



Problemas Hoja 4:

Módulos combinatoriales básicos

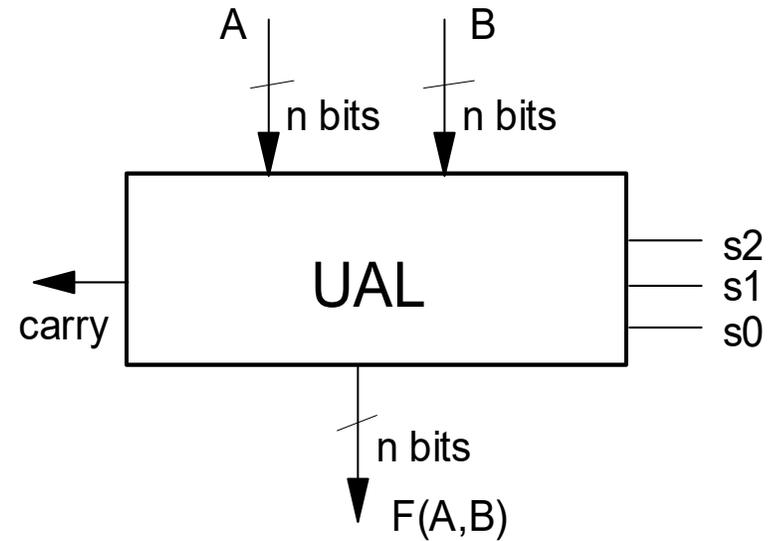
Fernando Castro Rodríguez

*Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid*





Ejercicio 1 Los computadores disponen de un circuito combinacional capaz de realizar las operaciones más elementales: AND, OR, Suma Aritmética, etc. sobre dos configuraciones binarias de n bits. Dicho circuito se denomina Unidad Aritmético Lógica (UAL). Diseñe una UAL como la mostrada en la figura, capaz de realizar las siguientes funciones: A AND B, A OR B, C1 B, A+B (suma aritmética), desplazamiento de B un bit a la derecha, desplazamiento de B un bit a la izquierda. El diseño se realizará para n=4 bits. La función que en cada caso realiza la UAL se selecciona mediante las entradas de control s2, s1, s0.



