

Circuitos combinacionales

Soluciones

1. Dar la tabla de verdad para la compuerta AND de 3 entradas.

A	B	C	ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

2. Dar la tabla de verdad para el circuito que compara dos números de dos bits. Las dos compuertas que aparecen ahí son XNOR, es decir la negación de XOR.

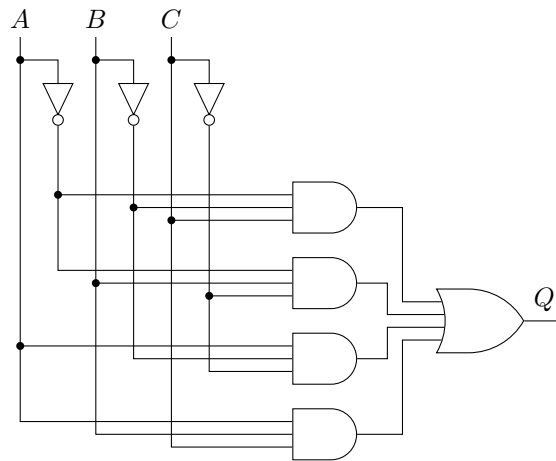
A_1	A_0	B_1	B_0	Q
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

3. Expresar la salida como Q del circuito comparador de forma algebraica (como una ecuación).

$$Q = \overline{(A_0 \oplus B_0)} \cdot \overline{(A_1 \oplus B_1)}$$

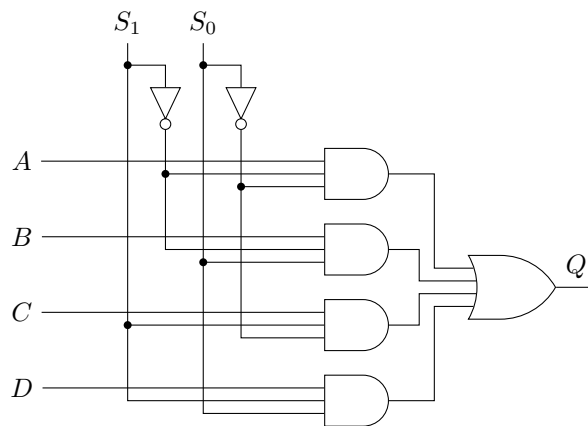
4. Dar un circuito de tres entradas y una salida que produzca un uno cuando haya un número impar de unos en sus entradas.

A	B	C	$Q = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



5. Dar el circuito y la tabla de verdad resumida para un multiplexor 4:1.

S_1	S_0	Q
0	0	A
0	1	B
1	0	C
1	1	D



6. Dar el circuito del decodificador 2:4. Dar circuito y tabla para el decodificador 3:8.

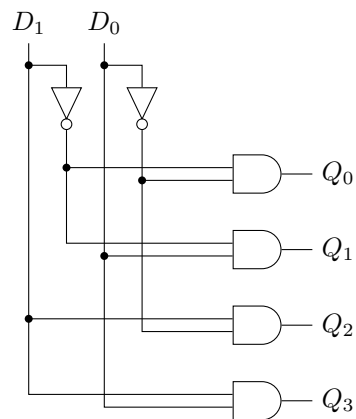
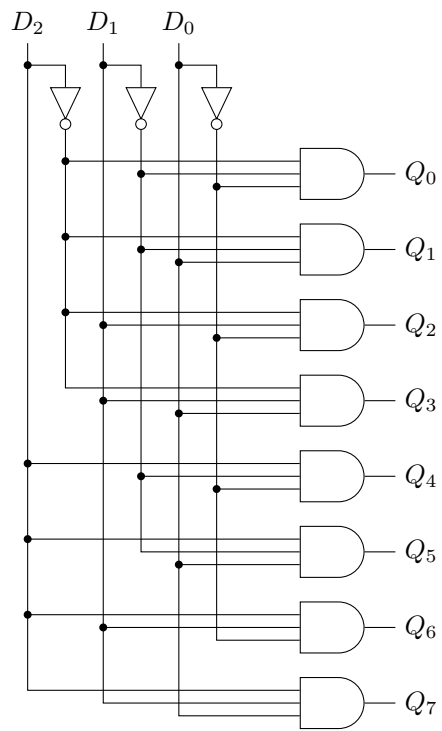
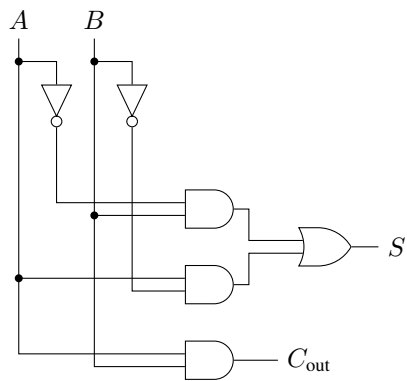


