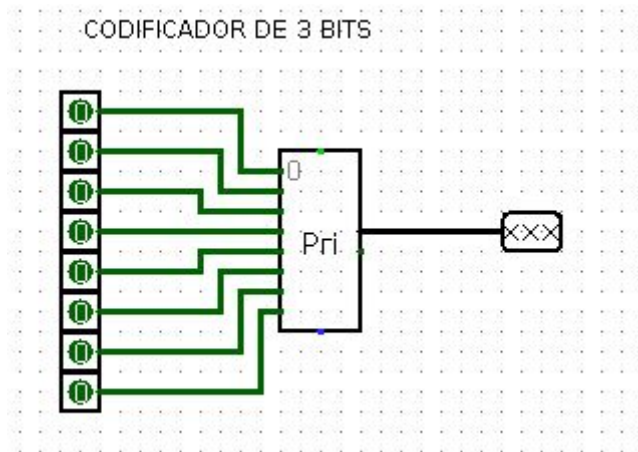


1-Crea la tabla de verdad de un codificador de 8 entradas. Prueba con el simulador logisim que la tabla de verdad es correcta. ¿Qué significa que el codificador de logisim tiene prioridad?

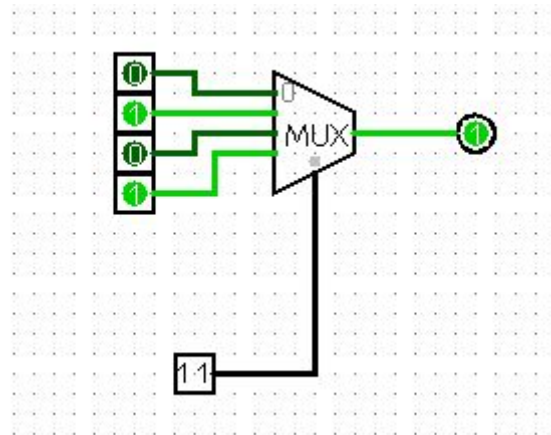


E7	E6	E5	E4	E3	E2	E1	E0	S2	S1	S0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

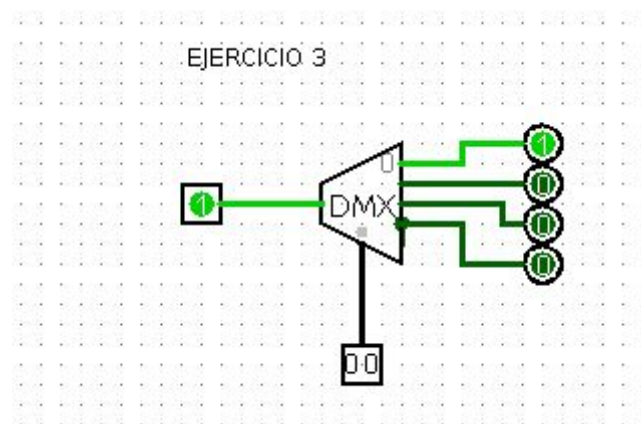
Cuando más de una entrada está activa la salida es indeterminada la codificación tiene prioridad y estará activa la de mayor de valor.

2-¿Qué salida presentará el siguiente multiplexor si A=0, B=1, C= 0, D=1, S 1 = 1 y S 2 =1?. Justifica tu respuesta y compruebala con el simulador logisim.

Al estar S1 =1 y S0=1 indica que la salida mostrará lo que valga el valor D ya que 11 en binario es el valor 3. En este caso D=1 por lo que en la salida se muestra un 1.



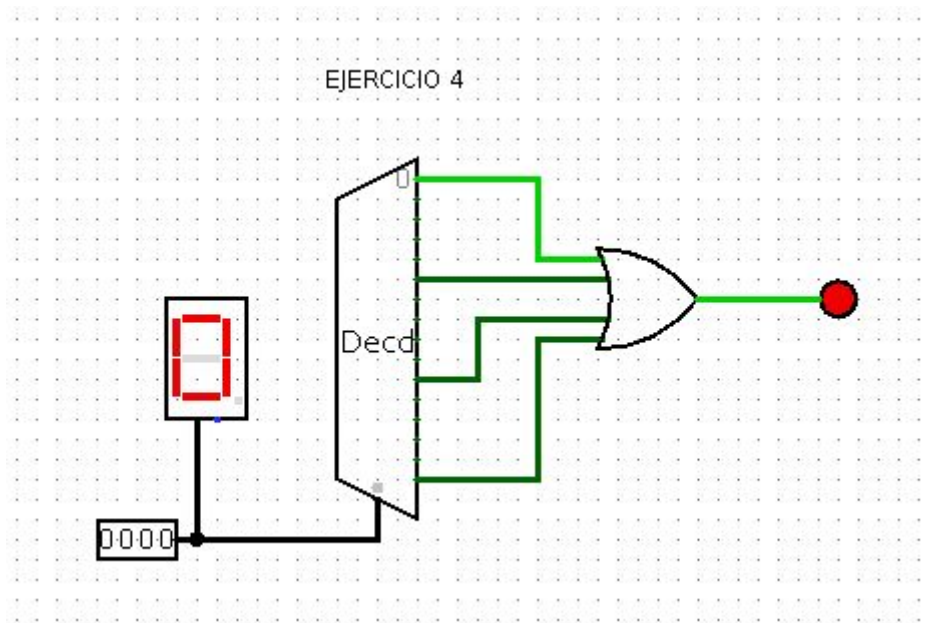
3-Prueba en el simulador logisim un demultiplexor de 4 salidas. Crea la tabla de verdad con los resultados obtenidos.



