



Arquitectura de Computadores 2023-II (CCOMP3-1)

Laboratorio 01b Lógica Digital: Circuitos Digitales Secuenciales

Yván Jesús Túpac Valdivia
Universidad Católica San Pablo, Arequipa – Perú
05 de setiembre de 2023

1 Introducción

1.1 Objetivos

- Verificar el funcionamiento de los principales bloques secuenciales, que pueden incluir componentes combinatoriales, usados en Sistemas de Computación, mediante simulación en CircuitVerse.

1.2 Material a usar

- Diapositivas de clase *Electrónica Digital: Logica Digital Secuencial*
- Libro Guía *Computer Architecture: From Microprocessors to Supercomputers* [Parhami, 2005]
- Simulador CircuitVerse, disponible en <https://circuitverse.org>

2 Circuitos de utilidad para implementar y analizar

2.1 Latch y Flip-Flop tipo D

[6pt, SO1]

- En la Figura 1 se observan dos circuitos, el de la derecha es un flip-flop tipo D y el de la izquierda es un Latch tipo D, cuyo funcionamiento fue explicado en clase y también se puede revisar en el material de clases.
- Se pide implementar en CircuitVerse estos dos circuitos en el mismo proyecto (uno al lado del otro)

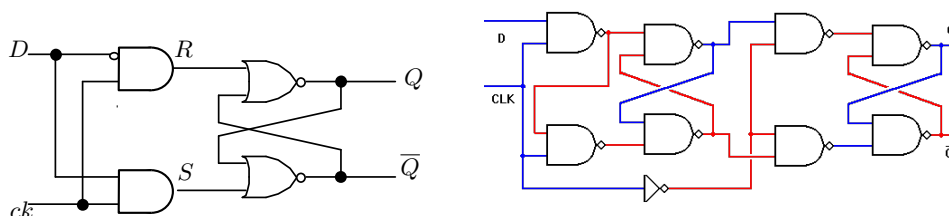


Figure 1: Izquierda: *Latch* tipo D, Derecha: *flip-flop* tipo D

- Conecte ambos circuitos al mismo reloj ck y misma entrada D . Esta entrada puede ser un botón para poder cambiarlo fácilmente de 0 a 1 y 1 a 0 usando el mouse.
- Para facilitar la visualización del comportamiento coloque Leds en las salidas Q y \bar{Q} de ambos circuitos.
- Configure el reloj a una frecuencia bastante baja (1 Hz o menos), tal que permita fácilmente observar el comportamiento del *latch* y *flip-flop* durante los diversos estados del reloj (intervalo con reloj en alto, en bajo y en los flancos de subida y bajada).

- Utilizando el botón como una entrada “asíncrona”, verifique lo siguiente:
 - i Comportamiento del *latch* y del *flip flop* durante el periodo bajo y alto del reloj ante entradas D .
 - ii En qué flanco (subida o bajada) el *flip flop* implementado “observa” el valor de D .
 - iii ¿Qué debería hacerse para que el *flip flop* observe el valor D en el otro flanco del reloj?
- Esta implementación y verificación deberá entregarse en el informe como link al proyecto.

2.2 Diseño de un conversor serial-paralelo de 8 bits

[6pt, SO6]

- Implemente en CircuitVerse el conversor de paralelo-serie para palabras de 8 bits de la Figura 2. Note que deberá usar el componente splitter de CircuitVerse para separar y juntar bits.

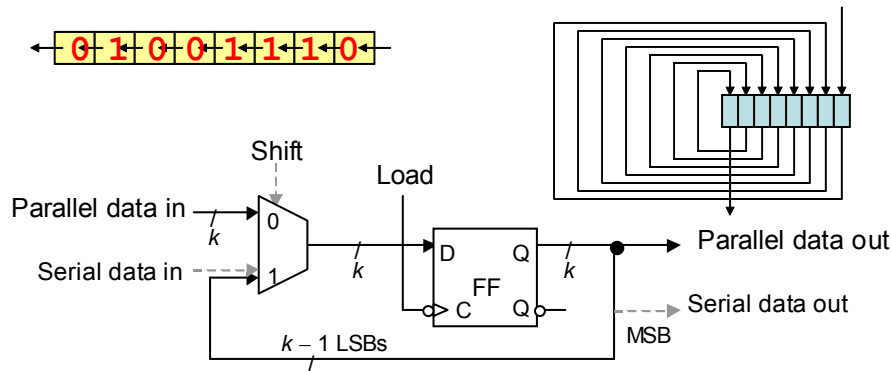


Figure 2: Conversion serie-paralelo

- Coloque los elementos de visualización más convenientes para poder observar entradas en serie/paralelo y salidas en serie/paralelo.
- Extienda este conversor implementando un contador de los bits seriales entrantes o salientes y permita “avisar” cuando se completó una entrada o salida serie. Esto implica definir un contador de 0 a $n - 1$, para $n = 8$ que, cuando llegue al valor $n - 1$ indique que se completó la carga o lectura, y debe esperar dos pulsos y luego de esos dos pulsos, retornar a 0 e iniciar una nueva carga o recarga de bits.
- Use el diagrama de tiempos de la figura 3 como referencia.

