

Capítulo 7

CONTADORES Y REGISTROS

7.1. CONTADORES

Un contador es un circuito secuencial cuya función es seguir una cuenta o conjunto predeterminado de estados como consecuencia de la aplicación de un tren de pulsos (reloj) en una de sus entradas. Los contadores son circuitos construidos a base de flip-flops, sincronizados a flancos, y de puertas lógicas para realizar la conexión entre los flip-flops. Las puertas lógicas en un contador se conectan de forma que fuercen a los flip-flops a seguir la secuencia prescrita de estados.

Dado que cada flip-flop es capaz de almacenar 2 estados, un sistema de n flip-flops podrá codificar 2^n estados (números de cuenta) diferentes. El paso del estado o número de cuenta actual al estado siguiente se realiza en sincronismo con la señal de reloj, de tal forma que el contador va avanzando por la secuencia de estados hasta que termina de completar un ciclo, es decir, una secuencia de conteo particular. Una vez terminado el ciclo, vuelve al estado de partida y comienza un nuevo ciclo. Como cada estado sólo posee un estado siguiente, es fácil deducir que todos los estados que recorre un contador en un ciclo o secuencia de conteo son diferentes. Si el número de estados diferentes que recorre es k , se habla de un contador módulo k . El número máximo de estados posibles es 2^n , por lo que $k \leq 2^n$. Un contador se denomina binario si $k=2^n$.

7.2. CONTADORES BINARIOS

Un contador, como todo circuito secuencial, se puede implementar como un autómata. Aunque un autómata muy especial, puesto que no posee entradas (cada estado sólo posee un único estado siguiente al que accede cuando existe una variación o flanco en la señal de reloj), y no posee salidas (la salida de un contador es su propio estado interno, es decir, el valor almacenado en sus biestables).

El método de diseño es el mismo que el usado en el tema anterior, e incluso algo más sencillo puesto que ya no es necesario minimizar ni asignar estados. Veamos un ejemplo y diseñemos un contador binario ascendente módulo 8, es decir, un contador que siga la secuencia $\{\dots, 0, 1, 2, 3, 4, 5, 6, 7, \dots\}$. La tabla de transiciones de estado del contador será:

Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
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

Para conocer las funciones J y K de cada biestable hay que utilizar la tabla de transiciones de estado de un flip-flop JK. Llegamos a la siguiente tabla:

Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	-	0	-	1	-
0	0	1	0	1	0	0	-	1	-	-	1
0	1	1	1	0	0	1	-	-	1	-	1
0	1	0	0	1	1	0	-	-	0	1	-
1	1	0	1	1	1	-	0	-	0	1	-
1	1	1	0	0	0	-	1	-	1	-	1
1	0	1	1	1	0	-	0	1	-	-	1
1	0	0	1	0	1	-	0	0	-	1	-

Sólo nos resta encontrar las expresiones de las entradas J y K de cada biestable en función del estado actual. Para ello debemos minimizar las funciones de la tabla anterior utilizando mapas de Karnaugh.

J_2	J_1	J_0
$Q_2^n \backslash Q_1^n Q_0^n$	$Q_2^n \backslash Q_1^n Q_0^n$	$Q_2^n \backslash Q_1^n Q_0^n$
00 01 11 10	00 01 11 10	00 01 11 10
0 0 0 1 0	0 0 1 - -	0 1 - - 1
1 - - - -	1 0 1 - -	1 1 - - 1
K_2	K_1	K_0
$Q_2^n \backslash Q_1^n Q_0^n$	$Q_2^n \backslash Q_1^n Q_0^n$	$Q_2^n \backslash Q_1^n Q_0^n$
00 01 11 10	00 01 11 10	00 01 11 10
0 - - - -	0 - - 1 0	0 - 1 1 -
1 0 0 1 0	1 - - 1 0	1 - 1 1 -

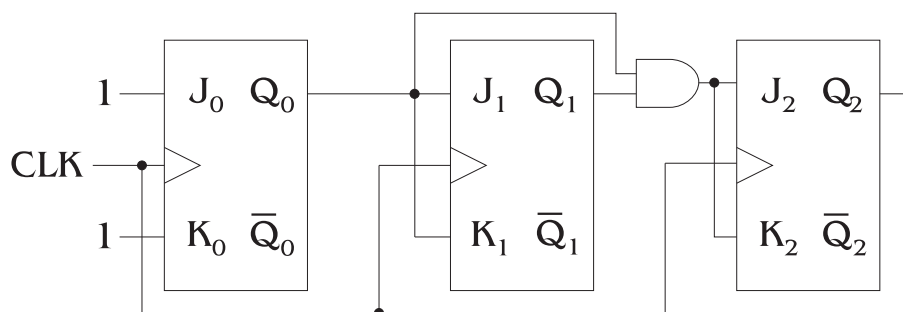


Figura 7.1: Contador binario.

Estas ecuaciones se pueden generalizar para secuencias de conteo mayores y para un número mayor de flip-flops de la siguiente forma:

$$\begin{aligned}
 T_0 &= J_0 = K_0 = 1 \\
 T_1 &= J_1 = K_1 = Q_0^n \\
 T_2 &= J_2 = K_2 = Q_1^n Q_0^n \\
 T_3 &= J_3 = K_3 = Q_2^n Q_1^n Q_0^n \\
 T_4 &= J_4 = K_4 = Q_3^n Q_2^n Q_1^n Q_0^n \\
 &\dots
 \end{aligned}$$

El contador se puede ver en la figura 7.1. El principal problema de este diseño es que, para un número elevado de biestables, se necesitan puertas AND cada vez con un número mayor de entradas y pueden aparecer problemas de fan-in y fan-out (recordemos los problemas del mismo tipo que aparecían en el sumador de acarreo adelantado).

Se puede plantear el problema inverso de tomar como partida el esquema de un contador ya diseñado y obtener su secuencia de conteo. El procedimiento a seguir es el mismo que el utilizado en el análisis de autómatas.

Leyendo las conexiones del contador obtenemos las expresiones de las Js y las Ks de cada biestable en función del estado actual del contador (Q^n). Partiendo de la cuenta 000 (u otros valores concretos de las Q^n) y usando las ecuaciones anteriores obtenemos los valores concretos de las J_i y K_i . Dados estos valores se determina a partir de la Tabla de Verdad del JK los nuevos valores de los Q^{n+1} , que son los correspondientes a la cuenta siguiente. Usando estos nuevos valores de cuenta (valores de Q^n) se recalculan los nuevos valores de J_i y K_i , y así sucesivamente hasta completar la secuencia de conteo, es decir, hasta que se repite un estado.

Podemos ver el funcionamiento de este proceso aplicándolo al contador que hemos diseñado anteriormente (ver figura 7.1). Obtenemos las funciones de onda que se pueden ver en la figura 7.2.

