FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

DECODIFICADORES Y MULTIPLEXORES

Prof. L. F. Caram

DECO Y MUX

Ejercicio 2: BCD a 7 segmentos

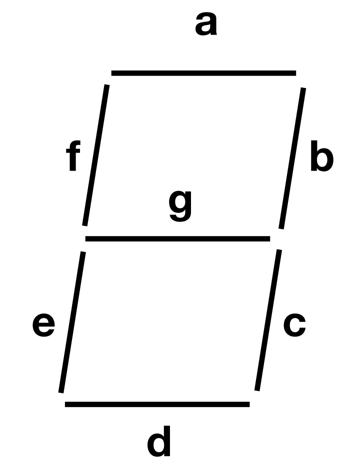
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

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

$$a = BD + \overline{B}\,\overline{D} + A + C$$

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

$$b = CD + \overline{C}\,\overline{D} + A + \overline{B}$$



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

$$c = D + \overline{C} + A + B$$

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

$$d = C\overline{D} + \overline{B}\,\overline{D} + A + \overline{B}C + B\overline{C}D$$

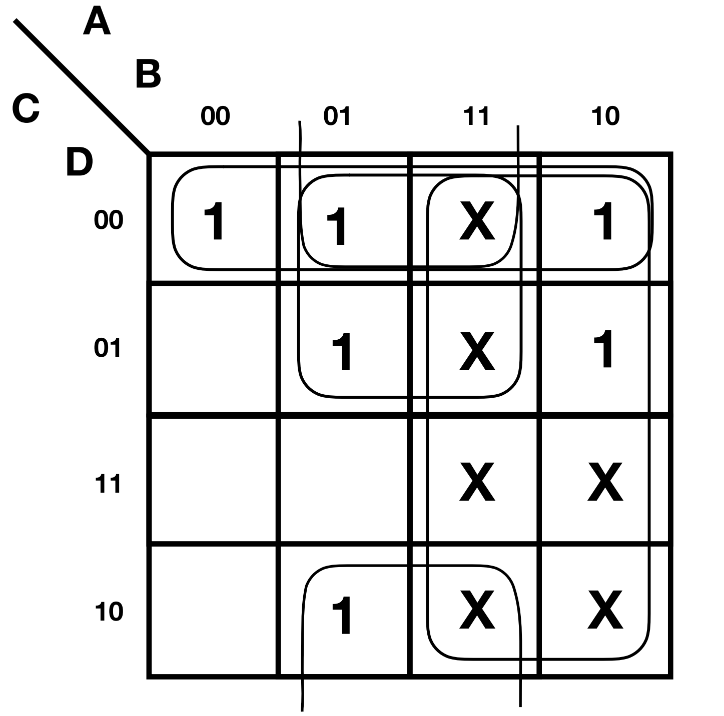
CD \ AB	00	01	11	10
00	1		X	1
01			X	
11			X	X
10	1	1	X	X

$$e = C\overline{D} + A\overline{D} + \overline{B}\,\overline{D}$$

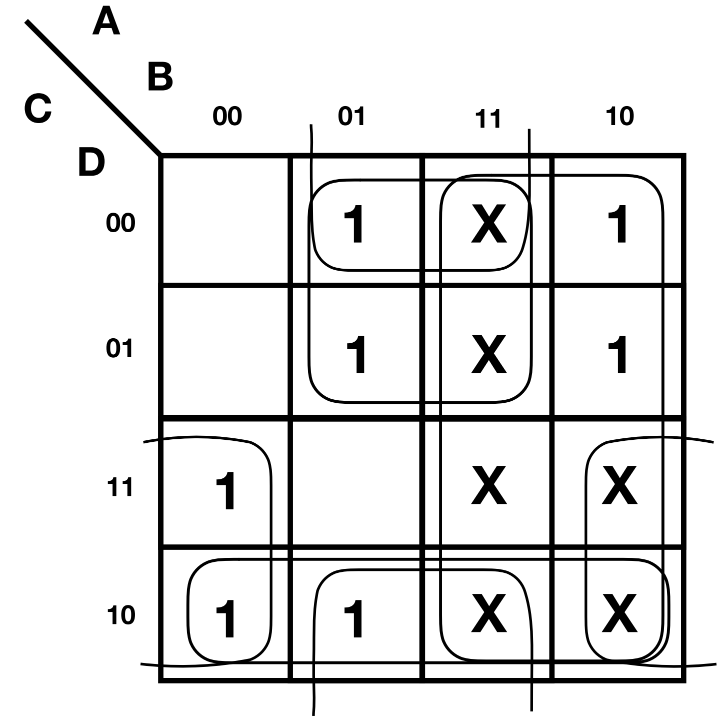
DECO Y MUX

Ejercicio 2: BCD a 7 segmentos

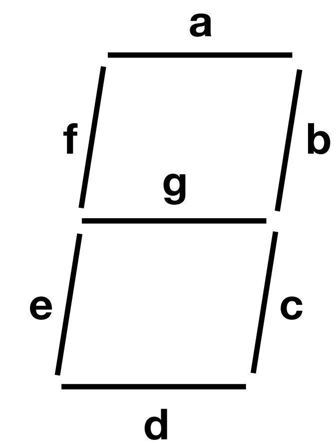
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X



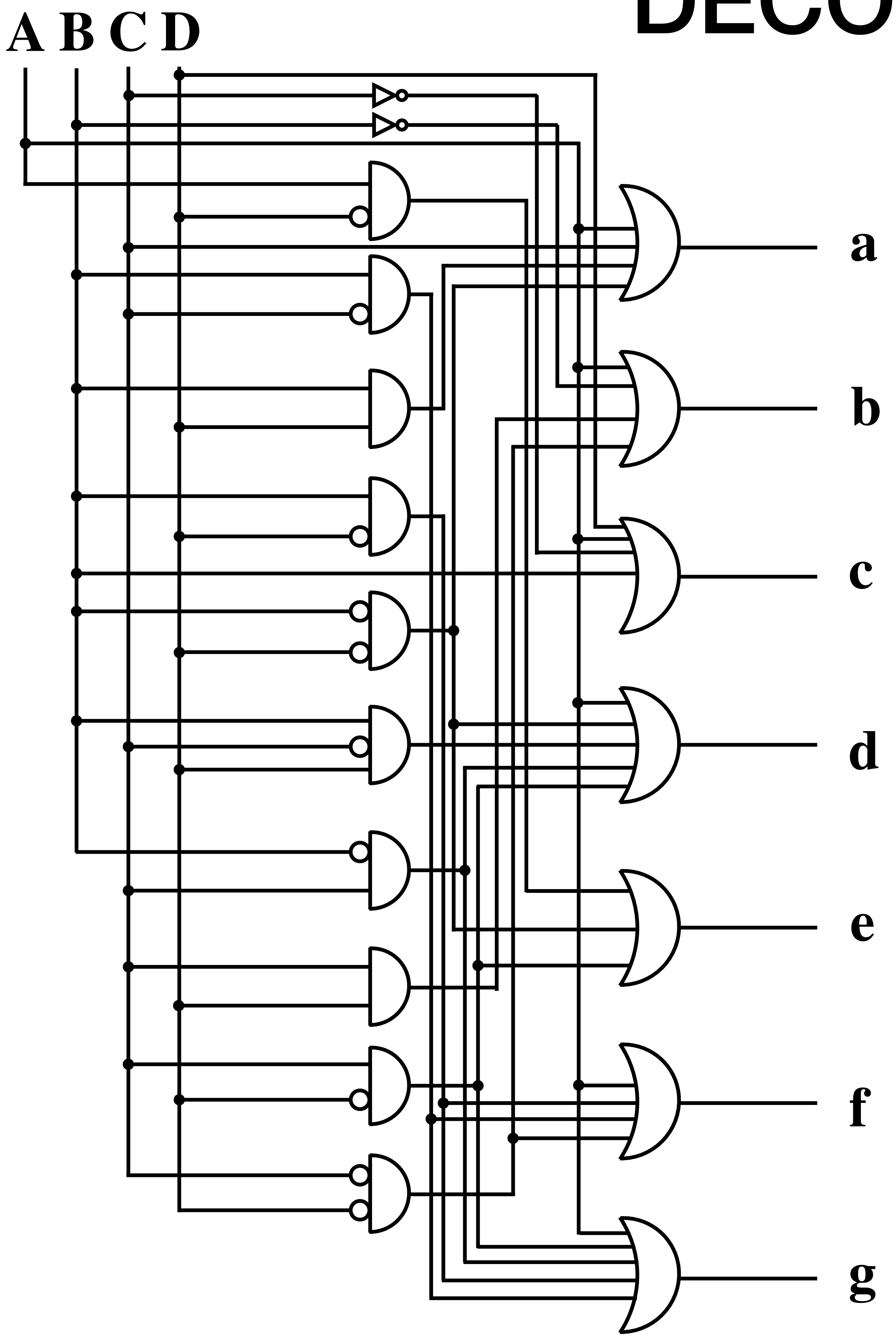
$$f = \overline{C}\,\overline{D} + B\overline{D} + A + B\overline{C}$$



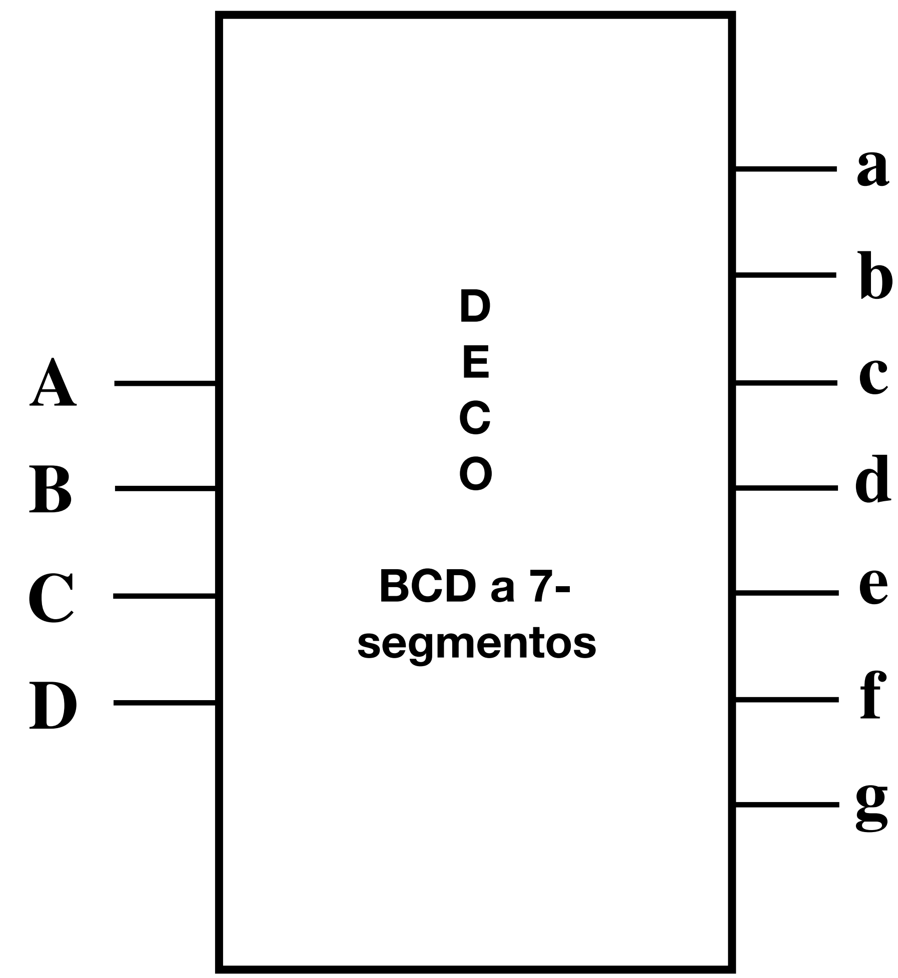
$$g = C\overline{D} + B\overline{D} + A + \overline{B}C + B\overline{C}$$

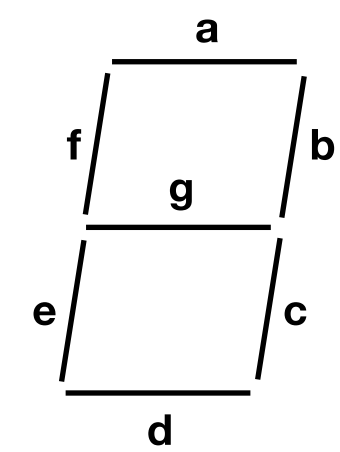


DECO Y MUX



≡

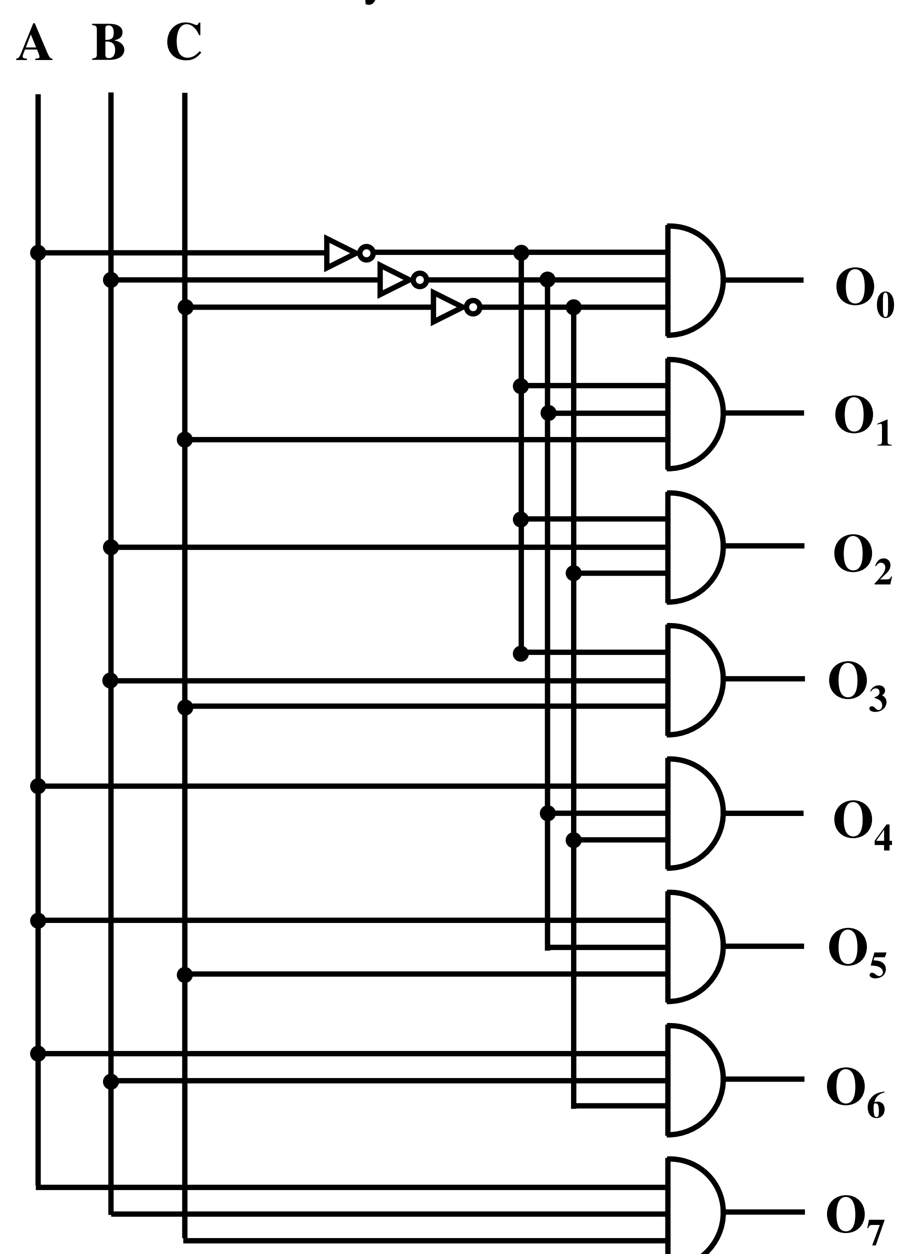




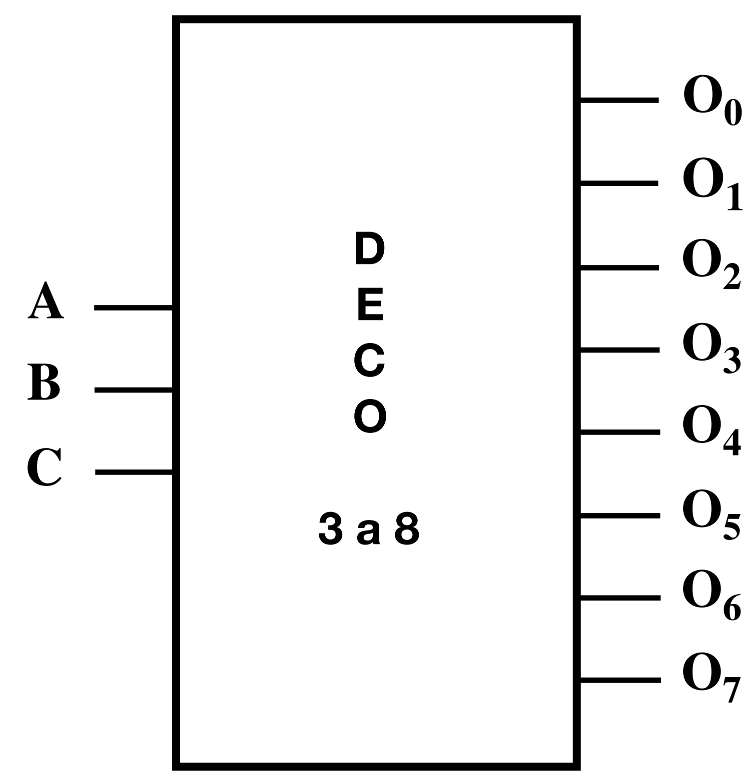
DECO Y MUX

Los decodificadores son dispositivos que colocan una salida en 1/0 (activo alto/bajo) y el resto en 0/1 cuando el código binario de entrada representa el valor decimal correspondiente a la salida. Existen activo alto salida activa en 1 y el resto en 0 ,como en este ejemplo y el activo bajo salida activa en 0 y el resto en 1 como el la página siguiente. Se puede observar la tabla de verdad, el circuito interno y el símbolo circuital.

A	B	C	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



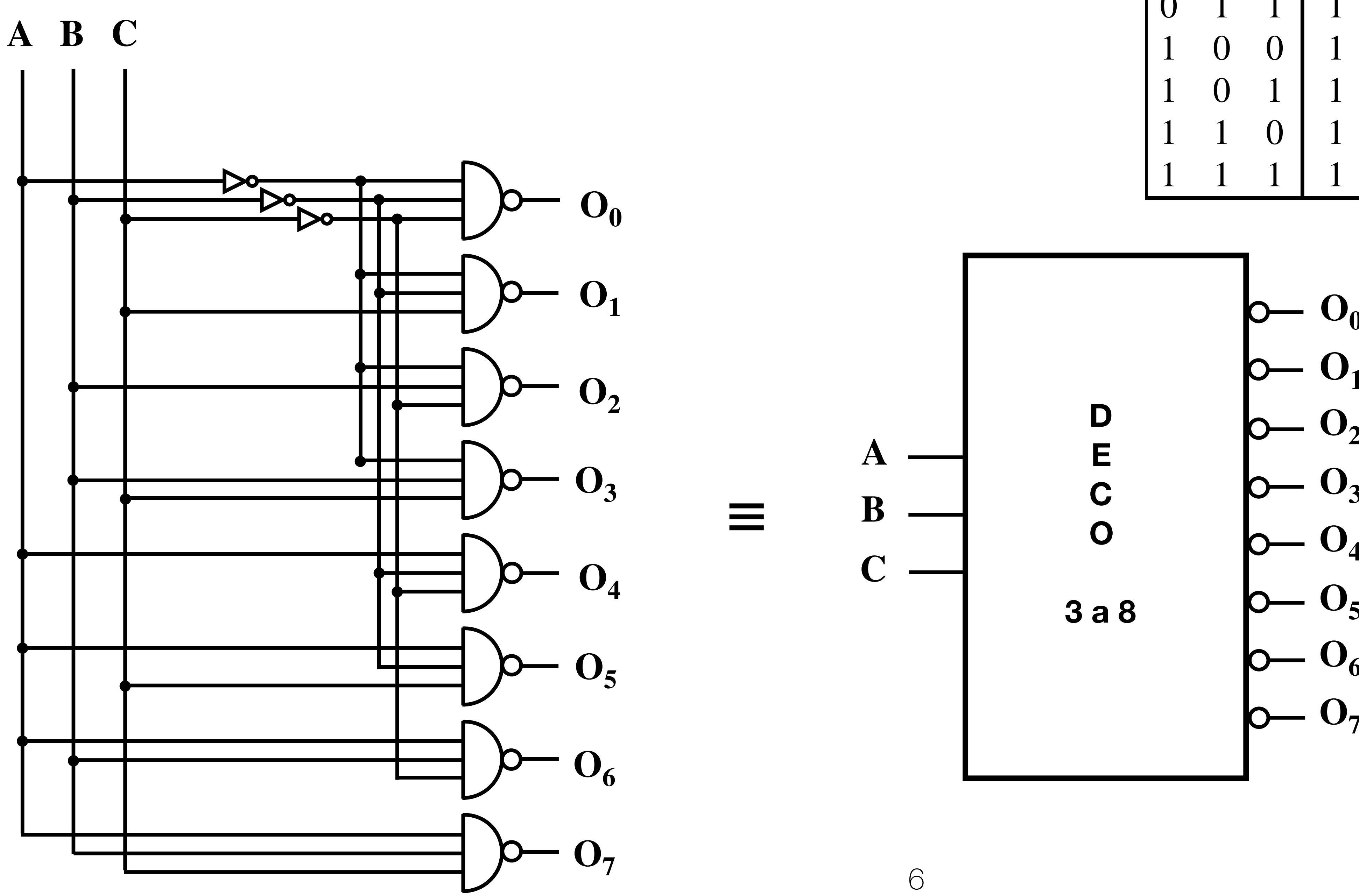
≡



DECO Y MUX

Decodificador activo bajo salida activa en 0 y el resto en 1. Se puede observar la tabla de verdad, el circuito interno y el símbolo circuital.

A	B	C	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

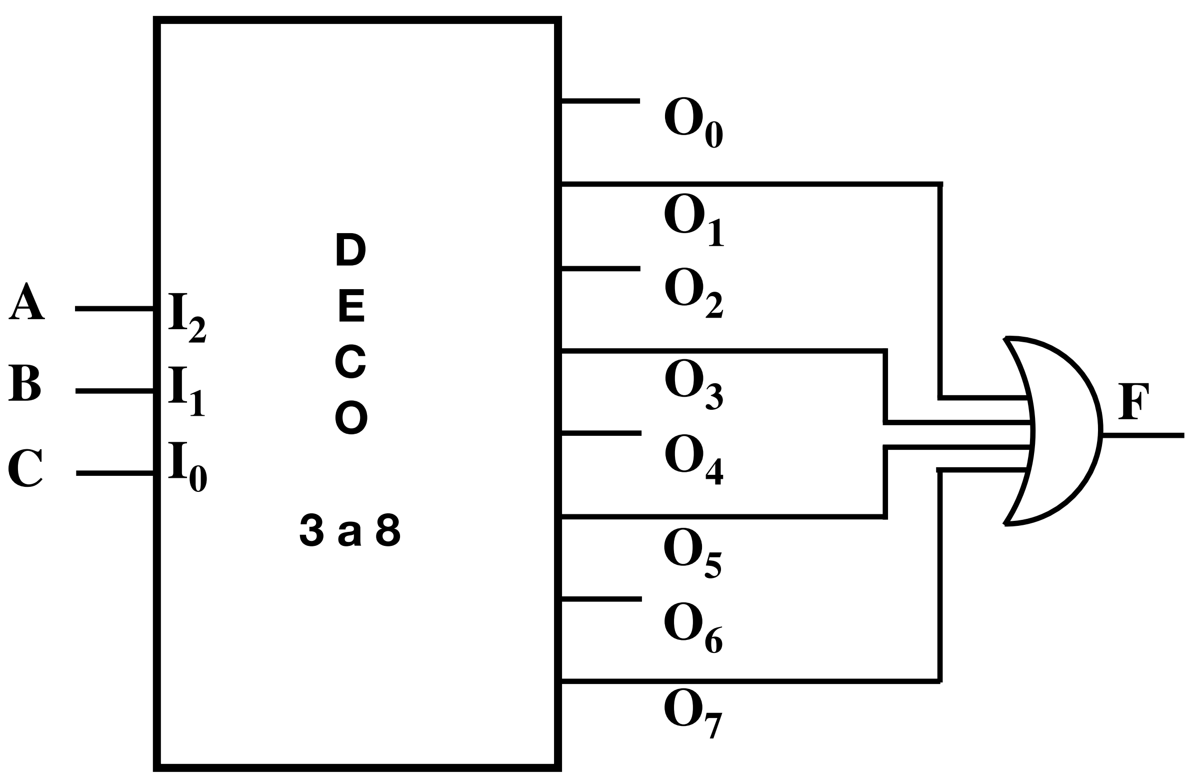


DECO Y MUX

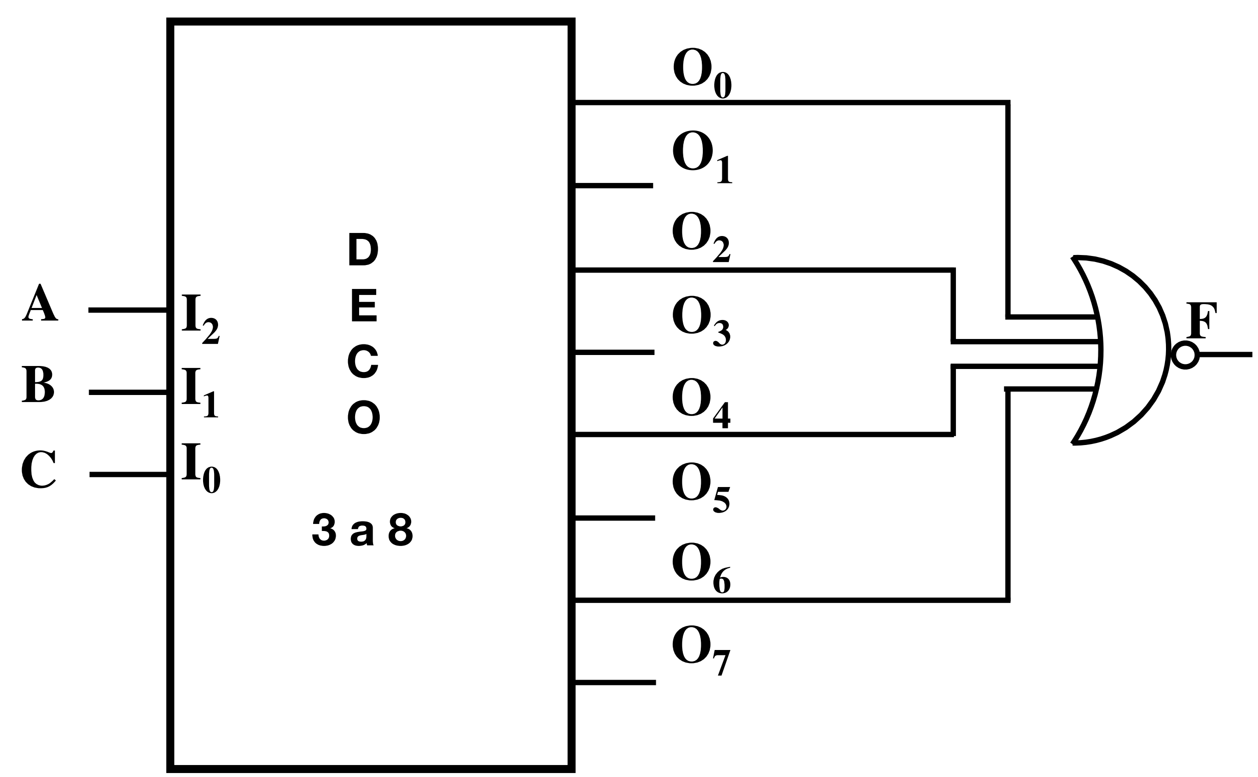
Veremos un ejemplo de implementación de una función con distintos tipos de salidas de decodificadores y distintas compuertas.