F0	F1	S3	S2	S1	S0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

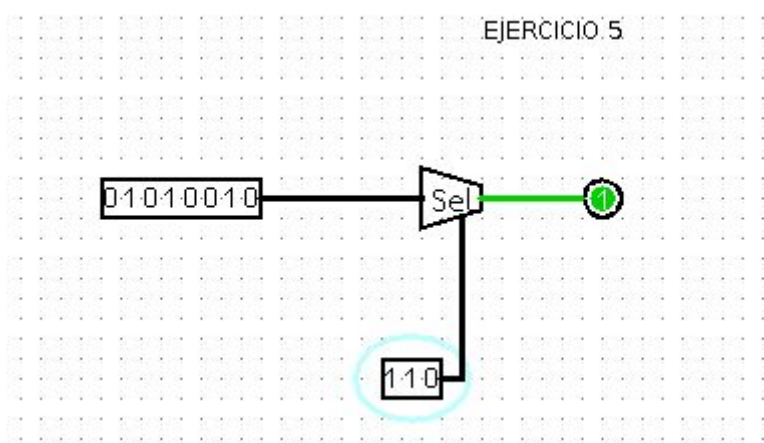
Si la entrada del multiplexor es 0, da igual el valor de F0 y F1, siempre será 0 en las salidas. Sin embargo, si en la entrada es 1, dependerá de los valores de F0 y F1 para saber en qué salida su valor será 1 tal y como se indica con la tabla de verdad.

4-Diseña un circuito que tenga una entrada de 4 bits, muestre el dígito hexadecimal correspondiente y encienda un led cuando el dígito sea 0, 5, A o F.

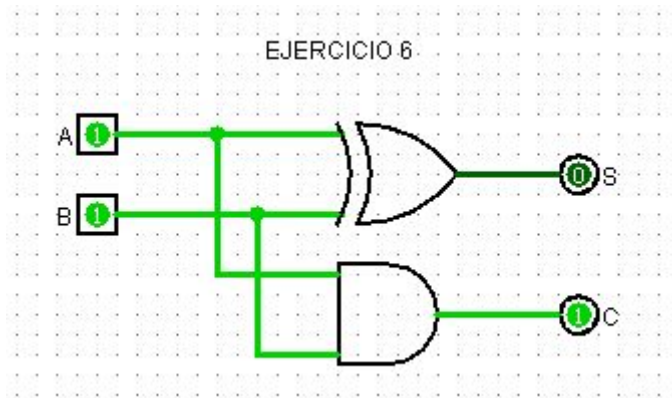


5-Prueba con el logisim un “selector de bit” y explica cuál es su utilidad. ¿Es un circuito combinacional o secuencial?

El selector de bit nos permite seleccionar que muestre por la salida el valor de un bit en concreto. Por ejemplo en la imagen vemos en la entrada 01010010 y estamos indicando que nos muestre el valor del bit 110 (el Bit 6), por eso la salida es 1 y por lo tanto se puede decir que es combinacional.



