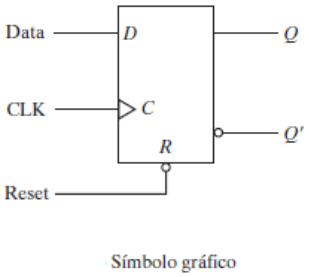
PRÁCTICO 5 - Circuitos Secuenciales

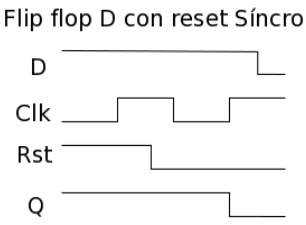
Flip flop D con reset Asíncrono:

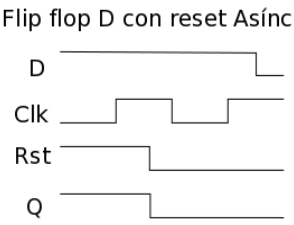


Símbolo gráfico

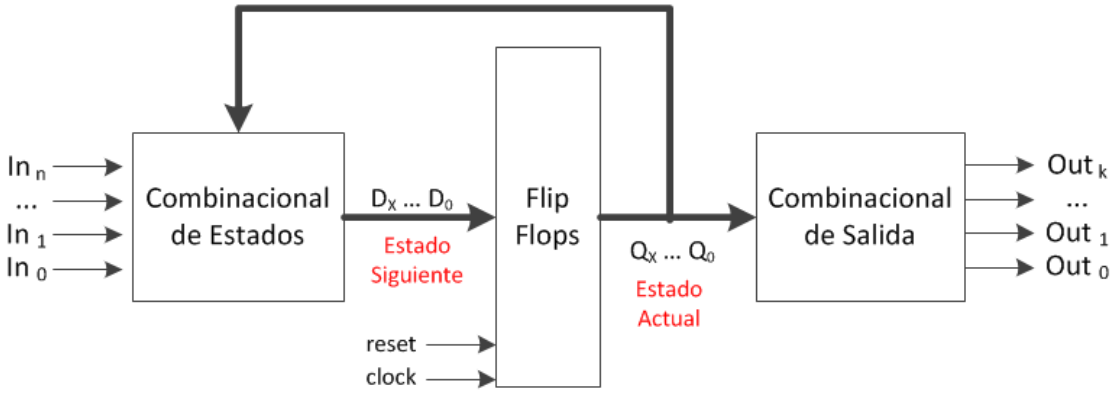
R	C	D	Q	Q'
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

Tabla de función



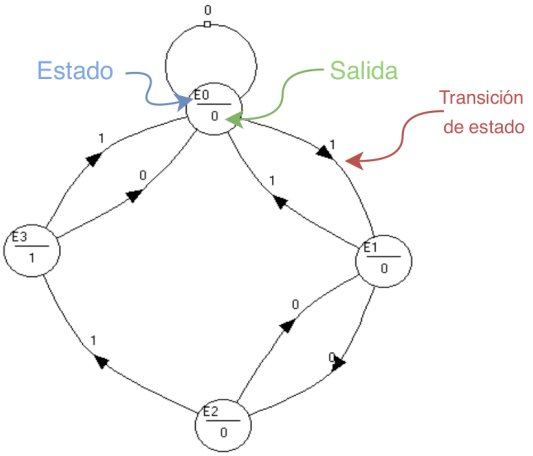


Implementación de circuitos secuenciales:



Formato de tablas para la resolución de ejercicios de circuitos secuenciales:

Entradas del combinacional de estados				Salidas del combinacional de estados	
Estado Actual		Entradas		Estado siguiente	
Q_X	Q_0	In_n	In_0	D_X	D_0
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...

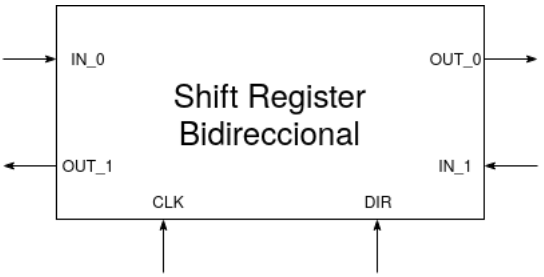


Estado Actual			Salidas del circuito	
Codificación	Q_X	Q_0	Out_K	Out_0
E0	...	...	...	...
E1	...	...	...	...
...	...	...	...	...

Ejercicio 1:
Implementar un registro de entrada y salida en paralelo de 4 bits con Flip-flops tipo D.

Ejercicio 2:
Implementar un Shift Register unidireccional de 5 bits con Flip-flops tipo D.

