

Tema 5: SISTEMAS SECUENCIALES

Grado en Informática

EJERCICIOS RESUELTOS

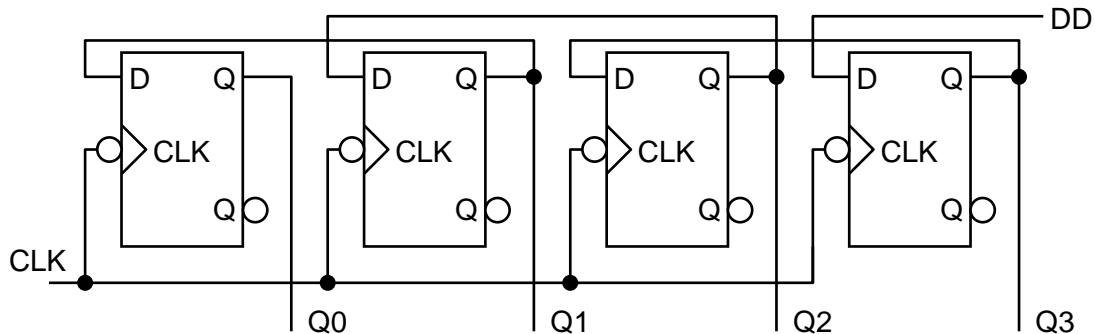
5.1 – Sistemas secuenciales básicos	pág. 2
5.2 –Análisis de sistemas secuenciales síncronos	pág. 8
5.3 –Diseño de autómatas de Moore	pág. 15

5.1 – Sistemas secuenciales básicos

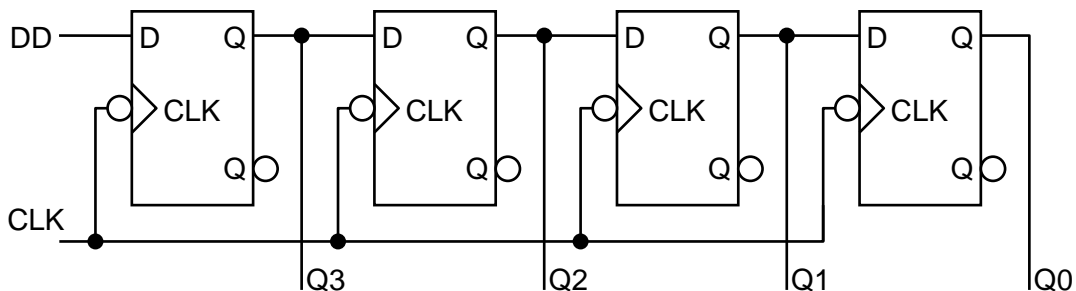
5.1.1.- El registro de desplazamiento con entrada serie y salida paralelo de la transparencia 18 del tema corresponde a un desplazamiento desde el LSB hasta el MSB. Utilice dicho circuito como patrón y diseñe el circuito de un registro de desplazamiento con entrada serie y salida paralelo de 4 bits con desplazamiento desde el MSB hasta el LSB. El nombre de la entrada de datos serie debe ser DD, el reloj CLK y el flanco activo debe ser el flanco de bajada.

Hay al menos dos maneras de dibujar el circuito que se pide.

La primera es poniendo los biestables del LSB al MSB ordenados de izquierda a derecha; así la entrada de datos DD entra a Q3 desde la parte derecha, y los bits se van desplazando hacia la izquierda con cada flanco de reloj, tal y como se muestra en la figura siguiente:



Una manera alternativa de realizar el diseño es cambiar el orden de los biestables, colocando el MSB a la izquierda; en este caso la entrada DD se coloca en la parte izquierda y los desplazamientos ocurren hacia la derecha, como se muestra en la figura siguiente:



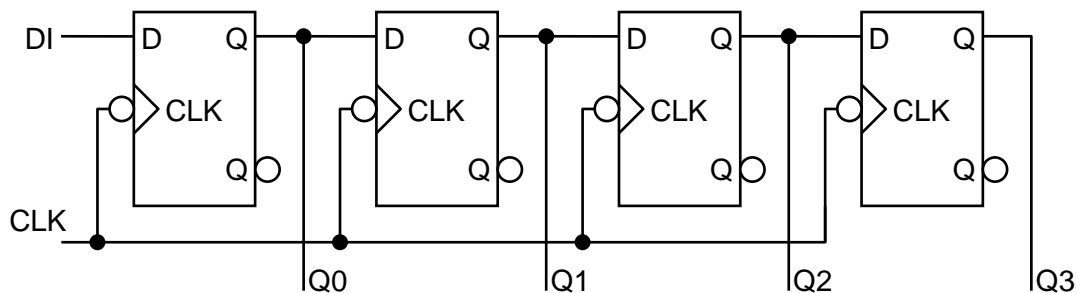
Ambos circuitos son funcionalmente idénticos, pues lo importante es que el valor de la entrada DD esté conectada en ambos circuitos a la entrada del biestable denominado Q3, la salida de Q3 a la entrada de Q2 y así sucesivamente.

5.1.2 .- Diseñe un registro de desplazamiento de 4 bits con dos entradas serie y salida paralelo; para ello debe “reunir” en un único circuito de 4 bits tanto las conexiones del registro de desplazamiento de la transparencia 18 del tema como las del ejercicio anterior. Siga las siguientes indicaciones:

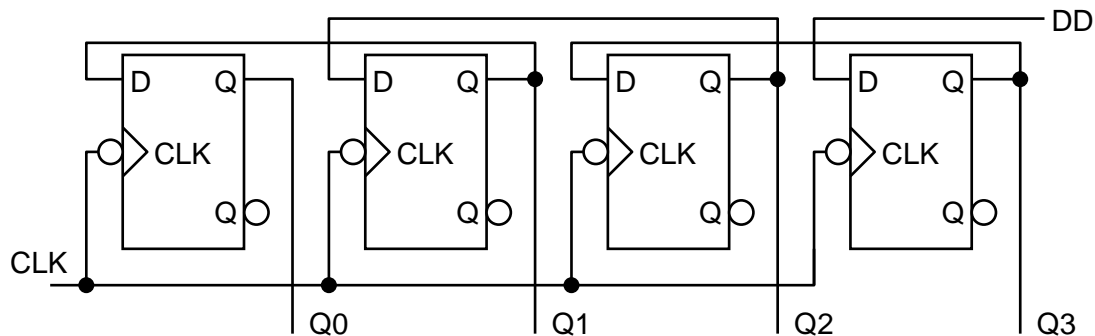
- Nombre CLK a la entrada de reloj. El flanco activo debe ser el flanco de bajada.
- Nombre DI a la entrada de datos serie desde el LSB hasta el MSB.
- Nombre DD a la entrada de datos serie desde el MSB hasta el LSB.
- Nombre DIR a la entrada de control que indique el sentido del desplazamiento, con el siguiente comportamiento:
 - el desplazamiento debe ser $DI \rightarrow Q0 \rightarrow Q1 \rightarrow Q2 \rightarrow Q3$ cuando $DIR = 0$,
 - y debe ser $DD \rightarrow Q3 \rightarrow Q2 \rightarrow Q1 \rightarrow Q0$ cuando $DIR = 1$.
- Utilice un MUX 2a1 para cada uno de los biestables. La entrada de selección de todos ellos debe estar conectada a la entrada DIR, y la salida del MUX directamente conectada a la entrada de datos D del biestable correspondiente.
- Utilice las entradas de datos del MUX para conseguir el comportamiento deseado.

Los dos circuitos que hay que “reunir” son los siguientes:

El primero, un registro de desplazamiento del LSB al MSB, entrada de datos DI,

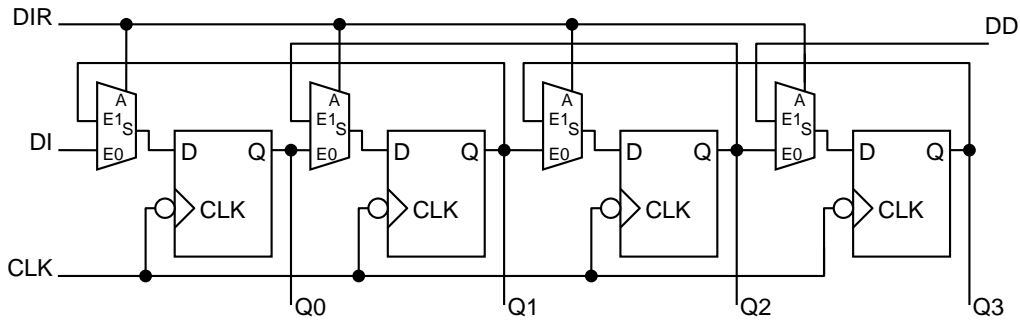


El segundo, un registro de desplazamiento del MSB al LSB, entrada de datos DD,

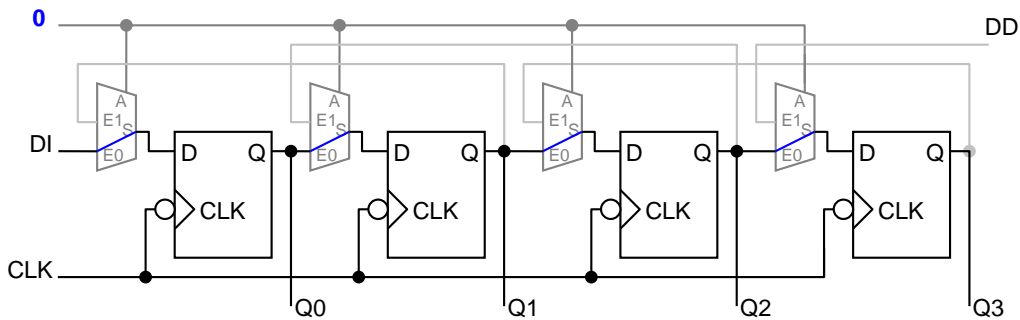


En los dos circuitos mostrados más arriba se ha mantenido el mismo orden de los biestables (colocando el LSB a la izquierda) para que la unión de ambos mediante multiplexores resulte más evidente.

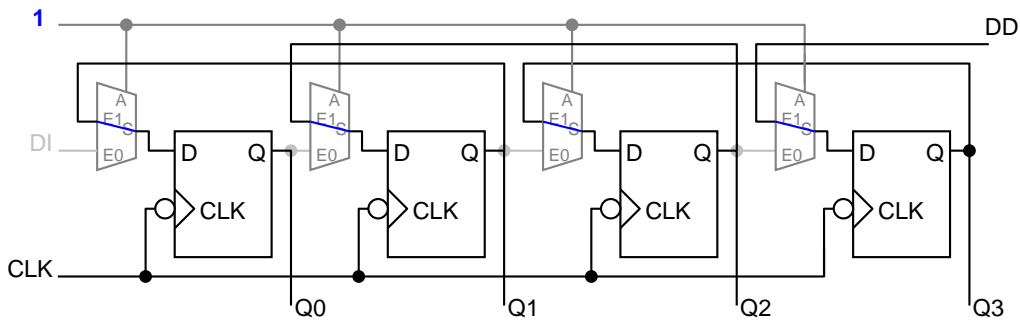
Ahora basta aplicar la indicación de utilizar un MUX 2a1 para determinar el valor de la entrada D de cada biestable con la selección gobernada por el valor de la entrada DIR, para conseguir el circuito mostrado a continuación



Se observa fácilmente que este circuito se comporta como el primero cuando $DIR = 0$ (véase la figura siguiente)



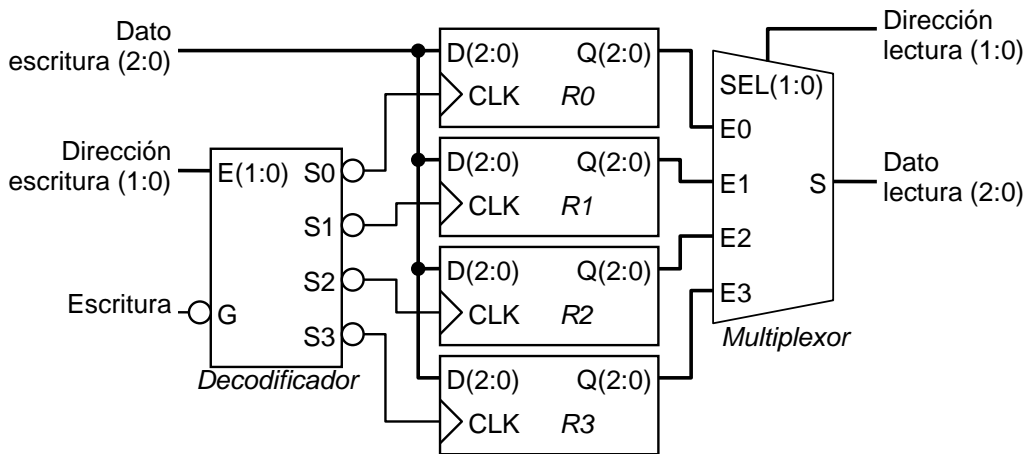
Y como el segundo cuando $DIR = 1$ (véase la figura siguiente)



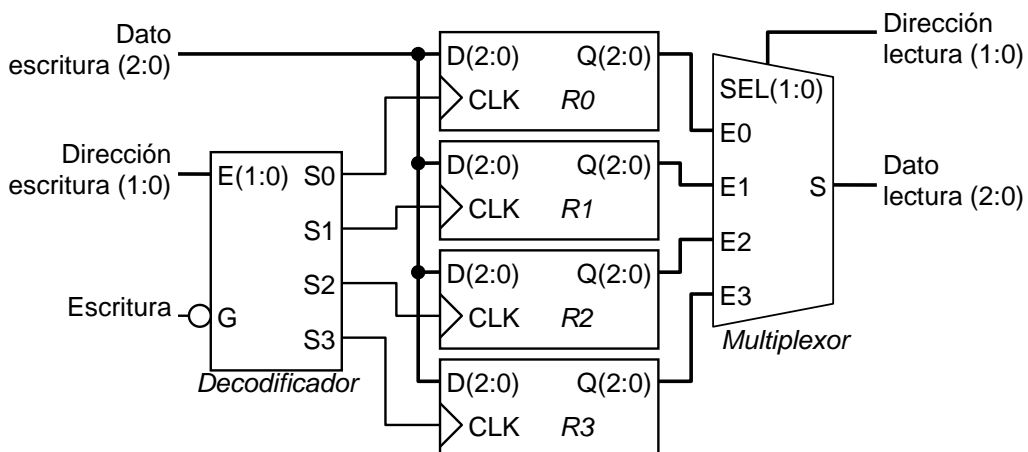
5.1.3.- Diseñe el circuito interno de un banco de registros de 4 registros de 3 bits utilizando únicamente registros, decodificadores y multiplexores con un puerto de lectura y otro de escritura. Los registros son activos por flanco de subida, y la entrada de habilitación del decodificador es a nivel bajo. Dibuje el símbolo lógico que representa a este sistema, e indique claramente si la señal de escritura es activa: i) a nivel alto, ii) a nivel bajo, iii) por flanco de subida, o iv) por flanco de bajada.