Ejemplo: implementar la función $f(A,B,C) = \sum_m (1,3,5,7)$ con un deco 3x8 y compuertas.

El mas simple es usar un deco con salidas activas altas y todos los minitérminos, uso una compuerta OR.

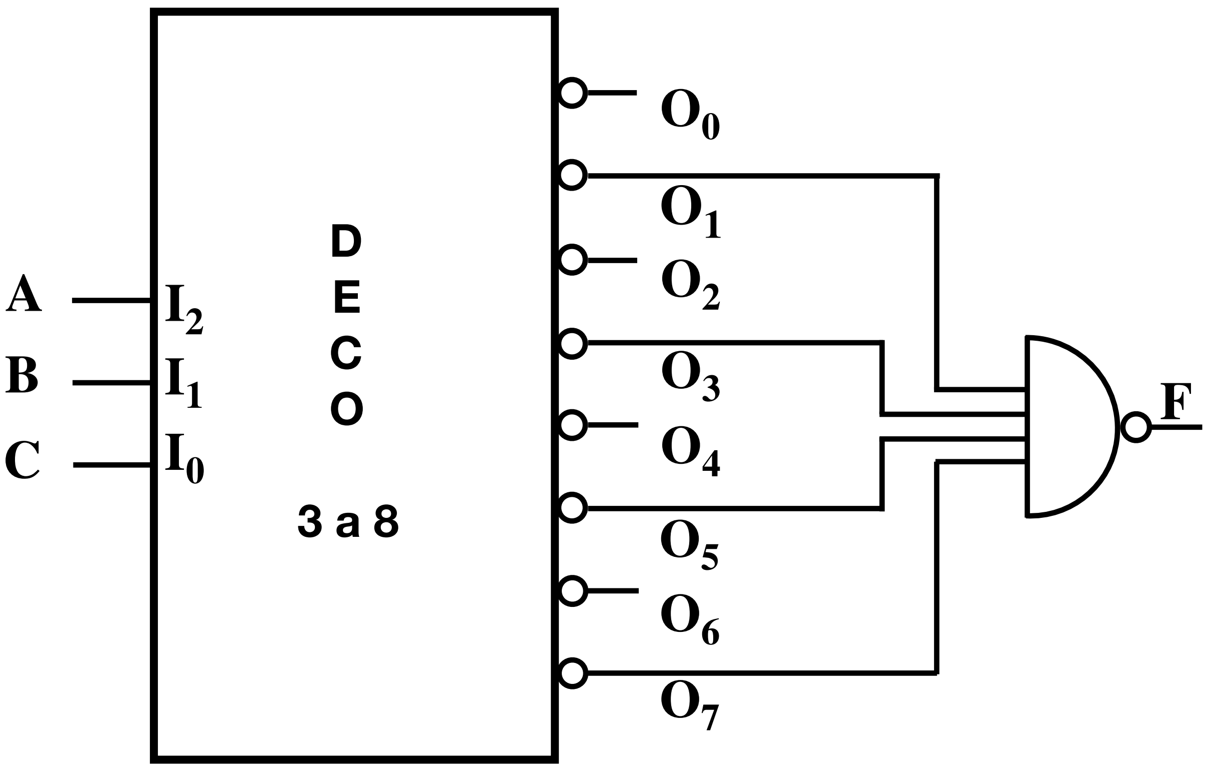


Si tomo el mismo deco pero tomo los maxitérminos, debo usar compuerta NOR.

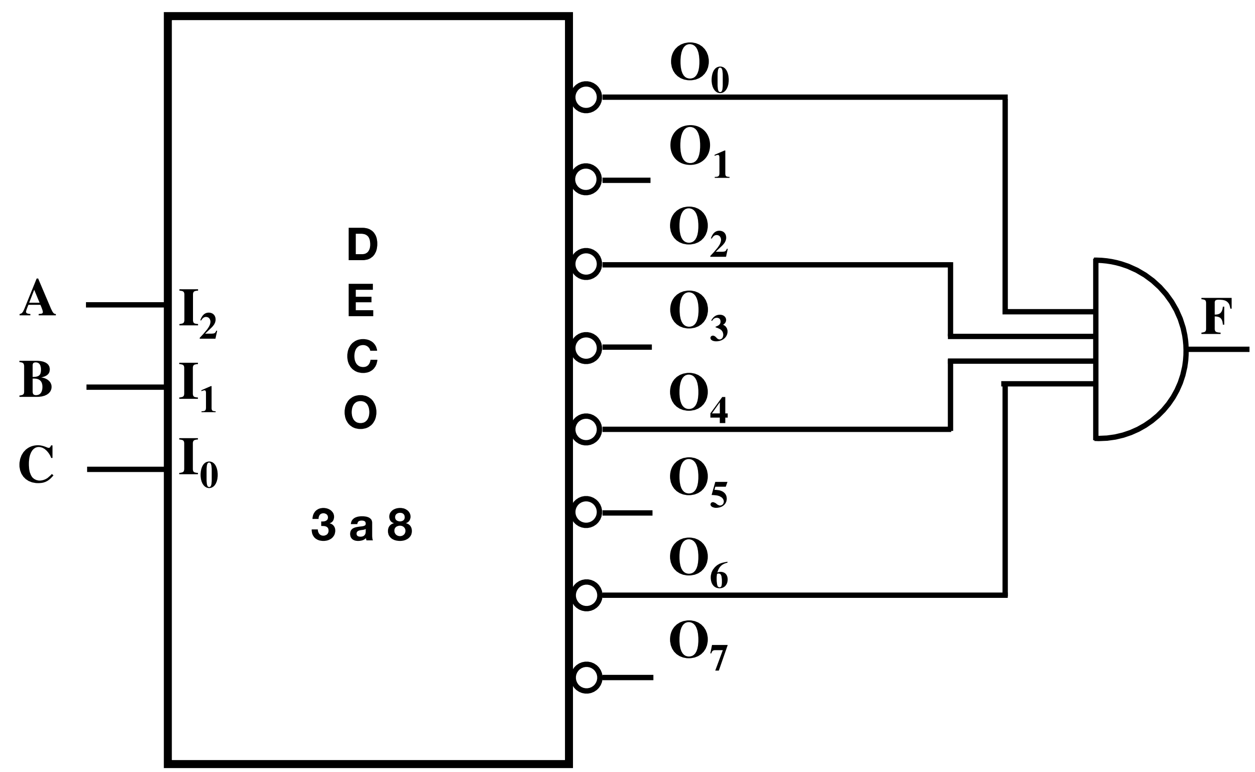


DECO Y MUX

Si ahora tengo un deco con salidas activas bajas y todos los minitérminos, debo usar una compuerta NAND.



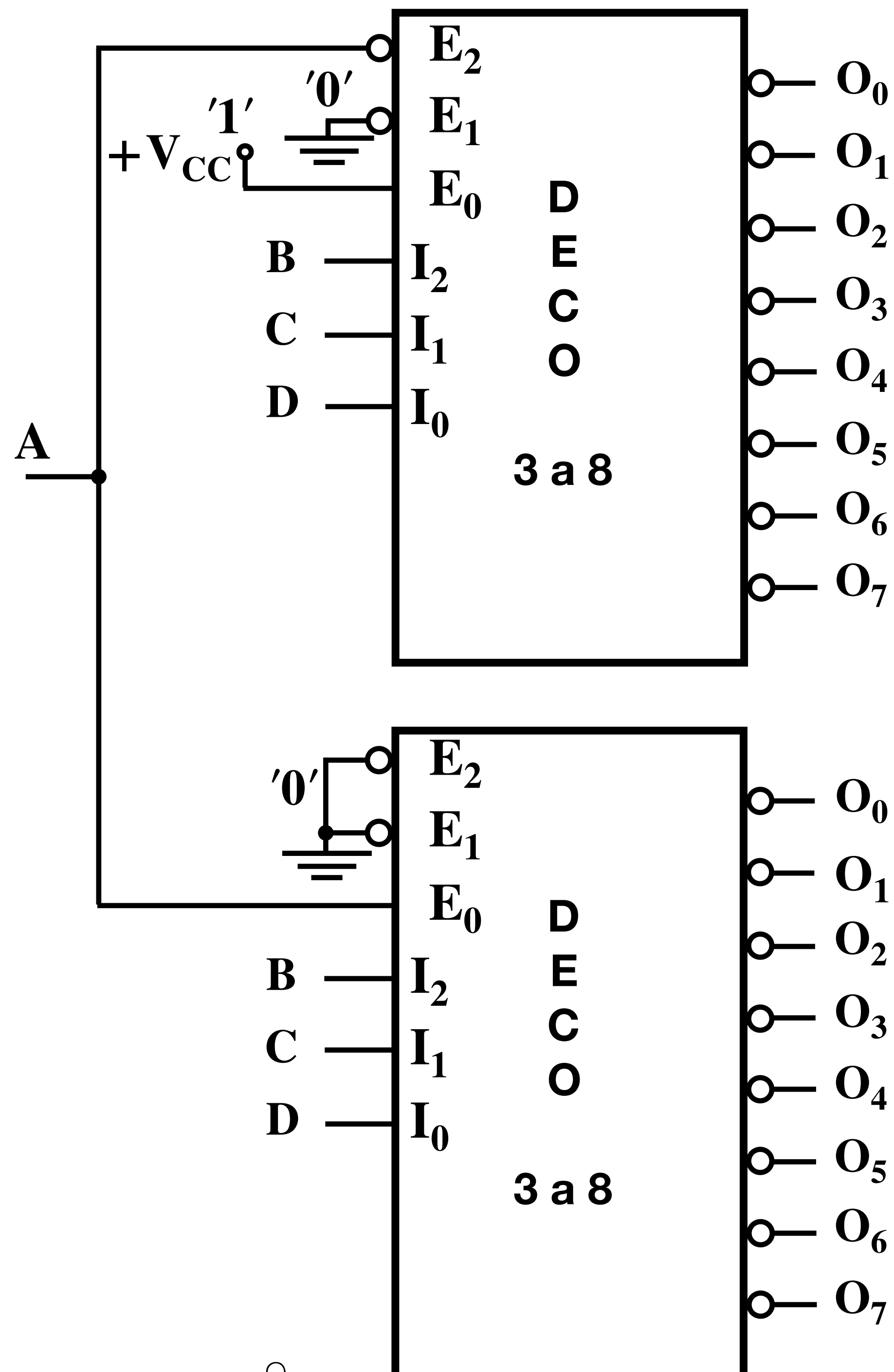
Si con el mismo deco tomo los maxitérminos, debo usar compuerta AND.



DECO Y MUX

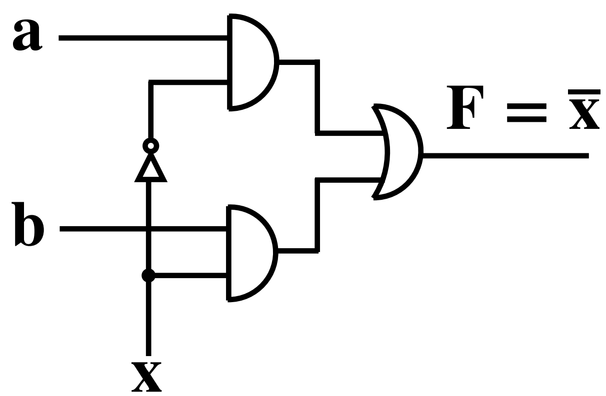
¿Qué pasa cuando necesito más salidas?, por ejemplo tengo un decodificador 74HC138 y quiero hacer uno de 16 salidas. Como se ve en el apunte del Ing. Fuchs en página 28, este deco tiene 8 salidas activas bajas y 3 entradas de control (2 activas bajas y 1 activa alta).

