Soluciones examen 1ª oportunidad
Electrónica Digital
2021

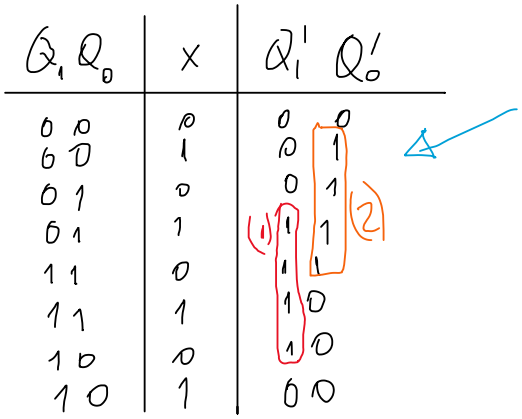
Juan J Pombo
Grado Robótica EPSE Lugo

Circuitos combinacionales útiles

1. Deseña un circuíto que use como elementos de memoria *flip-flop's* tipo D, con activación positiva de flanco de reloxio, e unha única entrada de habilitación X que se comporte da seguinte maneira:

a) cando X=0, o sistema non cambia de estado, b) cando X=1 o sistema cambia de estado de forma síncrona seguindo a seguinte secuencia $Q_{-1}Q_0$ = {00 -> 01 -> 11 -> 10 e volta a 00}

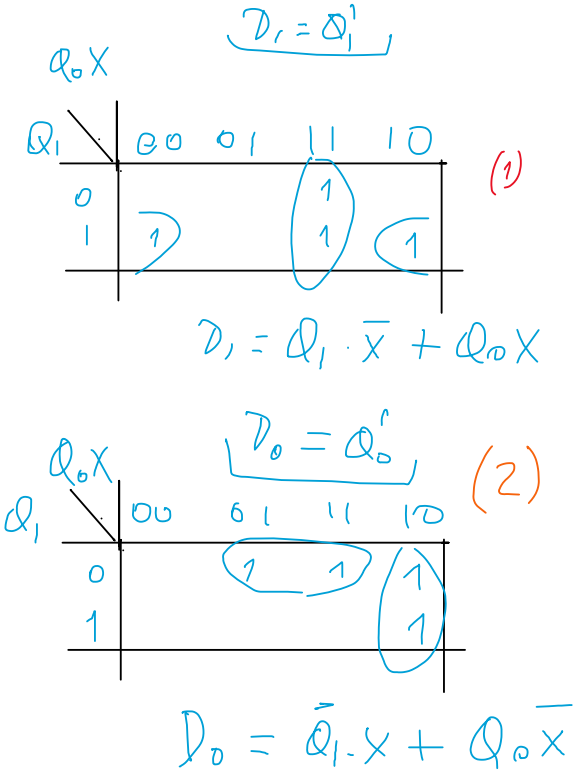
Obter diagrama e estados, tabla de cambio de estados, esquemático de circuito. Valórase número de portas mínimo.



1) Crear la tabla de transiciones a partir del enunciado

2) Generar unas funciones simplificadas para cada una de las transformaciones de estados: entradas a los biestables.

Como se pide utilizar biestables D, la entrada para el biestable es exactamente el valor que queremos almacenar.



Circuitos combinacionales útiles

Deseña un circuíto que use como elementos de memoria *flip-flop's* tipo D, con activación positiva de flanco de reloxio, e unha única entrada de habilitación X que se comporte da seguinte maneira:

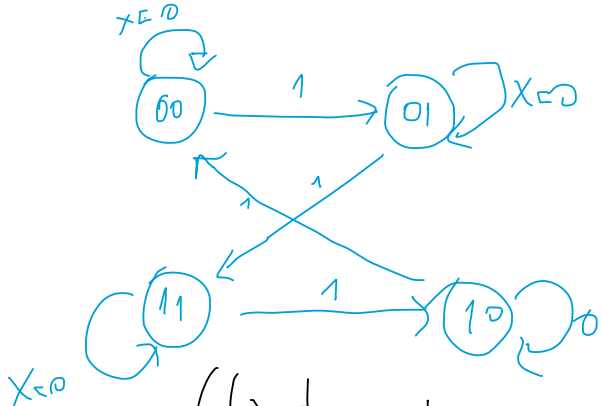
a) cando X=0, o sistema non cambia de estado, b) cando X=1 o sistema cambia de estado de forma síncrona seguindo a seguinte secuencia Q_{--1},Q_0 = {00 -> 01 -> 11 -> 10 e volta a 00}

Obter diagrama e estados, tabla de cambio de estados, esquemático de circuito. Valórase número de portas mínimo.