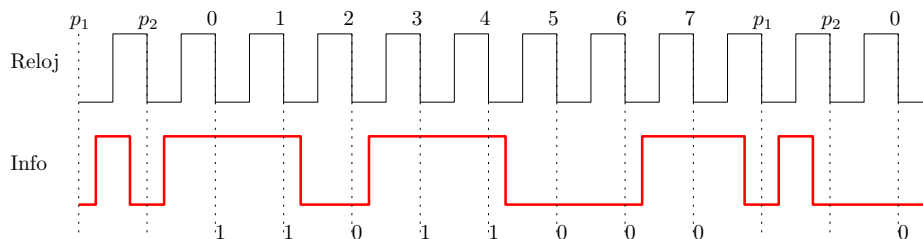


Figure 3: Ejemplo de transmisión serial

2.3 Contador de n -bits asíncrono

[8pt, SO1, SO6]

- Diseñe e implemente en CircuitVerse un contador asíncrono de n -bits, el cual consiste en una serie de *flip-flops* conectados en cascada donde el primer *flip flop* (FF_0) recibe la señal de reloj en su entrada ck_0 , su salida \bar{Q}_0 se conecta a la entrada D_0 y alguna de sus salidas \bar{Q}_0 o Q_0 se conecta a la entrada de reloj ck_1 del siguiente *flip-flop*, y conectar así sucesivamente hasta completar $n - 1$ *flip-flops*. Un ejemplo de esta conexión se muestra en la Figura 4
- Complete un contador con 4 *flip-flops*. Conecte las salidas Q_i a un decodificador de 7 segmentos para poder visualizar el valor de conteo.

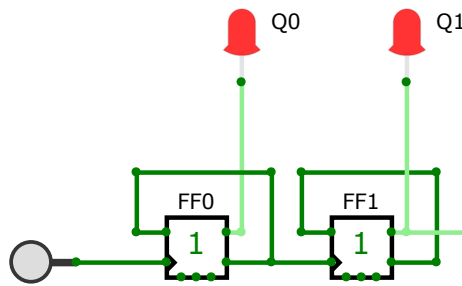


Figure 4: Ejemplo de contador asíncrono con 2 *flip-flops* tipo *D*

- Verifique qué ocurre con el conteo si las conexiones entre *flip-flops* sucesivos se hacen desde \bar{Q}_{i-1} hacia ck_i , o se hacen desde Q_{i-1} hacia ck_i .
- Los *flip-flop* tipo *D* suelen contar con entradas asíncronas, que hacen $Q_i = 0$ cuando esta entrada está en nivel alto, independiente del valor de ck . En CircuitVerse el *flip-flop* tipo *D* tiene la entrada *Asynchronous Reset*. Si ponemos nivel alto en todas las entradas *Asynchronous Reset*, todas las salidas Q_i harían un 0000, sirviendo como una forma de “resetear” el contador.
- Se puede hacer una lógica que mande resetear los *flip-flops* ante un valor determinado, por ejemplo 1010, haciendo que el contador vaya de 0000 a 1001 y cuando llegue a 1010, inmediatamente cambiará a 0000, obteniendo así un contador de 0 a 9.
- Implemente la lógica combinacional que permita obtener un contador de 0 a 9 y colóquela en el circuito en CircuitVerse
- Vea cómo podría hacerse para que, mediante un *bit* de control, se pueda cambiar el sentido del conteo (ascendente \leftrightarrow descendente). ¿Qué circuito podría ayudar a implementar este cambio? Implemente esa nueva opción.

3 Entrega de Informe

Se debe entregar un informe mediante el Aula Virtual (archivo PDF de preferencia, incluso si son fotografías de hojas trabajadas en clase) conteniendo por cada pregunta:

- El desarrollo de la pregunta.
- Link al proyecto en CircuitVerse de las implementaciones realizadas, tener cuidado de no enviar el *link* desde el editor del circuito, sino desde la página del proyecto: Project Page en el Menu Project.
- Explicación de las pruebas realizadas, si es que se solicitó.
- Revise el plazo en el Aula Virtual para no entregar de forma tardía.

References

[Parhami, 2005] Parhami, B. (2005). *Computer Architecture: From Microprocessors to Supercomputers*. The Oxford Series in Electrical and Computer Engineering. OUP USA.