



IIC2343 – Arquitectura de Computadores (II/2018)

Tarea 3 - Pauta

Fecha de entrega: lunes 3 de septiembre de 2018 a las 11:59 AM

Rúbrica: Disponible en [syllabus/Pautas/RúbricaGeneral.pdf](#)

Parte programación

Para la forma A y B se hizo uso de los mismos *tests*, los que corresponden a:

1. **Test 1:** Un solo *flip-flop*, manejo total desde el *input* (`test1.json`).
2. **Test 2:** Un solo *flip-flop*, *output* igual a la negación del estado de este (`test2.json`).
3. **Test 3:** Un solo *flip-flop*, pero va alternando su valor al recibir la negación de su estado. Corresponde al ejemplo del enunciado (`test3.json`).
4. **Test 4:** Igual al anterior, pero cambia su estado a partir de una compuerta XOR (`test4.json`).
5. **Test 5:** Un solo *flip-flop* que recibe como dato un NAND a través de una compuerta AND + NOT, ya que no existe `nand` en la nomenclatura utilizada (`test5.json`).
6. **Test 6:** Igual al anterior, pero haciendo uso de un NOR (`test6.json`).
7. **Test 7:** Dos *flip-flops* en serie, para que la salida del segundo vaya cambiando a la mitad de la frecuencia (`test7.json`).
8. **Test 8:** Dos *flip-flops* en paralelo que conectan sus estados a una compuerta final para obtener un resultado (`test8.json`).

Forma A. En este caso, se aceptan varias opciones, entre ellas: **(1)** Que consideren todo estado proveniente de un *flip-flop* como *input*, ya que el valor de este es el único que tiene poder sobre el *output* del circuito, sin importar la composición interna de estas componentes. **(2)** Hacer uso de una versión combinacional del *flip-flop*, haciendo que la señal **C** funcione como control para una iteración dada. **(3)** Otras alternativas, siguiendo supuestos que no vayan en contra del enunciado y que estén estipulados en el **README**.

Forma B. Notará que todos los *tests* hicieron uso del *clock* con constantes flancos de subida y bajada. Esto, con el objetivo de poder ver la variación del circuito en cada iteración. En la evaluación se considera, además de la correctitud de las salidas, la inclusión tanto de los valores de estas como de los estados de los *flip-flops*.

Parte práctica

Primero, se revisa que se cumplan las conexiones pedidas. Especialmente, que se cumpla lo pedido en la sección “Descripción del componente general”. A continuación, los *tests* realizados:

1. **Test 1:** Revisar que al ingresar $D = 1$ al FFD, Q cambia a 1 al haber flanco de subida del *clock*.
2. **Test 2:** Revisar que al ingresar $D = 0$ al FFD, Q cambia a 0 al haber un flanco de subida del *clock*.
3. **Test 3:** Revisar que al ingresar $R = 0, S = 1$ al FFSR, Q cambia a 1 al haber flanco de subida del *clock*.
4. **Test 4:** Revisar que al ingresar $R = 1, S = 0$ al FFSR, Q cambia a 0 al haber flanco de subida del *clock*.
5. **Test 5:** Revisar que al ingresar $R = 1, S = 1$ al FFSR, Q mantiene su valor.
6. **Test 6:** Revisar que al ingresar $J = 1, K = 0$ al FFJK, Q cambia a 1 al haber flanco de subida del *clock* y que al ingresar $J = 0, K = 1$ al FFJK, Q cambia a 0 al haber flanco de subida del *clock*.
7. **Test 7:** Revisar que al ingresar $J = K$ (ya sea 0 o 1), el valor de Q se mantiene.
8. **Test 8:** Revisar que en el caso serial, los *leds* tengan diferentes frecuencias de encendido y pagado (esto sucede porque conectar FFD en serie hace un contador que aumenta con cada flanco de subida del *clock*).