

Universidad de Costa Rica

Tarea 4

IE-0523 Circuitos Digitales II

Prof. Enrique Coen Alfaro

Amy Herrera Mora  
B53473

2 de junio de 2024

# Índice

<b>1. Resumen</b>	<b>3</b>
<b>2. Descripción Arquitectónica</b>	<b>3</b>
2.1. Diagrama de bloques . . . . .	3
2.2. Codificación e implementación de la Máquina de Estados . . . . .	4
<b>3. Plan de Pruebas</b>	<b>5</b>
<b>4. Instrucciones de utilización de la simulación</b>	<b>6</b>
<b>5. Resultados y Análisis</b>	<b>6</b>
5.1. Comportamiento conductual del protocolo de comunicación SPI . . . . .	6
5.1.1. Prueba 1: Enviar y recibir datos tanto en el máster como en el periférico en el modo 0 . . . . .	6
5.1.2. Prueba 2: Enviar y recibir datos tanto en el master como en el periférico en el modo 1 . . . . .	7
5.1.3. Prueba 3: Enviar y recibir datos tanto en el master como en el periférico en el modo 2 . . . . .	7
5.1.4. Prueba 4: Enviar y recibir datos tanto en el master como en el periférico en el modo 3 . . . . .	8
5.2. Comportamiento conductual del protocolo de comunicación SPI en configuración Daisy Chain . . . . .	8
5.2.1. Enviar y recibir datos tanto en el master como en el periférico en los modos 0 y 1 . . . . .	9
5.2.2. Enviar y recibir datos tanto en el master como en el periférico en los modos 2 y 3 . . . . .	9
<b>6. Conclusiones y Recomendaciones</b>	<b>10</b>
6.1. Conclusiones . . . . .	10
6.2. Recomendaciones . . . . .	10
<b>Referencias</b>	<b>11</b>

## 1. Resumen

En este trabajo se ha desarrollado y simulado un transmisor y un receptor para la interfaz serial periférica (SPI) siguiendo las especificaciones del documento SPI Block Guide V4 de NXP. El objetivo principal fue diseñar estos componentes en Verilog y verificar su correcto funcionamiento mediante un banco de pruebas exhaustivo. El transmisor y receptor diseñados cumplen con las siguientes características: Interfaces del generador que incluyen señales clave como CLK, RESET, CS, SS, SCK, MOSI, MISO, CKP, y CPH, las cuales permiten la correcta comunicación serial; y Protocolo SPI con comunicación serial de cuatro cables, permitiendo transmisión full-duplex de datos. Para la verificación, se realizó una descripción conductual en Verilog, creando módulos separados para el transmisor y el receptor, así como un banco de pruebas para simular las transacciones. Estas pruebas incluyeron: transacciones de prueba en las que se realizaron transacciones de escritura y lectura de dos bytes, utilizando dígitos específicos del número de carné; verificación en los cuatro modos de operación del protocolo SPI y en la Configuración Daisy Chain se realizaron pruebas con múltiples dispositivos periféricos conectados en cadena.

Como principales conclusiones se tienen que el sistema funcionó según lo previsto y demostró su viabilidad con éxito para su descripción conductual. Sin embargo, las pruebas realizadas no fueron tan complejas y se podrían agregar más tests de data más extensa. Además, se identificaron dos desafíos importantes: uno con el desfase de las señales que no permitía una buena transición de estados; y el tener que trabajar con dos relojes. El primero se resolvió modificando la lógica para darle tiempo a cada Flip-Flop de actualizarse en cada flanco de reloj correspondiente y el segundo al realizar todas las transacciones desde un bloque always que utilizara solamente el reloj generado para evitar que clk interfiriera de manera inadecuada ya que rige el bloque combinacional del master.