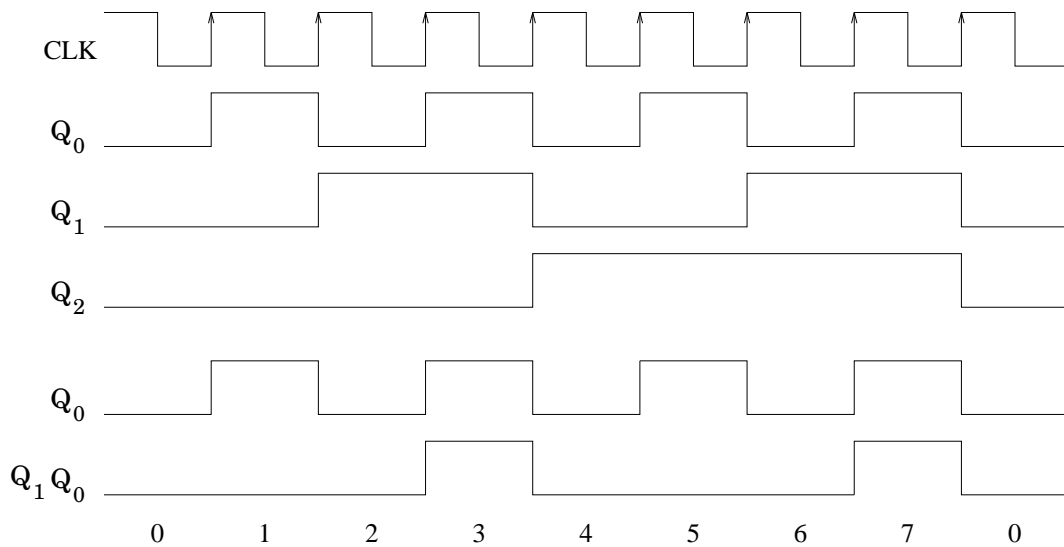


Figura 7.2: Salidas del contador binario.

7.3. CONTADORES NO BINARIOS

Hasta ahora hemos construido contadores binarios, esto, es con n flip-flops la secuencia que se ha implementado es $\{\dots, 0, 1, 2, \dots, 2^n-1, \dots\}$. Sin embargo, cabe la posibilidad de generar secuencias de conteo que no sigan el orden natural o que el número de cuentas sea menor que 2^n . En general, un contador que realice k cuentas distintas recibe el nombre de contador módulo k .

El método de diseño que hemos expuesto en el apartado anterior para contadores binarios es un método general que se puede aplicar para cualquier tipo de cuenta y se puede generalizar para cualquier tipo de flip-flop (JK, RS, T y D).

Como ejemplo de esta afirmación, construyamos a partir de flip-flops tipo T un contador que siga la secuencia $\{\dots, 0, 7, 5, 3, 4, 2, \dots\}$. El primer paso es obtener la Tabla de Transiciones de estado del flip-flop T:

$$\begin{array}{c|c} T & Q^{n+1} \\ \hline 0 & Q^n \\ 1 & \bar{Q}^n \end{array} \quad \begin{array}{l} Q^{n+1} = T\bar{Q}^n + \bar{T}Q^n \\ 0 = T1 + \bar{T}0 \\ 1 = T1 + \bar{T}0 \\ 0 = T0 + \bar{T}1 \\ 1 = T0 + \bar{T}1 \end{array} \quad \begin{array}{cc|c} Q^{n+1} & Q^n & T \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}$$

El segundo paso, y último, es la obtención de las expresiones de las T_i a partir de la cuenta actual y de la cuenta siguiente. En las cuentas no usadas ponemos indiferencias. El resultado final será:

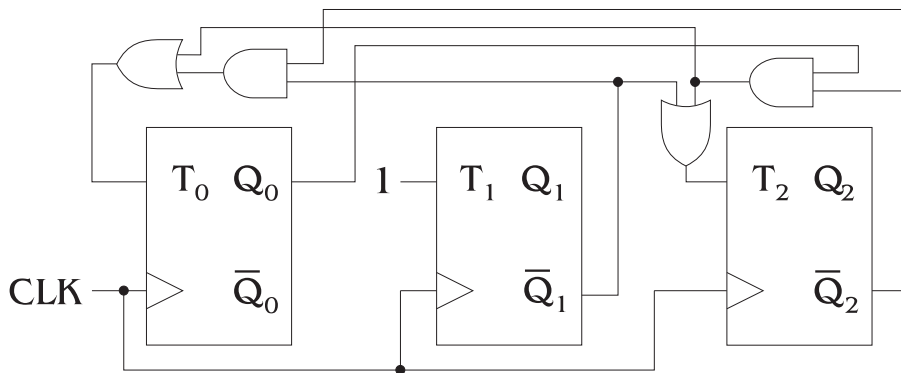


Figura 7.3: Contador de secuencia 0, 7, 5, 3, 4, 2.

Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	T_2	T_1	T_0
0	0	0	1	1	1	1	1	1
0	0	1	-	-	-	-	-	-
0	1	1	1	0	0	1	1	1
0	1	0	0	0	0	0	1	0
1	1	0	-	-	-	-	-	-
1	1	1	1	0	1	0	1	0
1	0	1	0	1	1	1	1	0
1	0	0	0	1	0	1	1	0

Para obtener las expresiones de las T_i en función de las Q^n minimizamos mediante mapa de Karnaugh.

		T_2				T_1				T_0							
Q_2^n	$Q_1^n Q_0^n$	00	01	11	10	Q_2^n	$Q_1^n Q_0^n$	00	01	11	10	Q_2^n	$Q_1^n Q_0^n$	00	01	11	10
0		1	-	1	0	0		1	-	1	1	0		1	-	1	0
1		1	1	0	-	1		1	1	1	-	1		0	0	0	-

Es decir, $T_2 = \overline{Q_1^n} + \overline{Q_2^n}Q_0^n$, $T_1 = 1$ y $T_0 = \overline{Q_2^n}\overline{Q_1^n} + \overline{Q_2^n}Q_0^n$. El circuito resultante se puede ver en la figura 7.3, y sus salidas están en la figura 7.4.

Una vez construido el contador podemos comprobar que la secuencia de conteo que realmente sigue es la que hemos diseñado. Cabe preguntarse que pasará si el contador cae, debido a un mal funcionamiento o comportamiento anómalo de sus componentes físicos, en alguna de las cuentas no utilizadas, en este caso 1 y 6.

