i} \quad A + B = 1 \quad \text{i} \quad A'B' = 0 \quad \text{i} \quad 0 = A' + B'$$

$$A = 1 \quad \text{i} \quad B = 1 \quad \text{i} \quad A' = 0 \quad \text{i} \quad B' = 0$$

Per tant, només es produirà un espuri estàtic quan l'entrada canviï de $(1, 1)$ a $(0, 0)$. La resposta a l'apartat A és *no*; la resposta a l'apartat B és *sí*.

Problema 2.18 Visualment es prepara el següent anàlisi modular:



Ara només queda analitzar el bloc TM i hauriem de ser capaços de donar una especificació en alt nivell de la funció del circuit. Escrivim la taula de veritat de TM:

$t_2 t_1 t_0$	$z_2 z_1 z_0$
000	001
001	010
010	011
011	100
100	101
101	110
110	111
111	100

Segons aquesta taula de veritat, sembla que TM suma 1 a l'entrada, en binari natural, però quan l'entrada és 7 retorna 4 i no 0 (possible inespecificació?). Una especificació en alt nivell seria:

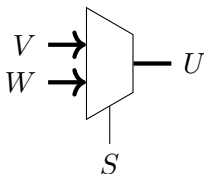
La xarxa suma 1 a Y i retorna el resultat. Excepcionalment, si $Y = 7$ es retorna 4.

En cas que $M = 0$, es fa servir X en comptes d' Y .

Problema 2.19

Apartat A La xarxa retorna la distància entre A i B : $C = |A - B|$

Apartat C El disseny per a M_2 és un multiplexor de 2 busos de 4 bits connectat apropiadament:



Apartat D Sabem que *per a aquest us en concret* de M_3 , $X \geq Y$, i per tant $0 \geq Z \geq 2^4 - 1$.

Es pot comprovar que un *restador en Ca2* es comporta com un *restador modular en binari natural*. I com que, per a les nostres entrades, la resta modular es comporta de la mateixa forma que la resta típica, podem fer servir un restador en Ca2 de 4 bits, sense preocupar-nos de fer cap conversió.

Un restador en Ca2 es pot fer amb un negador i un sumador amb carry d'entrada 1, connectats adequadament:

