



ARQUITECTURA DE COMPUTADORAS

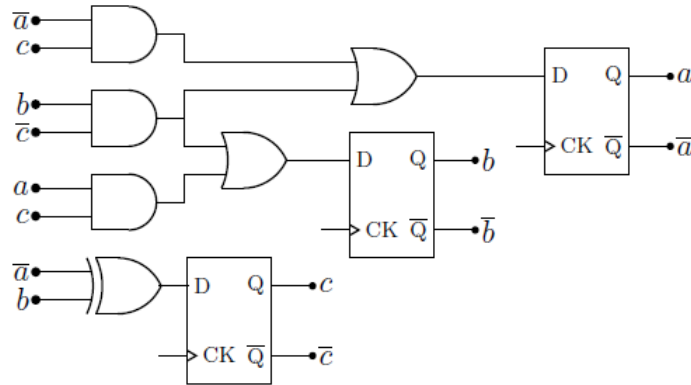
Trabajo Práctico N° 3
Circuitos Secuenciales¹
Primer Cuatrimestre de 2018

Ejercicios

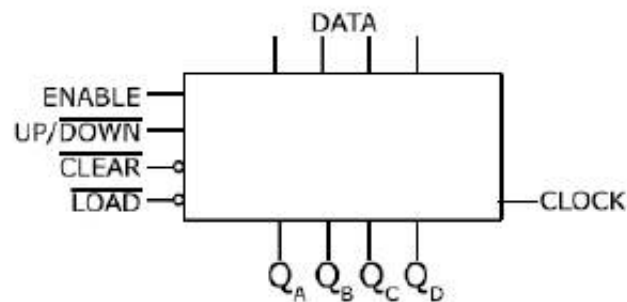
1. Emular los siguientes dispositivos:
 - a) Un flip-flop S-R a partir de un flip-flop D.
 - b) Un flip-flop J-K a partir de un flip-flop D.
 - c) Un flip-flop D a partir de un flip-flop T.
 - d) Un flip-flop J-K a partir de un flip-flop T.
 - e) Un flip-flop D a partir de un flip-flop J-K.
2. Implementar un contador binario sincrónico módulo 10 (esto es, de 0 a 9) a partir de los siguientes componentes:
 - a) Con flip-flops individuales tipo J-K, controlando sus entradas.
 - b) Con un contador de cuatro etapas *up-down* con carga en paralelo.
3. Un *contador Moebius* (esto es, un contador que que modifica un solo bit por cada cambio de estado), puede ser fácilmente implementado usando un registro de desplazamiento. Teniendo esto en cuenta:
 - a) Bosquejar un *contador Moebius* módulo seis a partir de un registro de desplazamiento implementado con flip-flops J-K.
 - b) Desarrollar las seis señales periódicas de salida.
 - c) Verificar que dividen el período de partida en seis intervalos iguales y que cada señal tiene un *duty cycle* de exactamente 50 %.
OBS: En un tren de pulsos ideal (uno compuesto de pulsos rectangulares), el *duty cycle* es la razón entre la duración del pulso y el período del mismo. Por caso, para un tren de pulsos en el que la duración del pulso es de $1\mu s$ y el período es de $4\mu s$, el *duty cycle* resulta ser del 25 %.
 - d) Indicar como se decodifican los estados 001, 110, 000 y 111.
 - e) Suponiendo que la transición del estado 011 al estado 001 recién deba realizarse cuando aparezca un 1 en una cierta señal *e* (esto es, *e* funciona como señal de *enable* para esa transición), ¿cómo hay que modificar al contador para que contemple a la señal *e*?

¹Fecha sugerida de finalización de este trabajo práctico: jueves 5 de abril de 2018.

- f) Modificar el control sobre la transición entre estados propuesta en el inciso anterior para que la señal e ahora regule el paso del estado 110 al estado 111.
4. Dada la siguiente mecanización, utilizando flip-flops D, encontrar la secuencia de salida que se produce partiendo de un estado inicial 000.