Tabla y circuito del decodificador 3:8

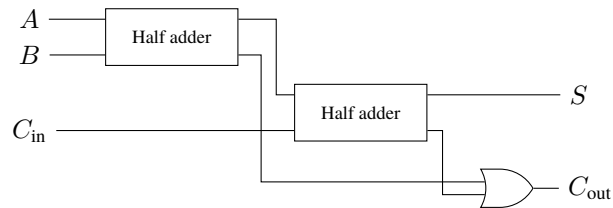
D_2	D_1	D_0	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



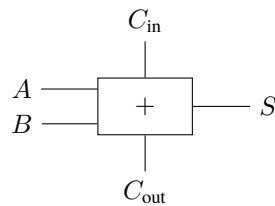
7. Dar el circuito de un *half adder* sin usar una compuerta XOR.



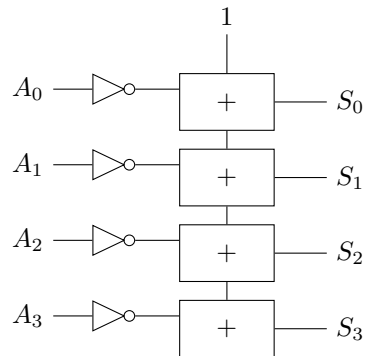
8. Dar el circuito de un *full adder*, teniendo en cuenta la tabla de verdad y que como su nombre lo indica hace uso de dos *half adder*.



En la próxima pregunta uso el siguiente símbolo con las entradas y salidas como se muestra para representar el *full adder*. Voy a usar la misma posición para las entradas y salidas, es decir el *carry in* arriba y el *carry out* abajo y todo tal como aparece en la figura.

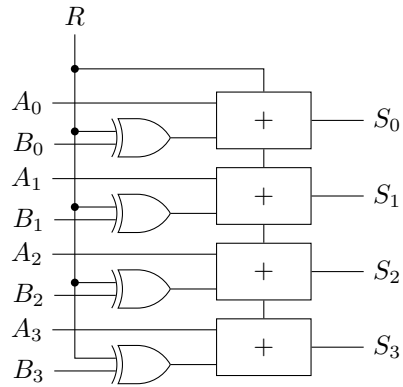


9. Implementar un circuito que me dé el complemento a dos de un número de 3 bits.



Este circuito solo funciona con números de 3 bits, el cuarto bit es necesario pero no puede dar el complemento a dos de números de 4 bits como el $10_{10} = 1010_2$. Para eso se necesitan 5 bits. El *carry in* del primer *full adder* siempre es 1. El complemento a dos del número A es el resultado de la suma S .

10. Implementar un circuito que sume o reste según se elija dos números de 3 bits.



Las compuertas XOR que están conectadas a la entrada de control R funcionan como si fueran multiplexores eligiendo entre B y \overline{B} . Para restar hay que poner la señal R en 1 y con eso obtenemos la suma de A y el complemento a dos de B . Por la misma razón del circuito de la pregunta anterior para trabajar con números negativos de 3 bits en realidad necesitamos 4 bits.