Para el diseño del Banco de Registros simplemente seguimos el patrón del circuito mostrado en las transparencias del tema.

Como en el enunciado no se indica nada sobre el nivel de activación de las salidas del decodificador, tenemos dos posibles soluciones. En la primera (que llamaremos solución A, mostrada a continuación), el decodificador utilizado tiene las salidas activas a nivel bajo:



En la segunda (que llamaremos solución B) el decodificador tiene las salidas activas a nivel alto (véase la figura siguiente)



En ambos casos la secuencia de etapas para realizar una operación de escritura son (por orden, y asumiendo que el decodificador está deshabilitado)

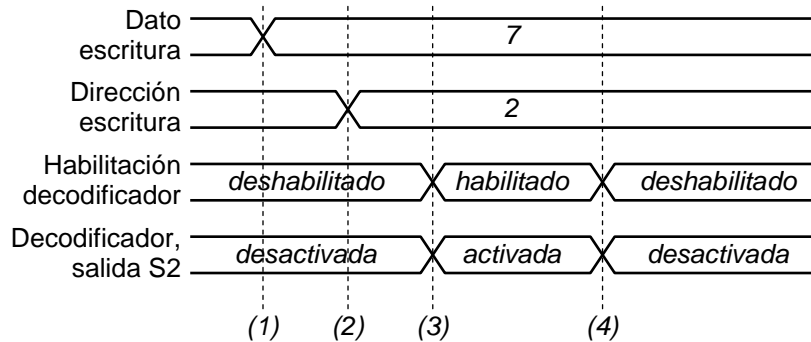
1) Establecer el valor del dato de escritura con el dato que se desea escribir
2) Establecer el valor de la dirección de escritura con el número de registro sobre el que deseamos escribir

3) Habilitar el decodificador

4) Deshabilitar el decodificador

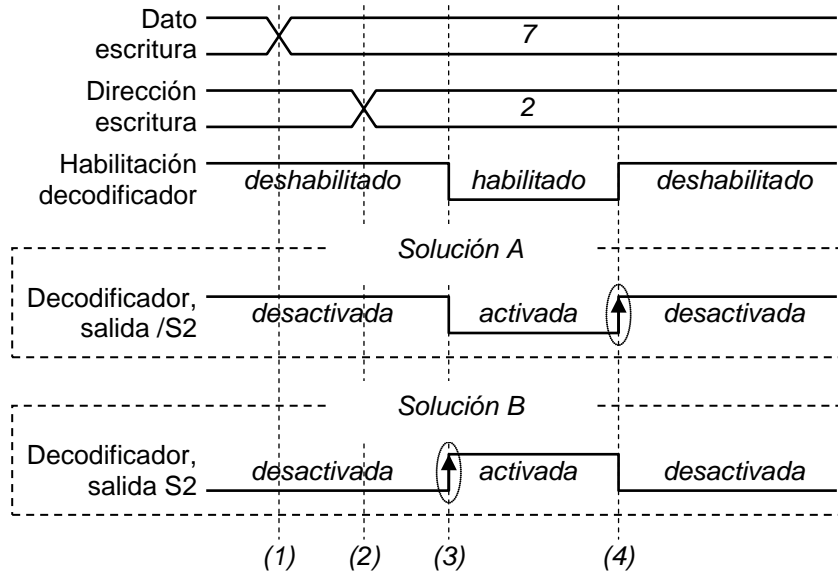
(Nota: Las etapas (1) y (2) se pueden realizar en cualquier orden, dado que estamos suponiendo que el decodificador está deshabilitado y, por tanto, hasta la etapa (3) no puede haber flancos en las líneas de reloj de los registros).

Estas etapas se muestran en el cronograma siguiente mediante un ejemplo (escritura del valor 7 sobre el registro 2). En este cronograma y de forma intencionada se han sustituido los valores 1 y 0 tanto para la habilitación del decodificador y como para la salida del decodificador conectada al reloj del registro 2 por indicaciones simbólicas de cuándo están activadas o desactivadas dichas líneas. De esta manera, el cronograma es aplicable a ambas soluciones.



La diferencia entre ambas soluciones está en el momento en el que se hace efectiva la escritura sobre el registro elegido; este momento puede ser en la etapa (3) o en la (4), dependiendo de: i) el nivel de habilitación de las salidas del decodificador, y ii) del flanco activo de los registros. Además, que la señal de escritura sea activa por flanco de subida o de bajada depende de si el registro se escribe en la etapa (3) o en la (4) y del nivel de habilitación de la entrada de habilitación del decodificador.

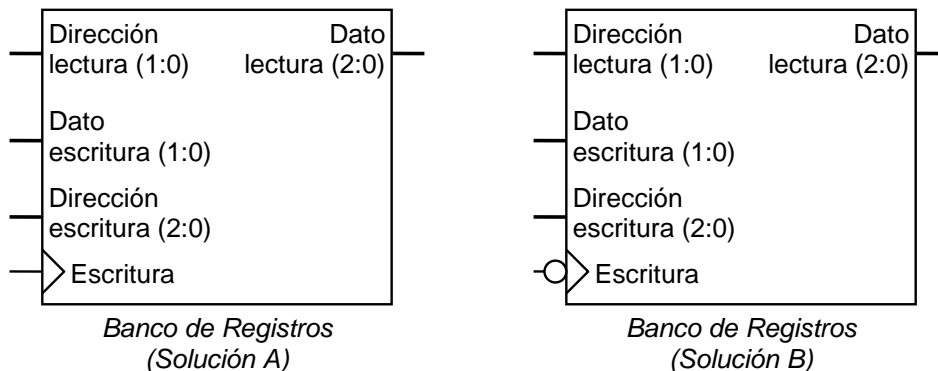
Si aplicamos estos detalles que acabamos de mencionar al cronograma anterior (escritura del valor 7 sobre el registro 2) obtenemos



En el cronograma anterior se ha marcado el flanco de subida de la salida 2 del decodificador, que es cuando la escritura se hace efectiva sobre el registro 2 (porque el registro es activo por flanco de subida).

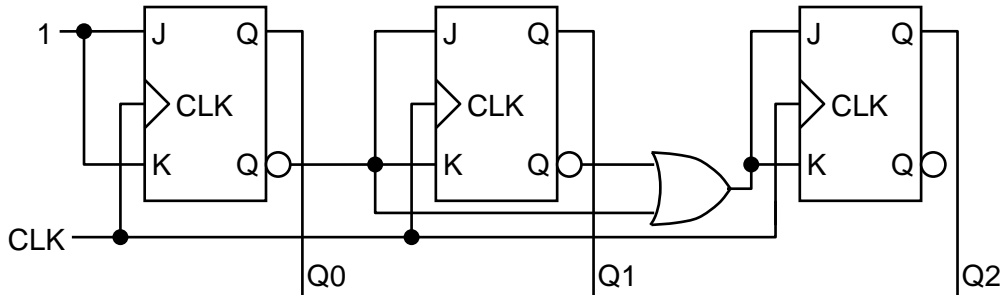
Si relacionamos ahora el instante en el que la escritura se hace efectiva con el flanco correspondiente de la habilitación del decodificador observamos que, para la solución A (escritura efectiva en (4)), el flanco correspondiente de la entrada de habilitación es el de subida. Mientras que en la solución B (escritura efectiva en (3)) el flanco en la entrada de habilitación en el momento de la escritura es el flanco de bajada.

Este análisis es necesario para determinar cómo reflejar correctamente en el símbolo lógico del banco de registros el flanco activo de la entrada de escritura. Así, los símbolos lógicos de ambas soluciones se muestran a continuación



5.2 –Análisis de sistemas secuenciales síncronos

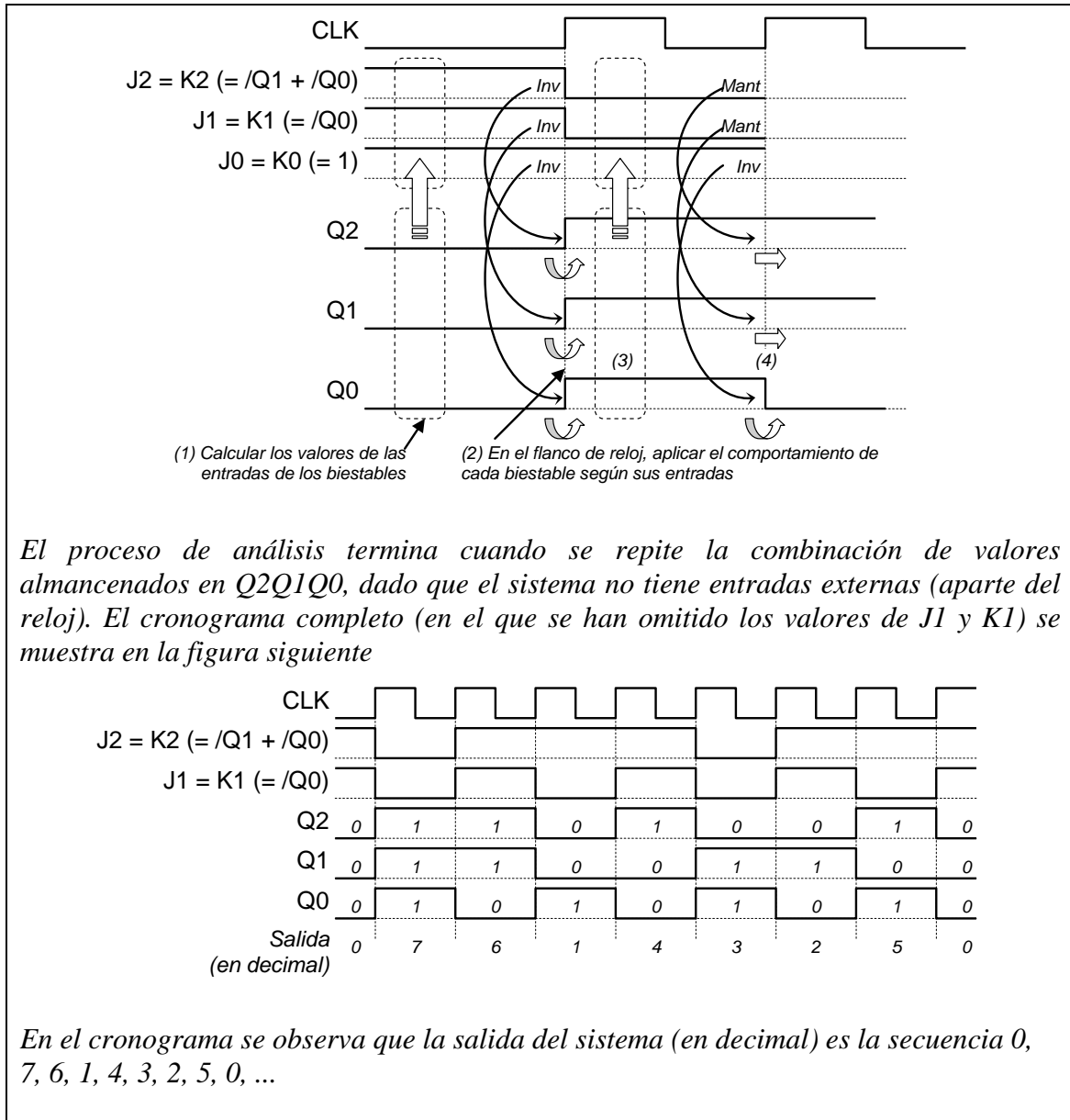
5.2.1.- Analice mediante un cronograma el siguiente circuito. El valor inicial de los biestables es $Q_2Q_1Q_0 = 000$.