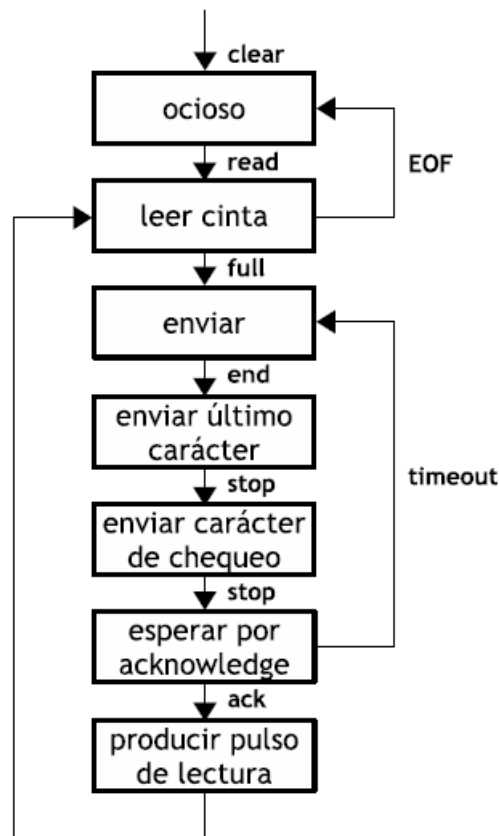


5. Utilizando compuertas y un contador up/down de 4 bits con carga paralela (con entradas enable y clear asincrónicas):
- Diseñar un contador binario módulo 7 (esto es, que siga la secuencia 0,1,2,3,4,5,6).
 - Modificar el circuito anterior para que dada una señal externa x_1 , pase del estado 2 al estado 6, del estado 5 al 0 y del estado 1 al 4 cuando $x_1 = 1$. En cualquier otro caso, debe pasar al estado siguiente módulo 7.



6. Usando un contador binario *up-down* con carga paralela, implementar los estados de control para un sistema que luego de leer registros de datos desde una cinta magnética en un buffer, los envíe a través de una línea de transmisión de datos, junto con un carácter de chequeo. Entre la transmisión de un registro y otro, debe esperar por una señal de recepción correcta **ack**. Si no se recibe esta señal dentro de un cierto tiempo **timeout**, el mensaje es repetido. Este ciclo se repite indefinidamente, hasta que se recibe el **ack**.

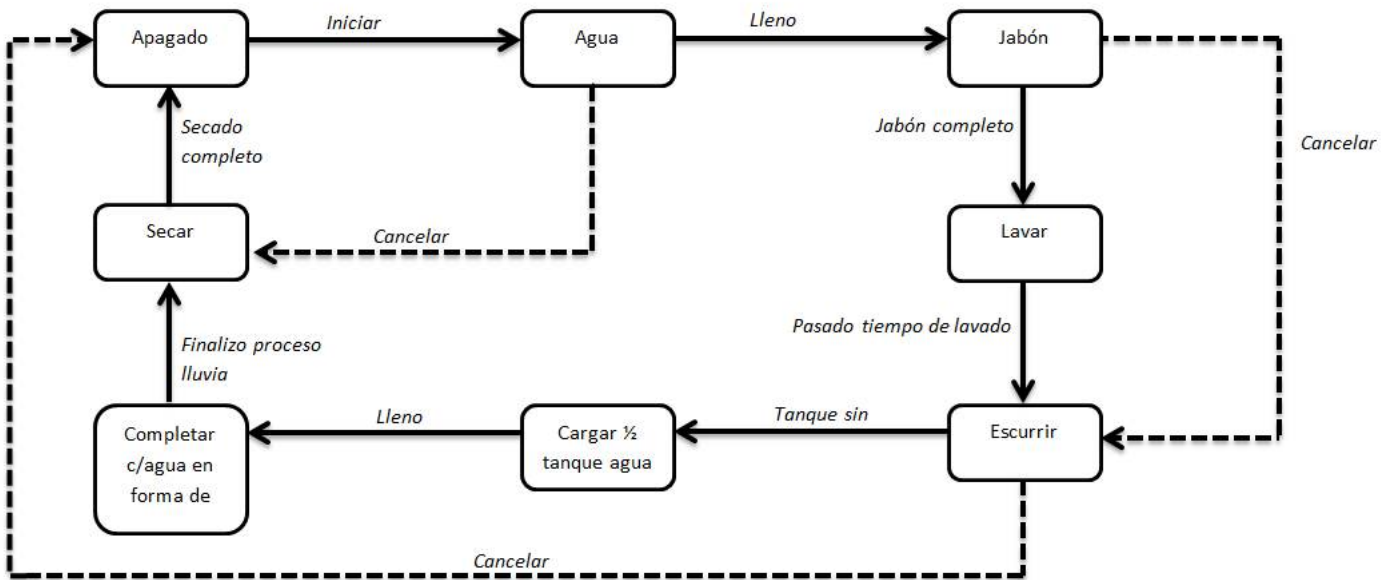
El diagrama de estados correspondiente al sistema es el siguiente:



