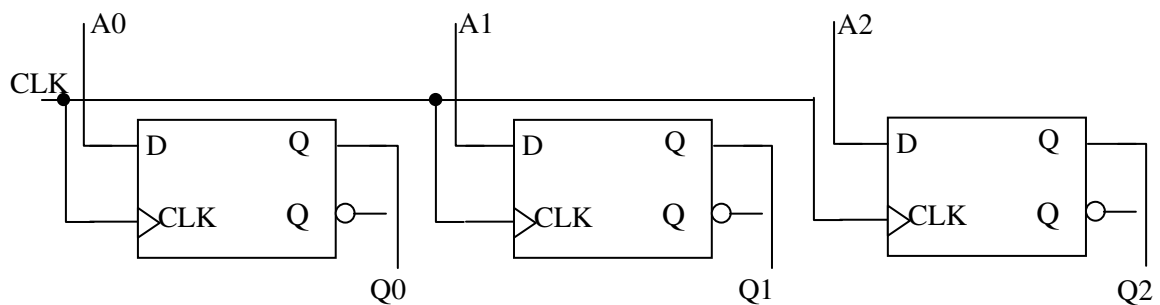


Tema 8. Circuitos secuenciales de Propósito general: REGISTROS Y CONTADORES

Una colección de dos o más biestables **D** con una entrada común se conoce como un **registro**. Los registros se usan para almacenar una serie de bits relacionados, como un byte (8 bits) de una computadora.

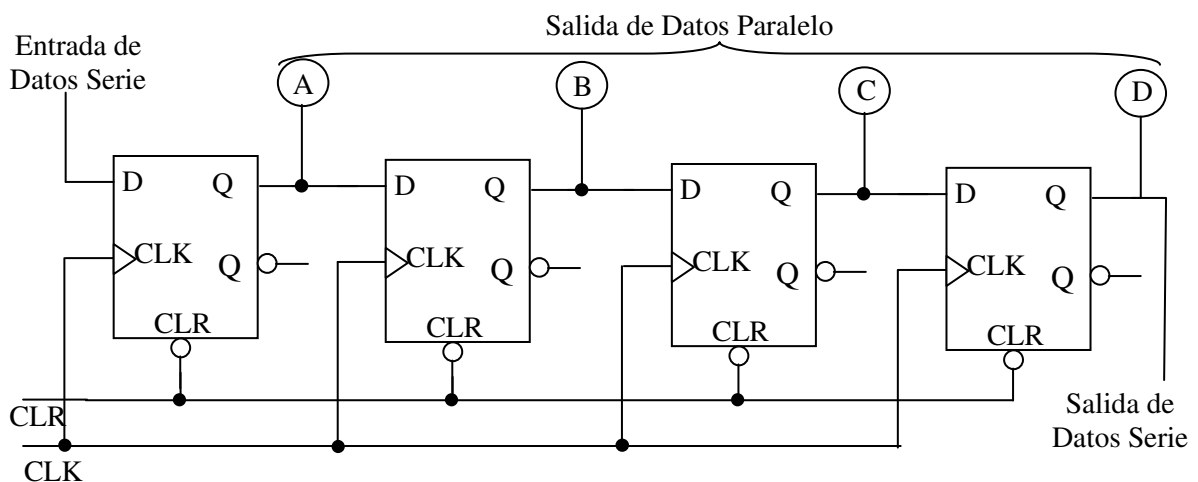
La figura muestra un registro de 3 bits. La palabra **A[0-2]** solo aparece en la salida **Q[0-2]**, si se produce el disparo del biestable, mediante el flanco de subida del reloj **CLK**:



Como vemos este circuito “registra” el dato de la entrada y lo transmite a la salida solo cuando se le da la orden. La transmisión del dato se hace de forma paralela.

Registros de Desplazamiento

Es un registro que ‘registra’ y ‘desplaza’ la información. La figura siguiente presenta un registro de desplazamiento a la derecha de 4 bits con carga serie:



Veamos el funcionamiento del circuito anterior:

La señal de borrado (**CLR**) coloca a los biestables en el estado **0**. Cada vez que llega un flanco de subida de la señal de reloj (**CLK**), cada biestable ‘captura’ lo que hay en su entrada, o sea, lo que hay en el biestable anterior, con lo que la información se va desplazando a la derecha.