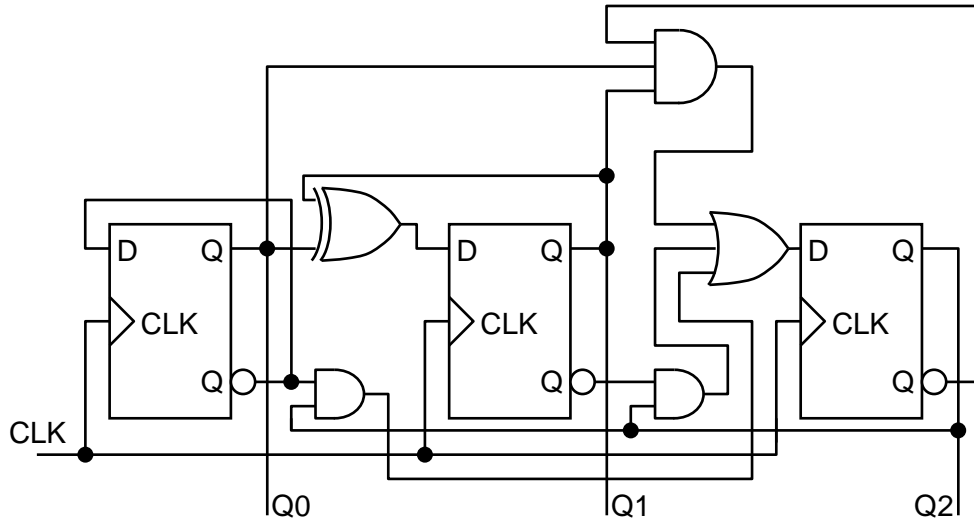
Para hacer más sencillo el análisis incluimos en el cronograma los valores de las entradas J y K de los biestables Q_1 y Q_2 : $J_1 = K_1 = \neg Q_0$ y $J_2 = K_2 = \neg Q_1 + \neg Q_2$

En cada paso de cálculo, en primer lugar y antes del flanco activo del reloj hay que calcular los valores de las entradas de los biestables (J_0 y K_0 son siempre 1, por lo que no es necesario llevar estas entradas al cronograma) para determinar cuál será el comportamiento de cada biestable cuando llegue el flanco activo del reloj. Y en segundo lugar se aplica el flanco activo del reloj a todos los biestables al mismo tiempo, puesto que se trata de un sistema síncrono.

Un ejemplo detallado de este proceso se muestra a continuación (véase figura siguiente). Al comenzar, $Q_2Q_1Q_0 = 000$ (según indicación del enunciado del problema). Con estos valores en (1) se obtiene que $J_1 = K_1 = \neg Q_0 = 1$ y que $J_2 = K_2 = \neg Q_1 + \neg Q_0 = 1$. Como $J_0 = K_0 = 1$ permanentemente, al aplicar en (2) el primer flanco activo del reloj los tres biestables invierten su estado, pasando la cuenta al valor 111. Con este nuevo valor de $Q_2Q_1Q_0$ se recalculan en (3) los valores de J_1 y K_1 (ahora 0) y J_2 y K_2 (ahora 0), por lo que Q_0 invertirá el estado y tanto Q_1 como Q_2 mantendrán el estado al aplicar el segundo flanco de reloj en (4).



5.2.2.- Analice mediante la tabla de estados el circuito siguiente. Obtenga el diagrama de estados correspondiente e indique la secuencia de salida del sistema. El valor inicial de los biestables es $Q_2Q_1Q_0 = 000$.



En primer lugar se obtienen las ecuaciones de las entradas de los biestables, que corresponden a la función de excitación del sistema.

$$D0 = \neg Q0; \quad D1 = Q1 \oplus Q0; \quad D2 = Q2 \cdot \neg Q0 + \neg Q2 \cdot Q1 + \neg Q2 \cdot Q1 \cdot Q0;$$

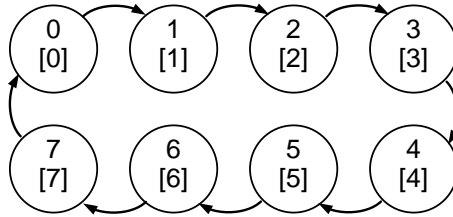
Con dichas ecuaciones se obtiene la tabla de verdad de la función de excitación sin más que sustituir en las ecuaciones anteriores las variables de entrada ($Q2Q1Q0$) por sus valores en cada una de las filas de la tabla.

Q2	Q1	Q0	D2	D1	D0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

La función de excitación coincide con el estado siguiente ya que se trata de biestables D. Es posible entonces obtener la tabla de estados con los estados codificados sin más que observar que la salida en cada estado coincide con el valor del propio estado.

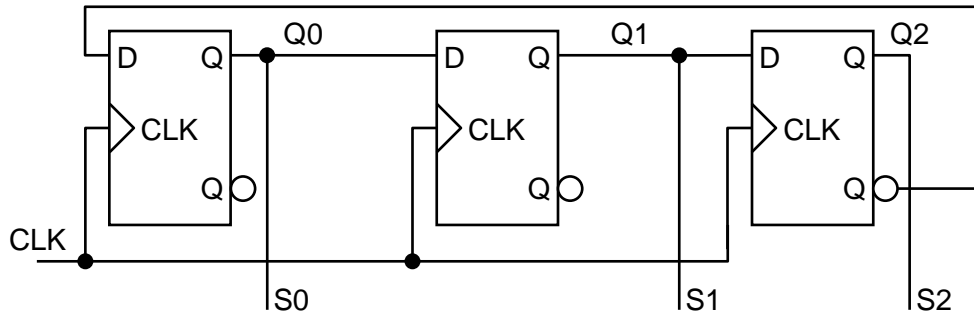
Estado actual			Estado siguiente			Salida		
Q2(t)	Q1(t)	Q0(t)	Q2(t+1)	Q1(t+1)	Q0(t+1)	Q2	Q1	Q0
0	0	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0	1
0	1	0	0	1	1	0	1	0
0	1	1	1	0	0	0	1	1
1	0	0	1	0	1	1	0	0
1	0	1	1	1	0	1	0	1
1	1	0	1	1	1	1	1	0
1	1	1	0	0	0	1	1	1

A partir de la tabla anterior, y utilizando los valores de estado (en decimal) como nombres para los estados es sencillo derivar el diagrama de estados del sistema.



Por último, a partir del estado inicial 000, se recorre el diagrama de estados para obtener que la secuencia de salida del sistema (en decimal) es 0, 1, 2, 3, 4, 5, 6, 7, 0, ...

5.2.3.- Analice el siguiente circuito. El valor inicial de los biestables es $Q_2Q_1Q_0 = 000$. Puede realizar el análisis mediante cronograma o tabla de estados; en este último caso debe obtener el diagrama de estados correspondiente antes de obtener la secuencia de valores de salida del sistema.



Para analizar este sistema (o cualquier otro) mediante la tabla de estados y el diagrama de estados, se obtienen en primer lugar las ecuaciones correspondientes a la excitación de los biestables (las ecuaciones correspondientes a sus entradas).

$$D_0 = \neg Q_2; \quad D_1 = Q_0; \quad D_2 = Q_1;$$

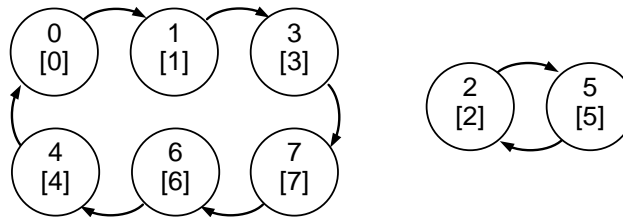
Con dichas ecuaciones se obtiene la tabla de verdad de la función de excitación sin más que sustituir en las ecuaciones anteriores las variables de entrada ($Q_2Q_1Q_0$) por sus valores en cada una de las filas de la tabla.

Q2	Q1	Q0	D2	D1	D0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	0

La función de excitación coincide con el estado siguiente ya que se trata de biestables D. Es posible entonces obtener la tabla de estados con los estados codificados sin más que observar que la salida en cada estado coincide con el valor del propio estado.

Estado actual			Estado siguiente			Salida		
Q2(t)	Q1(t)	Q0(t)	Q2(t+1)	Q1(t+1)	Q0(t+1)	S2	S1	S0
0	0	0	0	0	1	0	0	0
0	0	1	0	1	1	0	0	1
0	1	0	1	0	1	0	1	0
0	1	1	1	1	1	0	1	1
1	0	0	0	0	0	1	0	0
1	0	1	0	1	0	1	0	1
1	1	0	1	0	0	1	1	0
1	1	1	1	1	0	1	1	1

A partir de la tabla anterior, y utilizando los valores de estado (en decimal) como nombres para los estados es sencillo derivar el diagrama de estados del sistema.

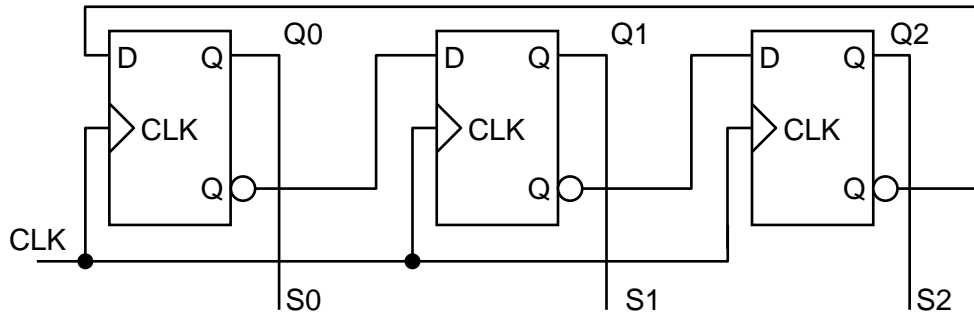


Por último, a partir del estado inicial 000, se recorre el diagrama de estados para obtener que la secuencia de salida del sistema (en decimal) es 0, 1, 2, 3, 7, 6, 4, 0, ...

Nótese que en este sistema los estados 2 y 5 no son alcanzables desde el resto de estados; eso significa que si el estado inicial hubiera sido 010, la secuencia de salida del sistema habría sido 2, 5, 2, ...

Generalmente estos estados inalcanzables aparecen en un sistema cuando, al realizar el diseño y durante la etapa de codificación, quedan códigos binarios sin utilizar. El estado siguiente para estos códigos binarios de estado no utilizados (no asignados a estados presentes en el diagrama de estados del sistema) es $xx...x$ (es decir, cualquier estado siguiente es válido). Dependiendo de cómo se implemente la función de excitación (con o sin simplificar, por ejemplo) dichas combinaciones indiferentes resultan en estados que sí aparecen en el diseño original o, como ocurre en este caso, con otro código no utilizado.

5.2.4.- Analice el siguiente circuito. El valor inicial de los biestables es $Q_2Q_1Q_0 = 000$. Puede realizar el análisis mediante cronograma o tabla de estados; en este último caso debe obtener el diagrama de estados correspondiente antes de obtener la secuencia de valores de salida del sistema.



Repetimos el proceso del ejercicio anterior para analizar este sistema mediante la tabla de estados y el diagrama de estados.

Primero las ecuaciones correspondientes a la excitación de los biestables.

$$D_0 = \neg Q_2; \quad D_1 = \neg Q_0; \quad D_2 = \neg Q_1;$$

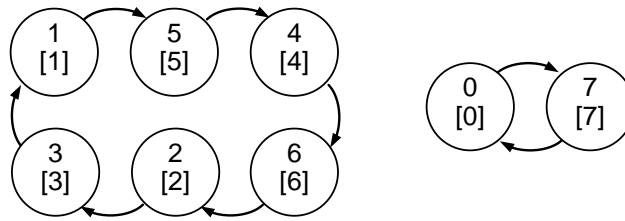
Después la tabla de verdad de la función de excitación.

Q2	Q1	Q0	D2	D1	D0
0	0	0	1	1	1
0	0	1	1	0	1
0	1	0	0	1	1
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	1	0	0	0

Posteriormente la tabla de estados con los estados codificados (de nuevo la salida en cada estado coincide con el valor del propio estado).

Estado actual			Estado siguiente			Salida		
Q2(t)	Q1(t)	Q0(t)	Q2(t+1)	Q1(t+1)	Q0(t+1)	S2	S1	S0
0	0	0	1	1	1	0	0	0
0	0	1	1	0	1	0	0	1
0	1	0	0	1	1	0	1	0
0	1	1	0	0	1	0	1	1
1	0	0	1	1	0	1	0	0
1	0	1	1	0	0	1	0	1
1	1	0	0	1	0	1	1	0
1	1	1	0	0	0	1	1	1

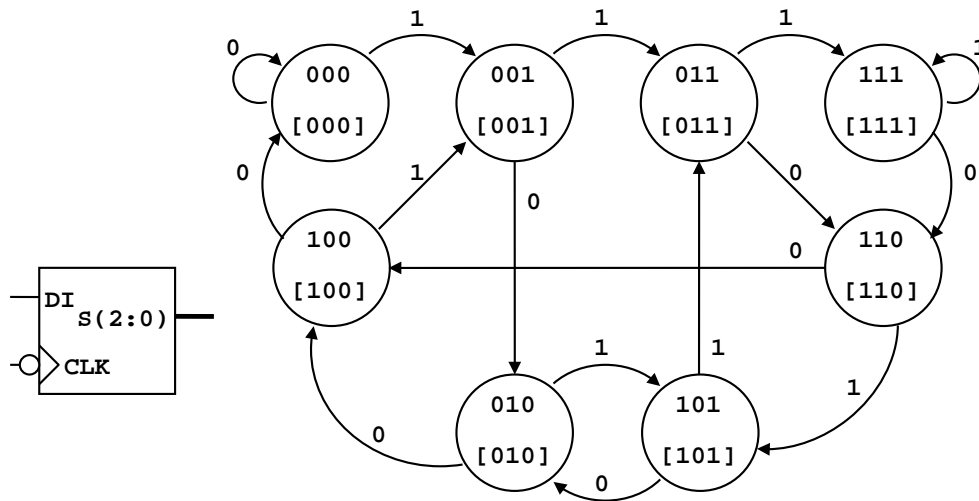
Y finalmente el diagrama de estados del sistema.



