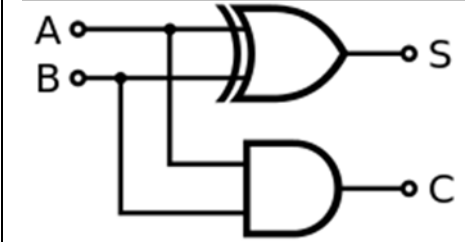


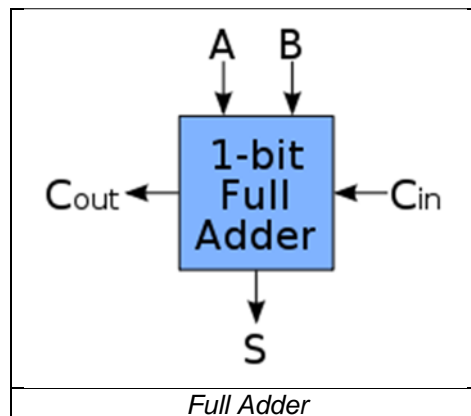
27. Diseñe un *Full Adder* con una ROM NOR. Dibuje la estructura interna e indique el tamaño de la memoria.

Los sumadores son circuitos habituales en tecnología de computadores y, como su nombre indica, permiten sumar números. Aunque pueden construirse para distintas representaciones, los más comunes son aquellos que operan con números binarios.

Para entender cómo funciona un *full adder* (sumador completo) se parte del diseño y tabla de verdad del circuito *half-adder* (o semisumador) que suma dos dígitos binarios A y B, proporcionando dos salidas, la suma (S) y el acarreo (C). La señal de acarreo representa un desbordamiento en el siguiente dígito dada una suma de varios dígitos.

	Entradas		Salidas	
	A	B	C	S
	0	0	0	0
	0	1	0	1
	1	0	0	1
	1	1	1	0
Semisumador		Tabla de Verdad		

A partir de dos semisumadores se puede construir un sumador completo de un bit, que es un circuito que suma números binarios junto con el acarreo de la etapa anterior (menos significativa). Por tanto, tiene como entradas tres números de un bit, A, B y Cin (acarreo de entrada) y como variables de salida, S (suma) y Cout (acarreo de salida). En general, como se verá al final, este *full-adder* es normalmente un componente de una cascada de sumadores de un bit que suma números de 8, 16, 32 bits...



En este circuito se he de cumplir:

- S es 1 si un número impar de las entradas es 1.
- Cout es 1 si dos o más de las entradas son 1.

Por tanto, la tabla de verdad del sumador completo queda como sigue:

Entradas (Variables)			Salidas (Funciones)			
C_{in} (Carry in)	B (Operando)	A (Operando)	C_{out} (Carry out)	S (Suma)		
0	0	0	0		0	
0	0	1	0		1	1
0	1	0	0		1	2
0	1	1	1	3	0	
1	0	0	0		1	4
1	0	1	1	5	0	
1	1	0	1	6	0	
1	1	1	1	7	1	7

Ponemos las 2 funciones del *Full Adder* como **suma de minitérminos**:

S (función suma) = $\sum (1, 2, 4, 7)$ (según tabla)

$$S = \sum (\overline{C_{in}} \overline{B} A, \overline{C_{in}} B \overline{A}, C_{in} \overline{B} \overline{A}, C_{in} B A)$$

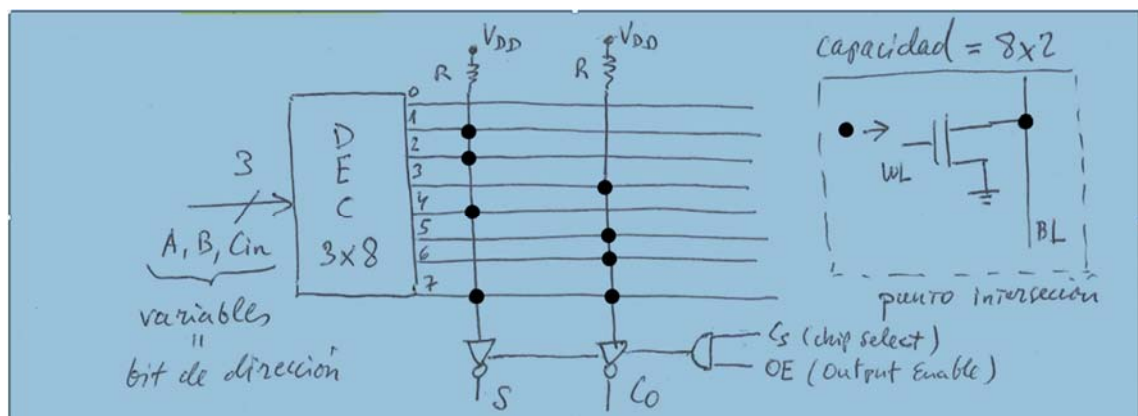
C_{out} (función acarreo de salida) = $\sum (3, 5, 6, 7)$ (según tabla)

$$C_{out} = \sum (\overline{C_{in}} B A, C_{in} \overline{B} A, C_{in} B \overline{A}, C_{in} B A)$$

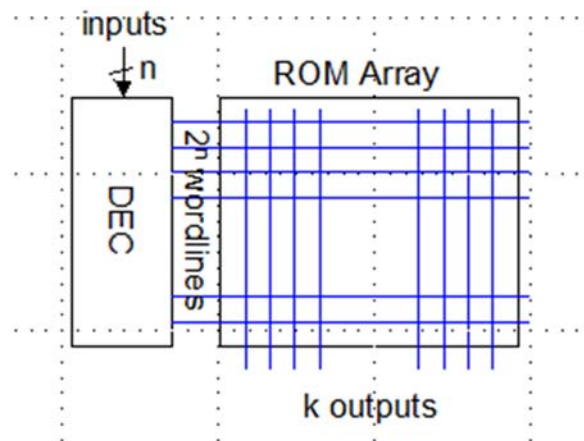
La **ROM** incluirá un **decodificador 3 a 8** para generar 8 líneas de palabra, que hacen el papel de minitérminos. Las entradas del *decoder* son las 3 variables: A y B (sumandos) y C_{in} (acarreo de entrada).

La **ROM tendrá 8 posiciones de memoria con 2 líneas de bit**, una para cada función, S y C_{out} . Se establecerán transistores de contacto en la intersección de la línea de bit con la línea de palabra correspondiente a los minitérminos de la función. Como los transistores NMOS colocan un cero en la celda, si se quiere que las funciones aparezcan sin negar, se ponen inversores a la salida de cada una de las líneas de bit.

El esquema particular del *full-adder* en estructura ROM NOR quedará así:



En general, dadas “n” entradas y “K” salidas, se tiene el siguiente esquema:



Este circuito *full-adder* de 1 bit es la base para construir un sumador de n bits. Cada sumador completo aporta una entrada C_{in} , que es el Cout del sumador anterior. Este tipo de sumador se denomina *ripple-carry adder* (RCA). Note que únicamente el primer sumador completo podría ser reemplazado por un semisumador (suponiendo que su acarreo de entrada $C_0 = 0$).