Q_1 Q_0	X	Q_1' Q_0'
0 0	0	0 0
0 0	1	0 1
0 1	0	0 1
0 1	1	1 1
1 1	0	1 0
1 1	1	1 0
1 0	0	1 0
1 0	1	0 0

(1) (2)

3) Dibujar el circuito y el diagrama de estados

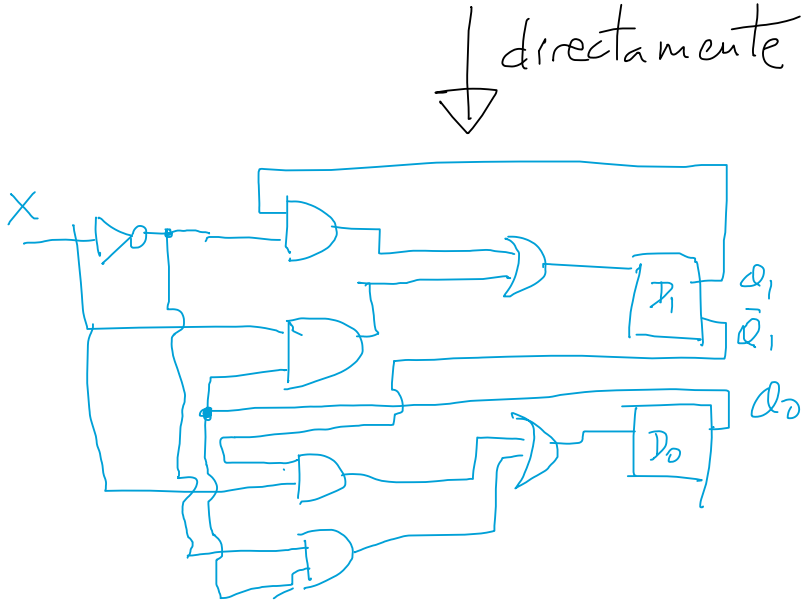


(directamente de la tabla)

$D_1 = Q_1 \cdot \bar{X} + Q_0 X$

$D_0 = \bar{Q}_1 \cdot X + Q_0 \bar{X}$

directamente



2. a) Deseña un circuíto full-adder de 1 bit utilizando aqueles elementos que necesites da seguinte lista:

- 4 multiplexores 4:1
- 4 inversores
- 4 portas NAND

Debuxa o esquemático e explica o procedemento de deseño que seguiches.

Nota: Busca una solución o mais optimizada que poidas en canto a número de compoñentes. Non é necesario utilizalos todos.

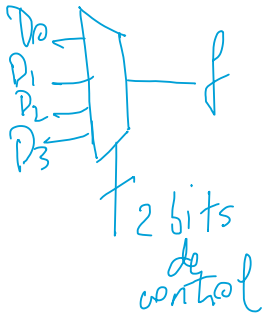
b) Utilizando o resultado do apartado a), ¿cómo poderías conectar varias instancias do circuíto anterior para facer un sumador de 3 bits? ¿ É suficiente o material que se propón no enunciado ?

A	B	C_{in}	C_{OUT}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

1) Un circuito full-adder es un sumador de un bit que tiene en cuenta un acarreo de entrada y genera uno de salida. Por tanto su tabla de verdad es la de la izquierda.

2) Se nos pide sintetizarlo usando los elementos indicados. Vamos a comenzar intentando realizar cada función con multiplexos 4:1

El MUX del que dispongo es el siguiente:



Como tengo 3 bits de entrada, usaré 2 como selección en el MUX para conectar uno de los 4 canales e entrada a la salida, pero, a su vez, cada una de estas entradas las pondré en función de la tercera variable.