1. Si el contador cae en $Q_2^n Q_1^n Q_0^n = 001$ (cuenta 1), sustituyendo en las ecuaciones, tendremos que $T_2 = 1$, $T_1 = 1$ y $T_0 = 1$ con lo cual el contador pasará al estado $Q_2^{n+1} Q_1^{n+1} Q_0^{n+1} = 110$ (cuenta 6).
2. Si el contador cae en $Q_2^n Q_1^n Q_0^n = 110$ (cuenta 6), sustituyendo en las ecuaciones obtenemos $T_2 = 0$, $T_1 = 1$ y $T_0 = 0$ y el siguiente estado será $Q_2^{n+1} Q_1^{n+1} Q_0^{n+1} = 100$

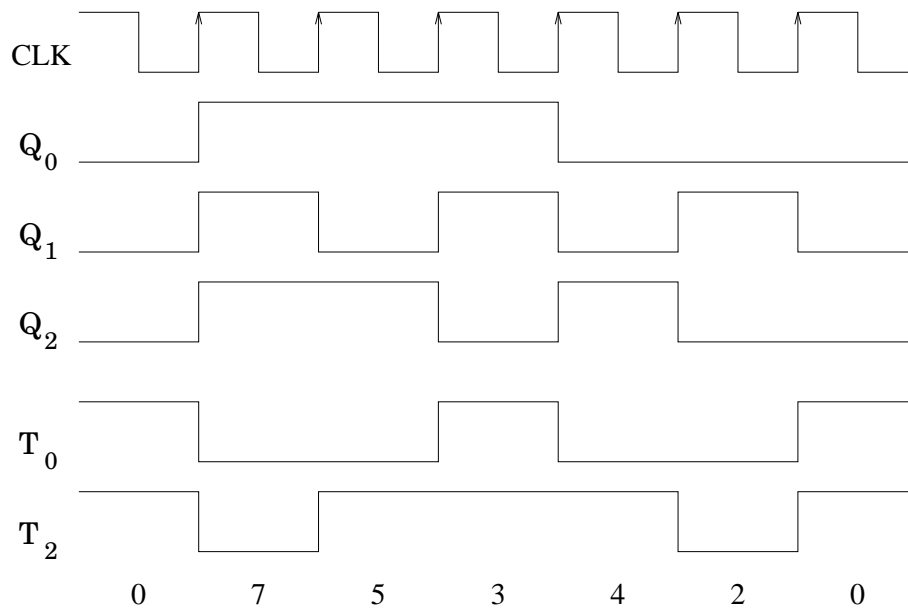


Figura 7.4: Salidas del contador de secuencia 0, 7, 5, 3, 4, 2.

(cuenta 4), que ya forma parte de la secuencia, con lo que el contador seguirá contando normalmente.

Una consecuencia de poner indiferencias en los estados (cuentas) no utilizados es que, si por casualidad el contador cae en uno de esos estados, desconocemos, a priori, el estado siguiente al que pasará. Cabe la posibilidad de que el contador produzca un ciclo por el que nunca llegue a un estado utilizado o válido. Esto puede ocurrir si el contador se queda en los ciclos $\{\dots, 1, 6, \dots\}$, $\{\dots, 1, \dots\}$, o $\{\dots, 6, \dots\}$. Un contador cuyos estados no utilizados tienen esta propiedad (de generar ciclos de estados o cuentas no válidas) se dice que puede bloquearse. En cualquier caso, en todo diseño de un contador es necesario comprobar el estado siguiente de cada estado no utilizado para comprobar que no se bloquea.

Una alternativa es la no utilización de indiferencias en el proceso de diseño del contador, colocando en las cuentas no utilizadas estados siguientes concretos (no necesariamente el mismo). Ello complicará el diseño, ya que eliminará indiferencias, pero garantiza un correcto funcionamiento del contador, aún en el caso de que, por cualquier causa, llegue a alcanzar un estado o número de cuenta no válido.

7.4. CONTADORES CON CARGA PARALELA

Los contadores utilizados en sistemas digitales a menudo requieren la capacidad para transferir un número binario inicial antes de la operación de conteo. Esta transferencia de un número de cuenta determinado al contador recibe el nombre de carga en paralelo.

En este apartado vamos a ver como realizar una carga en paralelo síncrona (en síncronismo con la señal de reloj). En el caso de una carga síncrona las únicas entradas sobre

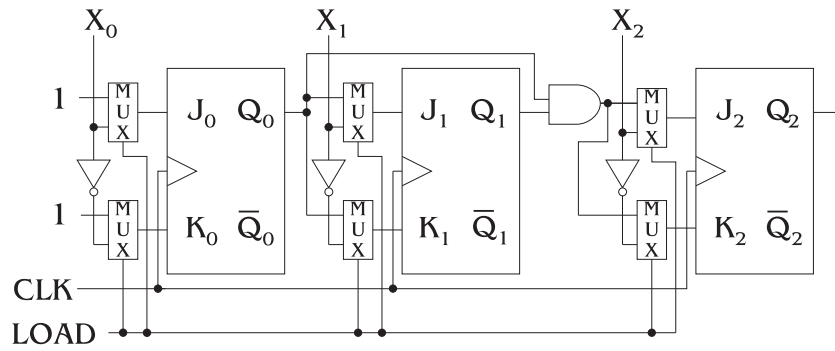


Figura 7.5: Contador binario con carga en paralelo.

las que se pueden actuar son J y K, ya que son las únicas que son síncronas. En este apartado sólo consideraremos flip-flops JK, pero el tratamiento es similar para los otros tipos de biestables.

Un contador con carga en paralelo, además de la señal de reloj, necesita una señal adicional que le indique cuando debe contar y cuando debe cargar el estado o principio de una nueva cuenta o ciclo. A esta señal se le suele denominar LOAD y su función es seleccionar, mediante MUXes 2 a 1, las entradas adecuadas de los biestables (J y K) en función de la tabla siguiente:

LOAD	Valores de J y K para
0	el contador cuente
1	realizar una carga en paralelo

Cuando $LOAD=0$ se seleccionan los valores de J y K obtenidos en el proceso de diseño del contador (el contador cuenta). Cuando $LOAD=1$ necesitamos poner en J_i y en K_i los valores adecuados para que el flip-flop i -ésimo cargue el valor X_i (el bit i -ésimo del estado que queremos cargar) en el siguiente pulso de reloj. Es decir, si $X_i = 0$ significa que en el biestable i -ésimo debemos poner un cero, y cuando $X_i = 1$ debemos poner un uno, independientemente del valor almacenado. Para ello tendremos que poner las entradas que se muestran en la siguiente tabla.

X_i	J_i	K_i
0	0	1
1	1	0

O lo que es lo mismo, $J_i = X_i$ y $K_i = \overline{X_i}$. Es decir, en la entrada J del primer flip-flop el MUX 2 a 1 escoge entre 1 y X_0 , mientras que en la entrada K otro MUX escoge entre 1 y $\overline{X_0}$. Algo similar ocurre en el resto de los biestables. El circuito completo se muestra en la figura 7.5.