## 2. Descripción Arquitectónica

### 2.1. Diagrama de bloques

Para realizar el diseño de esta tarea se hizo uso del diagrama de bloques expuesto en la figura (1). Aquí se observan las entradas en la parte izquierda que se le ingresarán a este protocolo de comunicación, y a su vez un periférico con sus entradas y salidas respectivamente.

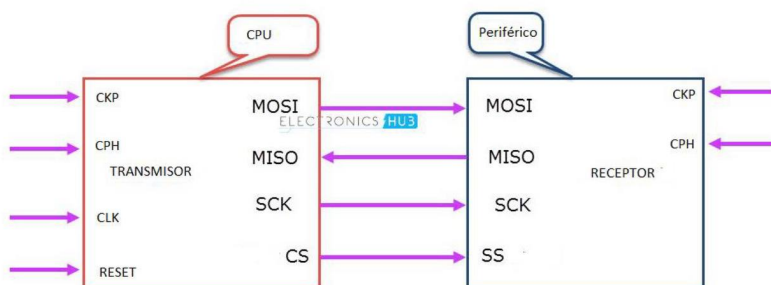


Figura 1: Diagrama de bloques para el protocolo de comunicación SPI de la Tarea 4. [1]

## 2.2. Codificación e implementación de la Máquina de Estados

Una vez analizado el diagrama de bloques, se obtuvieron 5 estados para implementar este controlador y se detallan a continuación de acuerdo a su codificación: IDLE = 3'b000, VERIFICACION = 3'b001, ENVIO\_DATOS = 3'b010, FINALIZACION\_ENVIO = 3'b011, INICIALIZACION = 3'b100. Con la figura (2) se explica el diagrama de estados y las transiciones que se quiere que esta Máquina de Estados cumpla con el fin de obtener un correcto funcionamiento.

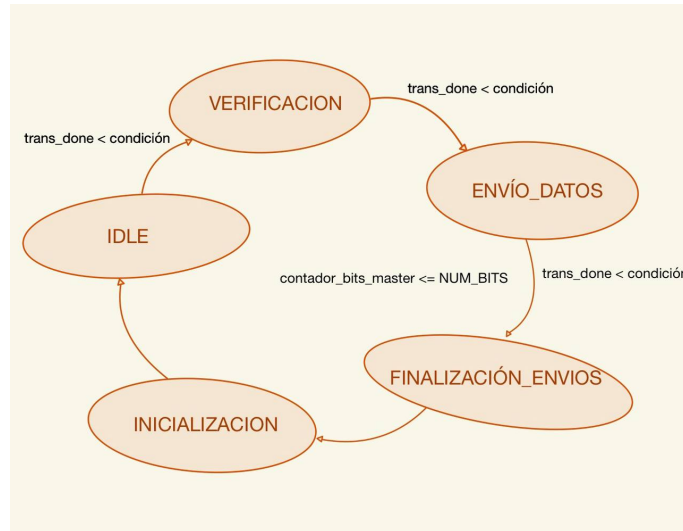


Figura 2: Diagrama de estados para el protocolo de comunicación SPI de la Tarea 4

En esta Máquina de Estados implementada con un bloque *always@(\*)* en Verilog, se inició con un bloque case que depende del estado actual del sistema y se procedió a describir la transición siguiendo el diagrama de estados anteriormente mencionado. Dicha transición se enumerará a continuación:

1. **IDLE:** En el estado IDLE, la señal CS se mantiene en 1 (lo que normalmente significa que el esclavo no está seleccionado). Si el número de transacciones completadas (*trans\_done*) es menor que 4, la FSM transita al estado VERIFICACION.
2. **VERIFICACION:** En el estado VERIFICACION, CS se pone a 0 (seleccionando el esclavo). Si CS es efectivamente 0, la FSM transita al estado ENVIO\_DATOS.
3. **ENVIO\_DATOS:** En ENVIO\_DATOS, si *trans\_done* es 4, CS se pone a 1. Si el contador\_bits\_master es menor o igual a NUM\_BITS, la FSM permanece en ENVIO\_DATOS para continuar enviando bits. Si no, la FSM transita a FINALIZACION\_ENVIO.
4. **FINALIZACION\_ENVIOS:** En FINALIZACION\_ENVIO, se asegura que MOSI esté en 0 bajo ciertas condiciones del reloj (CPH y CKP). Luego, la FSM transita a INICIALIZACION.
5. **INICIALIZACION:** En INICIALIZACION, CS se pone a 1, *sampled\_data\_MISO* se resetea a 0, el contador de bits se resetea a 0 y la FSM transita de nuevo al estado IDLE.
6. **DEFAULT:** Este es el estado por defecto, al cual la FSM se dirige si ocurre alguna condición inesperada.

Este protocolo, al utilizar un reloj de entrada *clk* para inicializar el Master que creaba un nuevo reloj al 25 % de *clk*, requirió la utilización de dos bloques *always @(posedge clk* y *always @(posedge SCK*, además de un *always @(negedge SCK* para manejar el valor de CPH para saber en cuál flanco de reloj realizar la transición de acuerdo a los modos descritos en la figura (3). Por lo tanto se realiza una transacción cada flanco de reloj respectivo según el modo en el que se esté operando. Además, al ser un protocolo de comunicación SPI, se tuvo que realizar un módulo para el master, y otro módulo para ejemplificar un periférico que puede ser instanciado las veces que sean para crear comunicación entre master-periférico y periférico-periférico. Cabe recalcar que todas las transacciones utilizando Flip-Flops se realizan teniendo como guía el reloj generado a menor frecuencia.

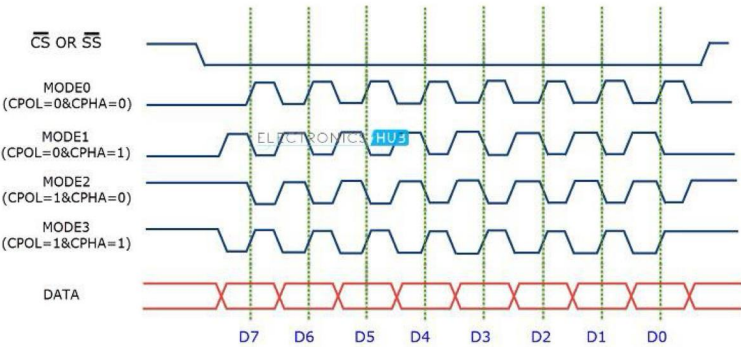


Figura 3: Diagrama de los modos para el protocolo de comunicación SPI de la Tarea 4. [1]

3. Plan de Pruebas

En la tabla (1) se puede observar el Plan de Pruebas mínimo para garantizar el correcto funcionamiento del protocolo SPI. En ella, se hace referencia al número de prueba, si falló o no, y la descripción de cada una en orden. Dichas pruebas fueron implementadas con un único archivo *tester.v*. Esto se hizo en un bloque de inicial donde se reseteaba la máquina de estados para inicializar y se cambiaban los valores de las entradas de CPH y CKP para verificar el valor de las salidas con la herramienta *gtkwave*. Dichas pruebas se llevaron a cabo siguiendo los modos expuestos en la (3).

Prueba	Falló	Descripción
1	No	Enviar y recibir datos tanto en el master como en el periférico en el modo 0. CKP=0 y CPH=0 donde SCK está en bajo mientras no haya transacción y la data se envía en el flanco positivo del reloj.
2	No	Enviar y recibir datos tanto en el master como en el periférico en el modo 1. CKP=0 y CPH=1 donde SCK está en bajo mientras no haya transacción y la transacción se realiza en el flanco negativo del reloj.
3	No	Enviar y recibir datos tanto en el master como en el periférico en el modo 2. CKP=1 y CPH=0 donde SCK está en alto mientras no haya transacción y la transacción se realiza en el flanco negativo del reloj.
4	No	Enviar y recibir datos tanto en el master como en el periférico en el modo 3. CKP=1 y CPH=1 donde SCK está en alto mientras no haya transacción y la transacción se realiza en el flanco positivo del reloj.