	A	B	C_{in}	C_{OUT}	Σ
①	0	0	0	0	0
	0	0	1 ②	0	1
	0	1	0	0	1
	0	1	1	1	0
①	1	0	0	0	1
	1	0	1 ②	1	0
	1	1	0	1	0
	1	1	1	1	1

Suma Σ

Probamos usando, p.ej:, B y Cin para seleccionar en el MUX

① Selec. 00

A	B	Cin	Σ
0	0	0	0
1	0	0	1

$\Sigma = A$

② Selecc 01

A	B	Cin	Σ
0	0	1	1
1	0	1	0

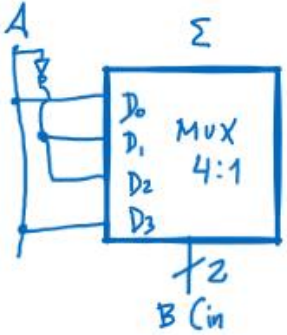
$\Sigma = \bar{A}$

de la misma manera:

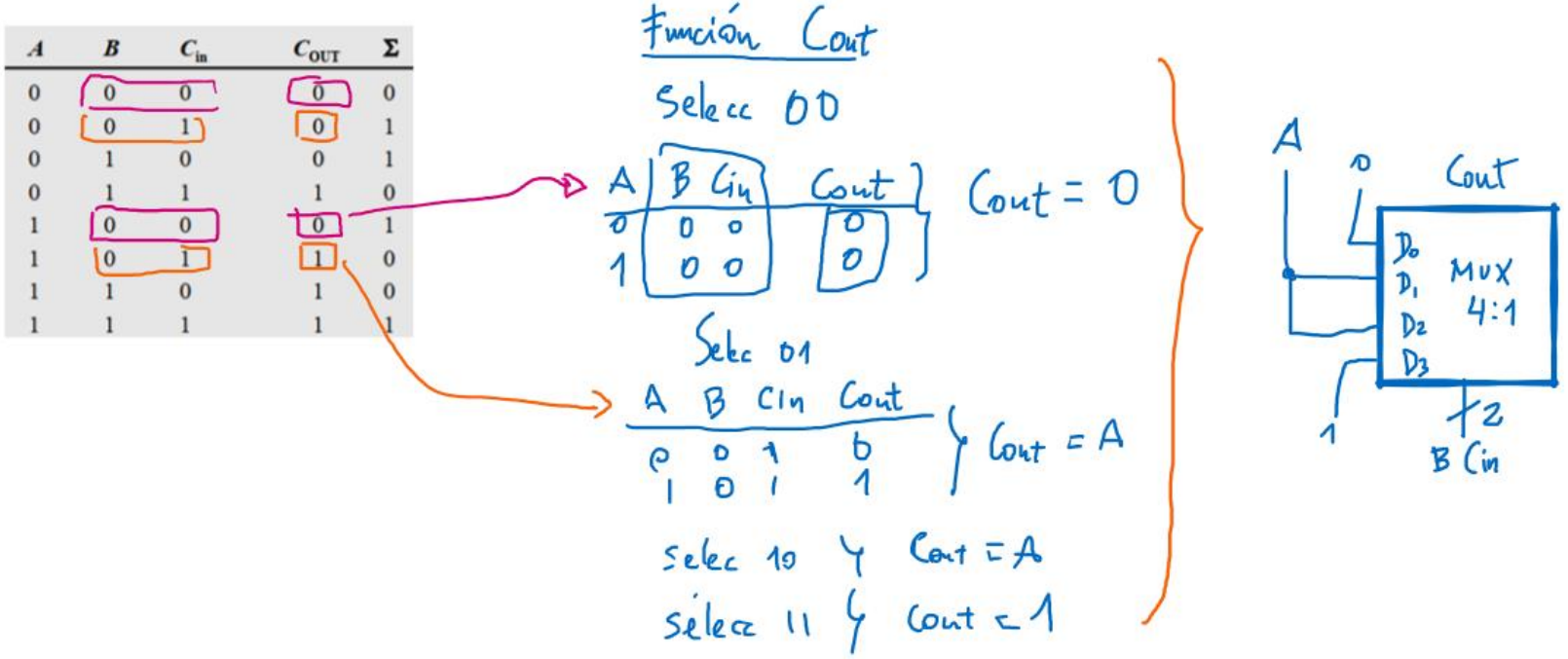
Selec 10 $\Sigma = \bar{A}$

Selecc 11 $\Sigma = A$

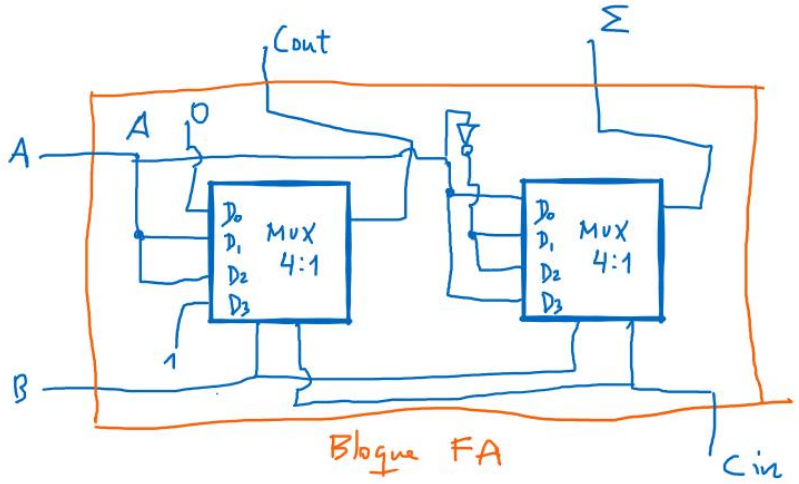
Por tanto la suma puede implementarse así:



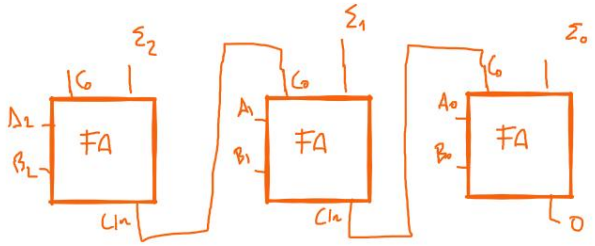
Del mismo modo determino una solución con otro MUX para la función del *carry* de salida:



El *full-adder* completo queda así:



Para hacer el sumador de 3 bits, uso 3 MUX y 3 inversores, con lo que me basta el material indicado:



$$\begin{array}{r} A \\ + \; B \\ \hline \Sigma \end{array} = \begin{array}{c} A_2\, A_1\, A_0 \\ B_2\, B_1\, B_0 \\ \hline \Sigma_2\, \Sigma_1\, \Sigma_0 \end{array}$$

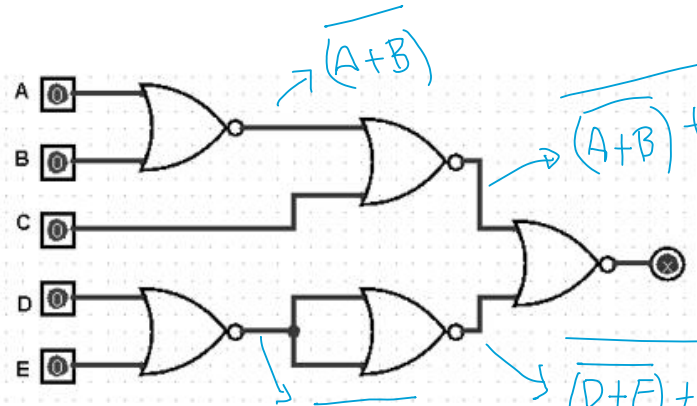
3. (**1,5 puntos**) Para o circuíto da figura comproba cal das seguintes expresión é correcta:

a) ((A+B)' + C) . (D'.E'))

b) ((A+B)' + C) . (D.E'))

c) (A+(B+C)') . (D'.E)

e) (A + B + C') . (D'.E')



$\overline{(A+B)}$

$\overline{(A+B)} + C$ } (1)

$\overline{(D+E)} + \overline{(D+E)} = \overline{\overline{(D+E)}} = D+E$ (2)

$\overline{(D+E)}$

(1) $\overline{\left(\overline{(A+B)} + C\right)} = (A+B)\cdot\overline{C}$

↑ de Morgan

Poniendo (1) + (2) $\Rightarrow$ $F = \overline{\left((A+B)\cdot\overline{C} + (D+E)\right)} = \left(\overline{(A+B)\cdot\overline{C}}\right)\cdot\overline{(D+E)} =$

de Morgan ↓ de Morgan ↓

$= \left(\overline{(A+B)} + C\right)\cdot(D'\cdot E') \Rightarrow$ Solución: Opc (a)

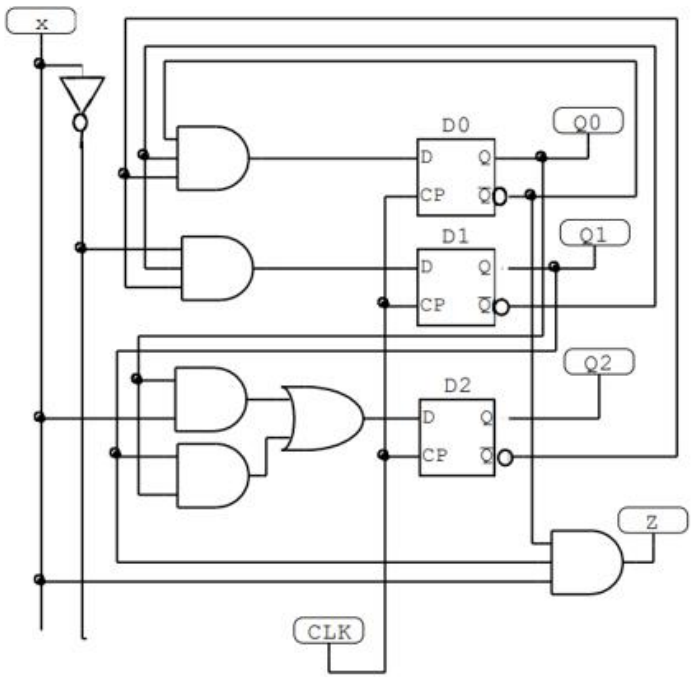
4. (2,5 puntos) No circuíto da seguinte figura **x** e a entrada o sistema, **z** a saída, e o estado actual ven dado pola palabra que definen os elementos de memoria que se amosan $Q_2Q_1Q_0$.

a) obter as funcións que definen a entrada a cada *flip-flop* tipo D.

b) obter a táboa de transición de estados e indicar si pertence a un modelo de *Mealy* ou *Moore* xustificando a resposta.

c) Obter un diagrama de estados e describir qué función realiza este circuíto. Descarta aqueles estados que non aporten ningunha operación útil ó circuíto.

e) Obter a saída para a seguinte secuencia de entrada: 000101001011101.



Inspeccionando el circuito las funciones de estado siguiente quedan:
(son biestables D, $Q_i = D_i$)