Solución: En este caso hay varias soluciones posibles (por tener más de 1 entrada de habilitación). Una es usar tres variables independientes para cada deco y la variable independiente más significativa usarla para habilitar y deshabilitar cada uno de los chips. Por ejemplo: tomemos A como variable más significativa y D como la menos.

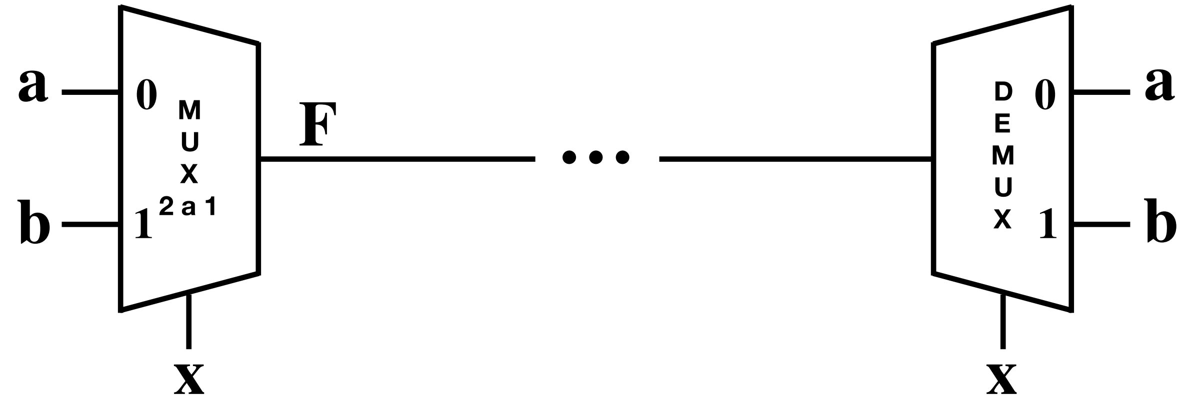


DECO Y MUX

<u>Ejercicios 9 y 13:</u>



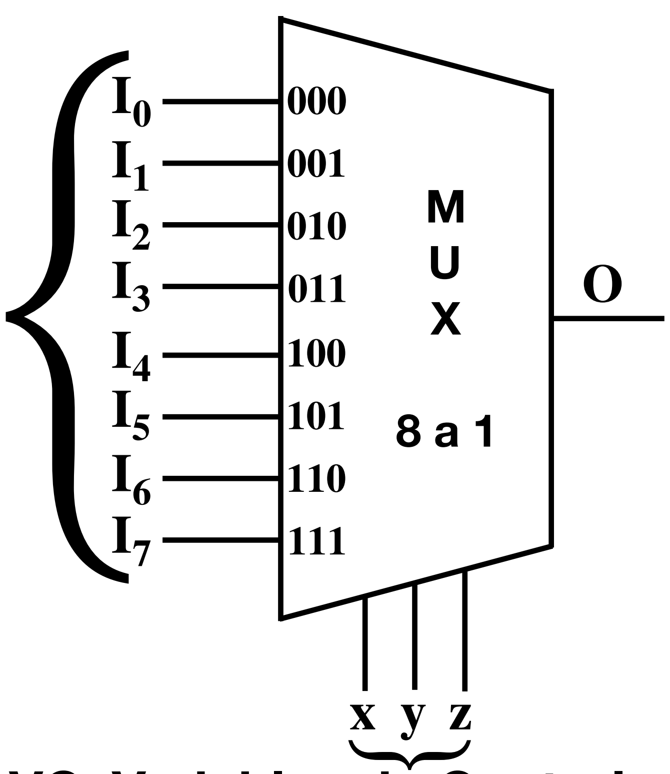
$F = \overline{x}\,.\,a + x\,.\,b$ $\equiv$



Los multiplexores (MUX)internamente tienen arreglos de compuertas AND'S, OR'S e INVERTERS. Como puede verificarse en la función de salida, la VC lleva ese nombre ya que su cambio de estado afecta directamente en la función de salida y como puede verse se repite en ambos términos negada y sin negar. El cambio de estado de VC de 0 a 1, hace que F=a pase a F=b, es como si se cambiara o conmutara la selección de la entrada que va a la salida. De esta forma se logra multiplexar en el tiempo las señales de las entradas y poder asi compartir un mismo medio de transmisión enviando pequeños slots de tiempo de cada una de las entradas.

$\boxed{\#VE = 2^{\#VC}}$

VE: Variables de Entrada



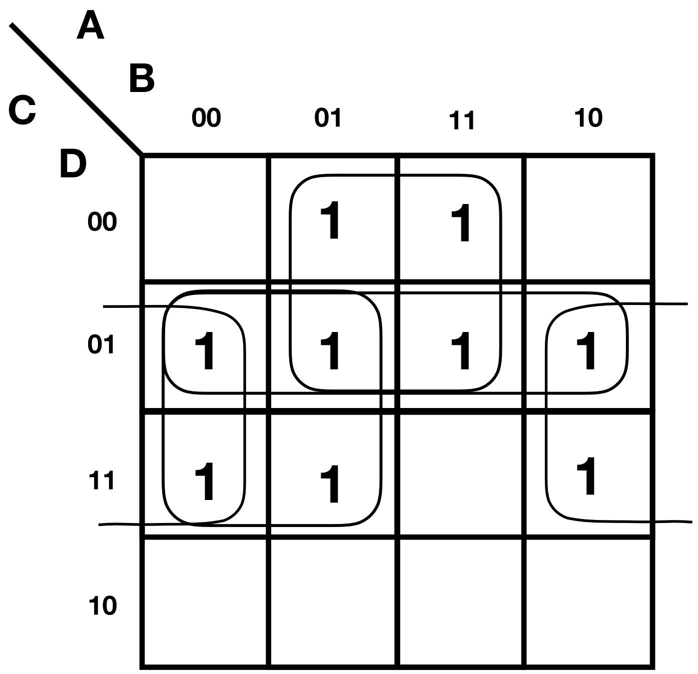
VC: Variables de Control

Los multiplexores son dispositivos que optimizan el medio de transmisión cuando se utilizan en comunicaciones. Funcionan como una llave electrónica que selecciona la VE según el estado de las VC y la conecta a la salida. Todos los MUX tienen 1 sola salida y tantas VE como la potencia de 2 de la cantidad de VC.
En comunicaciones se usan MUX en el extremo de envío de información y un DEMUX o demultiplexor en el extremo de recepción, éste posee 1 sola entrada y las VC están sincronizadas con las del MUX, tiene tantas salidas como 2 elevado al número de VC.

DECO Y MUX

Ejercicio 15: Implementación de funciones con MUX

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0



$$\boxed{F_{min} = \overline{A}D + B\overline{C} + \overline{B}D + \overline{C}D}$$

Para implementar funciones con multiplexores existen muchas opciones, según el tipo de MUX que tenga que usar, es decir de acuerdo a la cantidad de VC que tengan los MUX a usar. Una vez definidos, deberemos elegir del total de VI (variables independientes, en este caso A B C D) las VC "más adecuadas" según el MUX que tengamos que usar. Para ello existen varias posibilidades que se resuelven con el número combinatorio. Es decir número de VI, tomadas de a número de VC que necesite y como puede verse, de acuerdo al MUX a usar, tendremos distintas elecciones posibles y no todas serán óptimas, es decir no todas tendrán una cantidad mínima de multiplexores.

Veremos este ejercicio que pide usar MUX de 3 VC con todas la alternativas posibles para ver las diferencias entre las diferentes implementaciones.

El procedimiento consiste en dividir el mapa de Karnaugh en tantas regiones como VE del MUX a usar, una vez que elegí la o las VC, divido el mapa donde esas variables cambian de estado como veremos.

Elección de las VC

$$\binom{\#VI}{\#VC} = \#POSIBILIDADES$$

$$\binom{4}{4} = \frac{4!}{4!0!} = 1 \qquad \binom{4}{3} = \frac{4!}{3!1!} = 4$$

$$\binom{4}{2} = \frac{4!}{2!2!} = 6 \qquad \binom{4}{1} = \frac{4!}{1!3!} = 4$$

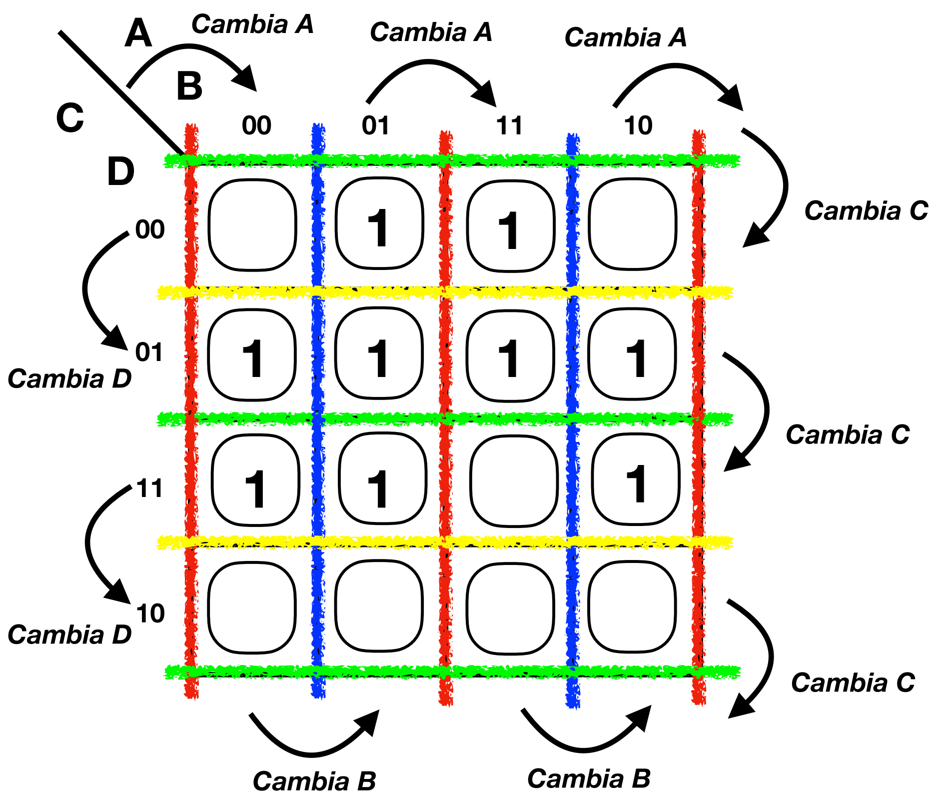
DECO Y MUX

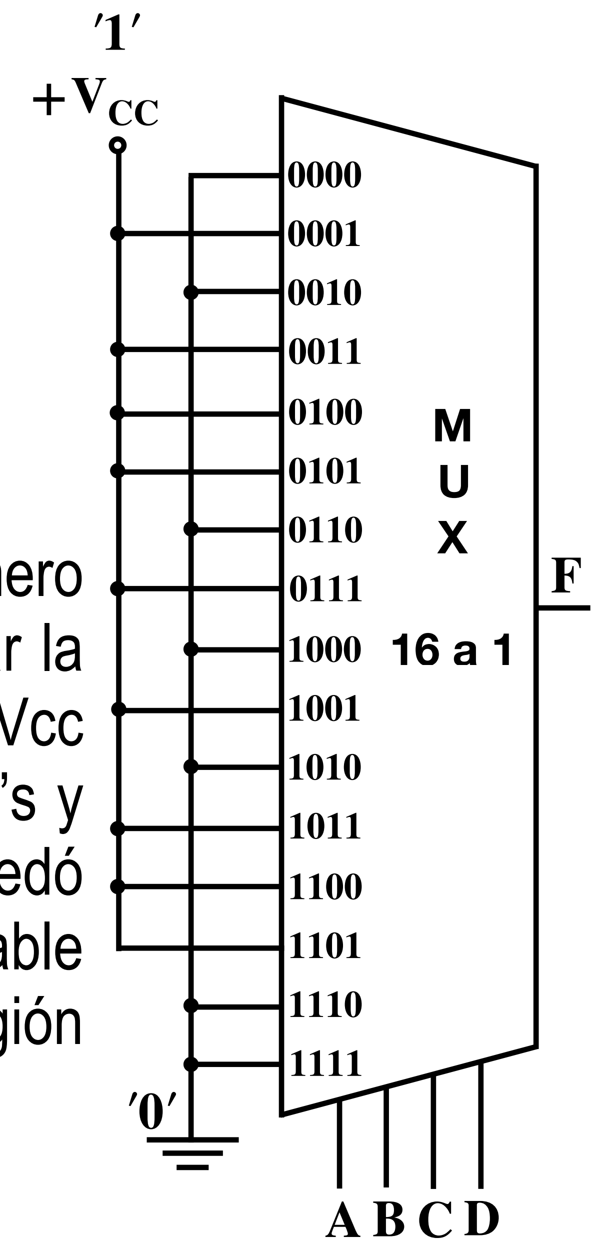
Ejercicio 15:

MUX de 4 VC VC: ABCD

MUX de 4 VC: tendré en las entradas solamente 1’s y 0’s, tabla de verdad.