Un ejemplo: Vamos a registrar y desplazar la palabra de 4 bits **0101**. Esta sería la secuencia de funcionamiento del circuito:

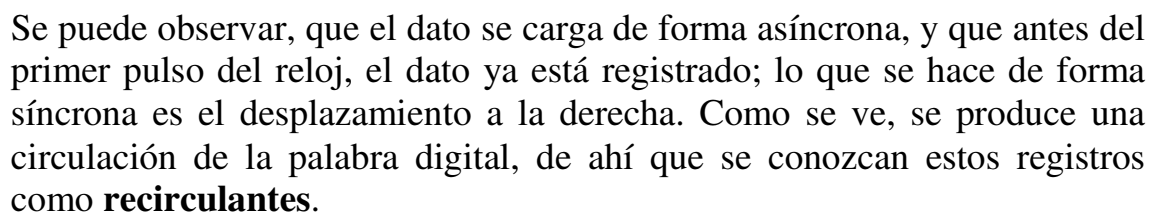
Salidas	
ABCD	
CLR = 0	0 0 0 0
CLR = 1 / 1 ^{er} Pulso de CLK	1 0 0 0
CLR = 1 / 2 ^o Pulso de CLK	0 1 0 0
CLR = 1 / 3 ^{er} Pulso de CLK	1 0 1 0
CLR = 1 / 4 ^o Pulso de CLK	0 1 0 1

La palabra **0101** ha sido cargada en el registro, de hecho la tenemos disponible en las salidas paralelo usando 4 pulsos de reloj. Para que tuviéramos toda la palabra disponible en la salida serie, debemos esperar 3 pulsos más:

Salidas Serie	
D	
CLR = 1 / 4 ^o Pulso de CLK	1
CLR = 1 / 5 ^o Pulso de CLK	0
CLR = 1 / 6 ^o Pulso de CLK	1
CLR = 1 / 7 ^o Pulso de CLK	0

La palabra se ha cargado en el registro bit a bit, a esto se le denomina carga serie.

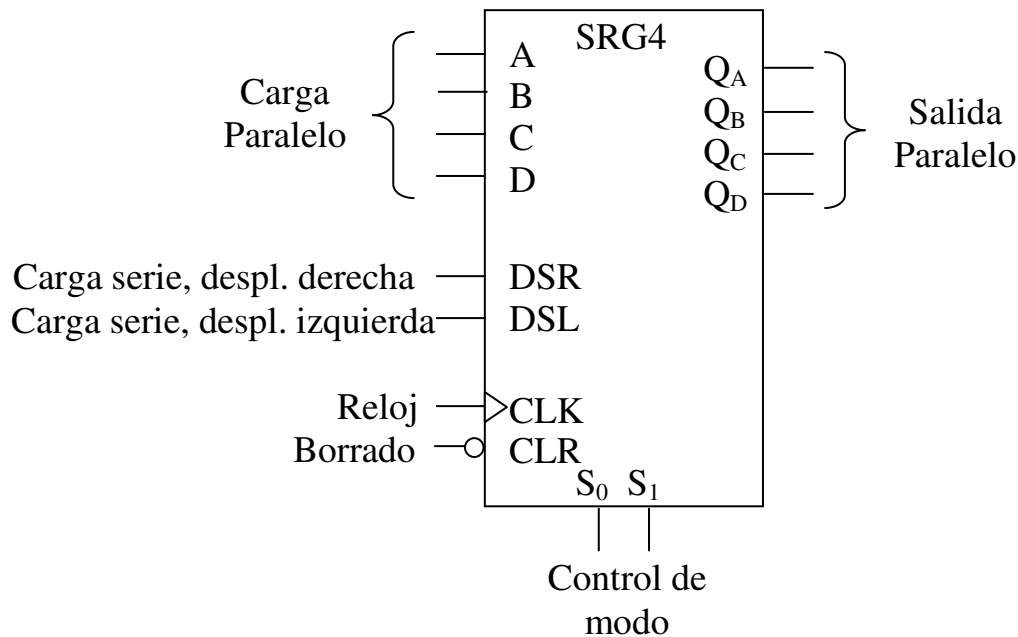
El siguiente circuito es un registro de desplazamiento de 4 bits de carga paralelo y desplazamiento a la derecha:



Registro de Desplazamiento Universal

Se trata de un circuito integrado, que dispone de un registro de desplazamiento, que permite carga serie, carga paralela, desplazamiento a izquierda y a derecha, mediante el uso de unas señales de control.

La figura muestra un Registro de Desplazamiento Universal de 4 bits (74194).



Modo de operación	S ₀	S ₁
Mantenimiento	0	0
Despl. Izquierda	0	1
Despl. Derecha	1	0
Carga paralelo	1	1

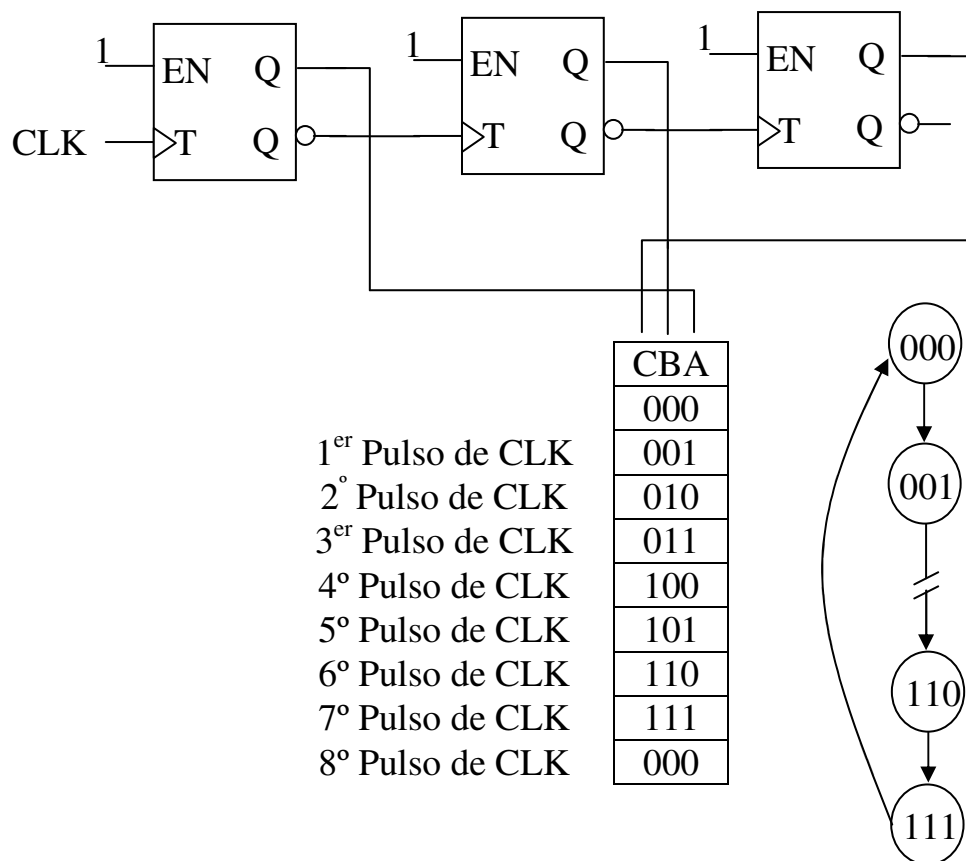
El funcionamiento de este dispositivo es similar a los descritos anteriormente; cabe hacer notar que cuando se selecciona la operación de desplazamiento a la derecha el bit que se carga (entrada: **DSR**) queda registrado en la posición A (salida: **Q_A**), mientras que si seleccionamos la operación de desplazamiento a la izquierda, el bit que se carga (entrada: **DSL**) queda registrado en la posición D (salida: **Q_D**).