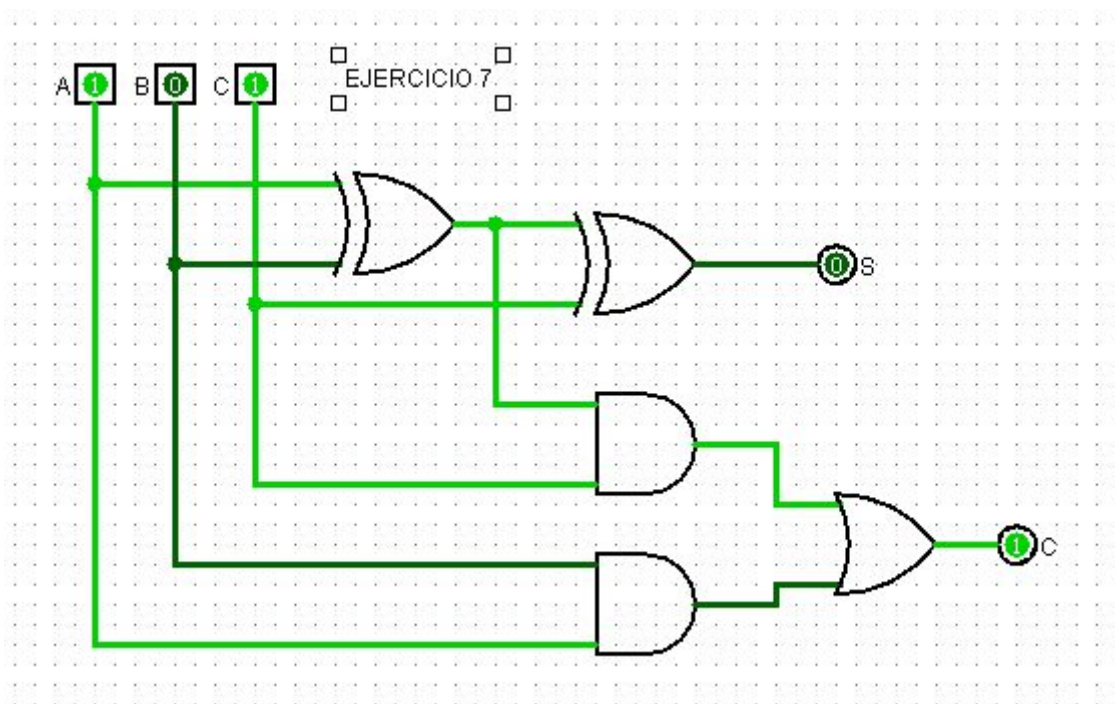
6-Diseña con puertas lógicas un sumador (HA) de 2 bits teniendo en cuenta la salida y el acarreo, no tengas en cuenta el acarreo de entrada. Compruébalo con el simulador.



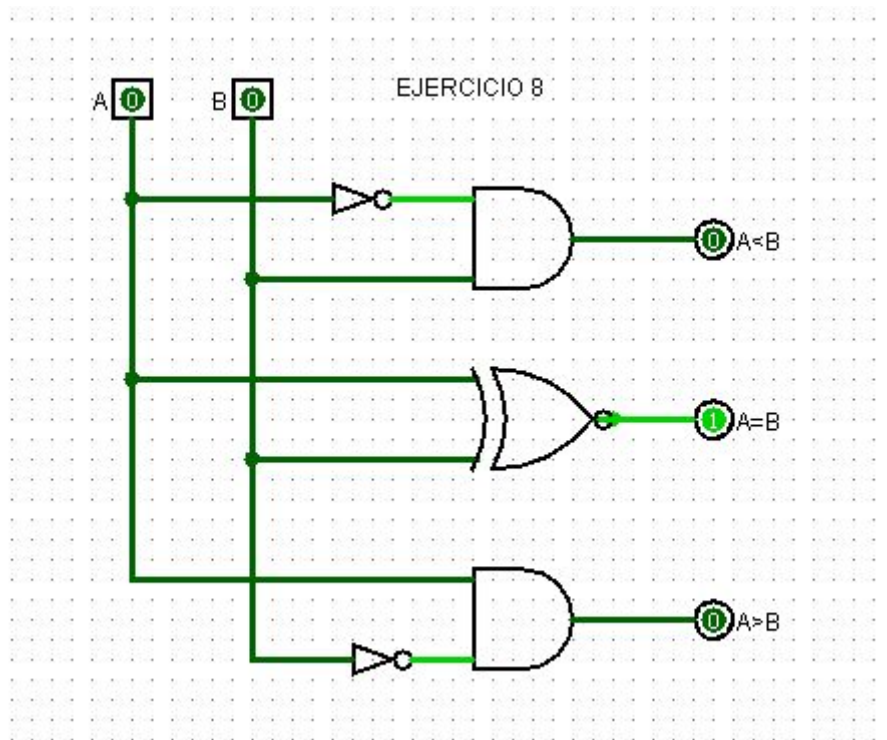
Como podemos observar en la imagen gastando una puerta XOR y otra AND podemos crear un HA donde S es la salida(resultado de la suma) y C es el Acarreo

7-Diseña con puertas lógicas un sumador (FA) de 2 bits teniendo en cuenta la salida y el acarreo tanto de entrada como de salida. Compruébalo con el simulador.

