

## Práctica I

### Contador Johnson con autocorrección

Un contador Johnson es un contador de desplazamiento (basado en registros de desplazamientos), cuyo módulo (nº de cuentas) es igual al doble de los elementos de memoria que tiene el registro. Este tipo de contadores es muy sensible ya que si partimos de un estado inicial erróneo, o se llega a un estado erróneo (por alguna causa externa al circuito), nunca se volverá a reproducir la secuencia correcta. Por ello, se han ideado una serie de estrategias para solventar la dificultad anterior, dando lugar a contadores con capacidad de autocorrección (capaces de volver a la secuencia correcta por sí solos). Un ejemplo de estos contadores de cinco bits se muestra en la figura 1.1.

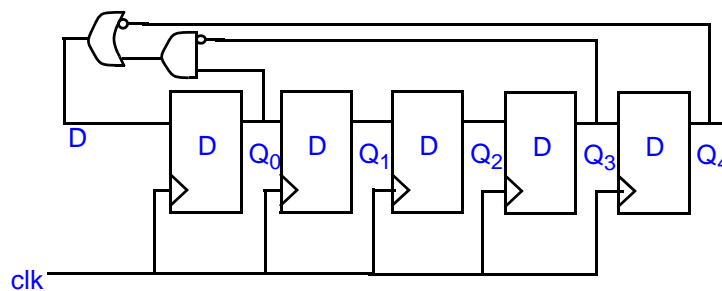


Figura 1.1.- Esquema de un contador Johnson de cinco bits con autocorrección.

En esta práctica se abordará el diseño microelectrónico de dicho contador. Dicho diseño deberá contar con los siguientes pasos:

- Completar el cronograma de la figura 1.2, tomando como valor inicial  $Q \leq "00000"$ .

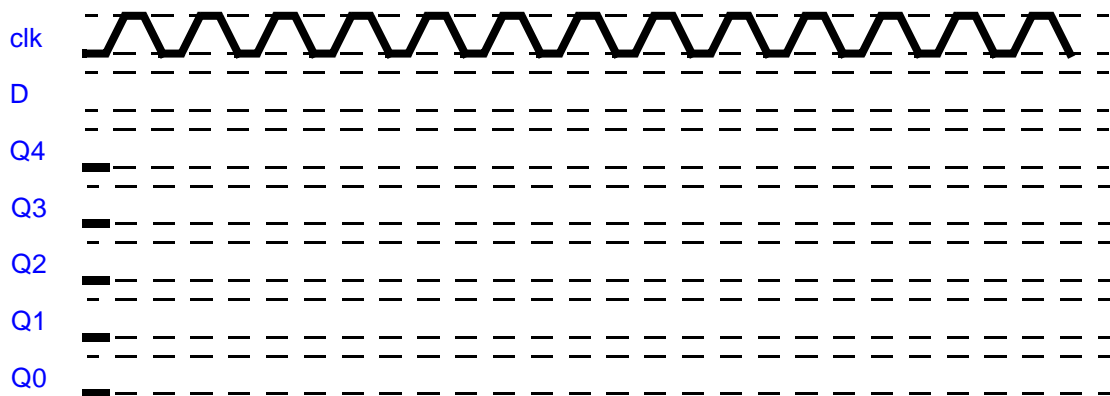


Figura 1.2.- Cronograma de comportamiento del contador Johnson con autocorrección.

- Completar el cronograma de la figura 1.3, tomando como valor inicial  $Q \leq "11001"$ . ¿La secuencia inicial es una combinación correcta del contador?, en caso negativo, ¿el contador se restaura a sus valores correctos?
- Modelar en VHDL el comportamiento del contador Johnson anterior, y mostrar las simulaciones correspondientes a los dos cronogramas anteriores. No olvidéis que para