$$\boxed{\binom{4}{4} = \frac{4!}{4!0!} = 1\,\text{posibilidad}}$$





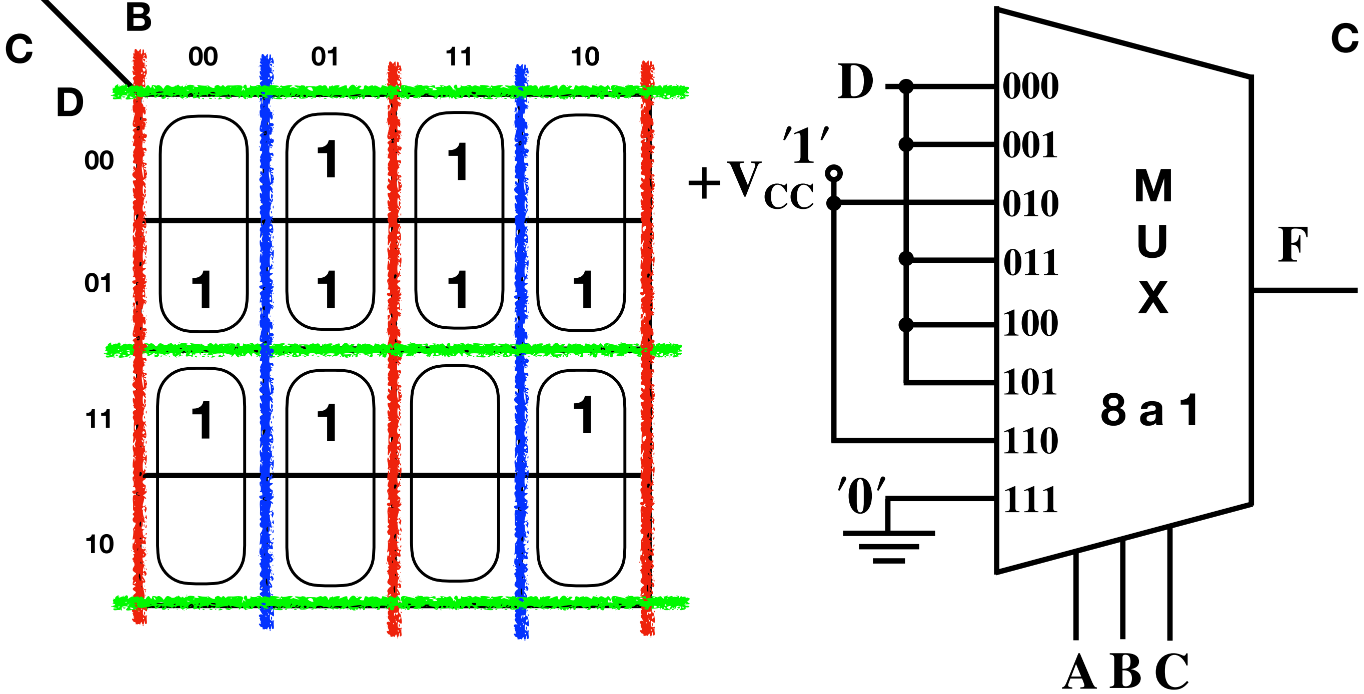
En esta caso cuando el MUX tiene el mismo número de VC que el número de VI de la función, la elección es única. Solamente tendré que colocar la tabla de verdad en las entradas del MUX, un ‘1' lógico lo conectaré a +Vcc y un ‘0' lógico lo pondré a masa. Asi las entradas posible serán 1's y 0’s y no necesito simplificar la función. Como se puede ver el mapa quedó dividido en 16 regiones que son las distintas VE del MUX. Cada variable tiene 2 cambios de estados de 0 a 1 y de 1 a 0. Dentro de cada región debo ver cuanto vale la función.

Ejercicio 15:

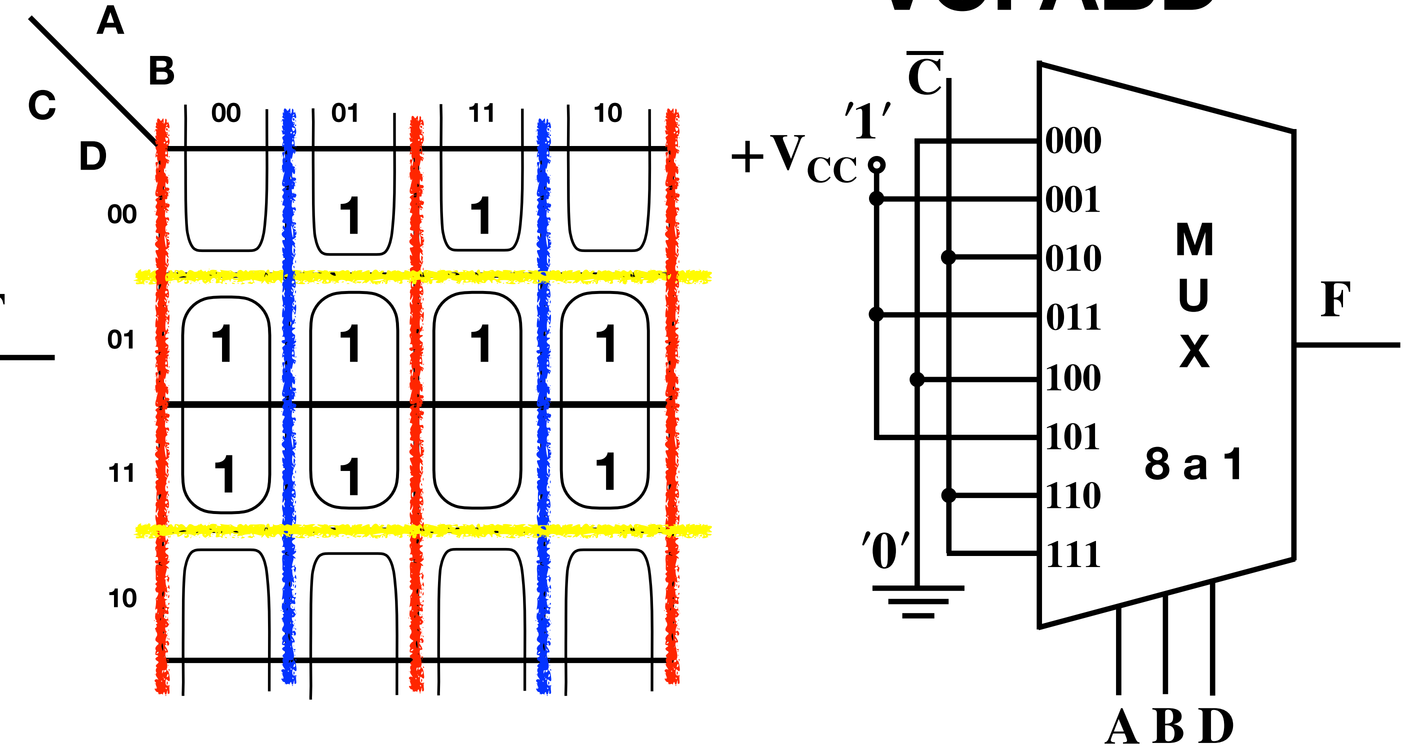
DECO Y MUX

MUX de 3 VC

VC: ABC



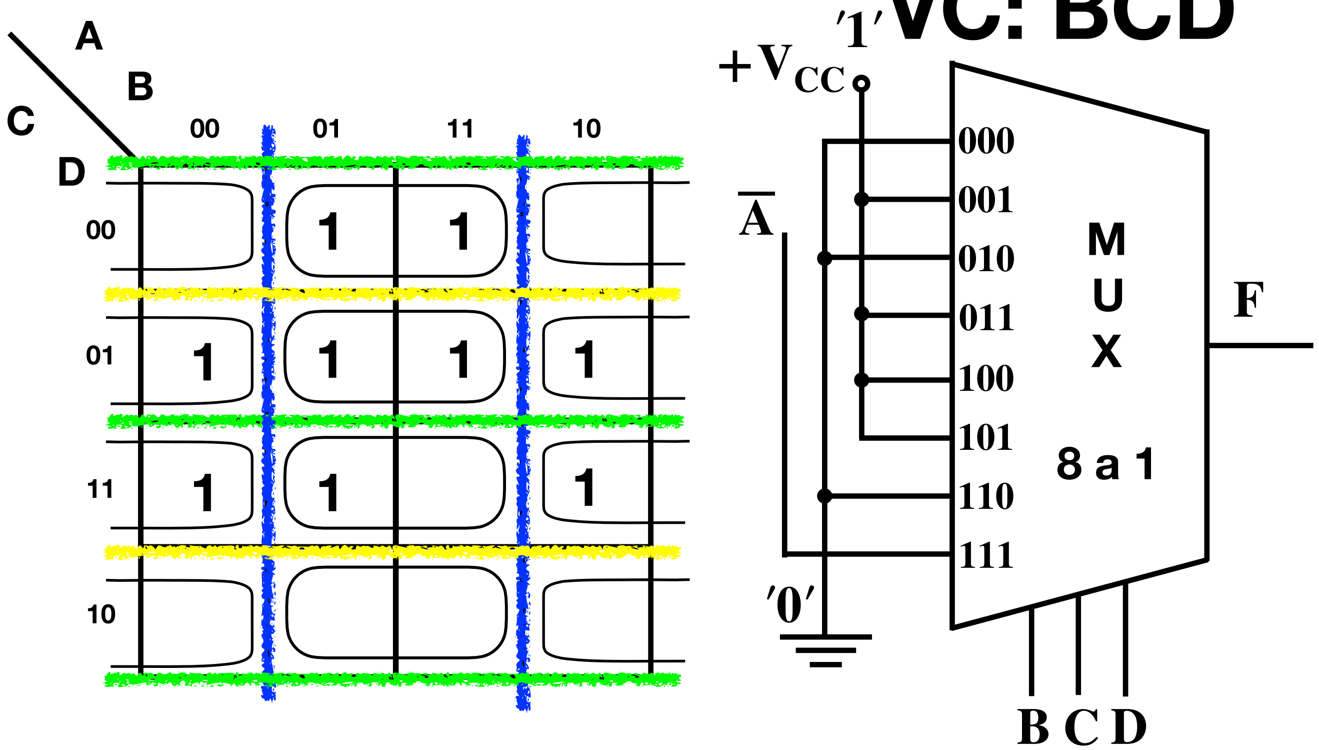
VC: ABD



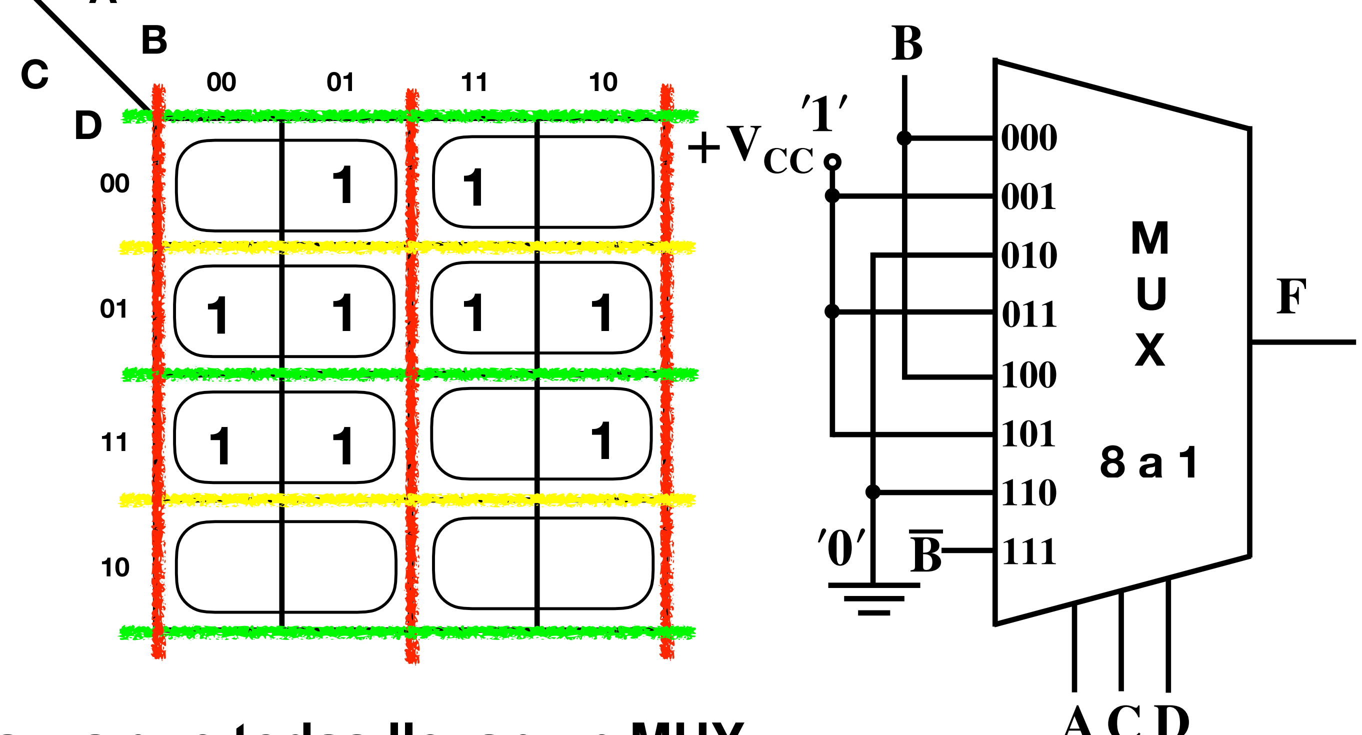
MUX de 3 VC: tendré en las entradas solamente 1's, 0's y la variable independiente no tomada negada y sin negar.

$$\binom{4}{3} = \frac{4!}{3!1!} = 4 \text{ posibilidades}$$

VC: BCD



VC: ACD

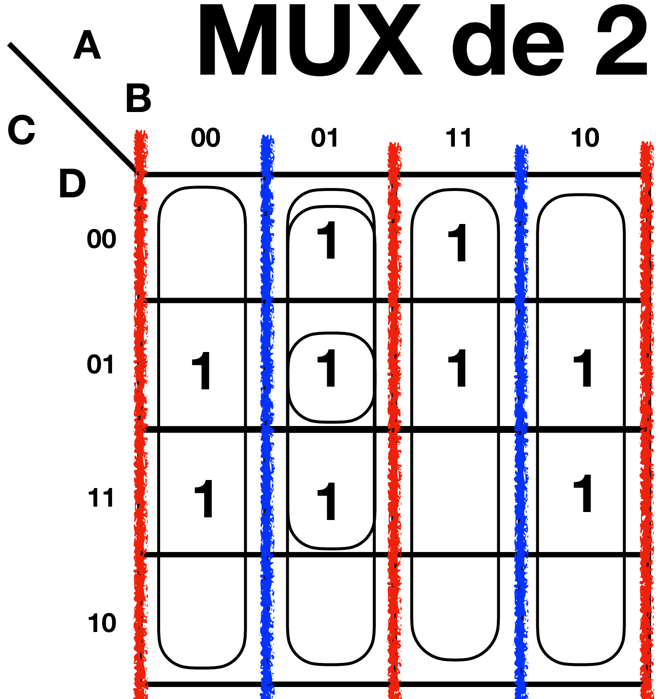


Con tres (n-1) VC, cualquier implementación es óptima, ya que todas llevan un MUX.