$$D_0 = Q_2' \cdot Q_1' \cdot Q_0'$$

$$D_1 = x' \cdot Q_2' \cdot Q_1'$$

$$D_2 = x Q_0 + Q_1 Q_0$$

$$Z = Q_1 \cdot Q_0' \cdot X$$

$D_0 = Q_2' \cdot Q_1' \cdot Q_0'$

$D_1 = x' \cdot Q_2' \cdot Q_1'$

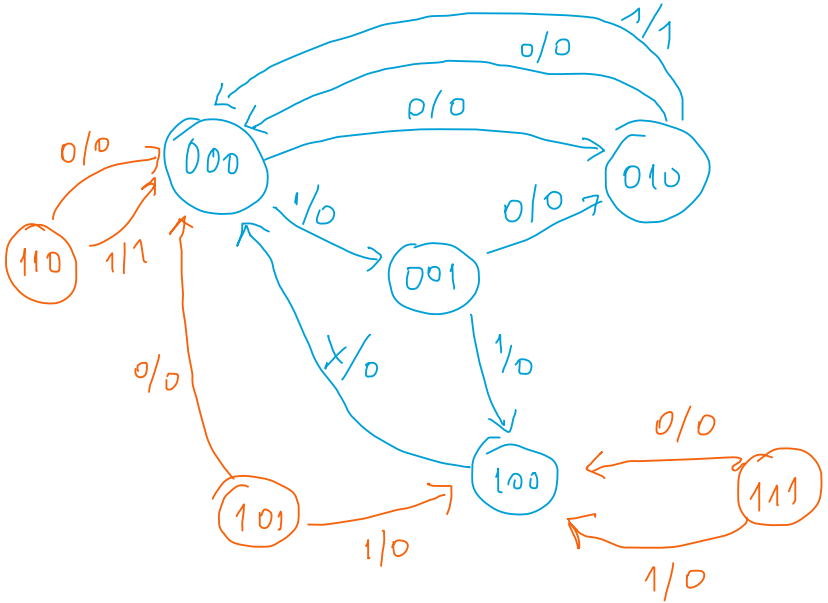
$D_2 = x Q_0 + Q_1 Q_0$

$Z = Q_1 \cdot Q_0' \cdot X$

El diseño es un modelo de Mealy claramente ya que la salida en cada momento se genera a partir del estado actual y de la entrada.
La tabla de transición de estados se puede obtener a partir de las ecuaciones directamente:

	X	
$Q_2 Q_1 Q_0$	0	1
000	011/0	001/0
001	010/0	100/0
010	000/0	000/1
011	100/0	100/0
100	000/0	000/0
101	000/0	100/0
110	000/0	000/1
111	100/0	100/0

$Q_2Q_1Q_0$	X = 0	X = 1
000	011 / 0	001 / 0
001	010 / 0	100 / 0
010	000 / 0	000 / 1
011	100 / 0	100 / 0
100	000 / 0	000 / 0
101	000 / 0	100 / 0
110	000 / 0	000 / 1
111	100 / 0	100 / 0



A los estados "naranjas" no se llega nunca por evolución del circuito. No nos interesan más que como un posible caso de condiciones iniciales

Ante la entrada:

X = 000 101 001 011 101 da:

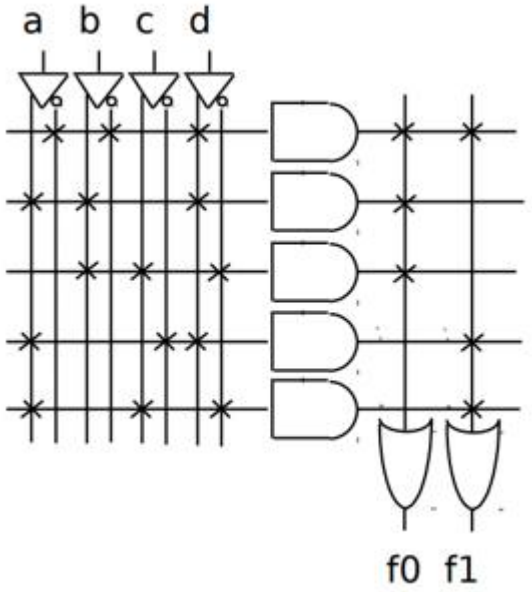
Z = 000 001 000 000 001

Detecta "101" en la secuencia de entrada

5. (2 puntos) Dado o circuíto PLA da figura obtén unha implementación alternativa das 2 función f_1 e f_0 das seguintes maneiras:

a) PAL con 4 funcións OR de salida (cada unha con catro términos de producto) e 4 entradas para obter ambas funcións.