Recorriendo el diagrama de estados se obtiene que, a partir del estado inicial 000, la secuencia de salida del sistema (en decimal) es 0, 7, 0, ...

5.3 –Diseño de autómatas de Moore

5.3.1.- A continuación se muestran tanto la interfaz como el diagrama de estados del autómata de Moore que representa a un registro de desplazamiento con entrada serie (llamada DI) y salida paralelo de 3 bits (llamada S) con desplazamiento desde el LSB hasta el MSB. Como nombre del estado se ha empleado el valor almacenado en el registro (en el orden Q2,Q1,Q0), por lo que estado y salida coinciden.



Realice las etapas de diseño restantes y compare el circuito resultante con el de la transparencia 18 del tema. Para terminar el diseño debe:

- Rellenar la tabla de estados.
- Escoger una codificación para los estados. Debe escoger una codificación que permita simplificar al máximo la función de salida del sistema.
- Rellenar la tabla de estados con los estados codificados.
- Rellenar las tablas de verdad de las funciones de excitación y salida.
- Calcular las ecuaciones de las funciones de excitación y salida.
- Dibujar el circuito resultante.

a) La tabla de estados se deriva fácilmente del diagrama de estados.

Estado actual			Estado siguiente			Salida		
			Q2(t+1)	Q1(t+1)	Q0(t+1)			
Q2(t)	Q1(t)	Q0(t)	Entrada DI		S2	S1	S0	
			DI = 0	DI = 1				
0	0	0	0 0 0	0 0 1	0	0	0	
0	0	1	0 1 0	0 1 1	0	0	1	
0	1	0	1 0 0	1 0 1	0	1	0	
0	1	1	1 1 0	1 1 1	0	1	1	
1	0	0	0 0 0	0 0 1	1	0	0	
1	0	1	0 1 0	0 1 1	1	0	1	
1	1	0	1 0 0	1 0 1	1	1	0	
1	1	1	1 1 0	1 1 1	1	1	1	

b) La codificación en realidad ya se ha hecho, puesto que en el enunciado del problema se indica que estado y salida coinciden.

c) Por tanto, la tabla de estados con los estados codificados es directamente la tabla de estados obtenida en el apartado (a).

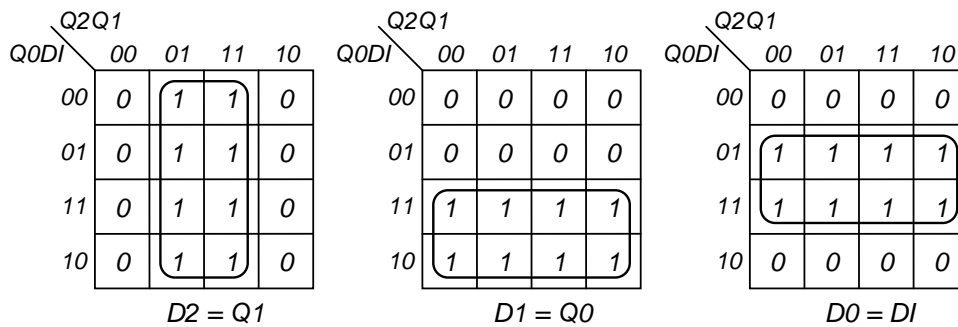
d) Las tablas de verdad de las funciones de excitación y salida se derivan de la tabla de estados con los estados codificados, separando por un lado la información sobre las transiciones entre estados (función de excitación) y por otro la información sobre el valor de las salidas en cada estado (función de salida).

Q2	Q1	Q0	DI	D2	D1	D0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	1	0	0
0	1	0	1	1	0	1
0	1	1	0	1	1	0
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	1	0	0
1	1	0	1	1	0	1
1	1	1	0	1	1	0
1	1	1	1	1	1	1

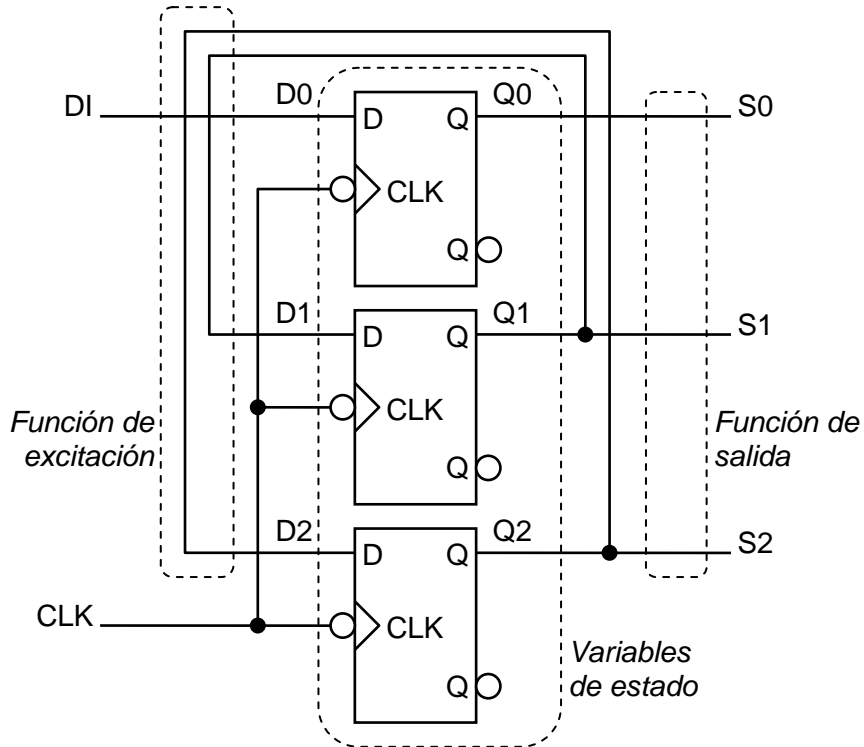
Función de excitación

Q2	Q1	Q0	S2	S1	S0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	1

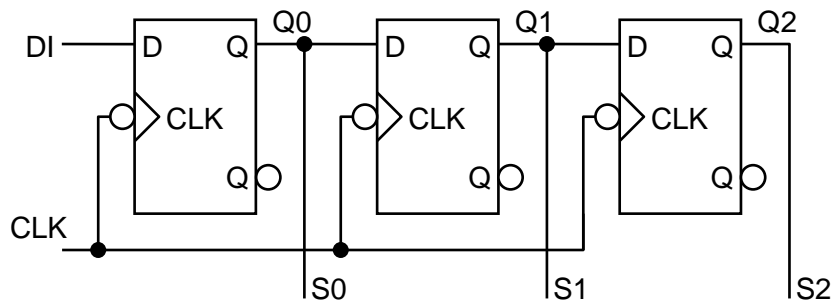
e) Utilizamos mapas de Karnaugh para simplificar la función de excitación y obtener las ecuaciones correspondientes. Esto no hace falta para la función de salida, puesto que coincide con el estado ($S2 = Q2$, $S1 = Q1$ y $S0 = Q0$).



f) Ahora se reúnen los diferentes elementos del sistema (variables de estados, función de excitación y función de salida) siguiendo la estructura de un autómata de Moore.



Simplemente reordenando los elementos del circuito anterior obtenemos el esquema típico de un registro de desplazamiento (véase el primer circuito de la solución del ejercicio 5.1.2, con la única salvedad de que aquél es de cuatro bits)



5.3.2.- Utilice el diagrama de estados mostrado en el ejercicio anterior como modelo para realizar todas las etapas del diseño de un registro de desplazamiento con entrada serie y salida paralelo de 3 bits con desplazamiento desde el MSB hasta el LSB. El nombre de la entrada de datos serie debe ser DD. Para ello debe:

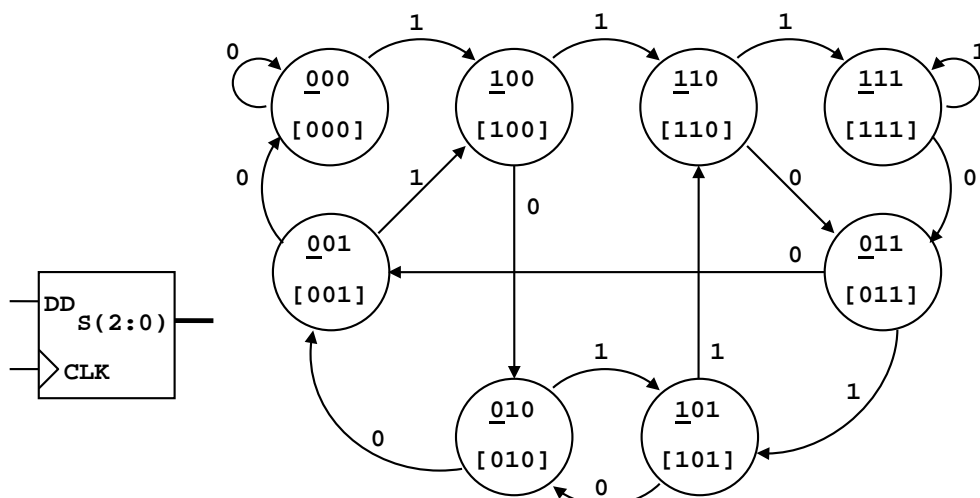
- Dibujar la interfaz del sistema. La entrada de datos serie debe llamarse DD, el reloj CLK (activo por flanco de subida) y la salida S(2:0).
- Dibujar el diagrama de estados. Utilice para este diagrama el valor (de 3 bits) que debe almacenar en cada momento el circuito como nombre de estado (en el orden Q2,Q1,Q0).
- Rellenar la tabla de estados.
- Escoger una codificación para los estados. Debe escoger una codificación que permita simplificar al máximo la función de salida del sistema.
- Rellenar la tabla de estados con los estados codificados.
- Rellenar las tablas de verdad de las funciones de excitación y salida.
- Calcular las ecuaciones de las funciones de excitación y salida.
- Dibujar el circuito resultante.

Una vez realizados todos los pasos, compare el circuito resultante con el del ejercicio 5.1.1.

a,b) La interfaz del sistema es prácticamente idéntica a la del circuito del ejercicio anterior, con las siguientes salvedades: i) El reloj es activo por flanco de subida, y ii) el nombre de la entrada de datos serie es, en este caso, DD.

En cuanto al diagrama de estados, es muy similar, excepto que como en este caso el dato de entrada (DD) entra al bit MSB y la información almacenada se va desplazando hacia el LSB, los valores de estado en este diagrama de estado corresponden a los del diagrama del ejercicio anterior pero en orden inverso (es decir, el estado alcanzado desde el 000 cuando la entrada es DD = 1 ahora es 100, cuando en el ejercicio anterior era 001).

Para facilitar la comprensión de este diagrama se ha subrayado en cada estado el dato recién incorporado a la memoria del registro.



b) Tabla de estados.

Estado actual Q2(t) Q1(t) Q0(t)			Estado siguiente Q2(t+1) Q1(t+1) Q0(t+1)					
			Entrada DD		Salida			
			DD = 0	DD = 1	S2	S1	S0	
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	1	0	1	0
0	1	1	0	0	1	0	1	1
1	0	0	0	1	0	1	0	0
1	0	1	0	1	0	1	0	1
1	1	0	0	1	1	1	1	0
1	1	1	0	1	1	1	1	1

c, d) De nuevo la codificación ya se ha hecho y la tabla de estados con los estados codificados es directamente la tabla de estados obtenida en el apartado anterior.

e) Tablas de verdad de las funciones de excitación y salida.

Q2	Q1	Q0	DD	D2	D1	D0	Función de excitación
0	0	0	0	0	0	0	
0	0	0	1	1	0	0	
0	0	1	0	0	0	0	
0	0	1	1	1	0	0	
0	1	0	0	0	0	1	
0	1	0	1	1	0	1	
0	1	1	0	0	0	1	
0	1	1	1	1	0	1	
1	0	0	0	0	1	0	
1	0	0	1	1	1	0	
1	0	1	0	0	1	0	
1	0	1	1	1	1	0	
1	1	0	0	0	1	1	
1	1	0	1	1	1	1	
1	1	1	0	0	1	1	
1	1	1	1	1	1	1	

Función de salida					
Q2	Q1	Q0	S2	S1	S0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	1

f) Mapas de Karnaugh para simplificar la función de excitación y obtener las ecuaciones correspondientes. De nuevo no hace falta para la función de salida, puesto que coincide con el estado ($S2 = Q2$, $S1 = Q1$ y $S0 = Q0$).