Tabla 1: Descripción de las pruebas realizadas.

## 4. Instrucciones de utilización de la simulación

Al extraer el archivo B53473.zip se encontrarán 2 directorios. En primer lugar se encuentra *SPI* que contiene la descripción conductual del master **generador.v**, la descripción conductual del periférico **objetivo.v**, el **testbench.v** y el **tester.v**; y para su debida compilación y simulación se debe escribir en la terminal de linux desde el dir B53473 **cd SPI** seguido de un **make**. De igual manera se utiliza para el directorio que realiza la configuración margarita *DC\_configuration*. Dicho directorio contiene los archivos pertinentes para la configuración margarita y la simulación de las pruebas al escribir en la terminal desde el dir B53473 **cd DC\_configuration** seguido de un **make** para observar el funcionamiento.

## 5. Resultados y Análisis

En la presente tarea se realizó el diseño de un protocolo de comunicación SPI. En primer lugar, se hizo la descripción conductual tanto del módulo Máster como del periférico y se aplicaron las 4 pruebas necesarias para analizar su comportamiento y garantizar su correcto funcionamiento. Seguidamente, instanciando dos periféricos se implementó la configuración Daisy Chain de la figura (4) y se probaron de igual manera los cuatro modos.

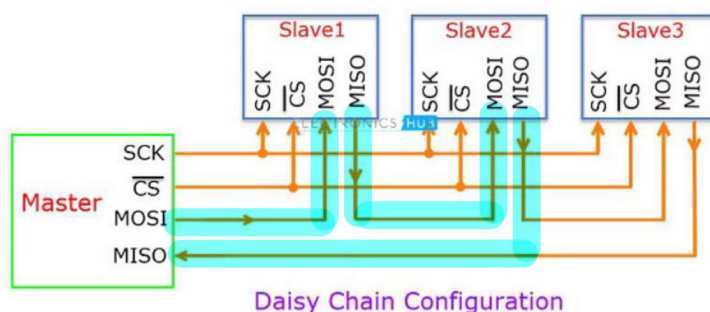


Figura 4: Módulos SPI conectados en configuración Daisy Chain. [1]

### 5.1. Comportamiento conductual del protocolo de comunicación SPI

#### 5.1.1. Prueba 1: Enviar y recibir datos tanto en el máster como en el periférico en el modo 0

De acuerdo al protocolo de comunicación, desde el máster se envía la señal MOSI que contiene la data 0304 que manda dicho módulo, mientras que desde el periférico se envía simultáneamente mediante MISO 0703 al ser full duplex. Seguidamente, para garantizar la comunicación se tiene un registro en cada módulo que atrapa bit a bit los datos que le llegan a cada uno. Al observar la figura (5), se muestra en la variable sampled que a master sí le llegó 0703 mediante MISO y que al periférico efectivamente le llegó 0304 mediante la señal MOSI. El contador de bits de cada módulo indexa y refleja tanto en MOSI como en MISO la transacción. En este caso se refleja en el flanco positivo, y cuando CS/SS VALEN 1, el IDLE es efectivamente en bajo.

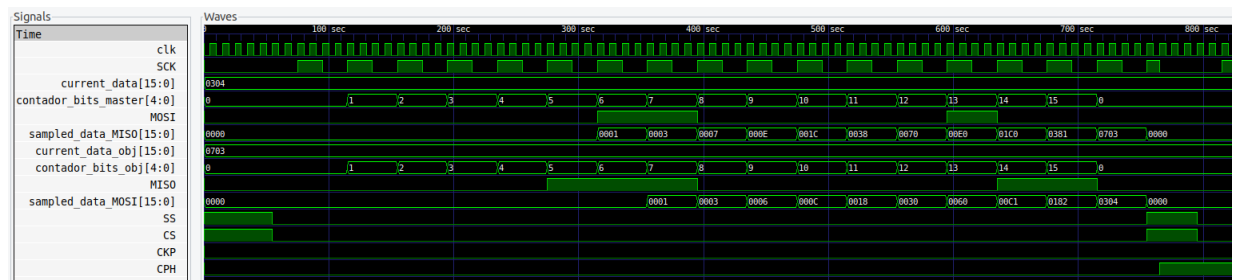


Figura 5: Simulación del correcto funcionamiento de la descripción conductual del protocolo SPI para la prueba 1.

### 5.1.2. Prueba 2: Enviar y recibir datos tanto en el master como en el periférico en el modo 1

De acuerdo al protocolo de comunicación, desde el master se envía la señal MOSI que contiene la data 0304 que manda dicho módulo, mientras que desde el periférico se envía simultáneamente mediante MISO 0703 al ser full duplex. Seguidamente, para garantizar la comunicación se tiene un registro en cada módulo que atrapa bit a bit los datos que le llegan a cada uno. Al observar la figura (6), se muestra en la variable sampled que a master sí le llegó 0703 mediante MISO y que al periférico efectivamente le llegó 0304 mediante la señal MOSI. El contador de bits de cada módulo indexa y refleja tanto en MOSI como en MISO la transacción. En este caso se refleja en el flanco negativo, y cuando CS/SS VALEN 1, el IDLE es efectivamente en bajo.

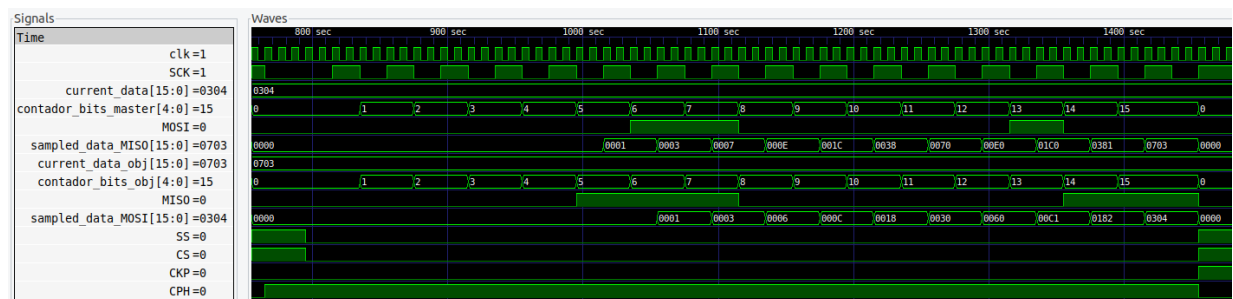


Figura 6: Simulación del correcto funcionamiento de la descripción conductual del protocolo SPI para la prueba 2.

### 5.1.3. Prueba 3: Enviar y recibir datos tanto en el master como en el periférico en el modo 2

De acuerdo al protocolo de comunicación, desde el master se envía la señal MOSI que contiene la data 0304 que manda dicho módulo, mientras que desde el periférico se envía simultáneamente mediante MISO 0703 al ser full duplex. Seguidamente, para garantizar la comunicación se tiene un registro en cada módulo que atrapa bit a bit los datos que le llegan a cada uno. Al observar la figura (7), se muestra en la variable sampled que a master sí le llegó 0703 mediante MISO y que al periférico efectivamente le llegó 0304 mediante la señal MOSI. El contador de bits de cada módulo indexa y refleja tanto en MOSI como en MISO la transacción. En este caso se refleja en el flanco negativo, y cuando CS/SS VALEN 1, el IDLE es efectivamente en alto.

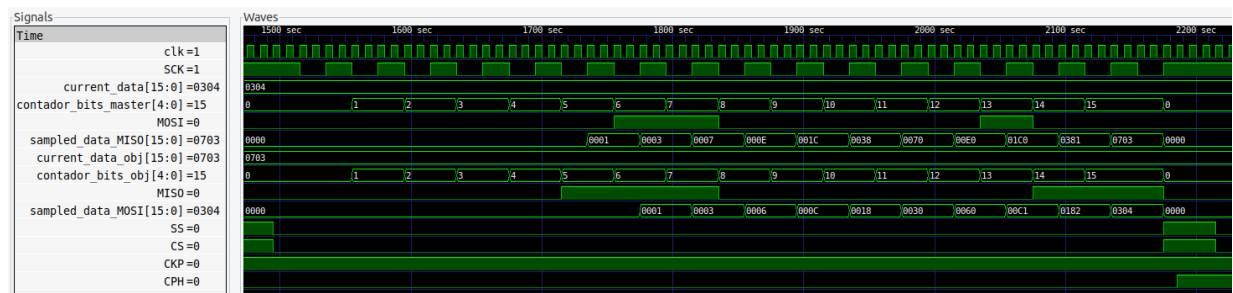


Figura 7: Simulación del correcto funcionamiento de la descripción conductual del protocolo SPI para la prueba 3.

#### 5.1.4. Prueba 4: Enviar y recibir datos tanto en el master como en el periférico en el modo 3

De acuerdo al protocolo de comunicación, desde el master se envía la señal MOSI que contiene la data 0304 que manda dicho módulo, mientras que desde el periférico se envía simultáneamente mediante MISO 0703 al ser full duplex. Seguidamente, para garantizar la comunicación se tiene un registro en cada módulo que atrapa bit a bit los datos que le llegan a cada uno. Al observar la figura (8), se muestra en la variable sampled que a master sí le llegó 0703 mediante MISO y que al periférico efectivamente le llegó 0304 mediante la señal MOSI. El contador de bits de cada módulo indexa y refleja tanto en MOSI como en MISO la transacción. En este caso se refleja en el flanco positivo, y cuando CS/SS valen 1, el IDLE es efectivamente en alto.

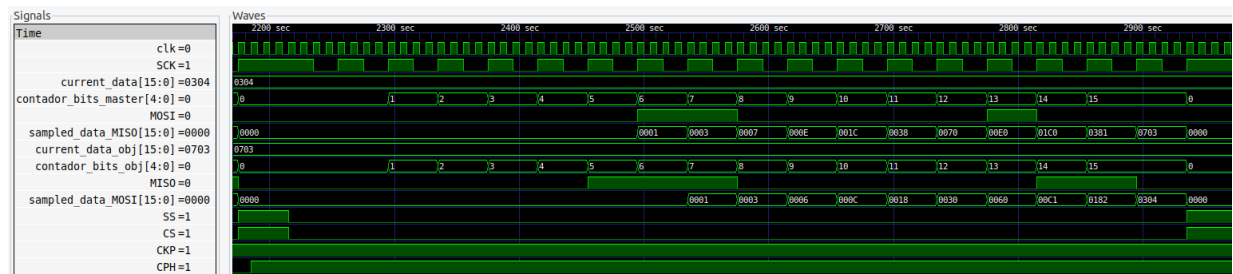


Figura 8: Simulación del correcto funcionamiento de la descripción conductual del protocolo SPI para la prueba 4.