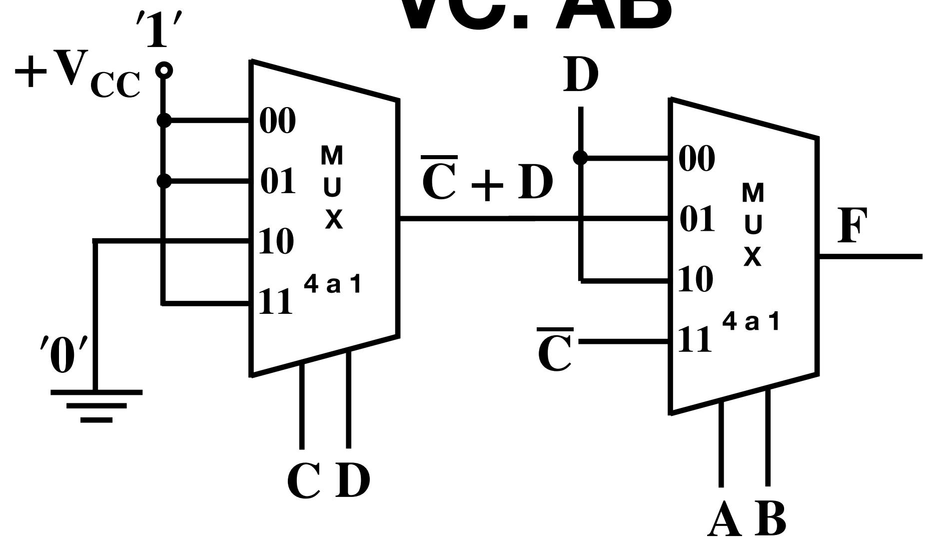
Ejercicio 15: $F_{min} = \overline{A}D + B\overline{C} + \overline{B}D + \overline{C}D$

DECO Y MUX

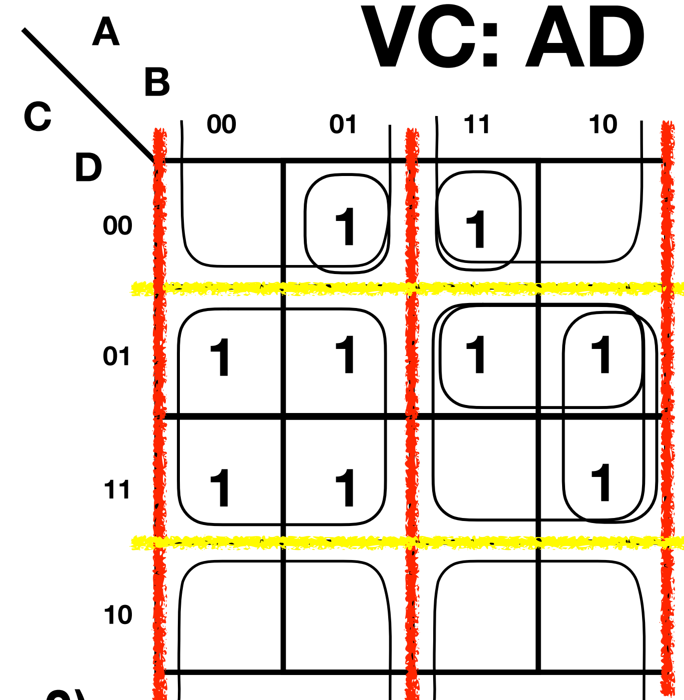
MUX de 2 VC



VC: AB

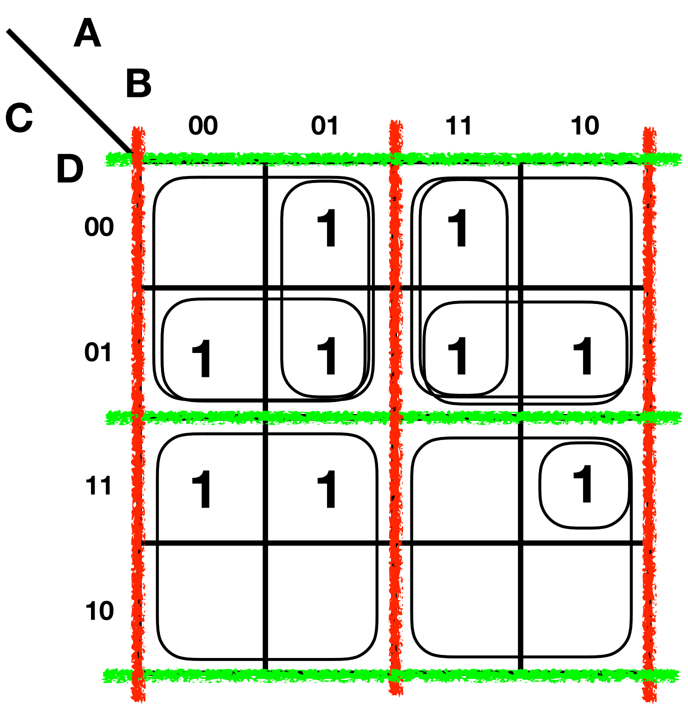


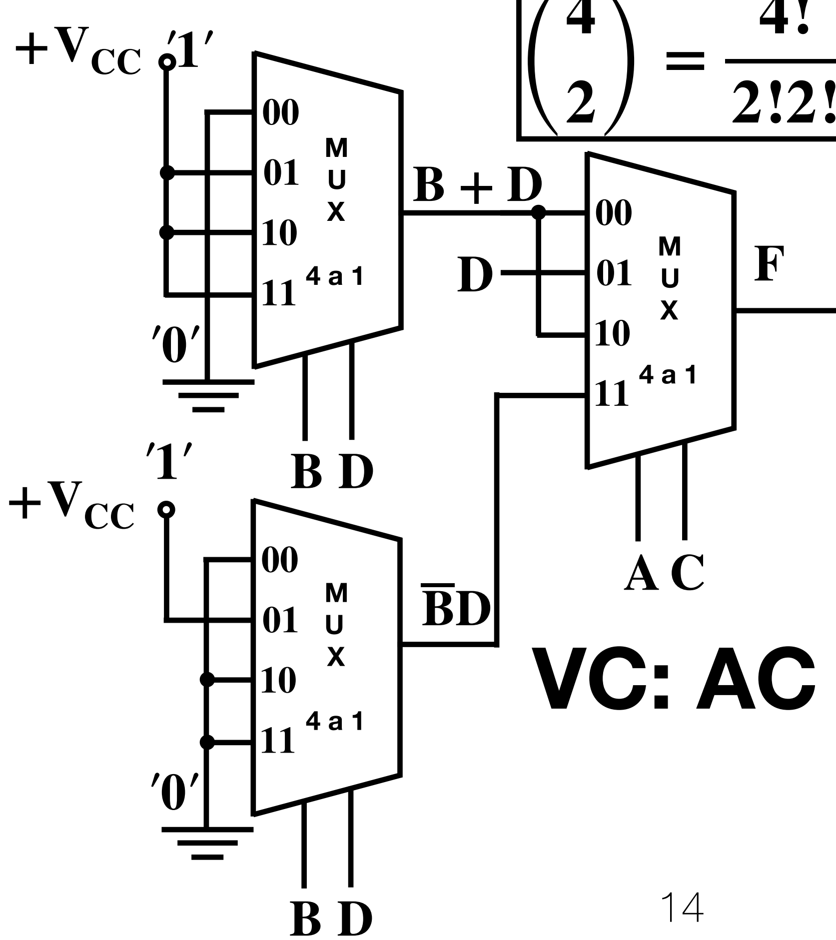
VC: AD



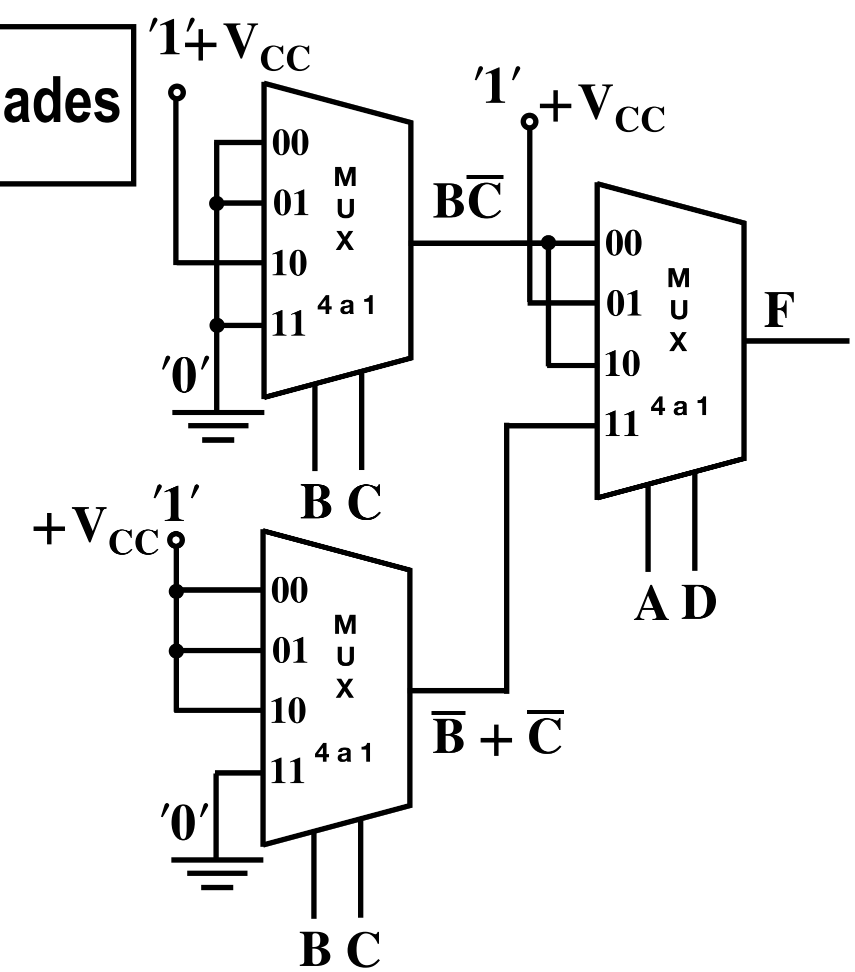
MUX de 2 VC: tendré en las entradas solamente 1's, 0's y las dos (n-2) variables independientes no tomadas negadas y sin negar y sumas y/o productos de las mismas que no puedo dejar expresados.

