

Laboratorio 2: Taller de Diseño Digital Tecnológico de Costa Rica

Bryan Esquivel Flores
2020035806
zayus@estudiantec.cr
Alajuela, Costa Rica

Freddy Mora Bolaños
2021040443
freddy.mora@estudiantec.cr
Alajuela, Costa Rica

Andrés Vargas Arce
2018151379
anbo80@estudiantec.cr
Alajuela, Costa Rica

October 2, 2024

1 Ejercicio 1

Se diseñaron los bloques en SystemVerilog para eliminar los rebotes y sincronizar las entradas de pulsadores e interruptores. El sistema incluye un contador sincronizado con el reloj, utilizando una señal de habilitación activa en alto (EN) y un reset activo en bajo (rst). La señal EN, que proviene de un pulsador, debe ser filtrada por un bloque de anti-rebote y sincronización para evitar conteos erróneos.

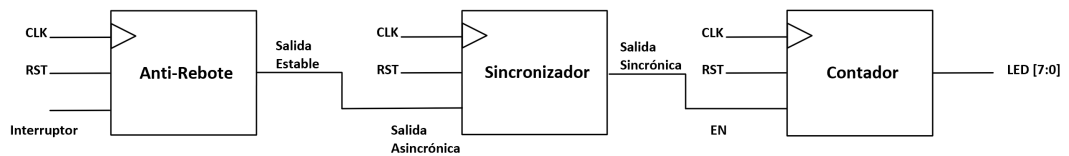


Figura 1: Diagrama de bloques.

1.1 Diagrama de flujo

Se realizó el diagrama de flujo del módulo antirrebote para comprender de una forma más clara el funcionamiento implementado en el diseño.

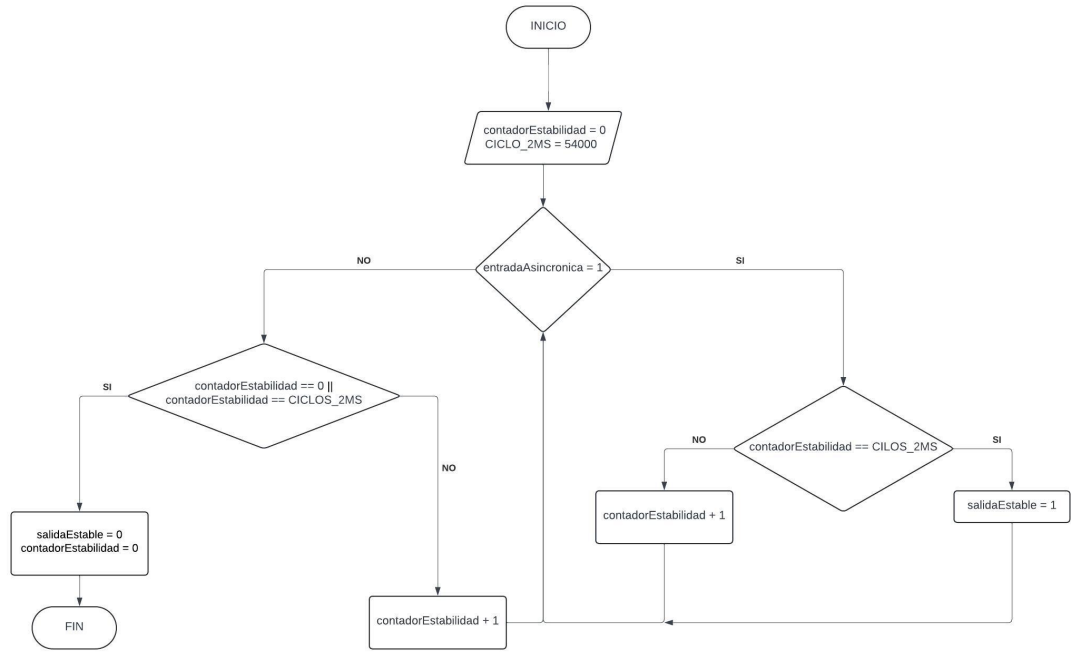


Figura 2: Diagrama de flujo del módulo antirrebote.

Se muestran los resultados obtenidos en la simulación del módulo antirrebote-sincronizador y el módulo top que incluye el contador de pruebas. Además, el código utilizado para los módulos y el TB se pueden encontrar en Repositorio.

1.2 Resultados del funcionamiento

Se realizó la simulación del módulo antirrebote sincronizador para verificar la eliminación de rebotes y la sincronización de reloj con la FPGA, producidos por un interruptor o pulsador.

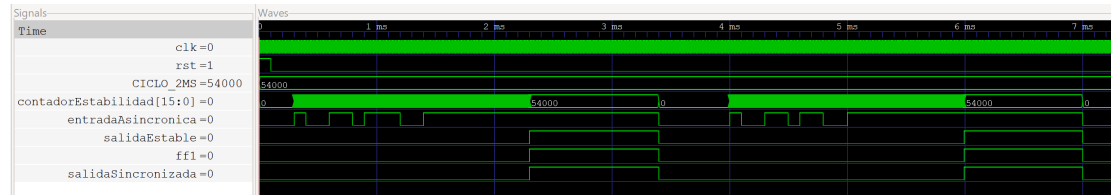


Figura 3: Simulación módulo antirrebote sincronizador.

Debido a la configuración de los LEDs de la FPGA, donde encienden en bajo por tener el ánodo fijo en Vcc, se tuvo que invertir la salida final del contador

de pruebas en la simulación final del diseño y disminuir la salida de los LEDs a 6 bits.

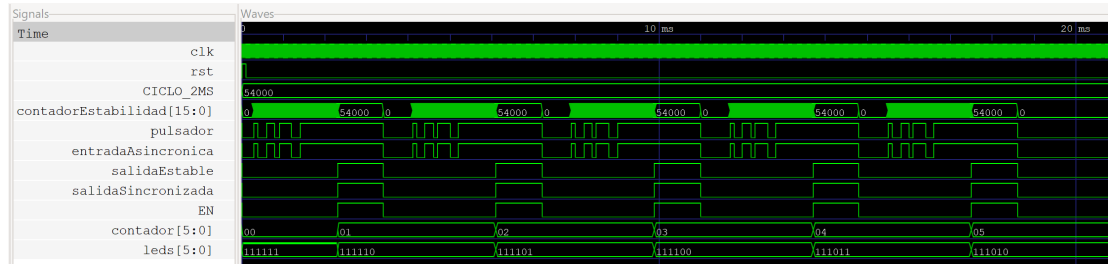


Figura 4: Simulación módulo top.

2 Ejercicio 2

Este ejercicio consiste en realizar la parte final de la interfaz de comunicación del teclado con nuestros pc's, para ello el primer módulo que se construirá será el responsable de la eliminación del rebote de teclas.

2.1 Key bounce elimination

Este módulo se encarga de verificar si la tecla ha sido presionada realmente o si ha sido algún rebote o error causado por la mecánica del circuito.

Para implementar este bloque se debe dividir el reloj con el que funciona la Fpga, este por defecto oscila **27Mhz**, cada período es de $37ns$. Se considera que 2ms es suficiente para que el mecanismo del teclado deje de oscilar, por lo tanto se debe esperar que pasen 27000 ciclos de reloj y luego se envían las señales una vez asegurada la señal.

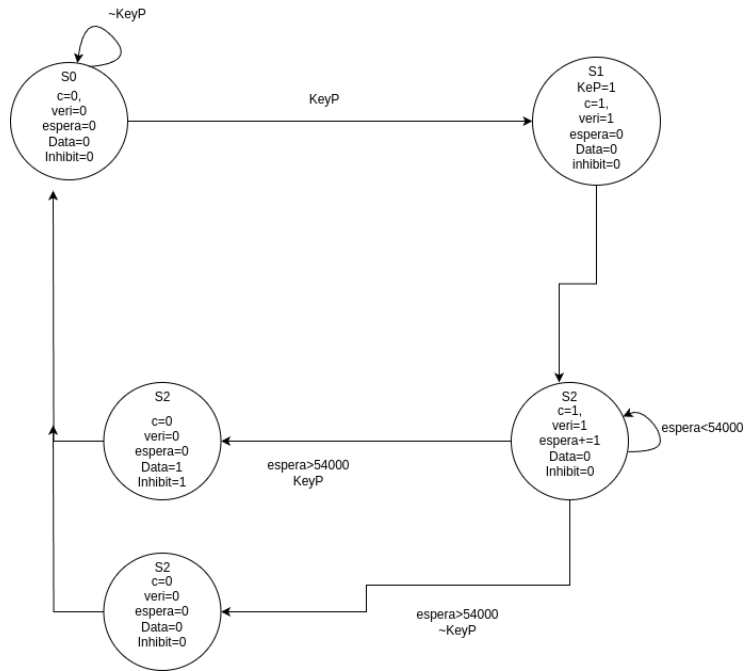


Figura 5: Máquina de estados del Key bounce elimination.

2.2 Contador

Un contador de 2-bit que suma 1 cada 1ms y se desactiva cuando la señal **inhibit=1**, este contador nos mostrará la columna presionada en el teclado.

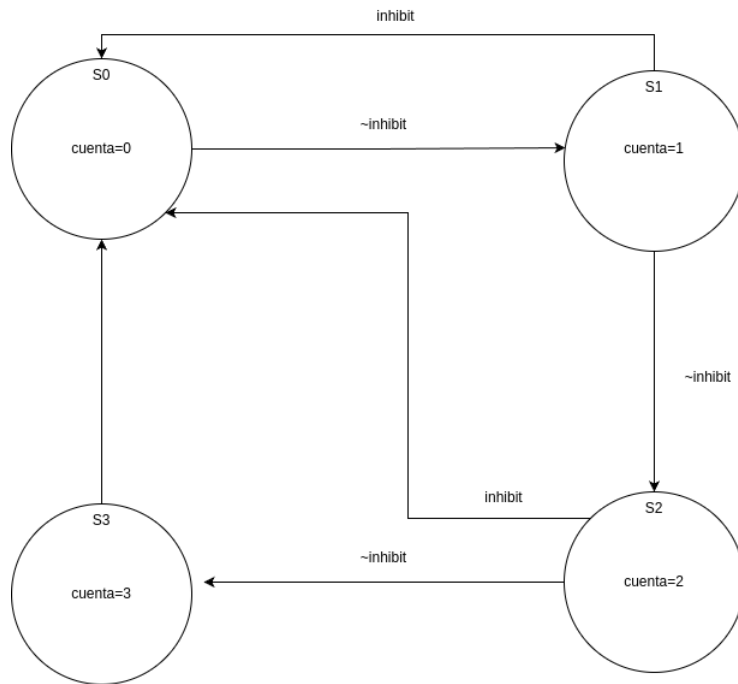


Figura 6: Máquina de estados del contador de 2-bits.

2.3 Sincronizador

El sincronizador cuenta con tres D flip-flops, este se encarga de recibir la señal **Data available** y sincronizar las salidas de las columnas y las filas del teclado.

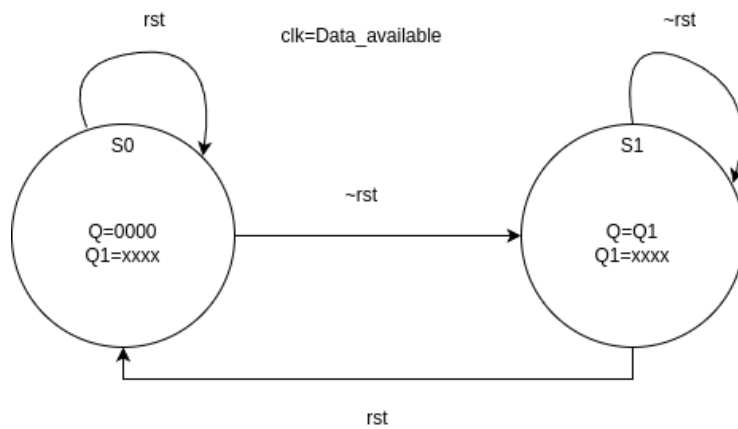


Figura 7: Máquina de estados del sincronizador.

2.4 Interfaz completa

El diagrama de funcionamiento de la interfaz completa sería el siguiente:

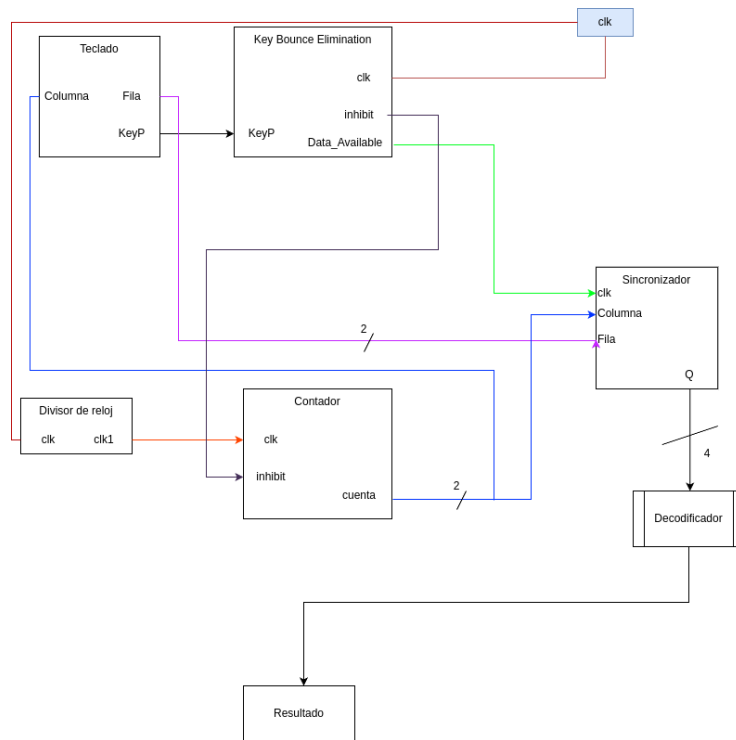


Figura 8: Diagrama final de todos los módulos.

3 Ejercicio 3

Para la UART se tomó el código dado por Lushaylabs, se modificó para que pudiera utilizar las señales enviadas por el teclado, su funcionamiento se basa en un máquina de estado, presentada a continuación:

Máquina de Estados UART

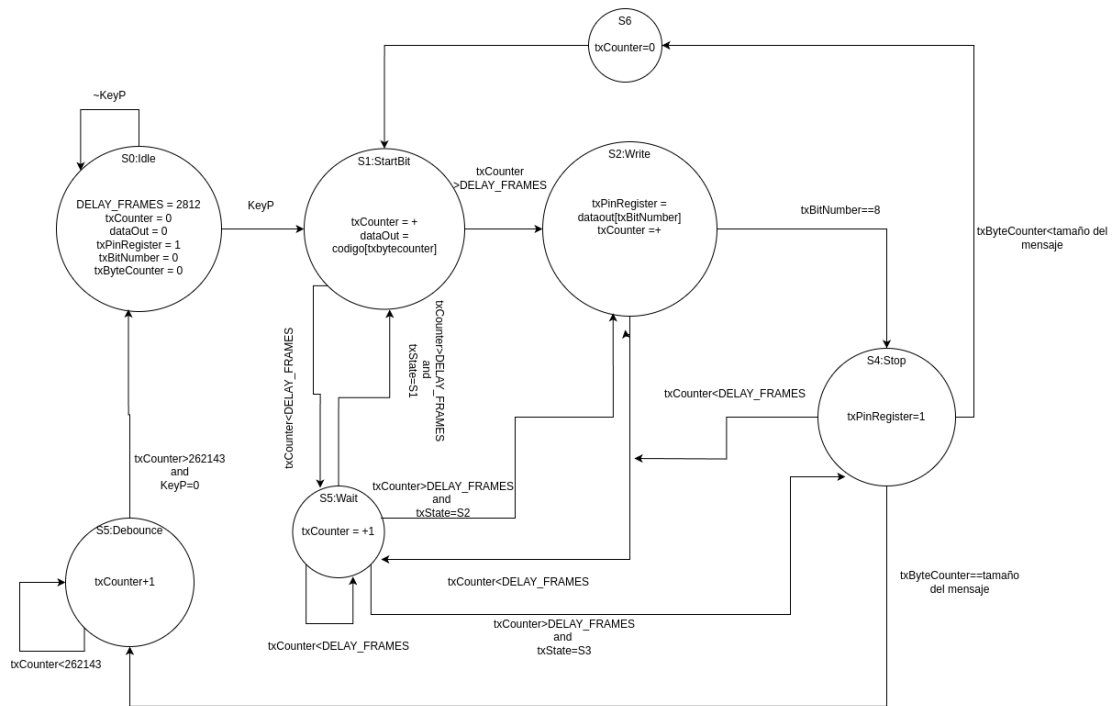


Figura 9: Máquina de Estados UART.

Esta UART funciona a 9600 baudios. Y el proceso de decodificación dura aproximadamente unos 10,646 ms.

4 Ejercicio 4

Para la SPI, se toma el código como base del siguiente repositorio: Repositorio SPI. el cual se modifica para lo que se quiere obtener, que es escoger dos posibles configuraciones de patrón de color. Cada configuración de color va a tomar únicamente dos posibles colores; rojo y azul, y otro de azul y verde. A partir de estos dos colores se pinta una grilla de colores en pantalla ante el accionar del teclado de la laptop.

Para las instrucciones de comando que se usaron de la SPI para su LCD, se tomaron de [], y están documentadas en el siguiente link: Comandos .