7.5. CONTADORES ASCENDENTES/DESCENDENTES

Para diseñar un contador reversible ascendente/descendente se comienza por diseñar dos contadores separados. El primero (ascendente) se diseña para que pase por la secuencia de estados que se especifique; el segundo (descendente) se diseña para que pase por la misma secuencia, pero en sentido inverso. Después se le añaden los multiplexos encargados de seleccionar los dos sentidos de cuenta. Se puede comprobar que el sentido de conteo puede conmutarse siempre que la señal de reloj no presente un flanco activo, de tal forma que no se altere la cuenta almacenada.

Supongamos, como ejemplo, que queremos diseñar un contador reversible módulo 3, con una secuencia ascendente $\{\dots, 0, 1, 2, \dots\}$ (y descendente $\{\dots, 2, 1, 0, \dots\}$). Diseñamos primero el contador ascendente:

Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	J_1	K_1	J_0	K_0
0	0	0	1	0	-	1	-
0	1	1	0	1	-	-	1
1	1	-	-	-	-	-	-
1	0	0	0	-	1	0	-

Es decir, $J_0 = \overline{Q_1}^n$, $K_0 = K_1 = 1$ y $J_1 = Q_0^n$

A continuación diseñamos el contador descendente:

Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	J_1	K_1	J_0	K_0
0	0	1	0	1	-	0	-
0	1	0	0	0	-	-	1
1	1	-	-	-	-	-	-
1	0	0	1	-	1	1	-

Es decir, $J_0 = Q_1^n$, $K_0 = K_1 = 1$ y $J_1 = \overline{Q_0}^n$

La selección de los valores respectivos de J y K se hará a través de MUXes 2 a 1, uno por cada entrada J y K de cada biestable. Definiremos la señal de control de los MUXes como $\overline{up}/down$. Si $\overline{up}/down=0$ la cuenta será ascendente (\overline{up}) y si $\overline{up}/down=1$ la cuenta será descendente ($down$). El circuito final es el de la figura 7.6.

7.6. CONTADORES DISEÑADOS A PARTIR DE BLOQUES

Existen varios tipos de contadores en circuitos integrados que cuentan en BCD ($\{\dots, 0, 1, 2, \dots, 9, 0, \dots\}$) o en binario ($\{\dots, 0, 1, 2, \dots, 15, 0, \dots\}$). Estos contadores pueden ser

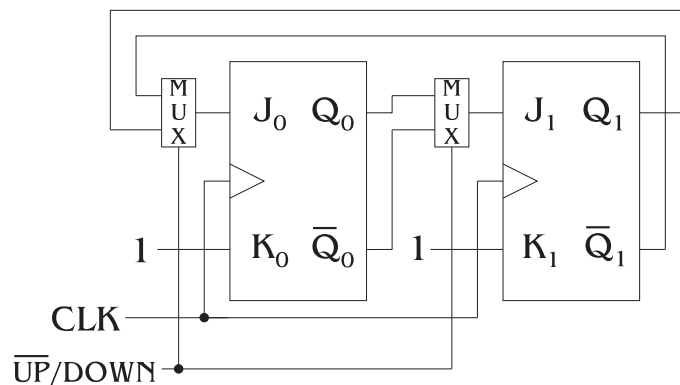


Figura 7.6: Contador reversible 0, 1, 2/ 2, 1, 0.

reversibles o no, tener la posibilidad de carga en paralelo (LOAD), borrado (CLEAR) y conteo (COUNT) (cuando el conteo es 0, el contador está parado y mantiene el valor de la cuenta). Lo más usual es que el borrado se realice mediante la activación de los Clear de los flip-flops (entradas asíncronas) y que, por tanto, sea una señal asíncrona. La carga en paralelo suele ser síncrona.

Además del valor de la cuenta, suelen tener como salida una señal de ACARREO, que se pone a 1 durante la última cuenta del contador (en la cuenta 9 si el contador es BCD o en la 15 si es binario), siendo 0 en los demás instantes. Esta señal (negada) se suele utilizar para conectar en cascada, varios contadores.

En la figura 7.7 se muestra el diagrama interno de un contador binario de 4 bits con carga en paralelo síncrona, borrado asíncrono y señal de conteo.

Este contador se puede utilizar como bloque fundamental para diseñar otros contadores a partir de él. Por ejemplo, se puede construir un contador módulo 6 de 4 formas distintas (ver figura 7.8), según se use la carga en paralelo síncrona o el borrado asíncrono.

(a) Se detecta el estado 0110 (6) y se impide la cuenta de este estado actuando sobre el CLEAR (señal asíncrona), el cual pone inmediatamente a cero el contador.

(b) Se detecta el estado 0101 (5) y se actúa sobre LOAD. La carga en paralelo (del 0), al ser síncrona, se realizará en el ciclo siguiente (en el siguiente flanco de reloj), por lo cual este estado se contará.

(c) Detectando el estado 1111 (15) mediante el acarreo y actuando sobre la carga en paralelo (cargando la cuenta 10).

(d) Detectando el estado 1000 (8) mediante Q_3 y actuando sobre la carga en paralelo para cargar el estado 3.