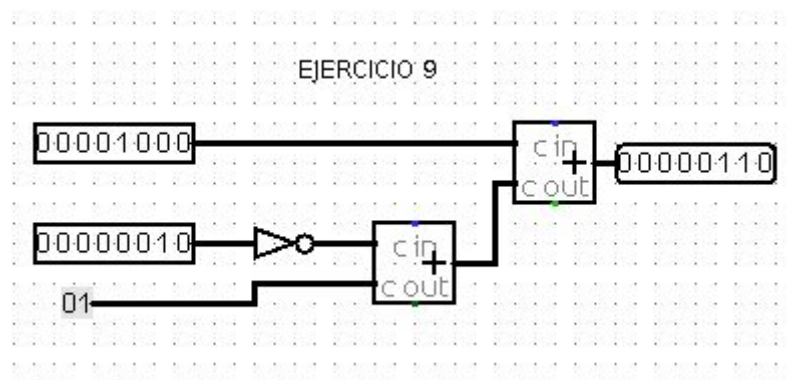
En el sumador FA, con una variable más para poder observar como arrastramos el Acarreo con las puertas AND y una final OR para obtenerlo(C), mientras que el resultado de la suma(S) es con dos XOR.



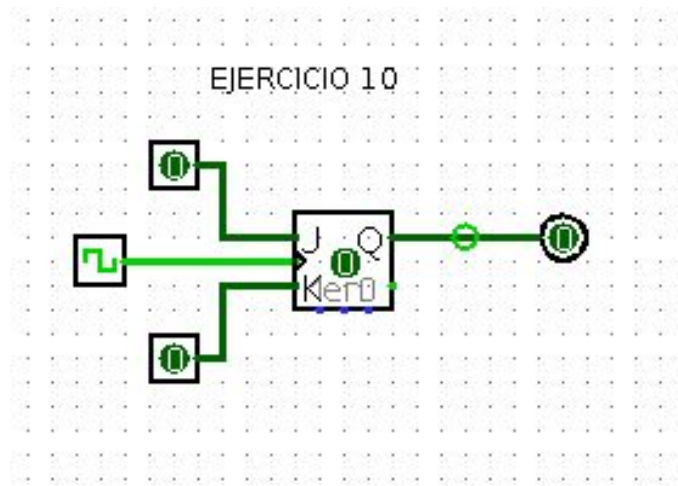
8. Diseña con puertas lógicas un comparador de 2 bits. Compruébalo con el simulador.



9. Crea un circuito restador de 8 bits en logisim, puedes utilizar todos los elementos menos el restador.



10.Crea la tabla de verdad de un biestable J-K teniendo en cuenta los posibles estados anteriores. Comprueba tus resultados con el simulador logisim.



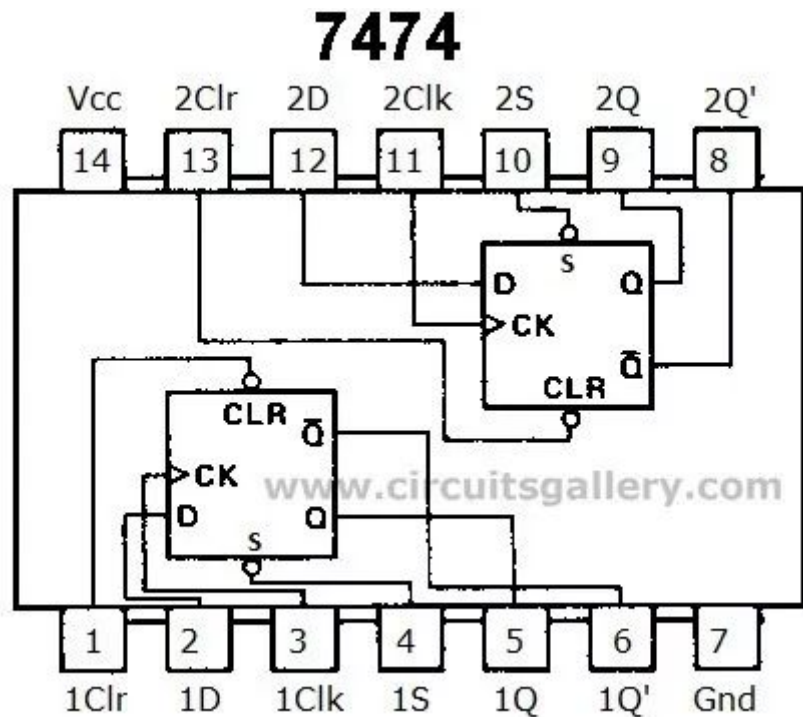
Estado anterior	K	J	Q	Q ₀ '
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

11. Busca en alguna tienda de electrónica un circuito integrado con biestables tipo D. Indica que circuito es, pon una captura del encapsulado del circuito y el diagrama de conexiones de los pines.

7474 Dual

Positive-edge-triggered

D-type Flip-flops With Pre



12. Crea en el simulador un contador del 0 al 15 con biestables J-K comprobando que funciona bien, en la salida debe aparecer la cuenta del 0 al 15 en binario de forma automática.