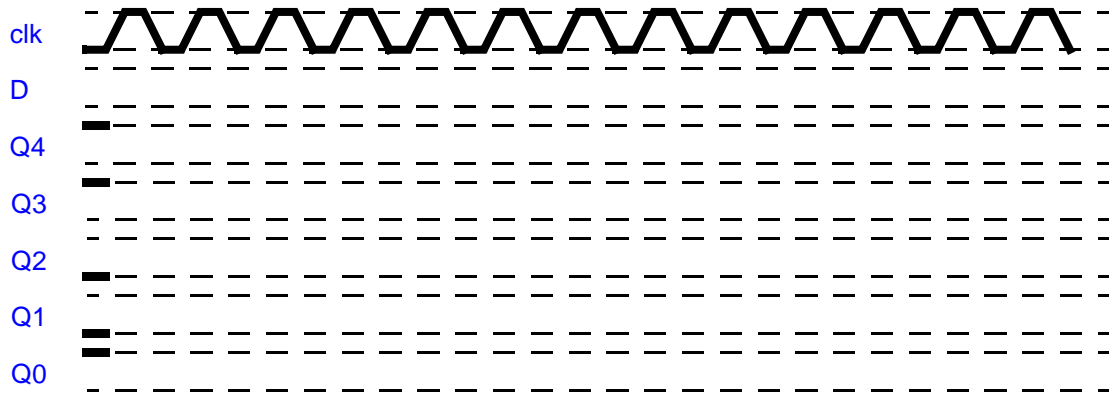


Figura 1.3.- Cronograma de comportamiento del contador Johnson con autocorrección.

el correcto funcionamiento hay que incluir una señal de reset para inicializar el contador, y que no parta de valores desconocidos.

- Realizar la tarea de síntesis con el WebPack de Xilinx e identificar los elementos necesarios para su implementación en el esquemático tecnológico. Indicar número de elementos y sus operaciones correspondientes.

Nota: Los buffers no hay que considerarlos (BUFGP, IBUF y OBUF).

- Implementar la/s puerta/s lógica/s (diferentes) en una familia CMOS. Mostrar el/los esquemáticos a nivel de transistores, así como sus simulaciones eléctricas. Para ello utilizar las siguientes dimensiones nominales (para las conexiones en serie hay que escalar dichas dimensiones, según el número de dichas conexiones):
  - Transistores NMOS:  $W_n = 0.9\mu$  y  $L_n = 0.6\mu$
  - Transistores PMOS:  $W_n = 3.6\mu$  y  $L_n = 0.6\mu$
- Implementar el/los elemento/s de memoria en la configuración TGMS (configuración maestro esclavo con puertas de transmisión). Hay que tener en cuenta que las puertas de transmisión deben tener señales de reset, por lo que dichas puertas tendrán los esquemas mostrados en la figura 1.4. Obtener las simulaciones eléctricas de dicho/s elementos. Para ello utilizar las dimensiones del caso anterior.

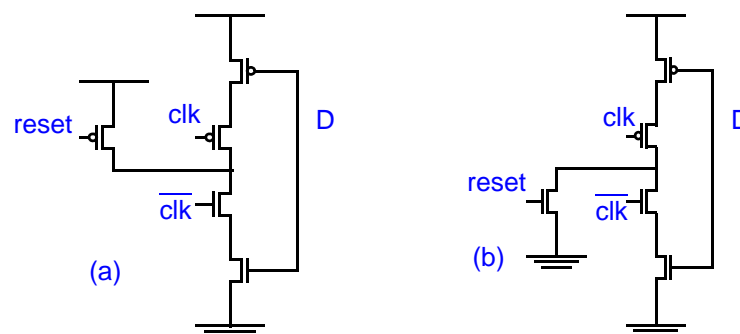


Figura 1.4.- Puertas de transmisión con reset asíncrono. (a) Pone en la salida un '1', y (b) pone en la salida un '0'.

- Obtener los layouts de las puertas y elementos de memoria de los dos puntos anteriores, utilizando el software de MICROWIND y las dimensiones indicadas anteriormente. Mostrar dichos layouts, así como las simulaciones de ellos.

- Obtener una colocación apropiada utilizando el algoritmo de clustering uniforme. Mostrar el esquema con la colocación anterior.
- Obtener el rutado utilizando el algoritmo del laberinto. Mostrar un esquema de dicho rutado, así como el número y tamaño de los diferentes canales que son necesarios.
- Utilizar la colocación y rutado anterior para generar el layout del contador completo. Mostrar el layout global así como su simulación.