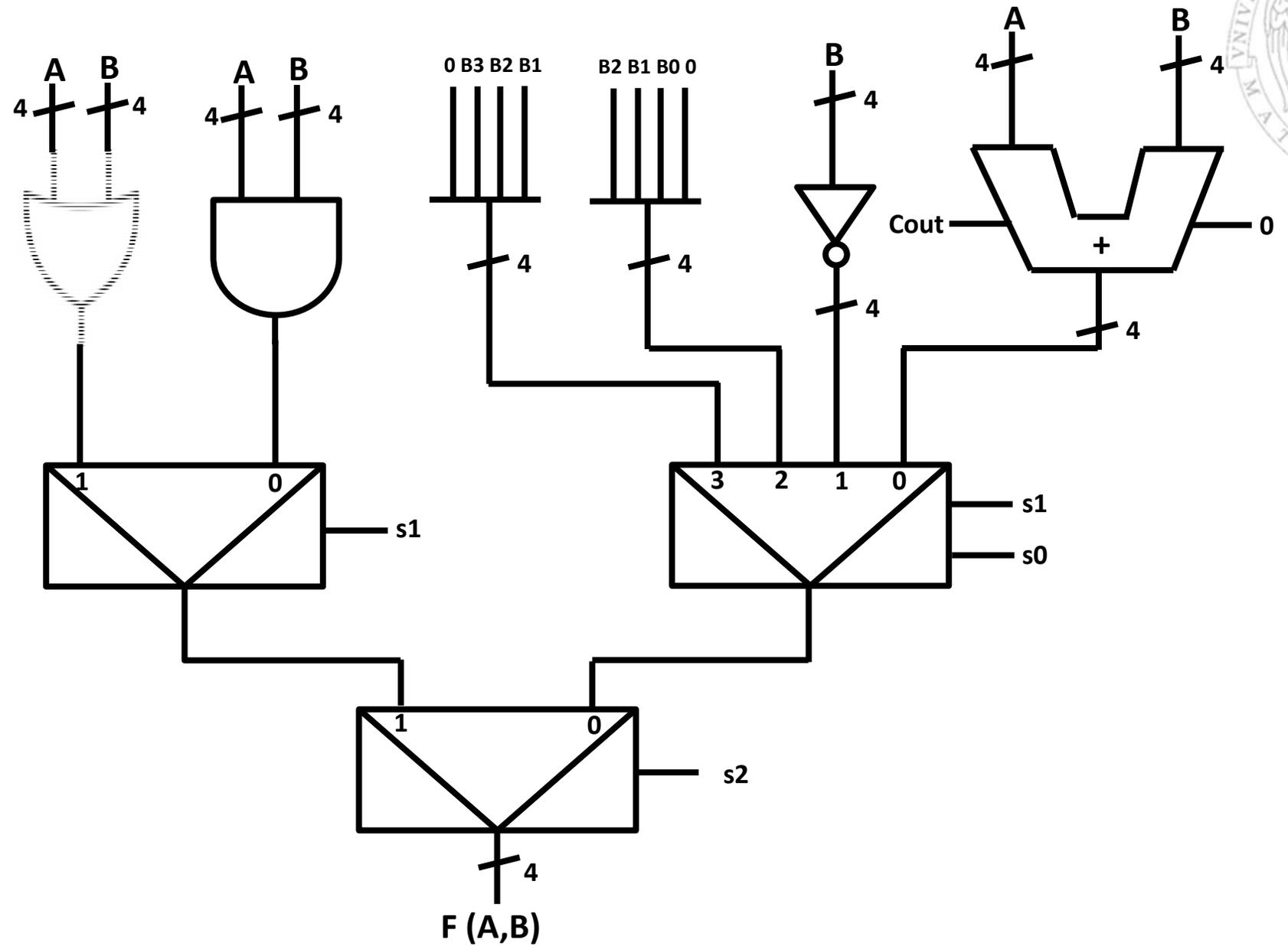
Tenemos que hacer un total de 6 operaciones distintas:

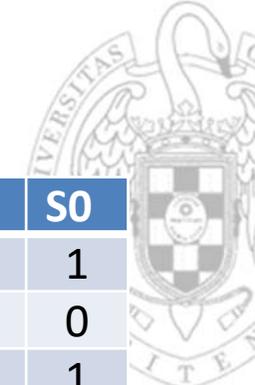
- 1) A and B
- 2) A or B
- 3) C1(B)
- 4) A+B
- 5) Desplaz a la dcha de B
- 6) Desplaz a la izquierda de B

Tenemos 3 señales de control (s2,s1,s0), con las que se podrían codificar hasta 8 operaciones diferentes. Elegimos la siguiente codificación para cada operación:

s2	s1	s0	Op
0	0	0	+
0	0	1	C1
0	1	0	Izq
0	1	1	Dcha
1	0	0	AND
1	0	1	AND
1	1	0	OR
1	1	1	OR

s2	s1	s0	Op
0	0	0	+
0	0	1	C1
0	1	0	lza
0	1	1	Dcha
1	0	0	AND
1	0	1	AND
1	1	0	OR
1	1	1	OR