Ejercicio 3:
Implementar un Shift Register bidireccional de 4 bits mediante el uso de Flip-flops tipo D y multiplexores de 2 entradas. El comportamiento es el siguiente: cuando en la entrada DIR hay un cero los datos se desplazan hacia la derecha, ingresan por IN_0 y salen por OUT_0; en el caso en que DIR vale 1 los datos se desplazan hacia la izquierda, ingresan por IN_1 y salen por OUT_1.



Ejercicio 4:
Implementar un registro **paralelo** de 4 bits que permita el intercambio (**swapping)** entre el par de bits más significativo y el par menos significativo de salida, utilizando Flip Flops tipo D y multiplexores de 2 entradas. Funcionamiento: Cuando la señal **swap** está activa (‘1’), se intercambian los dos bits más significativos con los dos bits menos significativos. Es decir, si la salida actual del registro es “1110” y **swap** = ‘1’, en el próximo flanco ascendente del **clk** la salida del registro cambiará a “1011”.

Ejercicio 5:
Diseñar un Shift Register de 4 bits (con entradas y salidas de datos en serie y paralelo) con dos señales de control $\mathbf{C_1}$ y $\mathbf{C_0}$ tales que:

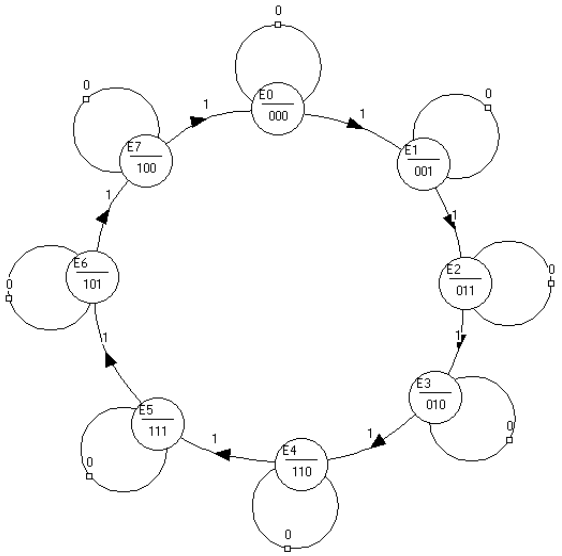
Si $\mathbf{C_1C_0}$ = “00”, el registro pone todas sus salidas a cero (reset).
Si $\mathbf{C_1C_0}$ = “01”, el registro desplaza 1 bit a la derecha.
Si $\mathbf{C_1C_0}$ = “10”, el registro mantiene la información.
Si $\mathbf{C_1C_0}$ = “11”, el registro carga información por su entrada en paralelo.

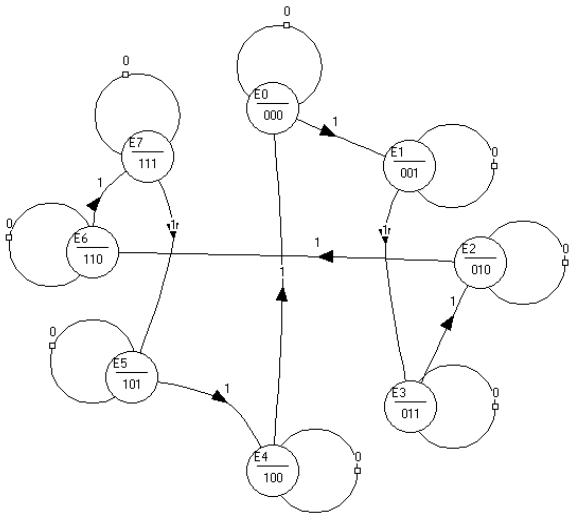
Ejercicio 6:
Implementar un contador de 3 bits de cuenta regresiva (“111” -> “110” -> “101” -> … “000”), con una entrada **R** (reinicio), que lleve el contador al estado “111” en el siguiente ciclo de reloj, si su valor es igual a ‘0’. Utilizar Flip-flops tipo D y las compuertas lógicas necesarias. Tener en cuenta que el contador es cíclico, es decir, que pasa del estado “000” al “111”.

Ejercicio 7:
Diseñar un circuito secuencial de 4 estados “00”, “01”, “10” y “11”, con 2 entradas **E** y **X**. Los valores de salida en cada estado son iguales a la codificación de dicho estado. Funcionamiento: si **E=** ‘0’ el circuito permanece en el mismo estado sin importar el valor de **X**. Cuando **E=** ‘1’ y **X=** ‘1’ el circuito pasa al siguiente estado (de “00” a “01” a “10” a “11” y

de vuelta al “00”). Cuando **E=** ‘1’ y **X=** ‘0’, el circuito vuelve al estado anterior (de “00” a “11” a “10” a “01” y de vuelta al “00”).
Minimizar las ecuaciones en caso de ser posible.
Implementar el circuito utilizando Flip-flops tipo D y las compuertas lógicas necesarias.