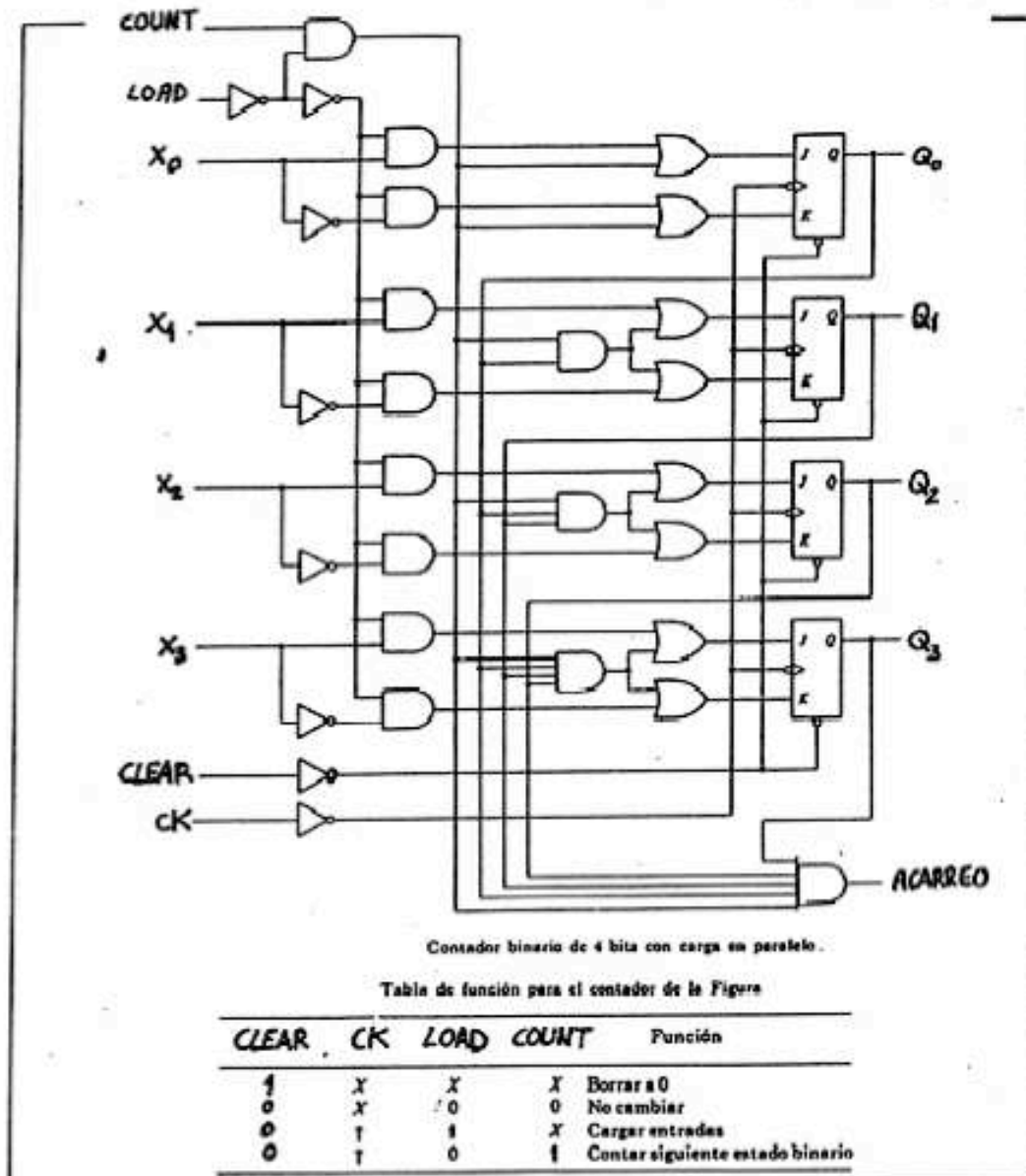


Figura 7.7: Contador binario de 4 bits integrado, con carga paralela síncrona, borrado asíncrono y señal de conteo.

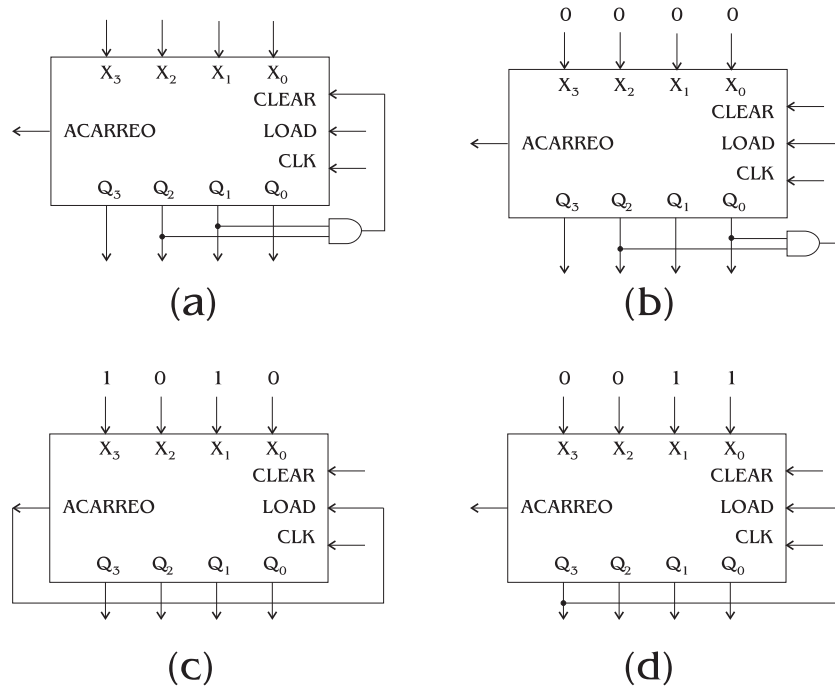


Figura 7.8: Cuatro formas de construir un contador módulo 6 a partir del contador integrado anterior.

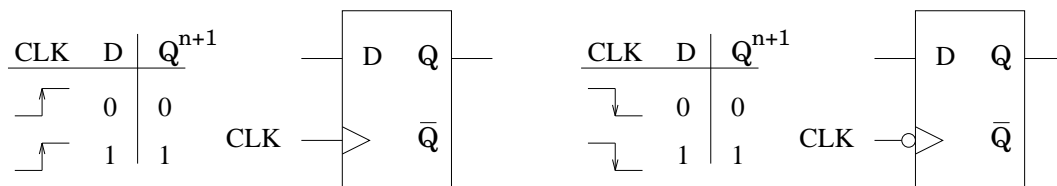


Figura 7.9: Símbolos y tablas del flip-flop tipo D.

7.7. REGISTROS DE DESPLAZAMIENTO

Un registro tiene como función primordial el almacenar información. La diferencia entre un registro y un flip-flop es que éste sólo puede almacenar un bit, mientras que un registro es capaz de almacenar n bits. Un registro consta, básicamente, de un conjunto de celdas de almacenamiento binarias (generalmente constituidas por flip-flops tipo D disparados a flancos) más un conjunto de puertas encargadas de realizar su conexión. Un flip-flop tipo D disparado a flancos, bien positivos, bien negativos, traslada el valor de la señal de entrada (D) a su salida cuando recibe un flanco activo de la señal de reloj. En los demás instantes actúa como una celda de almacenamiento, reteniendo el valor del bit que ha cargado.

La forma más sencilla de construir un registro de desplazamiento (a la izquierda) de n bits es conectar n flip-flops D uniendo la salida de cada flip-flop a la entrada del siguiente en la forma indicada en la figura 7.10, es decir, $D_i = Q_{i-1}$. El registro sólo tendrá una entrada externa, la del flip-flop menos significativo, D_0 o S , por la que entrarán los datos

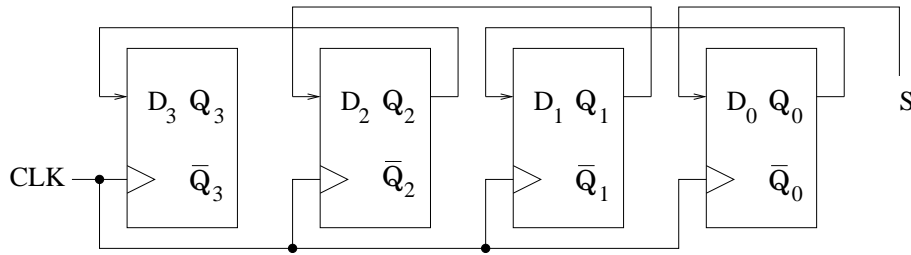


Figura 7.10: Registro de desplazamiento a la izquierda.

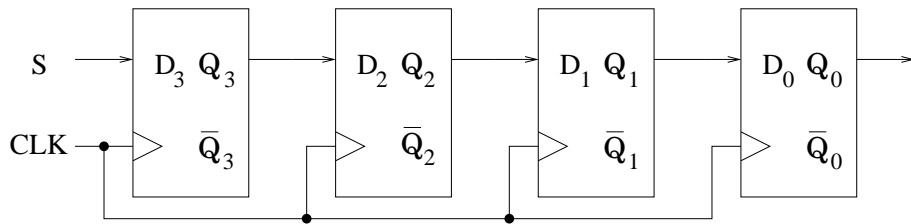


Figura 7.11: Registro de desplazamiento a la derecha.

al registro (uno en cada ciclo de reloj).

El registro que hemos diseñado realiza un desplazamiento de los bits hacia la izquierda, realizando a la inversa la conexión de los flip-flops podemos construir un registro de desplazamiento a la derecha. La salida de cada uno de los flip-flops se conectará a la entrada del situado a su derecha, es decir, $D_i = Q_{i+1}$. La entrada de los datos se realiza por el flip-flop más significativo, D_{n-1} o S , tal y como se ve en la figura 7.11. En cada ciclo de reloj van entrando nuevos bits y el contenido del registro se va desplazando a cada ciclo de reloj de flip-flop en flip-flop (figura 7.12).

7.7.1. Registro bidireccional con carga en paralelo y borrado

Incluyendo los dos tipos de conexiones entre los flip-flops D podemos construir un registro con posibilidad de ser desplazado tanto hacia la derecha como hacia la izquierda.

		Q_3	Q_2	Q_1	Q_0
ciclo 0	d_0	-	-	-	-
ciclo 1	d_1	d_0	-	-	-
ciclo 2	d_2	d_1	d_0	-	-
ciclo 3	d_3	d_2	d_1	d_0	-
ciclo 4		d_3	d_2	d_1	d_0

Figura 7.12: Funcionamiento de un registro de desplazamiento hacia la derecha.

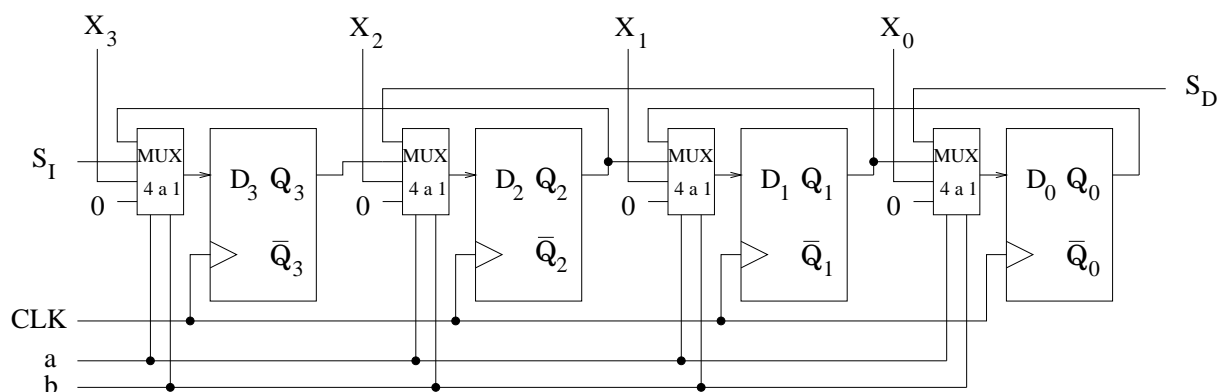


Figura 7.13: Registro bidireccional con carga en paralelo y borrado.

La selección de una u otra operación se realizará mediante MUXes en la entrada D de cada flip-flop. La carga en paralelo y el borrado se pueden añadir usando MUXes 4 a 1, conectando a una de las entradas un dato procedente del exterior (para la carga en paralelo) o un “0” para el borrado. Ambas operaciones, realizadas de esta forma, son síncronas.

Selección del MUX		Acción
a	b	
0	0	Desplazamiento a la izquierda
0	1	Desplazamiento a la derecha
1	0	Carga en paralelo síncrona
1	1	Borrado síncrono

La disposición de los MUXes, sus entradas y sus conexiones se pueden ver en la figura 7.13.

Como ya sabemos, desplazar un número binario i veces a la izquierda equivale a multiplicarlo por 2^i . Igualmente, desplazar un número binario hacia la derecha i posiciones equivale a dividirlo entre 2^i (quedándonos únicamente con la parte entera de dicha división). Además de multiplicar y dividir números entre potencias de 2, los registros de desplazamiento se utilizan para la conversión de datos paralelo/serie y serie/paralelo (imprescindible en las comunicaciones serie). Veamos como se utilizan los registros de desplazamiento en tales aplicaciones.

7.7.2. Conversión serie/paralelo y paralelo/serie

La manipulación de datos constituidos por varios bits puede realizarse de dos formas distintas: en modo serie o en modo paralelo. Se dice que un sistema digital opera en modo serie cuando la información se transfiere y manipula bit a bit. Por ejemplo, cuando el contenido de un registro se transfiere a otro desplazando los bits de un registro al siguiente, un bit en cada ciclo de reloj. Para prevenir la pérdida de datos, se hace recircular

el contenido del primero de los registros. Si el registro es de n bits, serán necesarios n ciclos de reloj para la transferencia.

En el modo paralelo, la información se transfiere y manipula en todos sus bits a la vez. Por ejemplo, en una transferencia entre dos registros en modo paralelo, todos los bits se transfieren del primer registro al segundo en un solo ciclo de reloj, mediante una carga en paralelo. Si los datos son de n bits, los registros han de tener n salidas y n entradas.

Conversión de datos paralelo a serie

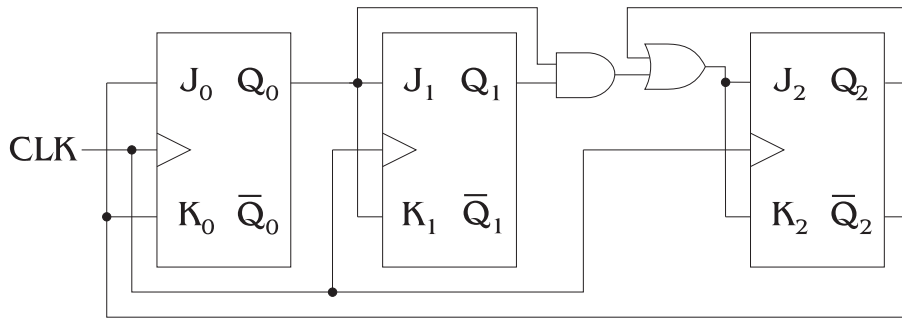
Los datos entran en el registro en un ciclo de reloj, todos los bits a la vez, mediante una carga en paralelo. Los bits de salida se toman en la salida Q del último flip-flop D (el menos significativo) uno en cada ciclo de reloj, a la vez que se va desplazando el registro hacia la derecha.

Conversión de datos serie a paralelo

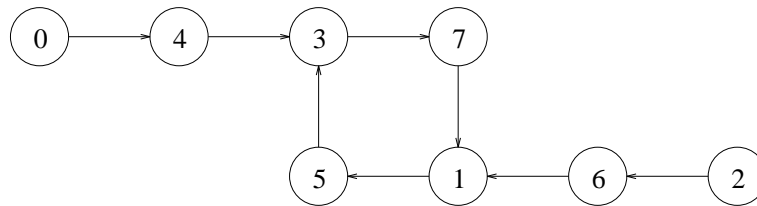
Los datos entran por la entrada serie del registro, un bit en cada ciclo de reloj. Una vez completada la carga de todos los bits, se toman como líneas de salida las Q de cada uno de los flip-flops D (se leen todos los bits a la vez).

EJERCICIOS

- 7.1. Construir un contador módulo 15 con un contador módulo 3 ($\{\dots, 0, 1, 2, \dots\}$) y un contador módulo 5 ($\{\dots, 2, 3, 4, 5, 6, \dots\}$).
- 7.2. Obtener la secuencia de conteo del siguiente contador:

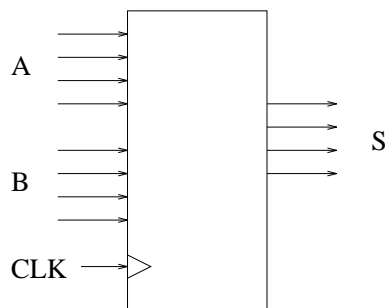


- 7.3. Diseñar un contador que siga la secuencia de estados de la siguiente figura utilizando flip-flops JK.



- 7.4. Construir un sistema secuencial con la estructura la siguiente figura, tal que realice continuamente las siguientes operaciones:
- i) durante 3 ciclos del reloj, la salida S debe ser igual a la entrada A.
 - ii) durante los 2 ciclos siguientes, S debe ser igual a la entrada B.

Estas operaciones se realizan continuamente, es decir, una vez realizada ii) vuelve a realizarse i).



- 7.5. Construir un sistema a partir de biestables activos en flancos positivos y circuitos combinatoriales que continuamente realice sobre 4 bits la siguiente secuencia de operaciones (una en cada ciclo de reloj):

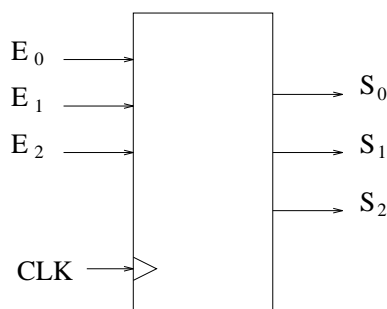
- I) Desplazamiento a la derecha.
- II) Desplazamiento a la derecha.
- III) Carga paralelo.
- IV) Desplazamiento a la izquierda.
- v) Desplazamiento a la izquierda.
- VI) Borrado de los cuatro bits.
- VII) Carga paralelo.

7.6. Construir un sistema secuencial con la estructura de la siguiente figura, tal que realice continuamente la conversión de datos:

$$\begin{array}{rcl} c_0 & b_0 & a_0 \rightarrow a_2 \quad a_1 \quad a_0 \\ c_1 & b_1 & a_1 \rightarrow b_2 \quad b_1 \quad b_0 \\ c_2 & b_2 & a_2 \rightarrow c_2 \quad c_1 \quad c_0 \end{array}$$

Es decir, en el **primer** ciclo del reloj, entran en el sistema el bit a_0 en la entrada E_0 , a_1 en E_1 y a_2 en E_2 ; en el **segundo** entran b_0 en E_0 , b_1 en E_1 y b_2 en E_2 ; en el **tercero** entran c_0 en E_0 , c_1 en E_1 y c_2 en E_2 . Entonces, en el **cuarto** ciclo de reloj ha de salir el bit a_0 por la salida S_0 , b_0 por S_1 y c_0 por S_2 ; en el **quinto** deben salir a_1 por S_0 , b_1 por S_1 y c_1 por S_2 ; y en el **sexto** deben salir a_2 por S_0 , b_2 por S_1 y c_2 por S_2 ;

La conversión de datos debe realizarse continuamente, es decir, en el séptimo ciclo debe comenzar una nueva conversión.



7.7. Contruir la siguiente señal a partir de un contador binario de 3 bits.



7.8. I) Diseñar un circuito que a partir de una señal cuadrada de 1Hz genere dos pulsos de anchura 0.5 sg separados por otro 0.5 sg cada 6 sg.

- II) Utilizando dos circuitos del tipo del apartado i) y puertas lógicas generar a partir de una señal cuadrada de 1Hz dos pulsos de anchura 0.5 sg separados por otro 0.5 sg cada 36 sg.
 - III) Utilizando tres circuitos del tipo del apartado i) y puertas lógicas generar a partir de una señal cuadrada de 1Hz dos pulsos de anchura 0.5 sg separados por otro 0.5 sg cada 216 sg.
- 7.9. Diseñar un reloj digital que tenga como salidas horas y minutos a partir de una señal cuadrada de 100Hz. La salida debe proporcionarse en **BCD**. Se debe permitir poner en hora el reloj, introduciendo valores arbitrarios de horas y minutos cuando se active la señal PONER_EN_HORA. Solo se permite utilizar contadores módulo 16 con carga en paralelo síncrona (no disponen de borrado) y circuitos combinacionales.
- 7.10. Construir a partir de biestables disparados en flancos positivos y con Preset y Clear activos a nivel alto un dispositivo que realice continuamente, sobre 4 palabras de 2 bits las siguientes operaciones, (una en cada ciclo de reloj):
- I) Carga de una palabra desde el exterior en la posición más a la izquierda, desplazando hacia la derecha las demás palabras, perdiéndose la última.
 - II) Desplazamiento a la izquierda de cada una de las cuatro palabras, recirculando la situada más a la izquierda a la última posición de la derecha.
 - III) Carga de la palabra 01 en la posición más a la derecha, desplazando a la izquierda las demás palabras y perdiéndose la primera.
 - IV) Carga de la palabra 10 en la posición más a la izquierda, desplazando hacia la derecha las demás palabras, perdiéndose la última.
 - V) Borrado del bit más significativo de las cuatro palabras.
- 7.11. Utilizando contadores módulo 16 con señales de carga en paralelo (LOAD), borrado (CLEAR) y conteo (COUNT) y una salida de ACARREO y puertas lógicas adicionales construye:
- Un contador módulo 7 que cuente de 0 a 6.
 - Un contador módulo 7 que siga la secuencia: 0,1,2,12,13,14,15.
 - Un contador módulo 35.
- 7.12. Utilizando el menor número de biestables JK y puertas lógicas adicionales, diseña un circuito **TOTALMENTE SÍNCRONO** que dé como salida la secuencia: {... , 5 , 20, 2, 9, 20, ...} (en binario natural).
- 7.13. Diseña un contador que haga la secuencia 0, 4, 6, 7, 3, 1 con un registro de desplazamiento a la derecha de 3 bits y puertas lógicas adicionales.
- 7.14. Diseña el circuito controlador de un cargador de CDs que tenga como salidas el número de CD y la canción seleccionada. El dispositivo dispone de 4 entradas: MARCHA, REPETICIÓN, SELECCIÓN y ESCANEADO que actúan del siguiente modo:

- I) Activando MARCHA se va cambiando de canción de forma secuencial, recorriendo primero todas las canciones de un CD, a continuación las del siguiente, y volviendo al primero una vez finalizada la última canción del último CD.
- II) Activando REPETICIÓN al acabar las canciones de un CD se vuelven a repetir las canciones de ese CD desde el principio sin pasar al siguiente CD.
- III) Activando SELECCIÓN se puede fijar externamente el número de CD y una canción determinada (usando entradas adicionales del controlador).
- IV) Activando ESCANEEO se selecciona la misma canción, primero en un CD, después en el siguiente y así hasta acabar los CDs. En ese momento se pasa a la siguiente canción y se repite el proceso. Al finalizar todas las canciones de todos los CDs se vuelve a la primera canción del primer CD.

Implementa el circuito suponiendo que el cargador tiene capacidad para 3 CDs y que el primer CD tiene 12 canciones, el segundo 10 y el tercero 14. Supón que solo una entrada está activa en cada momento. Si no hay ninguna activa, el circuito mantiene el último CD y canción seleccionados. Cada vez que acaba una canción o cambia una entrada se activa una señal que se utiliza como reloj (CK) para el controlador.

- 7.15. Diseña un circuito que recibe un número x , $0 \leq x \leq 3$, por sus líneas de entrada $D_1 D_0$ en cada ciclo. El circuito tiene un funcionamiento por periodos de 16 ciclos. En los quince primeros ciclos va contabilizando el número de veces que recibe cada uno de los valores posibles de su entrada. Estos valores están siempre disponibles en las salidas $S_{x3} S_{x2} S_{x1} S_{x0}$ para cada uno de los valores posibles de la entrada ($0 \leq x \leq 3$). Por lo tanto, el circuito tiene 16 líneas de salida. Tras los quince ciclos susodichos, hay un ciclo en el que la entrada es ignorada y que se emplea para restablecer el estado interno del circuito de cara al siguiente periodo de conteo.
- 7.16. Diseña un circuito con una entrada X y una salida Y (ambas de cuatro bits) que invierta el orden de los datos siguiendo un ciclo de cuatro flancos de reloj (frecuencia 1 Hz). Su funcionamiento sería el siguiente:

$$\begin{array}{l} \mathbf{X} : X_0 \quad X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5 \quad X_6 \quad X_7 \quad X_8 \quad X_9 \quad X_{10} \quad X_{11} \quad X_{12} \quad \dots \\ \mathbf{Y} : 0 \quad 0 \quad 0 \quad 0 \quad X_3 \quad X_2 \quad X_1 \quad X_0 \quad X_7 \quad X_6 \quad X_5 \quad X_4 \quad X_8 \quad \dots \end{array}$$

NOTA: Todos los datos X_i son números binarios puros de 4 bits.

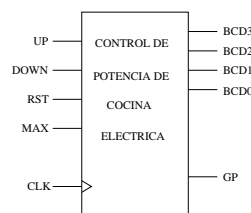
- 7.17. Diseña un circuito digital para el control de la velocidad de funcionamiento del limpiaparabrisas de un coche. El circuito recibe como entrada una onda cuadrada utilizada como base de tiempos y proporciona como salida otra onda cuadrada. La frecuencia de la onda de salida dependerá de la posición de un mando de control que está accionado por el conductor para seleccionar la velocidad. El limpiador se activa en los flancos positivos de la señal de salida y el mando de control puede estar en una de estas cuatro posiciones:

- $A \Rightarrow$ el limpiaparabrisas está parado.

- B \Rightarrow el limpiador se activa una vez cada dos segundos.
- C \Rightarrow el limpiador se activa una vez por segundo.
- D \Rightarrow el limpiador se activa cuatro veces por segundo.

Como parte de la solución del ejercicio se debe calcular qué frecuencia debe tener la onda utilizada como base de tiempos para poder obtener en la salida las temporizaciones solicitadas. El circuito contará además con una entrada que indique el estado del contacto del coche, y sólo funcionará cuando éste esté activado.

7.18. Diseña un circuito para controlar la potencia de una cocina eléctrica. El esquema general del sistema se muestra en la siguiente figura:



- i) Las señales BCD(3:0) nos indican el nivel de potencia expresado en BCD. El circuito debe proporcionar 10 niveles de potencia, que variarán desde apagado (nivel 0) a máxima potencia (nivel 9).
 - 1) Las señales UP y DOWN suben y bajan respectivamente el nivel de potencia en una unidad. Si estamos en el nivel 0 y seleccionamos DOWN pasamos al nivel 9; del mismo modo si estamos en el nivel 9 y seleccionamos UP pasamos a nivel 0. Si seleccionamos a la vez UP y DOWN el sistema no se mueve de nivel.
 - 2) Se dispone además de una señal de reset (RST) que pone el nivel de potencia a 0, y una señal de máxima potencia (MAX) que selecciona el nivel de potencia máxima sin necesidad de pasar por todos los niveles anteriores. Sólo una de estas dos señales puede estar activa a la vez.
- ii) La señal GP gradúa el nivel de potencia. La graduación se consigue generando una señal con un porcentaje mayor o menor de tiempo en activo (en alta) de la forma mostrada en la figura. La señal GP debe reiniciarse (comenzar desde el ciclo 0) cada vez que se produzca un cambio en el nivel de potencia.