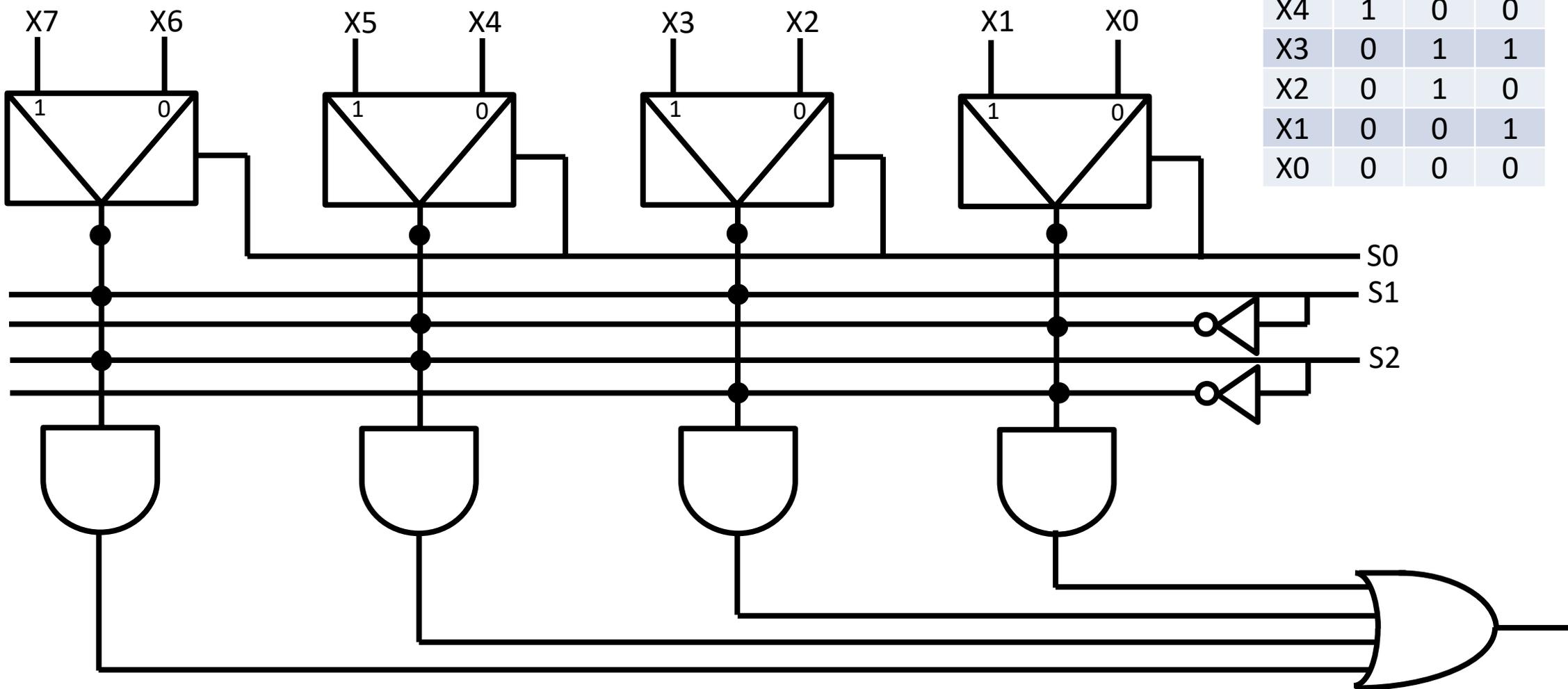


Ejercicio 2 Diseñe un multiplexor de 8 a 1 usando:

- a) 4 multiplexores de 2 a 1 y el mínimo número de puertas.
- b) 1 decodificador 3 a 8 y el menor número de puertas.