$$\binom{4}{2} = \frac{4!}{2!2!} = 6 \text{ posibilidades}$$





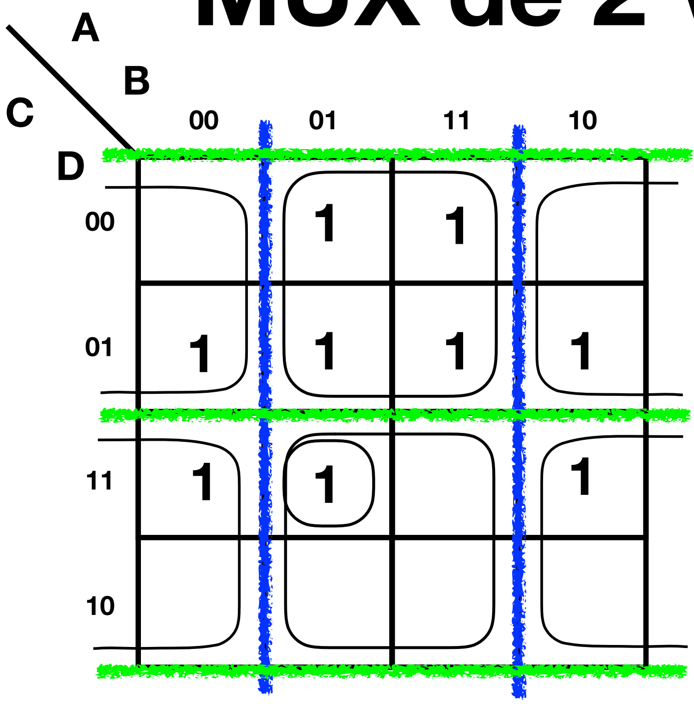
VC: AC



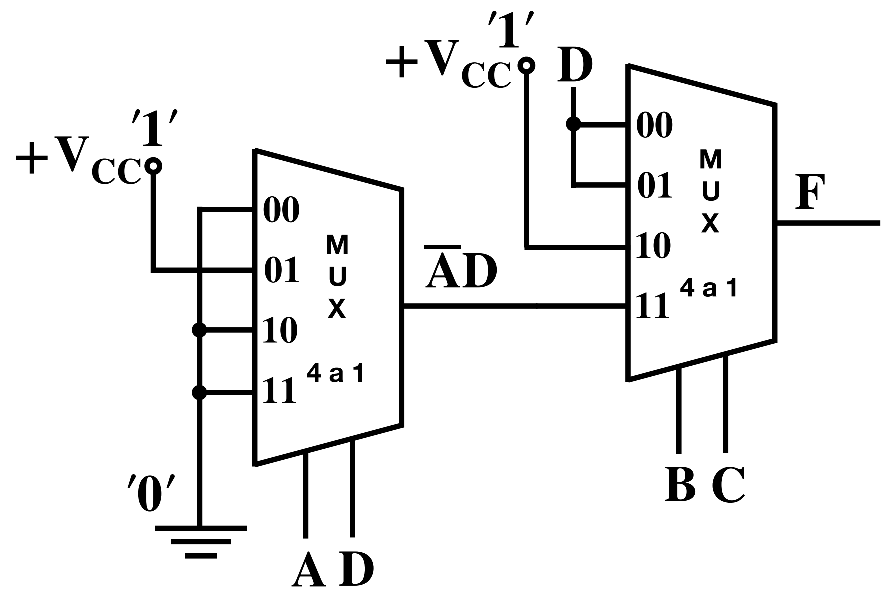
Ejercicio 15:

DECO Y MUX

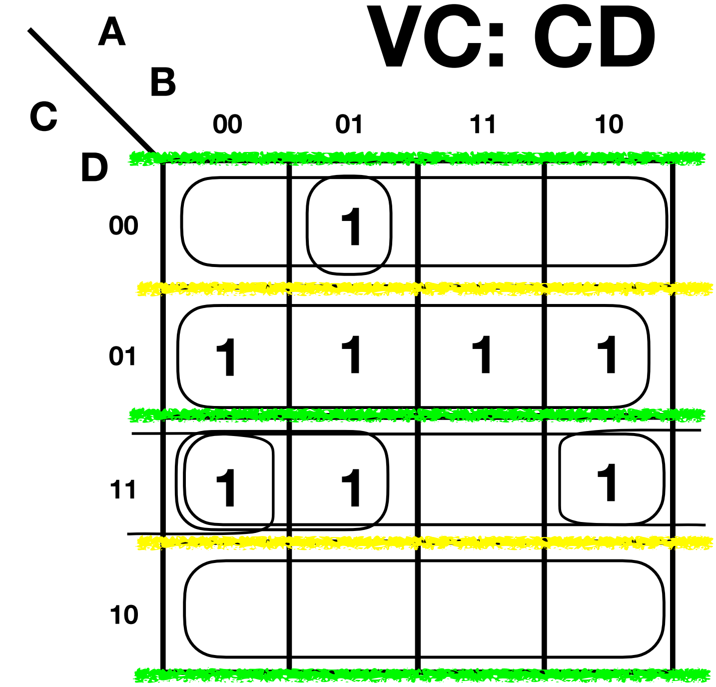
MUX de 2 VC



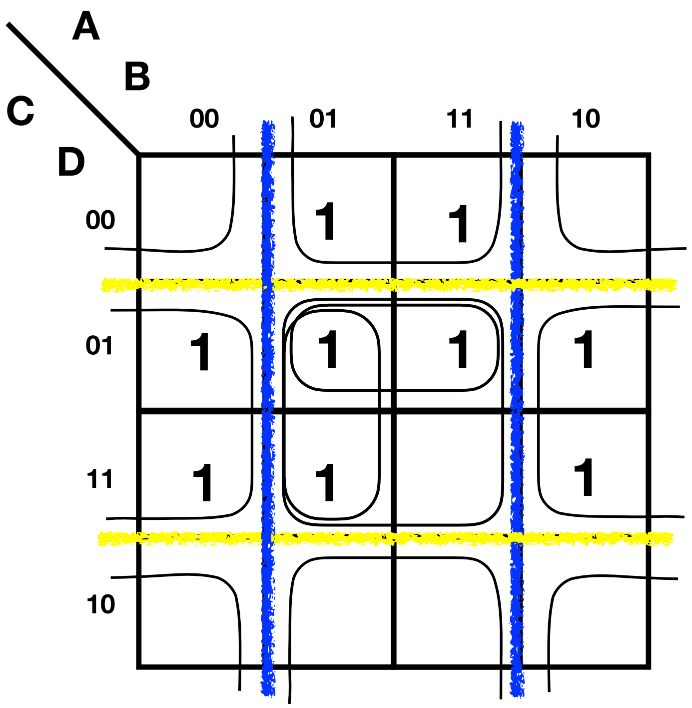
VC: BC

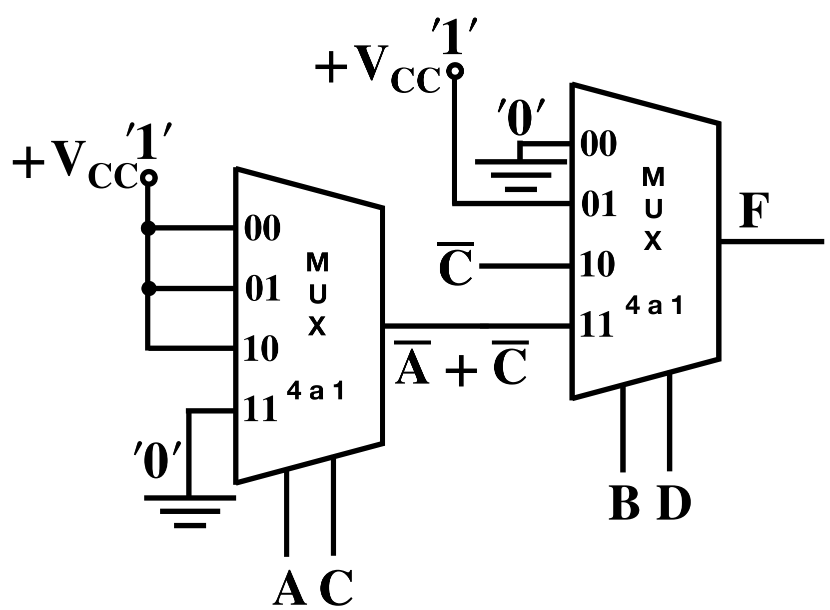


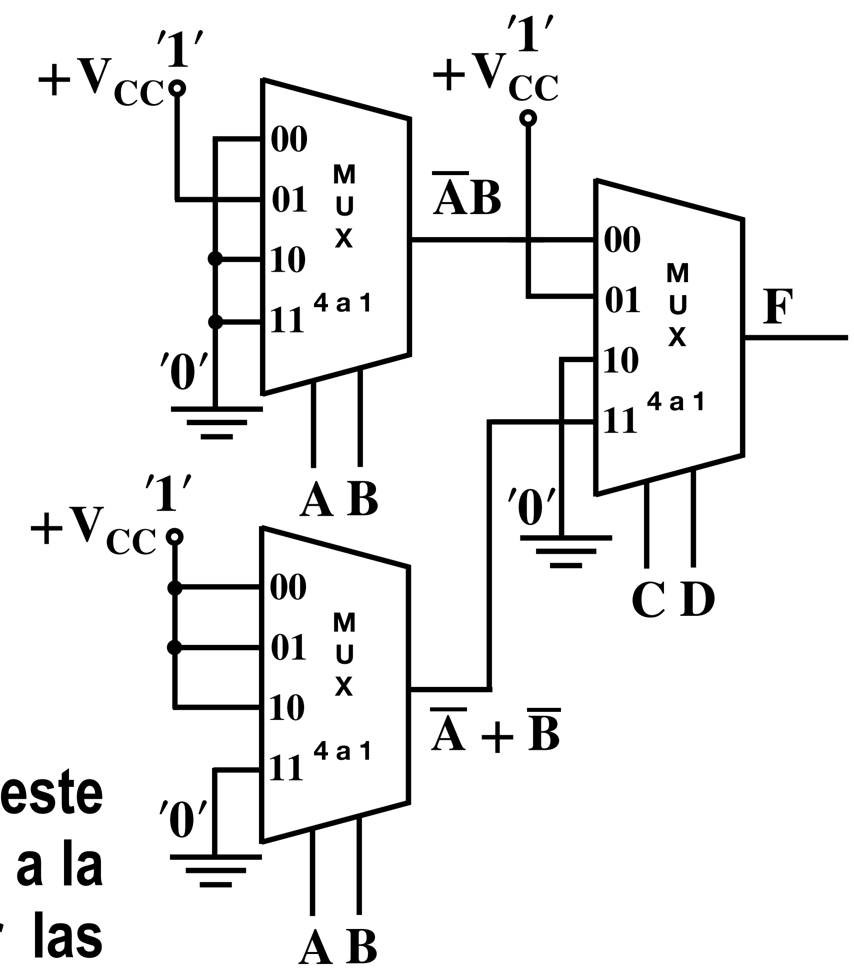
VC: CD



VC: BD







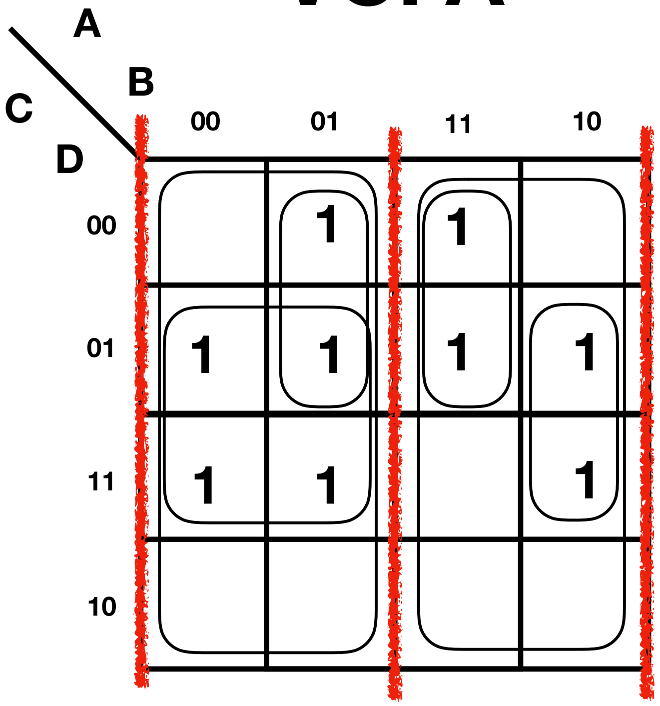
Con dos (n-2) VC, no todas las implementaciones son óptimas, en este caso hay solamente 3, AB, BC y BD que todas llevan dos MUX. Por eso a la hora de elegir las VC se debe observar la función mínima y tomar las variables que mas se repiten en todos los términos negadas y sin negar.

DECO Y MUX

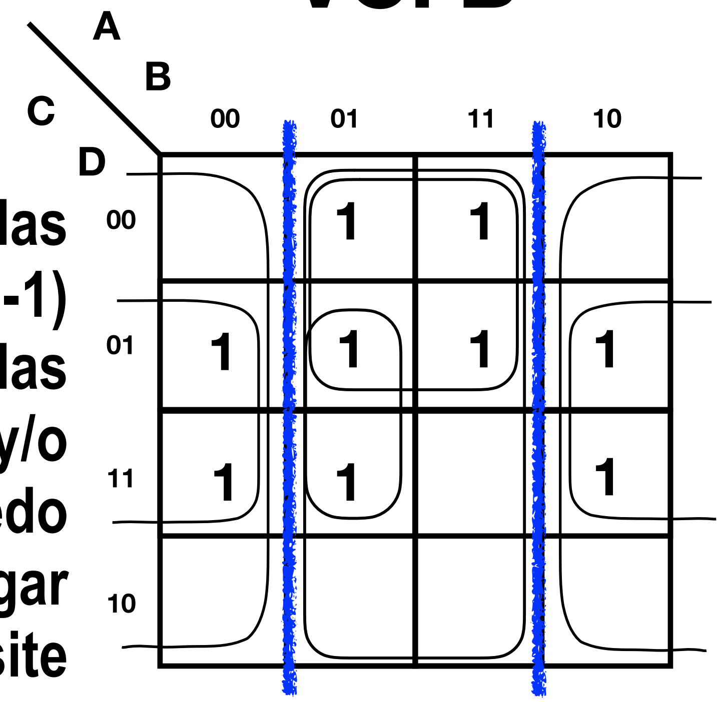
__Ejercicio 15:__ $F_{min} = \overline{A}D + B\overline{C} + \overline{B}D + \overline{C}D$