CONTADORES

El nombre de contador se usa para designar cualquier circuito secuencial que cuente **m** estados en un solo ciclo de funcionamiento. El número **m** de estados que cuenta se conoce como **módulo** del contador. Nuestro interés se centrará en los contadores binarios, aquellos que realizan una cuenta en binario de un número determinado de estados.

Por ejemplo, veamos un contador de módulo-8 (**mod-8**)

Este circuito cuenta 8 estados, con lo que necesita 3 bits, a cada pulso de reloj, avanza en la cuenta de estados:



Se le suele llamar A este tipo de contador contador de rizado

Un contador en general puede contar con una entrada que le permita contar tanto adelante como hacia atrás, incluso con entrada de inhibición que detenga la secuencia de conteo, y por último con un registro que permita establecer un estado inicial a partir del cuál comienza la cuenta.

Vamos a diseñar un contador síncrono binario que cuente de 0 a 7 y sea reversible. Emplearemos biestables JK y circuitos lógicos SSI o MSI.

Comencemos con la tabla de transición:

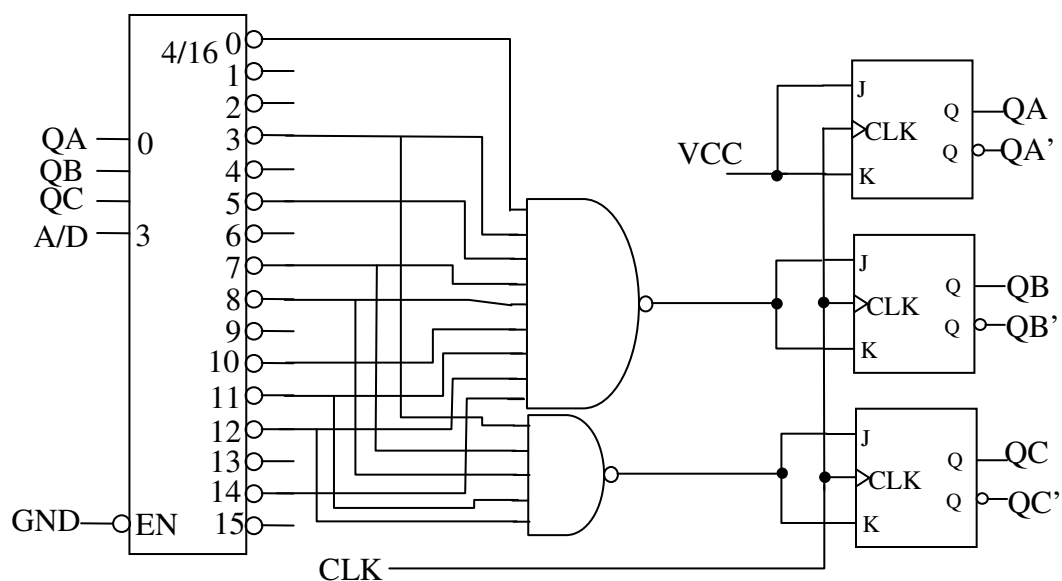
Ent A/D	Estado presente			Estado futuro			Entrada biestables					
	qc	qb	qa	QC	QB	QA	Jc	Kc	Jb	Kb	Ja	Ka
0	0	0	0	0	0	1	0	X	0	X	1	X
0	0	0	1	0	1	0	0	X	1	X	X	1
0	0	1	0	0	1	1	0	X	X	0	1	X
0	0	1	1	1	0	0	1	X	X	1	X	1
0	1	0	0	1	0	1	X	0	0	X	1	X
0	1	0	1	1	1	0	X	0	1	X	X	1
0	1	1	0	1	1	1	X	0	X	0	1	X
0	1	1	1	0	0	0	X	1	X	1	X	1
1	0	0	0	1	1	1	1	X	1	X	1	X
1	0	0	1	0	0	0	0	X	0	X	X	1
1	0	1	0	0	0	1	0	X	X	1	1	X
1	0	1	1	0	1	0	X	1	1	X	1	X
1	1	0	0	0	1	1	X	1	1	X	1	X
1	1	0	1	1	0	0	X	0	0	X	X	1
1	1	1	0	1	0	1	X	0	X	1	X	1
1	1	1	1	1	1	0	X	0	X	0	1	X

De la tabla se obtienen las funciones combinacionales que excitarán cada biestable, pudiéndose realizar con decodificadores de 4 a 16 líneas:

$$J_a = K_a = 1$$

$$J_b = K_b = \Sigma m(1,3,5,7,8,10,11,12,14)$$

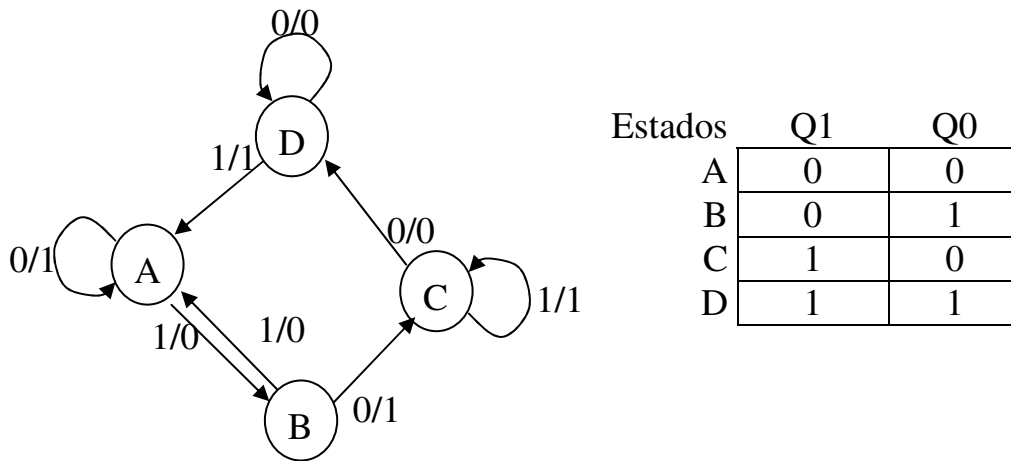
$$J_c = K_c = \Sigma m(3,7,8,11,12)$$



Diseño de Sistemas Secuenciales Síncronos con Registros

1. Codificación en binario natural y uso de un decodificador:

Supongamos el siguiente diagrama de flujo y la asignación de estados



Usaremos los biestables D que “contiene” el registro universal. Se comienza observando en que estados internos cada una de las variables Q1 y Q0 toman el valor 1, o sea, cuando se activan. Suponiendo X la entrada y Z la salida:

Q0 se activará cuando se llegue a los estados B y D; al estado B se llega desde A con entrada 1 (término AX); al estado D se llega desde C con entrada 0 (término CX') y desde el propio D con entrada 0 (término DX'), así la expresión booleana de Q0 será:

$$Q0_{n+1} = AX + CX' + DX'$$

Q1 se activará cuando se llegue a los estados C y D; al estado C se llega desde B con entrada 0 (término BX') y desde el propio C con entrada 1 (término CX), así la expresión booleana de Q1 será:

$$Q1_{n+1} = BX' + CX + CX' + DX' = BX' + C + DX'$$

De la misma manera se puede obtener Z como:

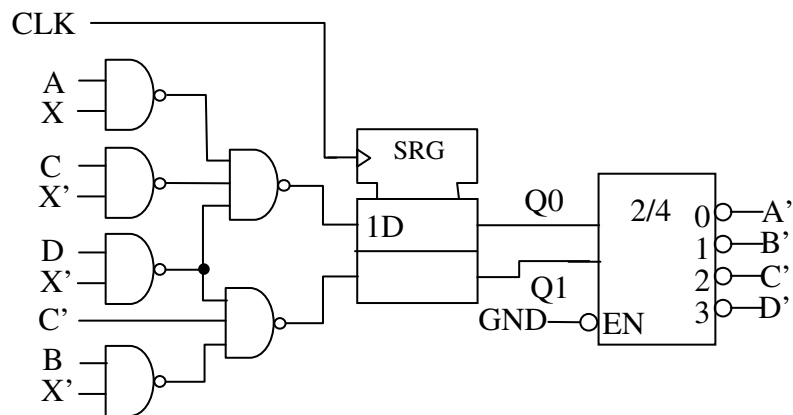
$$Z = AX' + DX + CX + BX'$$

Teniendo en cuenta que el registro contiene biestables D: $Q_{n+1} = D$

$$D0 = AX + CX' + DX'$$

$$D1 = BX' + C + DX'$$

A continuación implementemos el circuito (excepto la salida). La última etapa la forman el registro (en este caso de 2 bits) y un decodificador de 2 bits:



2. Codificación mediante un código de uno entre n (un FF por estado).

Este proceso de diseño es aún más sistemático y permite diseñar más rápidamente el sistema pero se deben usar más biestables (uno por cada variable de estado interno). Basándonos en el diagrama de flujo anterior:

Estados	Q3	Q2	Q1	Q0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

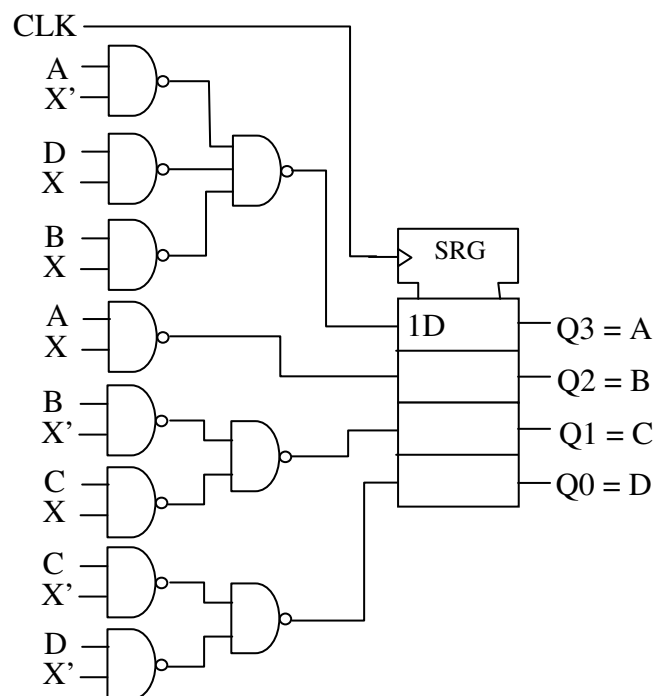
ECUACIONES:

$$Q3_{n+1} = AX' + DX + BX$$

$$Q2_{n+1} = AX$$

$$Q1_{n+1} = BX' + CX$$

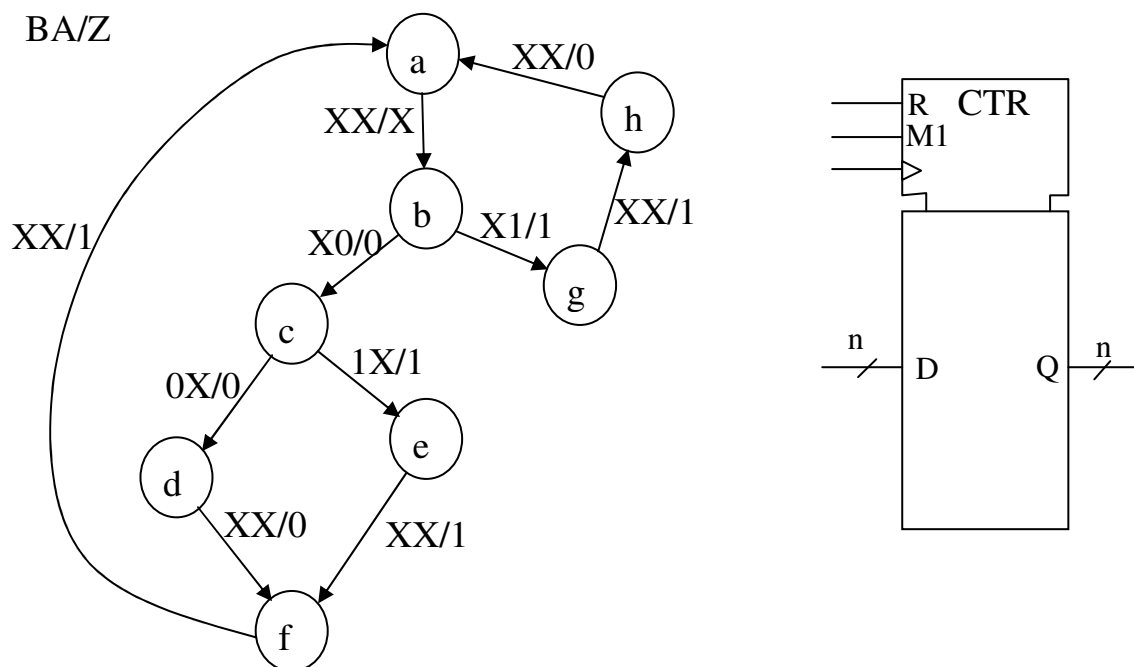
$$Q0_{n+1} = CX' + DX'$$



Diseño de sistemas secuenciales síncronos usando Contadores

En los diagramas de flujo de los sistemas secuenciales existen pasos de un estado a otro (como los contadores) y pasos al propio estado (como si el contador “parase”). Todas las posibilidades en un diagrama de flujo pueden ser cubiertas usando contadores de manera apropiada.

El contador síncrono que usaremos para nuestros ejemplos tendrá una entrada de reloj, una entrada de reset (R) que permita situar al contador en el estado inicial, una entrada de cuenta y no carga ($M1=0$) y no cuenta y carga en paralelo ($M1=1$) y n entradas de carga paralelo. Para ver el método de diseño nos basaremos en el diagrama de flujo siguiente:



Debemos generar entonces las entradas D y la señal M1.

Comenzaremos deduciendo el número de variables de estado Q que nos hacen falta para describir los estados. Este será número de biestables que contenga el contador. Debe cumplirse que se necesitan k biestables para n estados tal que $2^k \geq n$. La asignación de los códigos correspondiente a cada estado debe realizarse de manera el número de saltos entre estados que no tengan asignados códigos consecutivos sea mínimo.

Estado	Q2	Q1	Q0
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1

Estado	Q2	Q1	Q0
E	1	0	0
F	1	0	1
G	1	1	0
H	1	1	1

La tabla de estados que debemos realizar debe contener: las entradas, el estado presente, el estado futuro, M1 (no carga-0 o carga-1), el estado a cargar si no cuenta y la salida:

Entrada		Estado Presen.			Estado Futuro			Carga	Carga Estado			Sal
B	A	q2	q1	q0	Q2	Q1	Q0	M1	D2	D1	D0	Z
X	X	0	0	0	0	0	1	0	X	X	X	0
X	0	0	0	1	0	1	0	0	X	X	X	0
X	1	0	0	1	1	1	0	1	1	1	0	1
0	X	0	1	0	0	1	1	0	X	X	X	0
1	X	0	1	0	1	0	0	1	1	0	0	1
X	X	0	1	1	1	0	1	1	1	0	1	0
X	X	1	0	0	1	0	1	0	X	X	X	1
X	X	1	0	1	0	0	0	1	0	0	0	1
X	X	1	1	0	1	1	1	0	X	X	X	1
X	X	1	1	1	0	0	0	0	X	X	X	0

Ya solo queda deducir D2, D1, D0, M1 y Z por el método que se quiera:

$$D2 = q2'$$

$$D1 = q1'q2'$$

$$D0 = q0q1$$

$$M1(B,A,q2,q1,q0) = \Sigma m(3,5,9,11,13,18,19,21,25,26,27,29)$$

$$Z(B,A,q2,q1,q0) = \Sigma m(4,5,6,9,12,13,14,18,20,21,22,25,26,28,29,30)$$

El sistema quedaría como la figura indica. El circuito combinacional que implementa M1 y Z se realizará a modo de ejercicio en clase