Q2Q1		Q0DD			
		00	01	11	10
00	0	0	0	0	
01	1	1	1	1	
11	1	1	1	1	
10	0	0	0	0	

$D2 = DD$

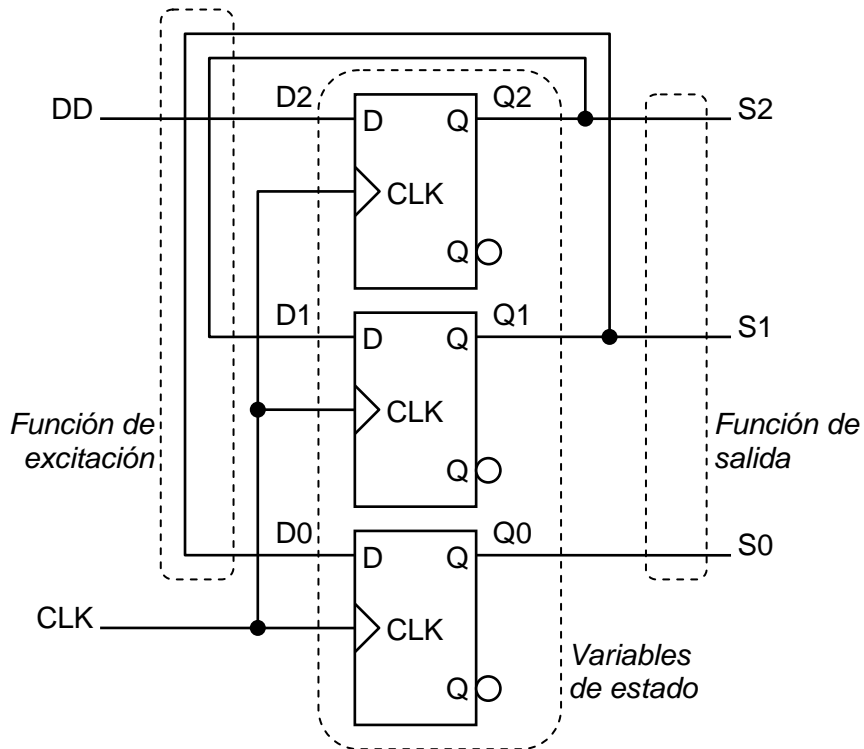
Q2Q1		Q0DD			
		00	01	11	10
00	0	0	1	1	
01	0	0	1	1	
11	0	0	1	1	
10	0	0	1	1	

$D1 = Q2$

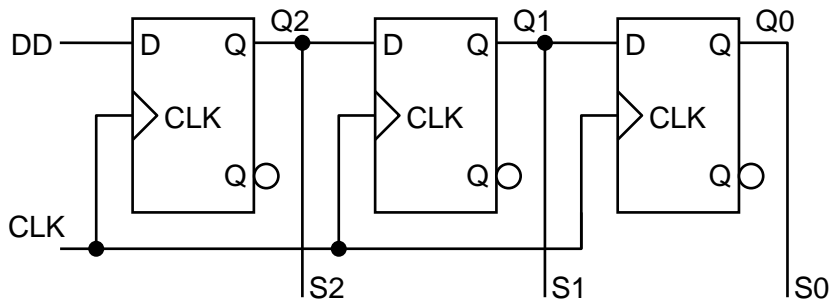
Q2Q1		Q0DD			
		00	01	11	10
00	0	1	1	0	
01	0	1	1	0	
11	0	1	1	0	
10	0	1	1	0	

$D0 = Q1$

g) Ahora se reúnen los diferentes elementos del sistema (variables de estados, función de excitación y función de salida) siguiendo la estructura de un autómata de Moore.



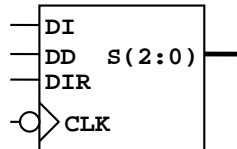
Y reordenando los elementos del circuito



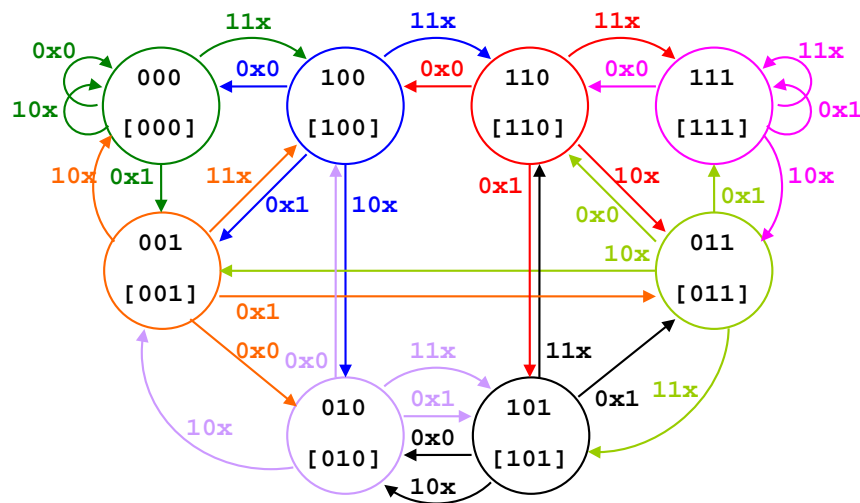
Que, evidentemente, coincide (con la salvedad de que éste es de tres bits y por flanco de subida) con el segundo circuito mostrado en la solución del ejercicio 5.1.1

5.3.3 .- Realice todas las etapas de diseño (dibuje la interfaz y el diagrama de estados, rellene la tabla de estados, escoja una codificación, rellene la tabla de estados con los estados codificados, rellene las tablas de verdad de las funciones de excitación y salida, calcule las ecuaciones de las funciones de excitación y salida y dibuje el circuito resultante) de un registro de desplazamiento de 3 bits como el descrito en el ejercicio 5.1.2. Terminado el diseño compare el circuito resultante con el del ejercicio 5.1.2.

La interfaz del sistema debe contemplar todas las entradas y salidas. Por lo tanto es



Para las transiciones del diagrama de estados pueden utilizarse tanto ecuaciones lógicas con las entradas del sistema como vectores de los valores de dichas entradas. Por comodidad se han empleado en este caso vectores en el orden DIR DD DI. El diagrama de estados se muestra en la figura siguiente; se han empleado diferentes colores para los distintos estados a fin de facilitar en lo posible la comprensión del diagrama.



De cada estado parten cuatro transiciones que representan las ocho posibles combinaciones de las entradas DIR, DD y DI agrupadas por parejas.

Estas parejas tienen en cuenta que cuando $DIR = 1$ (desplazamiento $DD \rightarrow Q2 \rightarrow Q1 \rightarrow Q0$) el valor de la entrada DI no es relevante y, por tanto, las transiciones que salen de un estado emplean los vectores 10x y 11x (el primero cuando $DIR = 1, DD = 0, DI$ irrelevante y el segundo cuando $DIR = 1, DD = 1, DI$ irrelevante).

Análogamente, los vectores de valores que representan las transiciones cuando $DIR = 0$ (desplazamiento $DI \rightarrow Q0 \rightarrow Q1 \rightarrow Q2$) emplean los vectores 0x0 ($DIR = 0, DI$ irrelevante, $DI = 0$) y 0x1 ($DIR = 0, DI$ irrelevante, $DI = 1$).

b) Como el sistema dispone de 3 entradas, en cada fila de la tabla de estados deben aparecer 8 valores de estado siguiente, uno por cada combinación de valores de las entradas DIR, DD y DI. Como esta tabla es demasiado grande, se puede dividir en dos.

La primera parte de la tabla de estados contiene la información de las transiciones entre estados cuando $DIR = 0$ (en estas transiciones, puesto que DD no es relevante, el estado siguiente para las combinaciones DD DI = 00 y 10 coinciden, así como también coinciden en los casos DD DI = 01 y 11)

Estado actual			Estado siguiente				Salida		
			Q2(t+1)	Q1(t+1)	Q0(t+1)				
Entradas DD, DI (sólo los casos DIR = 0)									
Q2(t)	Q1(t)	Q0(t)	00	01	10	11	S2	S1	S0
0	0	0	0 0 0	0 0 1	0 0 0	0 0 1	0	0	0
0	0	1	0 1 0	0 1 1	0 1 0	0 1 1	0	0	1
0	1	0	1 0 0	1 0 1	1 0 0	1 0 1	0	1	0
0	1	1	1 1 0	1 1 1	1 1 0	1 1 1	0	1	1
1	0	0	0 0 0	0 0 1	0 0 0	0 0 1	1	0	0
1	0	1	0 1 0	0 1 1	0 1 0	0 1 1	1	0	1
1	1	0	1 0 0	1 0 1	1 0 0	1 0 1	1	1	0
1	1	1	1 1 0	1 1 1	1 1 0	1 1 1	1	1	1

Y la segunda parte, cuando $DIR = 1$ (en la que se ve claramente que coincide el estado siguiente para las combinaciones DD DI = 00 y 01, y también coinciden para las combinaciones DD DI = 10 y 11)

Estado actual			Estado siguiente				Salida		
			Q2(t+1)	Q1(t+1)	Q0(t+1)				
Entradas DD, DI (sólo los casos DIR = 1)									
Q2(t)	Q1(t)	Q0(t)	00	01	10	11	S2	S1	S0
0	0	0	0 0 0	0 0 0	1 0 0	1 0 0	0	0	0
0	0	1	0 0 0	0 0 0	1 0 0	1 0 0	0	0	1
0	1	0	0 0 1	0 0 1	1 0 1	1 0 1	0	1	0
0	1	1	0 0 1	0 0 1	1 0 1	1 0 1	0	1	1
1	0	0	0 1 0	0 1 0	1 1 0	1 1 0	1	0	0
1	0	1	0 1 0	0 1 0	1 1 0	1 1 0	1	0	1
1	1	0	0 1 1	0 1 1	1 1 1	1 1 1	1	1	0
1	1	1	0 1 1	0 1 1	1 1 1	1 1 1	1	1	1

c, d) De nuevo la codificación ya se ha hecho y la tabla de estados con los estados codificados es directamente la tabla de estados obtenida en el apartado anterior.

e) No es necesario reflejar la tabla de verdad de la función de salida dado que estado y salida coinciden, y el circuito resultante es $S2 = Q2$, $S1 = Q1$ y $S0 = Q0$. En cuanto a la función de excitación, se trata de una función lógica con 6 variables (3 variables de estado y tres entradas externas).

Igual que en el caso de la tabla de estados, la función de excitación se puede dividir en dos para que resulte más manejable. Para que sea más sencillo trasvasar la información de la tabla de estados se ha utilizado la entrada DIR como la de mayor peso de la función de excitación, de manera que las dos partes en las que se ha dividido esta tabla coinciden con las dos partes de la tabla de estados.

Función de excitación

DIR	Q2	Q1	Q0	DD	DI	D2	D1	D0	DIR	Q2	Q1	Q0	DD	DI	D2	D1	D0
0	0	0	0	x	0	0	0	0	1	0	0	0	0	x	0	0	0
0	0	0	0	x	1	0	0	1	1	0	0	0	1	x	1	0	0
0	0	0	1	x	0	0	1	0	1	0	0	1	0	x	0	0	0
0	0	0	1	x	1	0	1	1	1	0	0	1	1	x	1	0	0
0	0	1	0	x	0	1	0	0	1	0	1	0	0	x	0	0	1
0	0	1	0	x	1	1	0	1	1	0	1	0	1	x	1	0	1
0	0	1	1	x	0	1	1	0	1	0	1	1	0	x	0	0	1
0	0	1	1	x	1	1	1	1	1	0	1	1	1	x	1	0	1
0	1	0	0	x	0	0	0	0	1	1	0	0	0	x	0	1	0
0	1	0	0	x	1	0	0	1	1	1	0	0	1	x	1	1	0
0	1	0	1	x	0	0	1	0	1	1	0	1	0	x	0	1	0
0	1	0	1	x	1	0	1	1	1	1	0	1	1	x	1	1	0
0	1	1	0	x	0	1	0	0	1	1	1	0	0	x	0	1	1
0	1	1	0	x	1	1	0	1	1	1	1	0	1	x	1	1	1
0	1	1	1	x	0	1	1	0	1	1	1	1	0	x	0	1	1
0	1	1	1	x	1	1	1	1	1	1	1	1	1	x	1	1	1

f) No se pueden emplear directamente mapas de Karnaugh para simplificar la función de excitación, ya que el número de variables de entrada es demasiado grande. Por otro lado, construir el circuito a partir de las ecuaciones sin simplificar de D2, D1 y D0 (empleando, por ejemplo la suma de los minitérminos o el producto de los maxitérminos que pertenecen a D2, D1 y D0 respectivamente) resultaría demasiado complejo.

A continuación se introducen un par de métodos para obtener ecuaciones de la función de excitación, métodos que quedan (estrictamente hablando) fuera del alcance de los contenidos de la asignatura, aunque interesantes por lo que tienen de ampliación.

En cualquier caso, para poder continuar será necesario aceptar que unas ecuaciones razonablemente simples para la función de excitación son

$$D2 = Q1 \cdot \overline{DIR} + DD \cdot DIR; \quad D1 = Q0 \cdot \overline{DIR} + Q2 \cdot DIR; \quad D0 = DI \cdot \overline{DIR} + Q1 \cdot DIR;$$

Las alternativas que quedan al diseñador para obtener el circuito de la función de excitación son

f.1) Emplear una aplicación software que emplee un método de simplificación alternativo a los mapas de Karnaugh y que permita simplificar funciones de muchas variables. Un ejemplo de este tipo de algoritmo de simplificación es el denominado Quine-McCluskey (véase el enlace de la Wikipedia sobre el algoritmo en http://es.wikipedia.org/wiki/Algoritmo_Quine-McCluskey y una explicación aplicada a un ejemplo en <http://foro.blogelectronico.com/sistemas-digitales/metodo-quine-mccluskey/?action=dlattach;attach=40>).