MUX de 1 VC

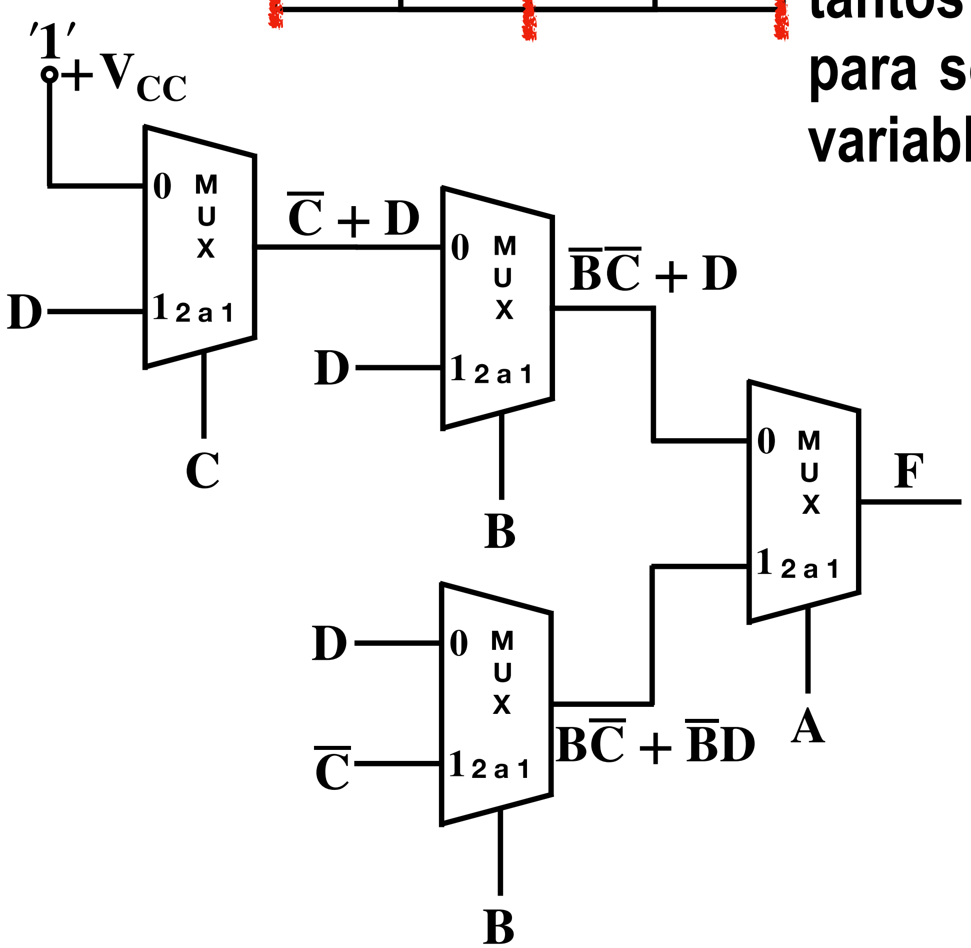
VC: A



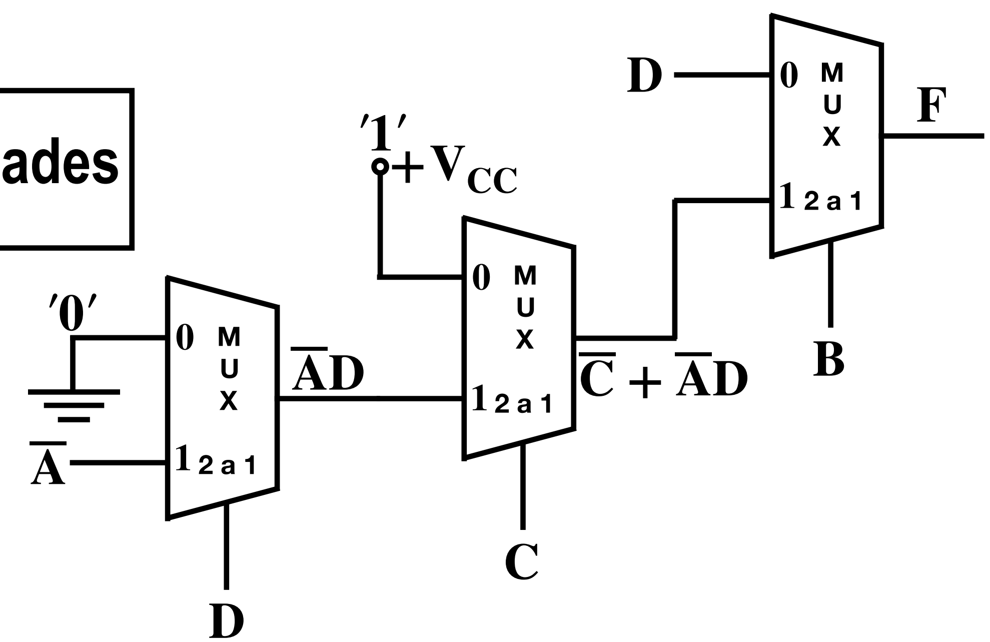
VC: B



__MUX de 1 VC__: tendré en las entradas solamente 1's, 0's y las tres (n-1) variables independientes no tomadas negadas y sin negar y sumas y/o productos de las mismas que no puedo dejar expresados. Por eso debo agregar tantos niveles de MUX como necesite para solamente tener 0's, 1's y una sola variable negada o no.



$$\binom{4}{1} = \frac{4!}{1!3!} = 4 \text{ posibilidades}$$



Ejercicio 15:

DECO Y MUX

MUX de 1 VC

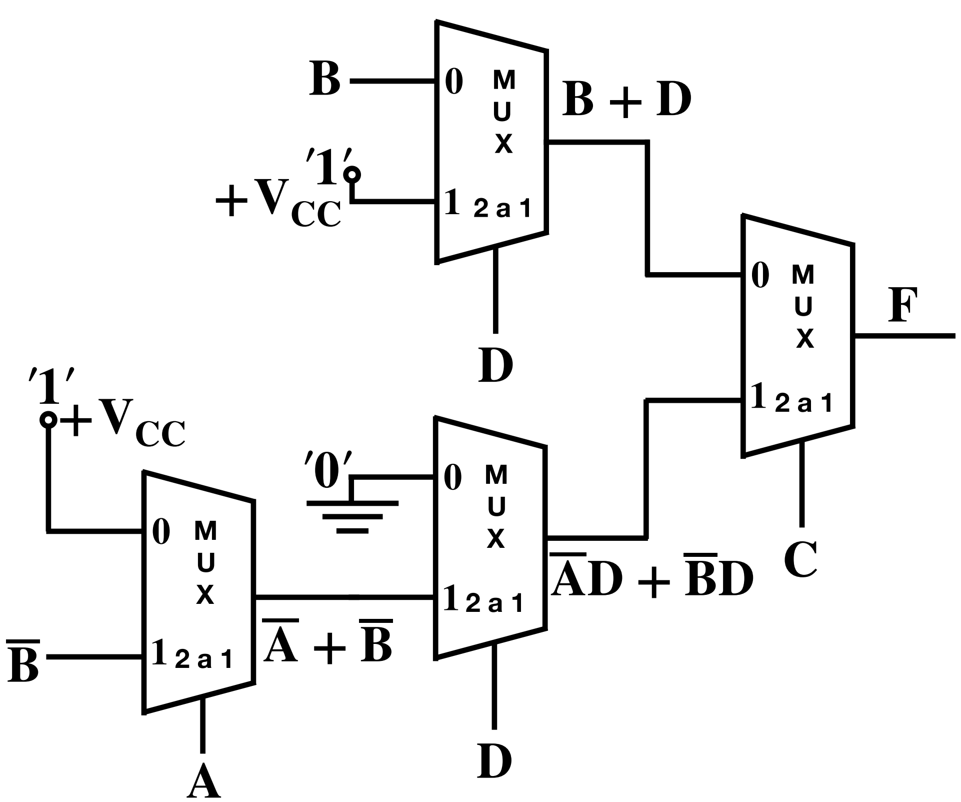
VC: C

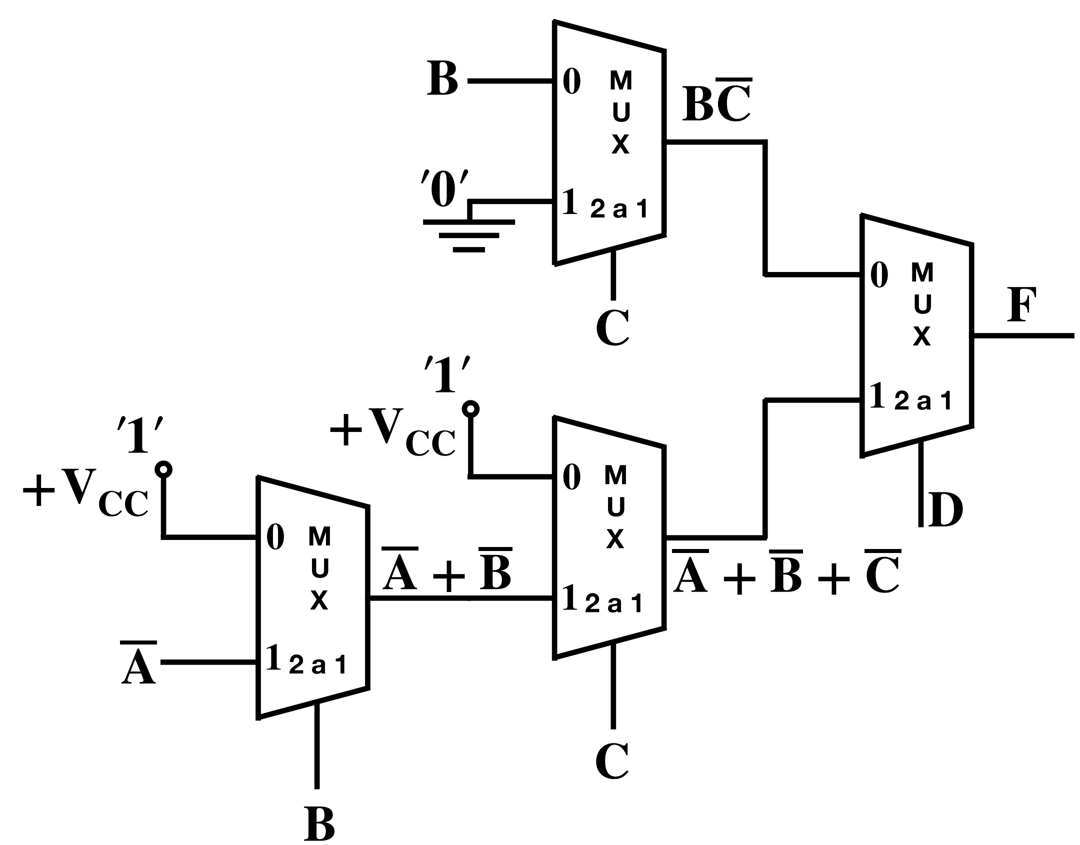
CD \ AB	00	01	11	10
00		1	1	
01	1	1	1	1
11	1	1		1
10				

VC: D

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

Con una (n-3) VC, no todas las implementaciones son óptimas, en este caso hay solamente 1, B, que lleva tres MUX. Por eso a la hora de elegir las VC se debe observar la función mínima y tomar las variables que mas se repiten en todos los términos negadas y sin negar.

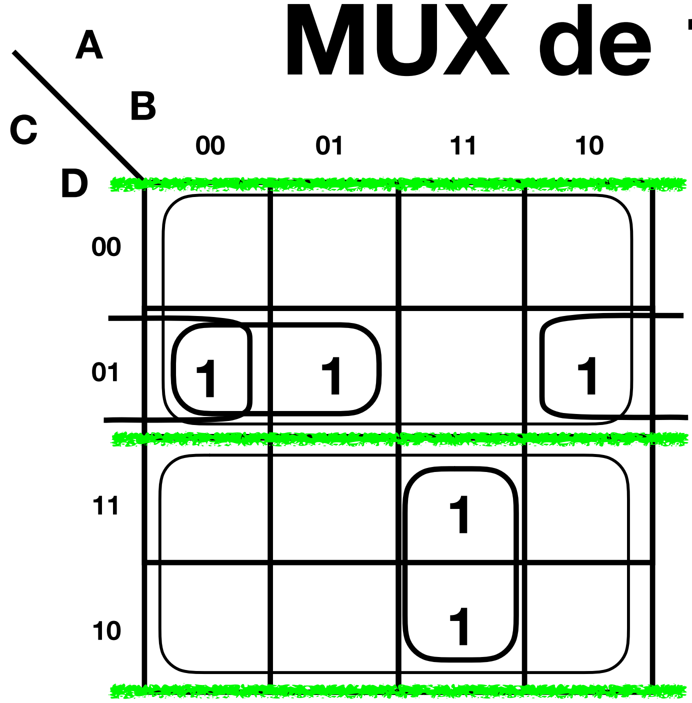




Modelo de Parcial Técnica Digital

<u>Ejercicio 5:</u> Implementar la función Z(A,B,C,D)= Σm (1,5,9,14,15) solo con varios MUX de 1 entrada de control. Utilizar A, B y C (con A=MSB) como variables de control. Simplificar.

MUX de 1 VC



Con una (n-3) VC, no todas las implementaciones son óptimas, en este caso hay solamente 1, C. Por eso a la hora de elegir las VC se debe observar la función mínima y tomar las variables que mas se repiten en todos los términos negadas y sin negar.

$$\boxed{F_{min} = \overline{A}\,\overline{C}D + \overline{B}\,\overline{C}D + ABC}$$

Conviene tomar a C como variable de control, ya que está negada y sin negar y por lo tanto controla mas a la función