Ejercicio 8:
Un código Gray es una secuencia de números binarios con la propiedad de que el salto de un elemento de la secuencia al siguiente es de un solo bit. Por ejemplo, un código Gray binario de 3 bits: 000, 001, 011, 010, 110, 111, 101 y 100.
Utilizando 3 Flip-flops tipo D y compuertas lógicas, construir un contador de código Gray con una entrada **inc** que hace que el contador pase a la próxima secuencia. Notar que el código es cíclico.
Realizar dos implementaciones de dicho contador a partir de los diagramas propuestos en las figuras y luego comparar los resultados obtenidos.





Ejercicio 9:
Diseñar un circuito secuencial que mediante una entrada **inc** produzca la siguiente secuencia de salida: 2, 3, 2, 4, 2, 3, 2, 4…

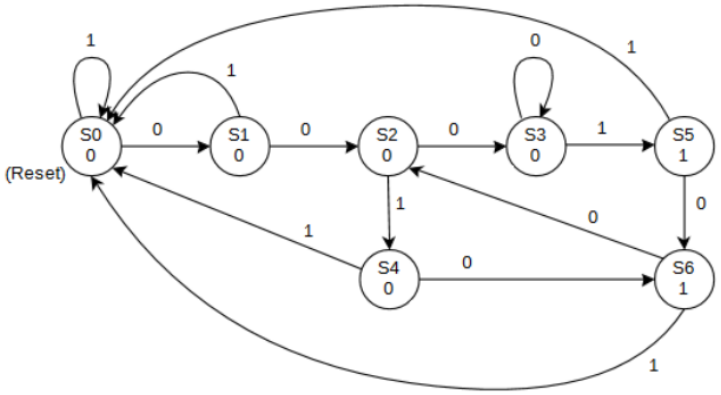
- Señales de entrada: **inc**.
- Señales de salida: $\mathbf{X_2}$, $\mathbf{X_1}$ y $\mathbf{X_0}$ (donde 2= “010”, 3= “011” y 4= “100”).
- Funcionamiento: Si **inc** = ‘0’, la secuencia repite el número que está mostrando y no avanza al próximo estado. Si **inc** = ‘1’, la secuencia avanza normalmente. Por ejemplo: …(**inc** = ‘1’) 2, 3, 2, (**inc** = ‘0’) 2, 2, 2, 2, 2, (**inc** = ’1’) 4, 2, 3, 2, 4, etc…

Ejercicio 10:
Diseñar un circuito secuencial que compruebe la paridad de una señal de entrada (**IN**) de un bit. Funcionamiento: en cada flanco de clk ingresa un nuevo bit, y la salida (**OUT**) pasa a ‘1’ cuando la cantidad de ‘1s’ ingresados desde el inicio de la secuencia es impar, caso contrario es ‘0’.

Ejercicio 11:
Diseñar una máquina de estados que funcione como detector del patrón “1011”. La máquina debería mostrar un ‘1’ como salida cada vez que se encuentra el patrón, y un ‘0’ en caso contrario.
No debe considerarse las superposiciones en la secuencia de entrada, es decir si:
“....1011011….” el output correcto es “....0001000….”.

Ejercicio 12:
El siguiente diagrama de estados representa la implementación de un circuito detector de patrones cuya salida inicia en ‘0’ y toma valor ‘1’ cuando se detecta un patrón válido.

a. Teniendo en cuenta que los patrones pueden solaparse, determinar cuáles son los patrones que detecta el circuito.



b. Considerando que ingresa la siguiente secuencia de entrada (In), determinar la salida (Out) para cada clock.

In:	1	0	0	1	1	0	0	0	1	0	0	1	0	1	1	1	0	1	0	0	1	0
Out:																						

Ejercicio 13:
Diseñar el diagrama de estados de un circuito de monitoreo de una secuencia de bits que se transmite en serie. La información en la secuencia representa distintos códigos de 3 bits.
El código “111” representa que ocurrió un error en el transmisor. La máquina de estados debe monitorear cada secuencia de 3 bits recibida y activar una señal de error “ERR_bar” de 1 bit, **activa por bajo**, si se detecta la secuencia “111”. En cualquier otro caso, la señal de error permanece inactiva.
Diagrama de tiempo de las señales, notar que siempre son paquetes de 3 bits: