Ayudantía 12: Verilog secuencial

Guido Muñoz (gtmunoz1@uc.cl)

17 de junio de 2020

Objetivos

1. Aprender comandos ocupados en bloques secuenciales.
2. Entender como codificar circuitos secuenciales en Verilog.

Problema 1 [A,A]

Escriba en verilog un módulo de flip-flop tipo D activado en flanco positivo, con capacidad de enable y con entrada reset asíncrono.

Problema 2[A]

Implemente un contador que siga la secuencia 0,3,2,1 en verilog.

Problema 3[C]

Un circuito secuencial tiene dos entradas, w_1 y w_2, y una salida, z. Este circuito compara ambas entradas, si durante cuatro ciclos cualquiera de clock $w_1 = w_2$, entonces $z = 1$, de forma contraria, $z = 0$. Los ciclos de reloj pueden traslapar entre sí para el propósito de comparar ambas entradas. Ejemplo:

$$w_1 : 0110111000110$$

$$w_2 : 1110101000111$$

$$z \ : 0000100001110$$

Dibuje la FSM para el circuito con el método que estime más conveniente, y codifíquela en verilog. No olvide considerar que w_1 y w_2 entran de manera serial, es decir, entran bit a bit.

Problema 4[C,A]

Derive un circuito que actue como un generador de bit de paridad para una secuencia de 3 bits. Cada 3 bits que sean observados en 3 ciclos de clock en una entrada w, la salida p generará el bit tal que la secuencia ahora de 4 bits sea par. Por ejemplo, si a w entra la secuencia 100 entonces $p = 1$.

Luego de derivar este circuito codifiquelo en verilog.

Desafio no tan largo[C]

Cree testbenches para los circuitos anteriores y verifique su funcionamiento.

Desafío largo [C]

El protocolo SPI (el cual enfrentarán durante SEP) es un protocolo de comunicaciones entre microcontroladores que sigue las siguientes reglas:

- Existe una señal de clock que, cuando no se envian datos, se mantiene en 1, llámela `SPI-CLK`. Al enviarse datos esta señal se activa a una frecuencia considerablemente menor al clock interno de los circuitos. Se activa funcionando como un clock, de forma que en los flancos negativos se guarda el dato a comunicarse.

- Existen dos señales, llamadas `MOSI` y `MISO` (master output slave input, y master input slave output respectivamente) que se encargan de la comunicacion entre un dispositivo llamado master (tipicamente un microcontrolador que se comunica con varios perifericos) y otro dispositivo llamado slave (tipicamente un dispositivo a ser controlado a momentos por el microcontrolador, en SEP, por ejemplo, este dispositivo corresponde a una pantalla y un lector de tarjeta SD).

- Existe una cuarta y ultima señal, llamada `SS`, (slave select), que elije el slave al que se comunica el master. Para este desafio puede ignorar esta señal, y asumir el master solo se comunica con un slave.

Diseñe un multiplicador que opere con el protocolo SPI. Este multiplicador recibe dos números de 8 bits de manera secuencial, es decir, este multiplicador recibe 16 bits de manera serial, y separa los primeros 8 bits de los siguientes 8 bits. Luego multiplica estos bits entre sí, y entrega una salida de 16 bits con el resultado. Por último cree una testbench para verificar el funcionamiento de este circuito. Para esto puede asumir que existe solo una linea de comunicación, e ignore la señal de `SS`, por lo que el SPI solo queda reducido a dos señales.

Tips: Este no es un desafio trivial, ¡De hecho es una adaptación de una tarea de un curso más avanzado que yo tuve que hacer! Se recomienda fuertemente partir haciendo un diagrama de bloques, luego hacer módulos correspondientes a cada cuadro de este diagrama de bloques e instanciar todo en un módulo top (Este consejo es extrapolable a cualquier implementación que realice en verilog). Se recomienda tener en mente que debe incorporar un registro de desplazamiento en la entrada.

Es un desafio largo, ¡Pero le ayudará mucho a entender como programar en Verilog!

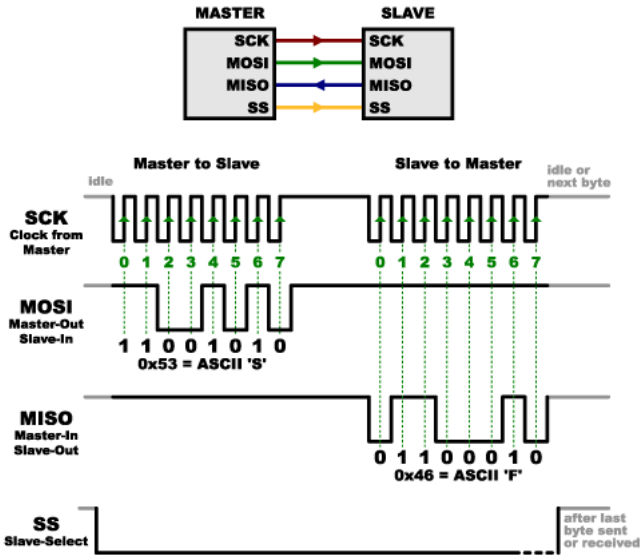


Figura 1: Operación del protocolo SPI entre dos dispositivos.