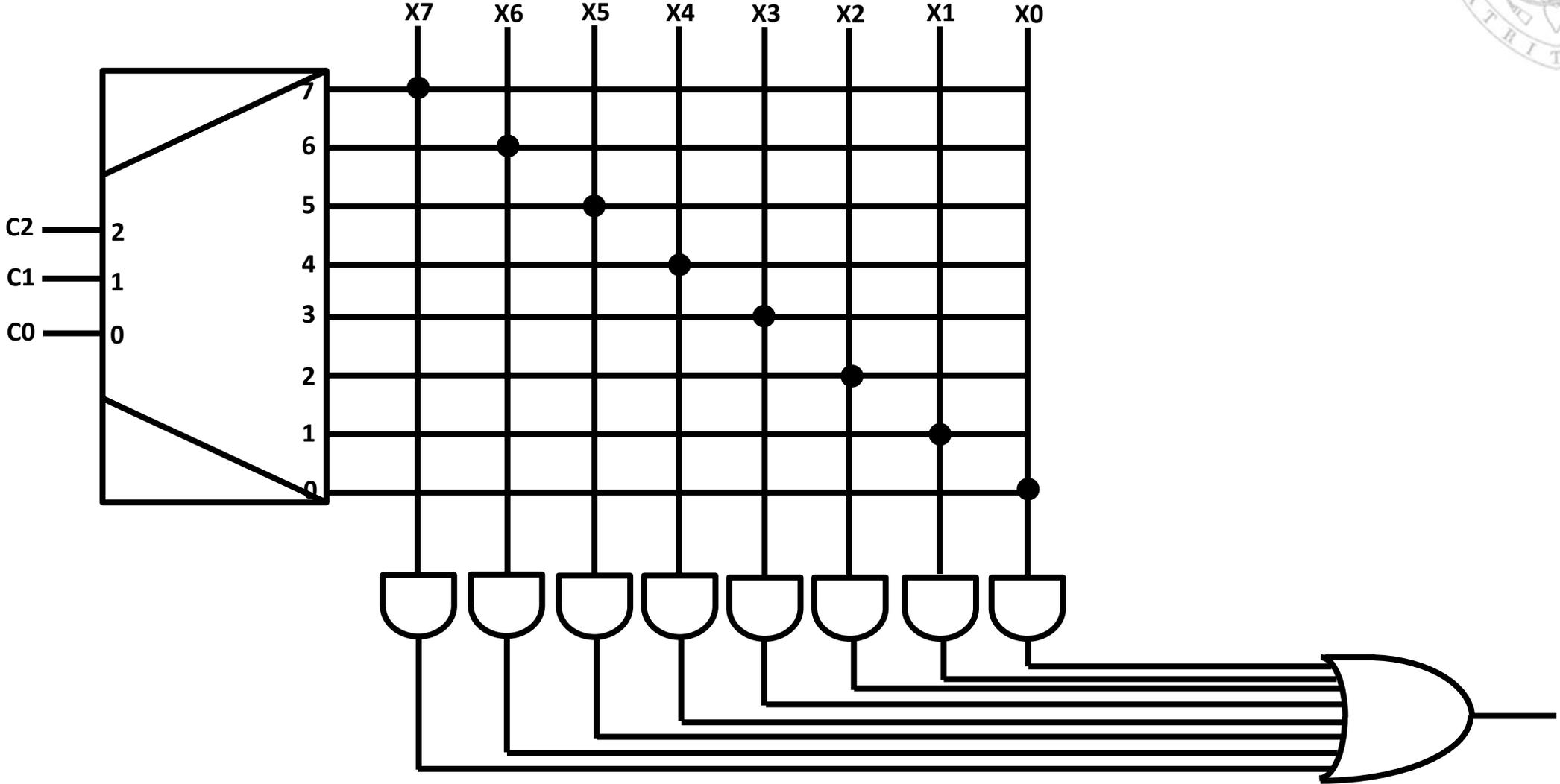
	S2	S1	S0
X7	1	1	1
X6	1	1	0
X5	1	0	1
X4	1	0	0
X3	0	1	1
X2	0	1	0
X1	0	0	1
X0	0	0	0

a)





b)

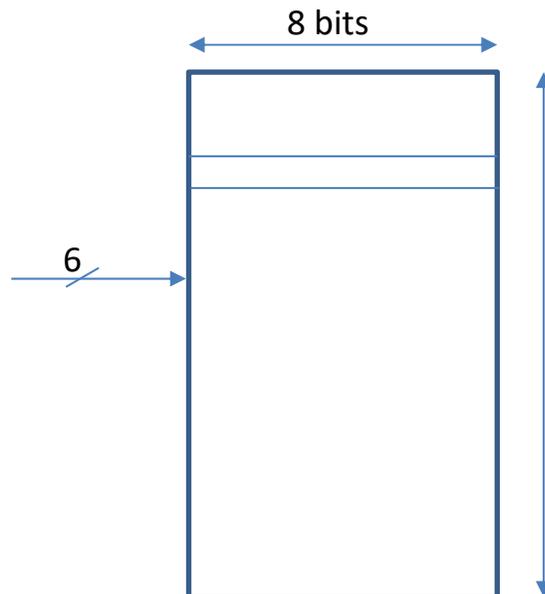




Ejercicio 3 Considere la implementación en una ROM de un sumador de números naturales codificados en binario puro con 3 bits que codifica el resultado en BCD. Discuta razonadamente el tamaño mínimo de la ROM e indique el contenido de la ROM en las direcciones $(3E)_{16}$ y $(1D)_{16}$.

Con 3 bits, y usando la como codificación binario puro, se pueden representar los números del 0 al 7, luego la suma tiene que producir como resultado algún número en el rango 0-14 (es decir, un máximo de 2 dígitos decimales). Sabiendo que cada dígito decimal se representa en BCD con 4 bits, el tamaño de cada fila en bits (palabra, o lo que es lo mismo número de columnas) que almacenará la memoria ROM en donde se guardan los resultados de las sumas será de 8.

Por otro lado, las posibles combinaciones de los dos números a sumar (cada uno de 3 bits), serán $2^6 = 64$, luego necesito 6 bits para indicar la dirección (fila de la memoria ROM) a la que quiero acceder.



Dirección $(3E)_{16}$: 0011 1110 \rightarrow suma de $(7)_{10} + (6)_{10} = (13)_{10} = (0001\ 0011)_{BCD}$

Dirección $(1D)_{16}$: 0001 1101 \rightarrow suma de $(3)_{10} + (5)_{10} = (8)_{10} = (0000\ 1000)_{BCD}$

2^6 palabras

Tamaño mínimo de la ROM:
 2^6 palabras x 8 bits = 512 bits



Ejercicio 4 Usando un sumador y puertas lógicas diseñe un conversor de MyS a C2 de 8 bits. Ídem para un conversor de C2 a MyS que tenga una salida adicional de overflow. ~~Generalice el diseño para que el tipo de conversión sea seleccionable por una entrada de control.~~

MyS \rightarrow C2:

- a) Los números positivos se expresan de la misma forma en ambas codificaciones
- b) Transformación de los números negativos:

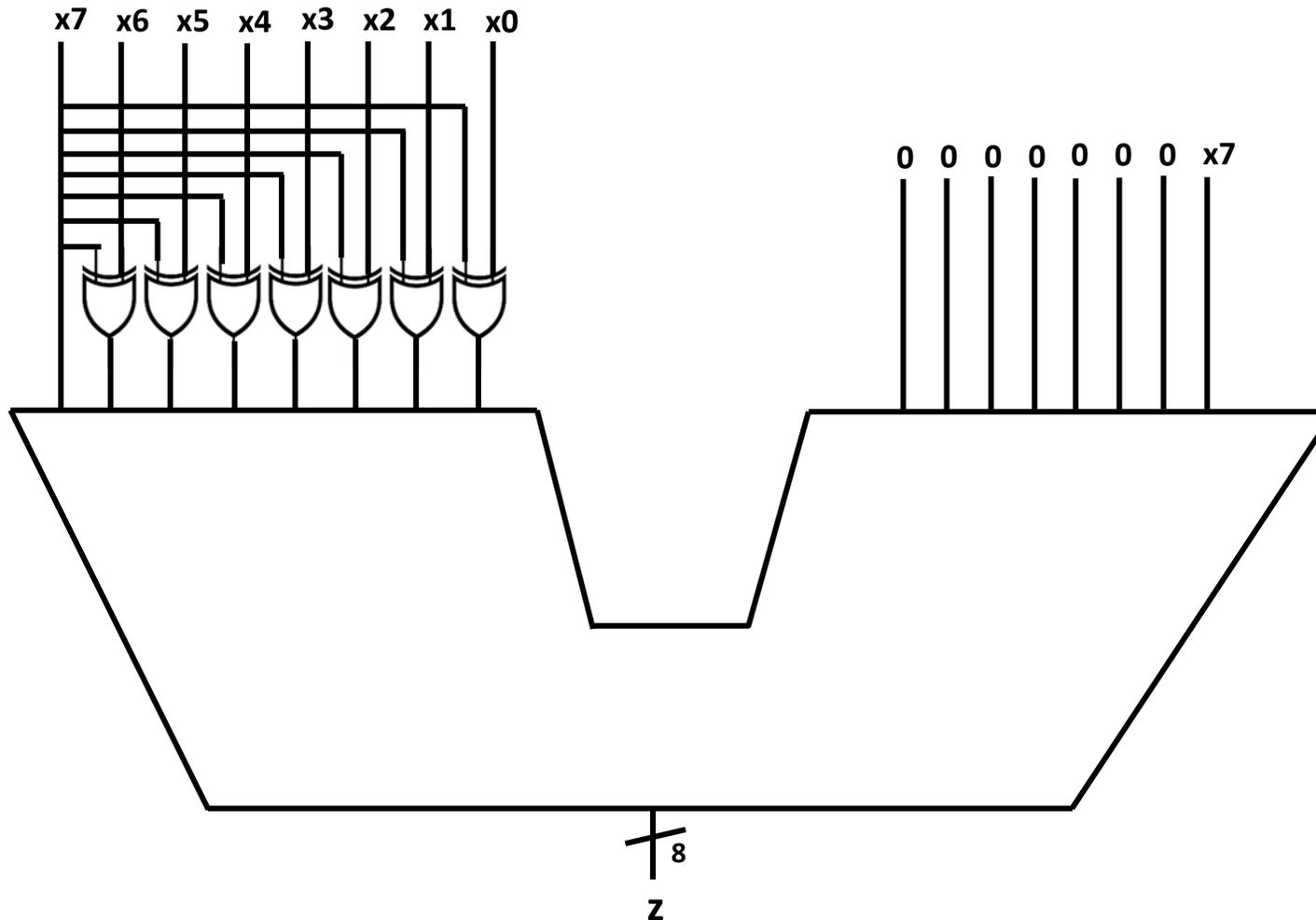
Ejemplo: $(-5)_{10} = (1101)_{\text{MyS-4bits}}$; $(5)_{10} = (0101)_{\text{MyS-4bits}} = (0101)_{\text{C2-4bits}} \rightarrow (-5)_{10} = \text{C2}(0101) = (1011)_{\text{C2-4bits}}$