No es difícil encontrar en Internet aplicaciones que implementan dichos algoritmos. Por ejemplo, en [http://en.literateprograms.org/Quine-McCluskey_algorithm_\(Java\)](http://en.literateprograms.org/Quine-McCluskey_algorithm_(Java)) se puede descargar una aplicación

Java que permite simplificar funciones lógicas a partir de sus minterminos utilizando simplificación mediante Quine-McCluskey.

Esta aplicación de simplificación para las columnas D2, D1 y D0 da como resultado el texto que se muestra a continuación (los ficheros de texto con los minterminos de cada columna están disponibles en PoliformaT con los nombres *d2_ejercicio_5_3_3.txt*, *d1_ejercicio_5_3_3.txt* y *d0_ejercicio_5_3_3.txt* para poder ejecutar el programa java mencionado anteriormente y obtener estos mismos resultados).

Simplificación D2:	Simplificación D1:	Simplificación D0:
2 terms, 6 vars	2 terms, 6 vars	2 terms, 6 vars
{0 X 1 X X X}	{0 X X 1 X X}	{0 X X X X 1}
{1 X X X 1 X}	{1 1 X X X X}	{1 X 1 X X X}

Téngase en cuenta que los términos indicados por el programa corresponden a la suma de términos producto con las variables en el orden de la tabla de verdad – DIR Q2 Q1 Q0 DD DI – y que, como términos producto, resultan de la simplificación de los minterminos de la función (por lo que se invierten las variables que valen 0).

Así, el programa de simplificación viene a decir que para obtener D2 basta con sumar dos términos producto (de algunas de las 6 variables de la función): en el primer término producto sólo son relevantes las variables 1ª (DIR, con valor 0) y 3ª (Q1, con valor 1) y en el segundo DIR (=1) y DD (=1); el resto de variables no importan. Escribiendo la ecuación de suma de estos términos producto nos queda

$$D2 = \overline{DIR} \cdot Q1 + DIR \cdot DD$$

De la misma forma las funciones de excitación de D1 y D0 quedan

$$D1 = \overline{DIR} \cdot Q0 + DIR \cdot Q2; \quad D0 = \overline{DIR} \cdot DI + DIR \cdot Q1;$$

f.2) Emplear los mapas de Karnaugh para simplificar la función de excitación, pero no directamente sino por partes, troceando los casos de la tabla de verdad. Una vez obtenidas las ecuaciones para cada una de las partes se pueden reunir en una única ecuación de un manera muy sencilla empleando el álgebra de Boole.

Téngase en cuenta que este procedimiento no permite garantizar que el resultado sea el más sencillo posible, dado que se van a simplificar separadamente combinaciones que son algebraicamente adyacentes y que, por tanto, podrían dar lugar a agrupaciones de más términos. Lo único que se va a poder obtener es lo que se denomina “solución sub-óptima”: una ecuación más sencilla que la simple suma de los minterminos, aunque sin garantías de que sea la ecuación más sencilla posible.

En primer lugar, y dado que la tabla de verdad de la función de excitación está partida en dos, es más sencillo obtener las ecuaciones de D2, D1 y D0 para cada una de las partes (como si se tratara de funciones independientes) para luego reunir los resultados en una única ecuación.

En la primera parte de la tabla de verdad de la función de excitación (los casos en los que DIR = 0) se observa muy fácilmente que el valor de la entrada DD es irrelevante, de manera que podemos descartar directamente esta entrada. Por otra parte, como pretendemos obtener las ecuaciones resultantes para D2, D1 y D0 sólo cuando DIR = 0 (en esta primera parte de la tabla de verdad), dado que el valor de DIR es constante no participa en dichas ecuaciones. Nos quedan, por tanto, Q2, Q1, Q0 y DI para obtener

		Q2Q1						Q2Q1						Q2Q1			
Q0DI		00	01	11	10	Q0DI		00	01	11	10	Q0DI		00	01	11	10
00		0	1	1	0	00		0	0	0	0	00		0	0	0	0
01		0	1	1	0	01		0	0	0	0	01		1	1	1	1
11		0	1	1	0	11		1	1	1	1	11		1	1	1	1
10		0	1	1	0	10		1	1	1	1	10		0	0	0	0
		D2 = Q1						D1 = Q0						D0 = DI			

El mismo razonamiento se puede emplear sobre la segunda parte de la tabla para, por un lado, descartar la variable de entrada DI y, por otro, excluir la entrada DIR de las ecuaciones, obteniendo

		Q2Q1			
Q0DD		00	01	11	10
00		0	0	0	0
01		1	1	1	1
11		1	1	1	1
10		0	0	0	0

$D2 = DD$

		Q2Q1			
Q0DD		00	01	11	10
00		0	0	1	1
01		0	0	1	1
11		0	0	1	1
10		0	0	1	1

$D1 = Q2$

		Q2Q1			
Q0DD		00	01	11	10
00		0	1	1	0
01		0	1	1	0
11		0	1	1	0
10		0	1	1	0

$D0 = Q1$

Lo que resta ahora es reunir las dos ecuaciones para D2 en una sola, y lo mismo para D1 y D0.

El razonamiento es muy sencillo: la primera ecuación para D2 determina el valor de D2 sólo cuando DIR = 0, la segunda sólo cuando DIR = 1; por tanto, podemos simplemente sumar ambas ecuaciones sin más que añadir a cada una un factor cualificador que las haga ciertas solamente cuando el valor de la entrada DIR sea el correcto. Las primeras ecuaciones deben ir, por tanto, multiplicadas por not(DIR) y las segundas por DIR, quedando las ecuaciones finales

$$D2 = Q1 \cdot \overline{DIR} + DD \cdot DIR; \quad D1 = Q0 \cdot \overline{DIR} + Q2 \cdot DIR; \quad D0 = DI \cdot \overline{DIR} + Q1 \cdot DIR;$$

Casualmente el resultado de este proceder ha coincidido con la simplificación por Quine-McCluskey, pero esto no será cierto en general.

g) A partir de las ecuaciones de excitación y salida

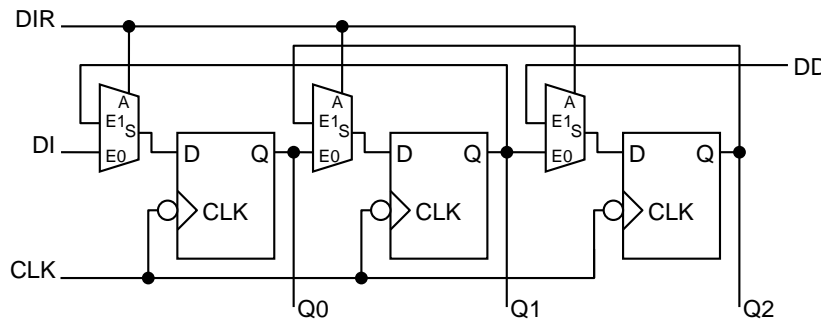
$$D2 = Q1 \cdot \overline{DIR} + DD \cdot DIR; \quad D1 = Q0 \cdot \overline{DIR} + Q2 \cdot DIR; \quad D0 = DI \cdot \overline{DIR} + Q1 \cdot DIR;$$

$$S2 = Q2; \quad S1 = Q1; \quad S0 = Q0;$$

se obtiene directamente el circuito, pues ya está determinada desde la codificación el número de variables de estado. Y un autómata de Moore no es más que una colección de variables de estado (biestables) alrededor de los cuales se implementa la lógica de las funciones combinacionales de excitación y salida.

Teniendo en cuenta que la ecuación $S = E0 \cdot \overline{SEL} + E1 \cdot SEL$ corresponde a la implementación de un MUX2a1 (S es la salida, SEL es la entrada de selección, E0 el dato de la entrada 0 y E1 el dato de la entrada 1), resulta evidente que las tres ecuaciones de excitación corresponden a sendos MUXes con la selección conectada a la entrada DIR del sistema.

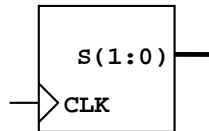
Así, el circuito queda



Que, evidentemente, coincide (con la salvedad de que éste es de tres bits) con el circuito mostrado en la solución del ejercicio 5.1.2

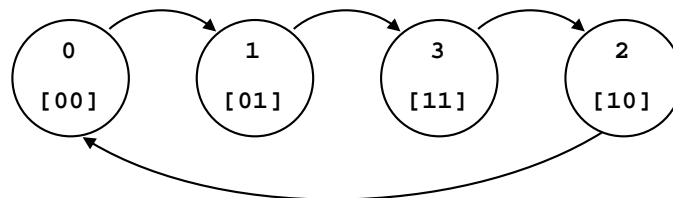
5.3.4.- Realice todas las etapas de diseño de un circuito contador que genere en su salida la secuencia 0 – 1 – 3 – 2 – 0 ...

a) La interfaz del sistema es



Nótese que como no se ha indicado nada al respecto del flanco activo del reloj, se ha elegido arbitrariamente que el flanco activo sea el de subida.

b) El diagrama de estados es muy sencillo, dado que sólo hay cuatro estados y ninguna entrada (los estados se recorren cíclicamente).



Se han empleado números distintivos como nombre de estado que coinciden con el valor de la salida, aprovechando que la salida no se repite en ningún estado.

b) Tabla de estados.

Estado actual	Estado siguiente	Salida	
		S1	S0
0	1	0	0
1	3	0	1
2	0	1	0
3	2	1	1

c, d) La codificación en realidad ya se ha hecho, sólo falta escribir la tabla de estados con los estados codificados utilizando los números de estado en binario con dos bits (como sólo tenemos cuatro estados es suficiente con dos variables de estado).

Estado actual		Estado siguiente		Salida	
Q1(t)	Q0(t)	Q1(t+1)	Q0(t+1)	S1	S0
0	0	0	1	0	0
0	1	1	1	0	1
1	0	0	0	1	0
1	1	1	0	1	1

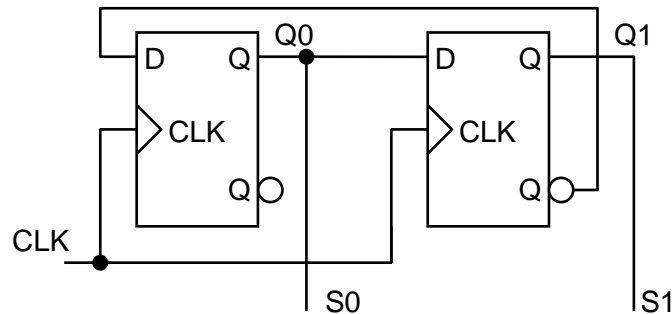
e, f) Tablas de verdad de las funciones de excitación y salida. Simplificación de las funciones de excitación y salida.

Q1	Q0	D1	D0
0	0	0	1
0	1	1	1
1	0	0	0
1	1	1	0

Q1	Q0	S1	S0
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

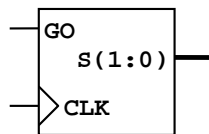
$D1 = Q0;$ $D0 = \overline{Q1}$ $S1 = Q1;$ $S0 = Q0$

g) Finalmente se reúnen los diferentes elementos del sistema.



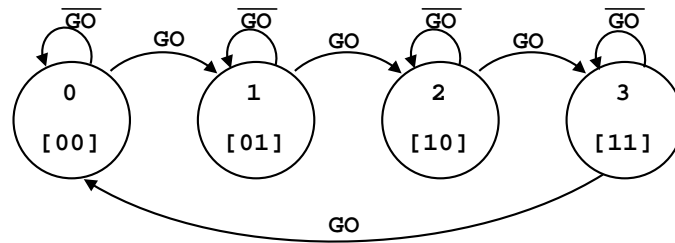
5.3.5.- Realice todas las etapas de diseño de un circuito contador que genere en su salida la secuencia 0 – 1 – 2 – 3 – 0 ... pero que solamente cambie el valor de la cuenta en aquellos flancos de reloj en los que una entrada (denominada GO, activa a nivel alto) esté activa; mientras GO esté desactivada la cuenta debe permanecer constante.

a) La interfaz del sistema es prácticamente la misma que la del ejercicio anterior, excepto que añadimos la entrada GO



Nótese cómo, de nuevo, al no haberse indicado nada al respecto del flanco activo del reloj, se ha elegido arbitrariamente que el flanco activo sea el de subida.

b) El diagrama de estados es relativamente sencillo: los estados se recorren cíclicamente siempre que $GO = 1$ y el estado no cambia cuando $GO = 0$. Por comodidad se ha realizado el diagrama de estados utilizando ecuaciones lógicas.



b) Tabla de estados.

Estado actual	Estado siguiente		Salida	
	GO = 0	GO = 1	S1	S0
0	0	1	0	0
1	1	2	0	1
2	2	3	1	0
3	3	0	1	1

c, d) La codificación en realidad ya se ha hecho, sólo falta escribir la tabla de estados con los estados codificados utilizando los números de estado en binario con dos bits (como sólo tenemos cuatro estados es suficiente con dos variables de estado).