A partir del estado inicial (**ocioso**), ante una señal de lectura (**read**) se alcanza el estado **leer cinta**. Por otro lado, si se detecta fin de archivo (**EOF**), denotando que no hay más información en la cinta, se retorna al estado **ocioso**. Cuando el buffer está lleno (**full**) con los datos obtenidos de la cinta, se pasa al estado de **enviar**, y se transmite la información hasta que la señal **end** indique que se ha llegado al último dato en el buffer. En ese momento se avanza al estado de **enviar último carácter**. Cuando se ha enviado el último bit del último dato, un bit de **stop** pone al sistema en condiciones de **enviar carácter de chequeo**. Luego, cuando se ha enviado el último bit del código de chequeo, una nueva señal de **stop** provoca que el sistema avance a **esperar por ack**. Si se recibe un **ack**, se entra en el estado de **producir un pulso de lectura**, el cual, con un pulso de reloj, genera una señal de **read** que causa que se vuelva a leer otro registro desde la cinta, en el estado **leer cinta**.

Esta secuencia se repite hasta que todos los registros son enviados, y una señal de **EOF** retorna al sistema a estado **ocioso**. Si no se recibe un **ack** dentro de $\frac{1}{3}$ de segundo de haber entrado en el estado **esperar por ack**, la señal **timeout** hace que el registro sea retransmitido.

7. Usando un contador binario up-down, implementar los estados de control de una *máquina lavavajillas*.



La máquina debe realizar un programa de lavado el cual inicia en un estado **Apagado**. Al recibir la señal *iniciar* pasa al estado **Agua** en el cual por medio de un proceso comienza a llenarse el tambor con agua. Cuando se llega al nivel de agua necesaria, se pasa al estado **Jabón** donde se procede a volcar el jabón, una vez completa esa etapa se pasa al estado **Lavar** en el cual se realiza el proceso de lavado. Una vez finalizado, se debe escurrir el contenido del tambor. Al recibir la señal indicando que se escurrió el tambor completo se pasa a llenar con agua, hasta recibir la señal *Lleno* para pasar al estado lluvia donde se continúa cargando agua pero en forma de lluvia. Luego y se vuelve a vaciar el tambor y echar aire caliente en el estado **Secar**.

El proceso puede cancelarse por el usuario al presionar un botón en los estados indicados en el gráfico.

8. Utilizando PLA y un contador up/down de 4 bits con carga paralela diseñar un contador binario módulo 6 (0...5) para que dada una señal externa y_1 pase del estado 0 al 3, en caso de encontrarse en los estados 2, 4 o 5 pase al 1; cuando $y_1 = 1$.

La función y_1 dependerá de 4 entradas A, B, C y D según la siguiente especificación.

$$y_1(A; B; C; D) = \sum(1; 2; 3; 8; 11), \sum_{op}(9; 13; 14)$$

| Estado actual | y_1 | Siguiente estado |
|---------------|-------|------------------|
| 0 | 1 | 3 |
| 2 | 1 | 0 |
| 4 | 1 | 0 |
| 5 | 1 | 2 |

9. Utilizando MUX y un contador up/down de 4 bits con carga paralela diseñar un contador binario módulo 7 (0..6), en caso que la función $f_1 = 1$ retorne al estado 4.

$$f_1(A; B; C; D) = \sum(0; 1; 5; 8; 10; 13), \sum_{op}(9; 15)$$

10. -*Opcional*- Dada la siguiente tabla de estados de un Ring Counter.

| Estado Actual | Siguiente Estado |
|---------------|------------------|
| ABC | ABC |
| 000 | 100 |
| 100 | 110 |
| 110 | 111 |
| 111 | 011 |
| 011 | 001 |
| 001 | 000 |

- a) Diseñar el circuito con flip-flop D, asumiendo que los estados que no se incluyen en la tabla son don't care.
 - b) Utilizando compuertas, agregar la lógica necesaria al circuito para inicializarlo en el estado 000 cuando se encienda.
 - c) Diseñar el circuito con un contador Up/Down.
11. -*Opcional*- Un FF *set-dominant* tiene entradas set y reset. Difiere de el convencional FF SR en que, cuando S y R son 1 el FF se setea en 1.
- a) Obtener la tabla de estados para este FF.
 - b) Diseñar el FF set-dominant usando un FF JK y compuertas.
12. Un FF JN tiene dos entradas J y N. La entrada J se mantiene como la entrada J de un FF- JK, y la entrada N es el complemento de la entrada K de un FF-JK.
- a) Obtener la tabla de estados para este FF.
 - b) Demostrar que conectando las dos entradas J y N juntas se obtiene un FF-D.

Referencias

- [Bla79] BLAKESLEE, T. R. *Digital Design with Standard MSI and LSI*, second ed. John Wiley & Sons, 1979.
- [Mar76] MARCUS, M. P. *Switching Circuits for Engineers*, third ed. Prentice-Hall, 1976.