## 5.2. Comportamiento conductual del protocolo de comunicación SPI en configuración Daisy Chain

En la configuración de Daisy Chain mostrada en la figura 4, los módulos SPI (Master y Periféricos) están conectados en el testbench de la siguiente manera: el pin MOSI del Master está conectado al pin MOSI del periférico 1, el pin MOSI del periférico 1 está conectado al pin MOSI del periférico 2, y el pin MOSI del periférico 2 está conectado al pin MOSI del Master. En dicha configuración, la data enviada desde el master va en dirección hacia el periférico1. Para esto, se nombraron los wires wire MOSI\_MOSI, wire MISO\_MOSI, wire MISO\_MISO que hiciera una conexión intuitiva y sencilla. De igual manera, la data enviada desde el periférico1 va en dirección al periférico2. Finalmente, los bits enviados desde el periférico2 van en dirección hacia el master. En las figuras (9) y (10) se observa que master guarda lo enviado por el periférico2 que es 0703, periférico1 recibe 0304



de master, y que periférico2 recibe y guarda los datos llegados desde el periférico1. Se observa el correcto funcionamiento de esta cadena margarita también los cuatro modos descritos en la tabla de las pruebas para esta tarea 4.

### 5.2.1. Enviar y recibir datos tanto en el master como en el periférico en los modos 0 y 1



Figura 9: Simulación del correcto funcionamiento de la descripción conductual del protocolo SPI para la prueba 4.

### 5.2.2. Enviar y recibir datos tanto en el master como en el periférico en los modos 2 y 3



Figura 10: Simulación del correcto funcionamiento de la descripción conductual del protocolo SPI para la prueba 4.

## 6. Conclusiones y Recomendaciones

### 6.1. Conclusiones

1. El sistema funcionó según lo previsto y demostró su viabilidad con éxito para su descripción conductual. Realizar un diseño paso a paso y realizar simulaciones en cada parte permitió un buen desarrollo de esta tarea. Sin embargo, las pruebas realizadas no fueron tan complejas y se podrían agregar más tests de data más extensa.
2. Se identificaron dos desafíos particulares: uno con el desfase de las señales que no permitía una buena transición de estados y dos el tener que trabajar con dos relojes. El primero se resolvió modificando la lógica para darle tiempo a cada Flip-Flop de actualizarse en cada flanco de reloj correspondiente, y el segundo al realizar todas las transacciones desde un bloque always que utilizara solamente el reloj generado.
3. Los problemas encontrados durante el proceso fueron superados con recursos disponibles. Las horas consulta de clase, la documentación, los archivos de clase sobre máquinas de estado en Verilog y la práctica fueron fundamentales para el éxito alcanzado en esta tarea.

### 6.2. Recomendaciones

1. Priorizar la fase de diseño inicial con la creación detallada de diagramas y la definición clara de requisitos y funcionalidades del sistema es fundamental para comenzar con el proceso. Dividir el problema total en pequeñas partes e ir armando el código poco a poco es indispensable, y esto se vuelve mucho más relevante en diseños más grandes y complejos.
2. Probar las señales una a una una vez terminada cada división del problema total, y luego comprobar su funcionamiento permite arreglar problemas de lógica de manera más fácil y rápida. Por ejemplo, al verificar que las señales que enviaban y recibían datos se tuvo especial cuidado con el bloque de lógica combinacional, ya que este se rige por el clk que inicializa el master, entonces las transacciones muchas veces no se sincronizan con el SCK generado si no se tiene precaución.
3. Realizar pruebas exhaustivas del sistema mediante simulaciones en diferentes escenarios y condiciones para garantizar su robustez y fiabilidad. Esto permite observar el funcionamiento del trabajo para poder identificar mejor las fallas, y así poder corregir cualquier problema que se presente.
4. Pedir ayuda externa, ya sea en horas consulta, es indispensable, pues existen muchos errores de funcionalidad que no pueden ser detectados por una única persona que diseña circuitos digitales. También el continuar aprendiendo y practicando el uso de Verilog y otras herramientas de diseño de hardware para mejorar constantemente las habilidades y la eficiencia en futuros proyectos similares.

## Referencias

1. “Basics of Serial Peripheral Interface (SPI),” *Electronics Hub*, Nov. 5, 2020. [Online]. Available: <https://www.electronicshub.org/basics-serial-peripheral-interface-spi/>. [Accessed: June 1, 2024].