\rightarrow Se mantiene el bit de signo, se invierten el resto de bits, y se suma 1



MyS \rightarrow C2 Sea $x = (x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ el n $^\circ$ expresado en MyS y $z = (z_7, z_6, z_5, z_4, z_3, z_2, z_1, z_0)$ el n $^\circ$ expresado en C2

- \rightarrow Los números positivos se expresan igual en ambas codificaciones
- \rightarrow Para los números negativos se mantiene el bit de signo, se invierten el resto de bits, y se suma 1



Si $x_7 = 0$, x es un n $^\circ$ positivo, y $(x_7 \oplus x_i) = (0 \oplus x_i) = x_i$, por lo que el primer sumando es x y el segundo sumando es 0 $\rightarrow z = x + 0 = x$

Si $x_7 = 1$, x es un n $^\circ$ negativo, y $(x_7 \oplus x_i) = (1 \oplus x_i) = \overline{x_i}$, por lo que el primer sumando mantiene el signo de x e invierte el resto de bits y el segundo sumando es 1 $\rightarrow z = x$ expresado en C2



C2 → MyS

a) Los números positivos se expresan de la misma forma en ambas codificaciones

b) Transformación de los números negativos:

Ejemplo: $(-5)_{10} = (1011)_{C2-4bits}$; $(5)_{10} = C2(1011) = (0101)_{C2-4bits} = (0101)_{MyS-4bits} \rightarrow (-5)_{10} = (1101)_{MyS-4bits}$

→ Se mantiene el bit de signo, se invierten el resto de bits, y se suma 1

C2 → MyS Sea $x = (x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ el nº expresado en MyS y

$z = (z_7, z_6, z_5, z_4, z_3, z_2, z_1, z_0)$ el nº expresado en C2

→ **Los números positivos se expresan igual en ambas codificaciones**

→ Para los números negativos se mantiene el bit de signo, se invierten el resto de bits, y se suma 1

Por tanto, ¡el diseño para el conversor de MyS a C2 es exactamente el mismo que para el conversor de C2 a MyS!



Acerca de *Creative Commons*

- Licencia CC ([Creative Commons](#))



- Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:



- Reconocimiento** (*Attribution*):

En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.



- No comercial** (*Non commercial*):

La explotación de la obra queda limitada a usos no comerciales.



- Compartir igual** (*Share alike*):

La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: <https://creativecommons.org/licenses/by-nc-sa/4.0/>