Estado actual		Estado siguiente		Salida	
		Q1(t+1)	Q0(t+1)		
Q1(t)	Q0(t)	Entrada GO		S1	S0
		0	1		
0	0	0	0	0	0
0	1	0	1	0	1
1	0	1	0	1	0
1	1	1	1	1	1

e, f) Tablas de verdad de las funciones de excitación y salida. Simplificación de las funciones de excitación y salida

Q1	Q0	GO	D1	D0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

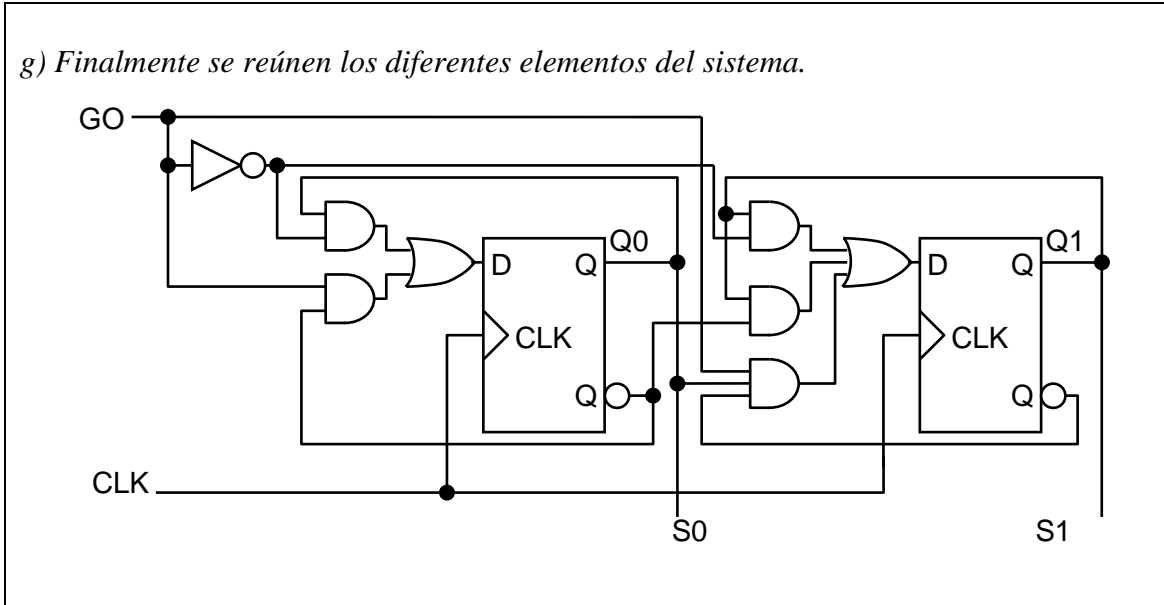
Q1	Q0	S1	S0
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

$S1 = Q1; \quad S0 = Q0$

$D1 = Q1 \cdot \overline{GO} + Q1 \cdot \overline{Q0} + \overline{Q1} \cdot Q0 \cdot GO$

$D0 = Q0 \cdot \overline{GO} + \overline{Q0} \cdot GO$

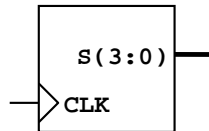
g) Finalmente se reúnen los diferentes elementos del sistema.



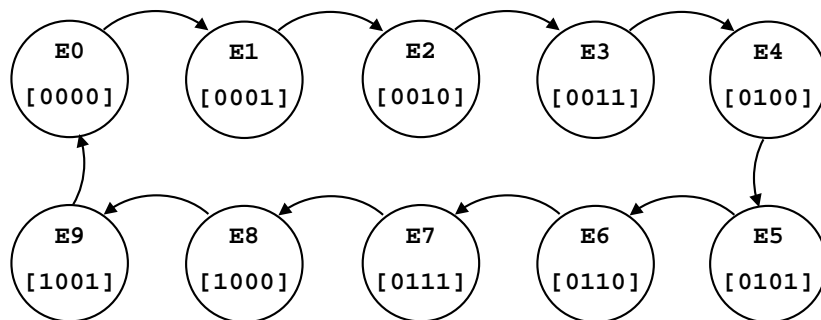
5.3.6.- Realice todas las etapas de diseño de un contador de décadas (de 0 a 9).

Un contador de décadas es un contador ascendente módulo 10 que cuenta desde 0 hasta 9. Como no se ha indicado nada al respecto del flanco activo del reloj, se ha elegido arbitrariamente que el flanco activo sea el de subida.

a) La interfaz del sistema es la misma que la de cualquier contador (sin entradas aparte del reloj). La salida debe disponer de tantos bits como sea necesario para representar todos los valores de salida; en este caso, serán necesarios 4 bits



b) El diagrama de estados es muy sencillo: los estados se recorren cíclicamente.



Aunque se han empleado nombres simbólicos para los estados también se podrían haber empleado los valores de salida para distinguir un estado de otro, aprovechando que en este sistema el valor de la salida nunca es igual para dos estados distintos.

b) Tabla de estados.

Estado actual	Estado siguiente	Salida			
		S3	S2	S1	S0
E0	E1	0	0	0	0
E1	E2	0	0	0	1
E2	E3	0	0	1	0
E3	E4	0	0	1	1
E4	E5	0	1	0	0
E5	E6	0	1	0	1
E6	E7	0	1	1	0
E7	E8	0	1	1	1
E8	E9	1	0	0	0
E9	E0	1	0	0	1

Para obtener el circuito que implementa este autómata se van a analizar dos variantes, en función de dos codificaciones diferentes. La primera variante (denominada variante A) emplea una codificación compacta con 4 variables de estado mientras que la segunda (denominada variante B) emplea una codificación no compacta (“one-hot”) con 10 variables de estado. Cada una de estas dos variantes se desarrollan hasta el final como ejercicio para la reflexión sobre el problema de optimización que se le presenta al diseñador a la hora de escoger una codificación.

Variante A: Codificación compacta, 4 variables de estado

c, d) Si se emplean cuatro variables de estado la codificación en realidad ya se ha hecho, sólo falta escribir la tabla de estados con los estados codificados utilizando los valores de la salida en binario con cuatro bits.

Estado actual				Estado siguiente				Salida			
Q3(t)	Q2(t)	Q1(t)	Q0(t)	Q3(t+1)	Q2(t+1)	Q1(t+1)	Q0(t+1)	S3	S2	S1	S0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	1
0	0	1	0	0	0	1	1	0	0	1	0
0	0	1	1	0	1	0	0	0	0	1	1
0	1	0	0	0	1	0	1	0	1	0	0
0	1	0	1	0	1	1	0	0	1	0	1
0	1	1	0	0	1	1	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1
1	0	0	0	1	0	0	1	1	0	0	0
1	0	0	1	0	0	0	0	1	0	0	1
1	0	1	x	x	x	x	x	x	x	x	x
1	1	x	x	x	x	x	x	x	x	x	x

e, f) Tablas de verdad de las funciones de excitación y salida. Simplificación de las funciones de excitación y salida

Función de excitación				Función de salida											
Q3	Q2	Q1	Q0	D3	D2	D1	D0	Q3	Q2	Q1	Q0	S3	S2	S1	S0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	0	1	0	1	0	0	1	0	0
0	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1
0	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0
0	1	1	1	1	1	0	0	0	0	1	1	0	1	1	1
1	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1
1	0	1	x	x	x	x	x	x	1	0	1	x	x	x	x
1	1	x	x	x	x	x	x	x	1	1	x	x	x	x	x

$$D3 = Q2 \cdot Q1 \cdot Q0 + Q3 \cdot \overline{Q0}$$

$$S3 = Q3; \quad S2 = Q2$$

$$D2 = Q2 \cdot \overline{Q1} + Q2 \cdot Q1 \cdot \overline{Q0} + \overline{Q2} \cdot Q1 \cdot Q0$$

$$S1 = Q1; \quad S0 = Q0$$

$$D1 = Q1 \cdot \overline{Q0} + \overline{Q3} \cdot \overline{Q1} \cdot Q0$$

$$D0 = \overline{Q0}$$

Como se puede observar, aunque el circuito resultante de esta variante es realizable las ecuaciones correspondientes a la función de excitación tienen un coste notable.

g) El circuito final reuniendo los diferentes elementos siguiendo la estructura general de un autómata de Moore se deja como ejercicio para el lector.

Variante B: Codificación “one-hot”, 10 variables de estado

Las ecuaciones de excitación al emplear una codificación compacta (véase la variante A anterior) resultan bastante complejas. Es posible reducir dicha complejidad empleando otro tipo de codificación; en este caso se desarrolla el sistema con codificación “one-hot” para comparar ambas soluciones y decidir con cuál se obtiene un circuito más sencillo.

c', d') Al emplear codificación “one-hot” se necesitan tantas variables de estado como estados. Y al existir tantas variables de estado existen multitud de casos imposibles en las combinaciones de valores del estado. De las 1024 combinaciones posibles de 10 variables sólo se emplean 10 para codificar los 10 estados del sistema, todos los demás casos son códigos sin utilizar.

Estado	Codificación Q(9:0)	Estado actual Q(9:0)(t)	Estado siguiente Q(9:0)(t+1)	Salida S(3:0)
E0	0000000001	0000000001	0000000010	0000
E1	0000000010	0000000010	0000000100	0001
E2	0000000100	0000000100	0000001000	0010
E3	0000001000	0000001000	0000010000	0011
E4	0000010000	0000010000	0000100000	0100
E5	0000100000	0000100000	0001000000	0101
E6	0001000000	0001000000	0010000000	0110
E7	0010000000	0010000000	0100000000	0111
E8	0100000000	0100000000	1000000000	1000
E9	1000000000	1000000000	0000000001	1001
		0000000000	xxxxxxx	xxx
		Otro caso	xxxxxxx	xxx

Evidentemente no se han enumerado exhaustivamente todos los casos, sólo los más relevantes. Todos los casos no enumerados son imposibles, por lo que los valores de estado siguiente y salidas corresponden a secuencias de X's.

e') Tablas de verdad de las funciones de excitación y salida.

Función de excitación		Función de salida	
Q(9:0)	D(9:0)	Q(9:0)	S(3:0)
0000000001	0000000010	0000000001	0000
0000000010	0000000100	0000000010	0001
0000000100	0000001000	0000000100	0010
0000001000	0000010000	0000001000	0011
0000010000	0000100000	0000010000	0100
0000100000	0001000000	0000100000	0101
0001000000	0010000000	0001000000	0110
0010000000	0100000000	0010000000	0111
0100000000	1000000000	0100000000	1000
1000000000	0000000001	1000000000	1001
0000000000	xxxxxxx	0000000000	xxx
Otro caso	xxxxxxx	Otro caso	xxx

f') Obtención de las ecuaciones de excitación y salida.

Función de excitación:

Dado el número de entradas de las funciones de excitación y salida, intentar una simplificación empleando mapas de Karnaugh es inviable. Se podría recurrir a un método de simplificación de más variables (véase, por ejemplo, la solución del ejercicio 5.3.3), pero al emplear codificación "one-hot" el diseñador dispone de otras alternativas, que se desarrolla a continuación.

f.1') Para el cálculo de D9, puesto que es una función que contiene un único minitérmino, una primera ecuación (sin simplificar) es el minitérmino correspondiente (recuérdese que una función siempre se puede construir mediante la suma de sus minitérminos).

Por tanto, $D9 = \overline{Q9} \cdot Q8 \cdot \overline{Q7} \cdot \overline{Q6} \cdot \overline{Q5} \cdot \overline{Q4} \cdot \overline{Q3} \cdot \overline{Q2} \cdot \overline{Q1} \cdot \overline{Q0}$.

Pero ocurre que, dentro de los códigos de estado válido, si $Q8$ está a valor 1 todas las demás variables de estado están a valor 0. Por lo tanto la ecuación anterior es equivalente a $D9 = Q8$ (ciñéndonos a las combinaciones de valores válidos del estado).

Dicho de otra manera, las dos expresiones anteriores, $Q8$ y $\overline{Q9} \cdot \overline{Q8} \cdot \overline{Q7} \cdot \overline{Q6} \cdot \overline{Q5} \cdot \overline{Q4} \cdot \overline{Q3} \cdot \overline{Q2} \cdot \overline{Q1} \cdot \overline{Q0}$ sólo dan valores distintos para combinaciones binarias no válidas de las variables de estado.

De la misma manera, aunque la ecuación inicial de excitación para $D8$ sea $D8 = \overline{Q9} \cdot \overline{Q8} \cdot Q7 \cdot \overline{Q6} \cdot \overline{Q5} \cdot \overline{Q4} \cdot \overline{Q3} \cdot \overline{Q2} \cdot \overline{Q1} \cdot \overline{Q0}$, ésta se simplifica a $D8 = Q7$ porque si $Q7 = 1$ está garantizado que las demás Q 's valen 0.

De forma análoga se obtiene la excitación para todos los biestables, por lo que

$$\begin{array}{lllll} D9 = Q8; & D8 = Q7; & D7 = Q6; & D6 = Q5; & D5 = Q4; \\ D4 = Q3; & D3 = Q2; & D2 = Q1; & D1 = Q0; & D0 = Q9; \end{array}$$

f.2') Otra manera de obtener las ecuaciones de excitación es relacionar las transiciones del diagrama de estados con los valores binarios de estado siguiente; eso es sencillo cuando se utiliza codificación "one-hot", porque hay una correspondencia total entre un estado del autómata y la variable de estado que representa a dicho estado.

Por ejemplo, si $Q8 = 1$ significa que el autómata está en el estado asociado ($E8$) pues es el estado con el código "010000000" (el único código en el que $Q8$ toma el valor 1).

De la misma manera, la función de excitación debe conseguir que si el estado es actualmente $E8$, el estado siguiente sea el estado $E9$ (dado que en el diagrama de estado desde el estado $E8$ sale una única transición y ésta acaba en $E9$). Como al utilizar biestables D el estado siguiente coincide con la excitación, que el estado siguiente sea $E9$ (código "10000000", el único con $Q9 = 1$) es lo mismo que decir que la excitación (para ese caso) sea "10000000", único caso en el que $D9 = 1$.

En resumen, al emplear codificación "one-hot" la ecuación lógica que haga que $D_i = 1$ será la transición hacia el estado siguiente E_i ; y que en dicha ecuación aparezca el término Q_j viene a decir que el estado actual es E_j .

Aplicando estos razonamientos (recalcando que sólo son válidos para codificación "one-hot"), se pueden obtener las ecuaciones de excitación de forma sencilla, como se explica a continuación.

La excitación para $D9$ es una ecuación que debe dar 1 cuando el autómata tenga que transitar hacia el estado $E9$. Aplicado al caso concreto que nos ocupa, la única transición hacia $E9$ proviene del estado $E8$ (estado actual que en la ecuación aparece como $Q8$), y es una transición que en el diagrama de estados no tiene ecuación asociada (es decir, no depende de ninguna entrada) por lo que $D9 = Q8$.

De forma análoga se obtiene la excitación para todos los biestables, por lo que

$$\begin{array}{lllll}
 D9 = Q8; & D8 = Q7; & D7 = Q6; & D6 = Q5; & D5 = Q4; \\
 D4 = Q3; & D3 = Q2; & D2 = Q1; & D1 = Q0; & D0 = Q9;
 \end{array}$$

Función de salida:

Las ecuaciones de la función de salida tampoco se pueden simplificar por Karnaugh. Pero podemos aplicar razonamientos similares a los empleados para la excitación a fin de obtener las ecuaciones correspondientes.

Por ejemplo, la salida S2 debe activarse (a nivel alto) cuando el autómata esté en los estados E8 o E9. Teniendo en cuenta que las variables Q8 y Q9 valen 1 solamente cuando el estado del autómata es E8 y E9 respectivamente, se puede deducir fácilmente que $S2 = Q9 + Q8$.

De forma análoga se obtienen todas las ecuaciones de salida, por lo que

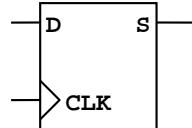
$$\begin{array}{l}
 S3 = Q9 + Q8; \\
 S2 = Q7 + Q6 + Q5 + Q4; \\
 S1 = Q7 + Q6 + Q3 + Q2; \\
 S0 = Q9 + Q7 + Q5 + Q3 + Q1;
 \end{array}$$

Obsérvese cómo, al emplear en esta variante una codificación distinta, las ecuaciones resultantes para las funciones de excitación y salida son radicalmente diferentes, y en este caso bastante más sencillas. Como contrapartida, se han de emplear 10 biestables para las 10 variables de estado, cuando en la variante anterior sólo eran necesarios 4.

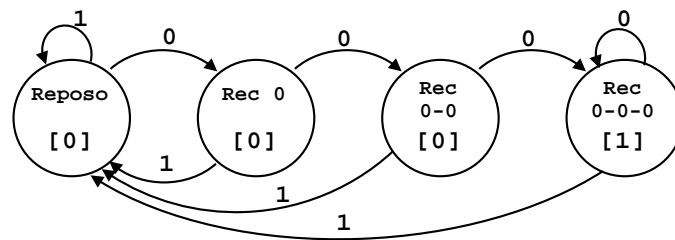
g') El circuito final reuniendo los diferentes elementos siguiendo la estructura general de un autómata de Moore se deja como ejercicio para el lector.

5.3.7.- Realice todas las etapas de diseño de un circuito capaz de detectar el patrón 000. La entrada de datos ocurre a razón de un bit por ciclo de reloj desde una entrada denominada D. El reloj es activo por flanco de subida. La salida del sistema (denominada S) debe ser 1 cuando en los tres ciclos de reloj previos la secuencia de valores de la entrada D haya sido 0 – 0 – 0.

a) La interfaz del sistema: entradas D y reloj, salida S



b) Diagrama de estados: Además de un estado de reposo, que modela la situación de no haber recibido ningún dígito del patrón buscado, es necesario incluir un estado para cada etapa de la detección (“recibido el primer 0”, “recibida la secuencia 0 – 0” y “recibida la secuencia 0 – 0 – 0”). Se han empleado nombres simbólicos para los estados y vectores de valores (con la única entrada D) en las transiciones.



Evidentemente, en cuanto se detecta un 1 a mitad de secuencia el sistema vuelve al estado de reposo. Nótese que, el autómata se mantiene en el estado “Rec 0-0-0” mientras la entrada D sea cero, dado que se ha especificado en la salida debe activarse siempre que la secuencia de la entrada en los tres últimos ciclos de reloj haya sido 0, 0 y 0.

b) Tabla de estados

Estado actual	Estado siguiente		Salida S
	D = 0	D = 1	
Reposo	Rec 0	Reposo	0
Rec 0	Rec 0-0	Reposo	0
Rec 0-0	Rec 0-0-0	Reposo	0
Rec 0-0-0	Rec 0-0-0	Reposo	1

c, d) Una codificación elemental es emplear dos variables de estado para representar los cuatro estados y codificación binaria (Reposo: 00, Rec 0: 01, etc.). Con esta codificación sólo es necesario sustituir los nombres simbólicos de los estados por sus códigos binarios para escribir la tabla de estados con los estados codificados.

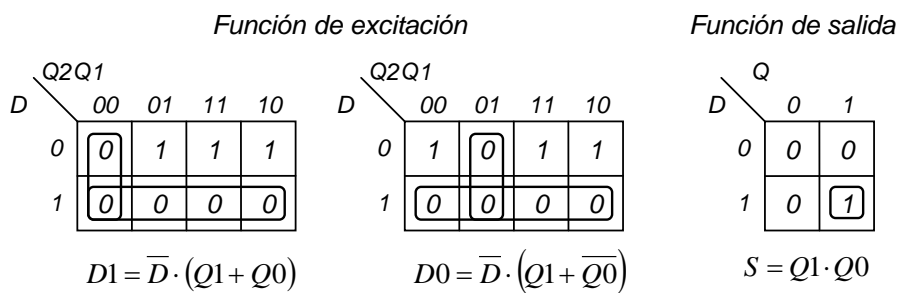
Estado	Codif.	Estado siguiente		Entrada D		Salida
		Q1(t+1)	Q0(t+1)	0	1	
Reposo	00	0	0	0	1	0
Rec 0	01	0	1	1	0	0
Rec 0-0	10	1	0	1	1	0
Rec 0-0-0	11	1	1	1	1	1

e) Tablas de verdad de las funciones de excitación y salida.

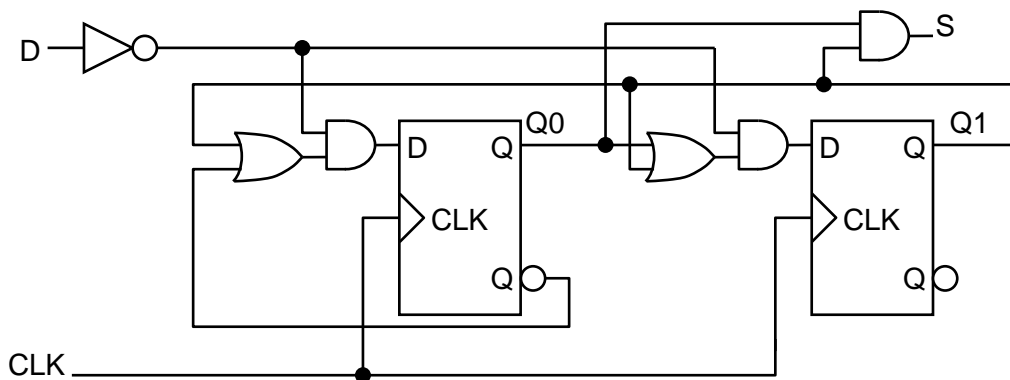
Q1	Q0	D	D1	D0
0	0	0	0	1
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Q1	Q0	S
0	0	0
0	1	0
1	0	0
1	1	1

f) Simplificación de las funciones de excitación y salida. En este caso se ha optado por simplificar la función de excitación mediante Karnaugh por ceros, por dar un resultado más simple.

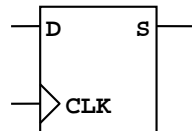


g) Finalmente se reúnen los diferentes elementos del sistema.

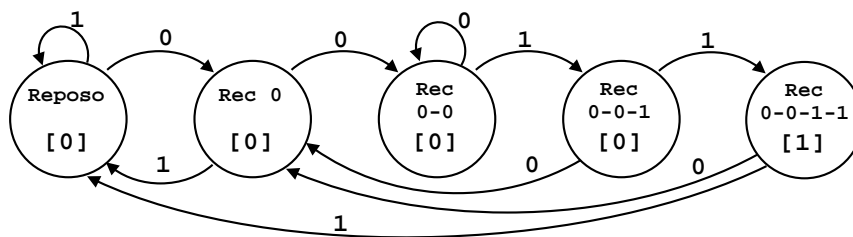


5.3.8.- Realice todas las etapas de diseño de un circuito capaz de detectar el patrón 0011. La entrada de datos ocurre a razón de un bit por ciclo de reloj desde una entrada denominada D. El reloj es activo por flanco de subida. La salida del sistema (denominada S) debe ser 1 cuando en los cuatro ciclos de reloj previos la secuencia de valores de la entrada D haya sido 0 – 0 – 1 – 1.

a) La interfaz del sistema es idéntica a la del ejercicio anterior: entradas D y reloj, salida S



b) Diagrama de estados: Se puede emplear también el ejercicio anterior como modelo, en el que se incluye un estado de reposo y un estado por cada dígito del patrón que se desea detectar.



Determinar la transición al estado siguiente cuando la entrada coincide con el patrón que se busca es inmediato. La única dificultad del diagrama de estados de este autómata es determinar a qué estado debe transitar cuando se detecta un dígito fuera de la secuencia buscada. Por ejemplo, cuando desde el estado que modela haber recibido el primer dígito del patrón (estado “Rec 0”) se recibe un 1, éste rompe la secuencia 0011 y es necesario volver a empezar.

Sin embargo, cuando el autómata se encuentra en el estado “Rec 0-0” que modela haber recibido (en los dos últimos ciclos de reloj) la secuencia 00, un nuevo valor 0 en la entrada rompe el patrón buscado, pero no tanto como para volver a empezar desde el estado de reposo, pues si los próximos valores de la entrada son 1 habrá de activarse la salida ya que se ha encontrado el patrón buscado.

Emplear el tipo de razonamiento para todos y cada uno de los estados en los que se ha recibido parte del patrón es el que permite determinar el estado siguiente cuando la entrada no cumple con lo que se busca.

b) Tabla de estados

Estado actual	Estado siguiente		Salida
	D = 0	D = 1	S
Reposo	Rec 0	Reposo	0
Rec 0	Rec 0-0	Reposo	0
Rec 0-0	Rec 0-0	Rec 0-0-1	0
Rec 0-0-1	Rec 0	Rec 0-0-1-1	0
Rec 0-0-1-1	Rec 0	Reposo	1

c, d) Una codificación sencilla es emplear tres variables de estado para representar los cinco estados y codificación binaria ordenada y ascendente (Reposo: 000, Rec 0: 001, etc.). Con esta codificación quedan tres códigos sin utilizar (101, 110 y 111) que se convertirán en casos imposibles en las funciones de transición y salida. Sustituyendo los nombres simbólicos de los estados por sus códigos binarios se escribe la tabla de estados con los estados codificados.

Estado	Codif.	Estado siguiente			Salida
		Q(2:0)(t+1)			
Est. actual		Entrada D		S	
Q(2:0)(t)		0	1		
Reposo	000	0 0 0	0 0 1	0 0 0	0
Rec 0	001	0 0 1	0 1 0	0 0 0	0
Rec 0-0	010	0 1 0	0 1 0	0 1 1	0
Rec 0-0-1	011	0 1 1	0 0 1	1 0 0	0
Rec 0-0-1-1	100	1 0 0	0 0 1	0 0 0	1

e) Tablas de verdad de las funciones de excitación y salida.

Función de excitación				Función de salida			
Q2	Q1	Q0	D	D2	D1	D0	S
0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0
0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	1	0
0	1	1	0	0	0	1	0
0	1	1	1	1	0	0	0
1	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0
1	0	1	x	x	x	x	x
1	1	x	x	x	x	x	x

f) Simplificación de las funciones de excitación y salida.

Función de excitación

		Q2Q1			
		00	01	11	10
Q0D	00	0	0	x	0
	01	0	0	x	0
	11	0	1	x	x
	10	0	0	x	x

$$D2 = Q1 \cdot Q0 \cdot D$$

		Q2Q1			
		00	01	11	10
Q0D	00	0	1	x	0
	01	0	1	x	0
	11	0	0	x	x
	10	1	0	x	x

$$D1 = Q1 \cdot \overline{Q0} + \overline{Q1} \cdot Q0 \cdot \overline{D}$$

Función de salida

		Q2Q1			
		00	01	11	10
Q0	0	0	0	x	1
	1	0	0	x	x

$$S = Q2$$

		Q2Q1			
		00	01	11	10
Q0D	00	1	0	x	1
	01	0	1	x	0
	11	0	0	x	x
	10	0	1	x	x

$$D0 = \overline{Q1} \cdot \overline{Q0} \cdot \overline{D} + Q1 \cdot \overline{Q0} \cdot D + Q1 \cdot Q0 \cdot \overline{D}$$

g) El circuito final reuniendo los diferentes elementos siguiendo la estructura general de un autómata de Moore se deja como ejercicio para el lector.