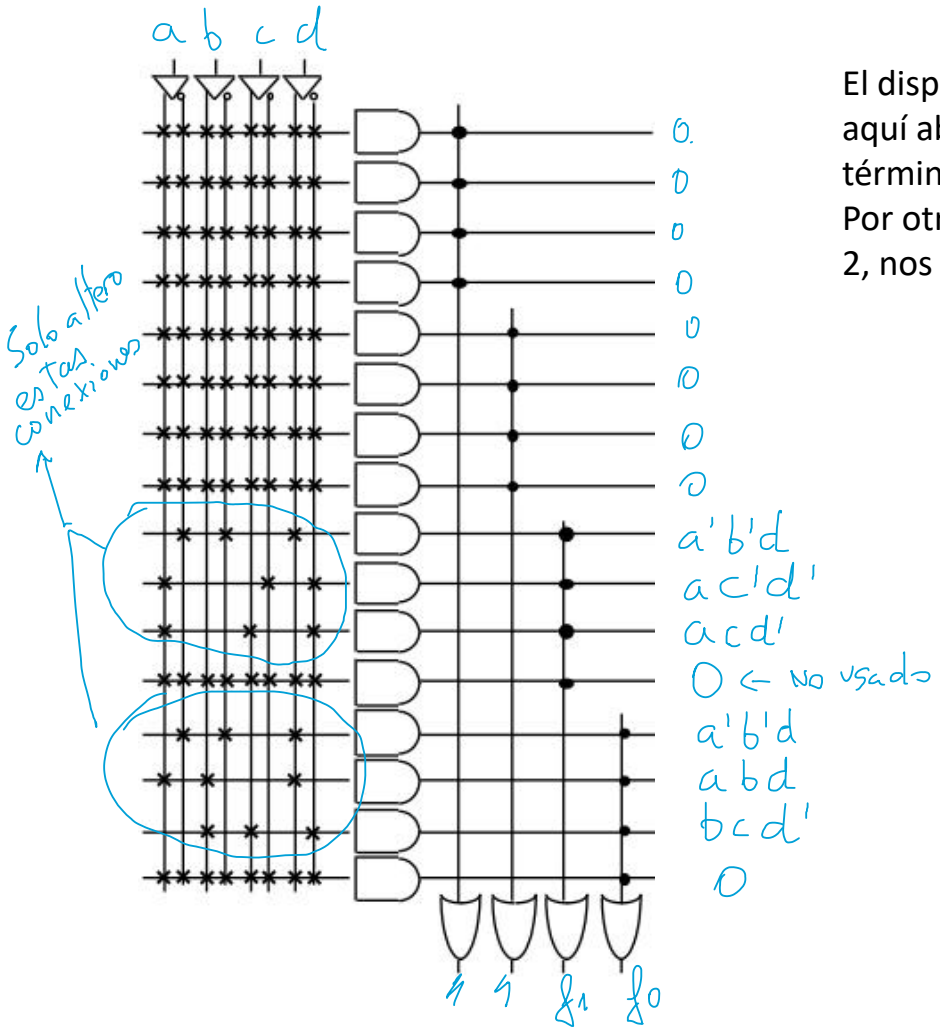
b) Multiplexores 4:1 (un para cada función).



Inspeccionando la PAL se obtienen las 2 funciones que implementa:

$f_0 = a'b'd + abd + bcd'$

$f_1 = a'b'd + ac'd + acd'$



El dispositivo PAL que se pide utilizar en el apartado a) es el que se muestra aquí abajo. La parte de la salida ya viene preprogramada. Tiene hasta 4 términos OR nuestras funciones sólo 3. Es más que suficiente.
Por otro lado implementa hasta 4 funciones de salida. Nosotros sólo tenemos 2, nos sobran los otros 2 canales de salida quedan sin usar.

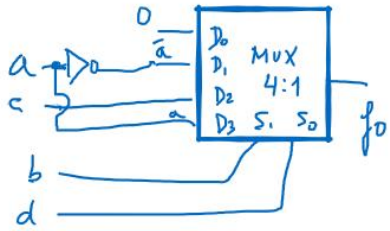
La implementación con MUX 4:1 es una vez más (ver ejercicio del Full-adder) una cuestión de usar 2 de los bits como selección y ver la dependencia de las otras 2 variables.
Vemos que los términos de la función sólo dependen de 3 variables así que la dependencia en cada caso es sólo de una “tercera” variable.
Para que quede un circuito sencillo es mejor usar como bits de selección aquellos que presenta mayor presencia en los términos POS.

$f_0 = a'b'd + abd + bcd'$

Un MUX a nivel algebraico se puede ver de la siguiente manera:

caso 00 → caso 01 → caso 10 → caso 11

$$f_0 = 0\cdot(b'd') + a'(b'd) + c(bd') + a(bd)$$



$f_1 = a'b'd + ac'd + acd'$

Para f_1 es mejor elegir a y d

$$f_1 = 0(a'd') + b'(a'd) + c(ad') + c'(ad)$$