

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0523 - Circuitos Digitales II

I ciclo 2023

Tarea 4

Diseño de una interfaz serial periférica (SPI)

Isaí David Vargas Ovarés B88263

Grupo 1

Profesor: Enrique Coen Alfaro

28 de mayo de 2023

1. Resumen

En la interfaz SPI, la comunicación síncrona se establece entre un dispositivo maestro (transmisor) y uno o más dispositivos esclavos (receptor). Los dispositivos están conectados por líneas de señal separadas que permiten la transmisión y recepción de datos, por lo que el diseño del transmisor y receptor SPI (Serial Peripheral Interface) está diseñado de acuerdo a las especificaciones del documento NXP SPI Block Guide V4 [1].

2. Descripción Arquitectónica

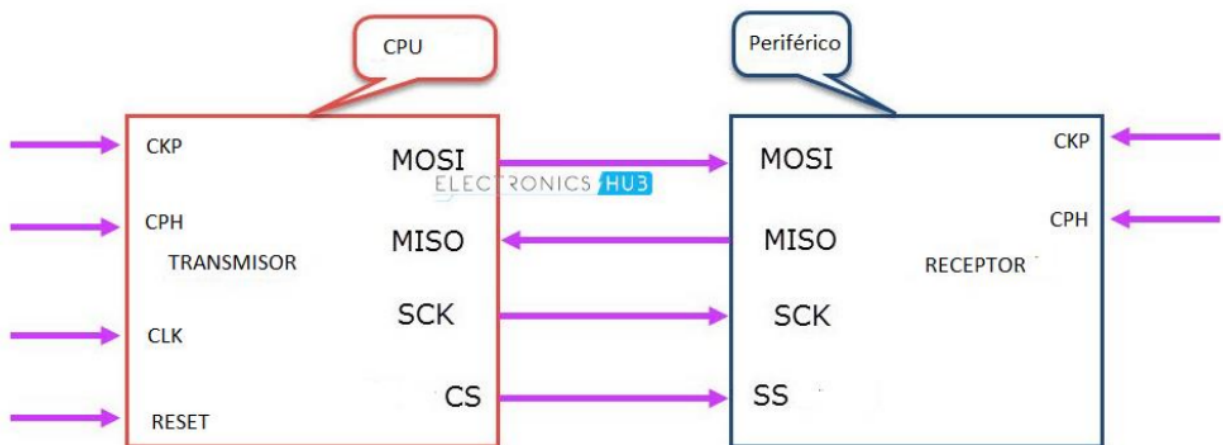


Figura 1: Interfaz serial periférica SPI.

Como se menciona en [2] SPI (Serial Peripheral Interface) es un protocolo de comunicación utilizado para transferir datos entre dispositivos digitales. La guía de bloques NXP SPI V4 contiene especificaciones e instrucciones para implementar el módulo transceptor SPI. En este caso, el módulo debe tener las siguientes entradas y salidas:

2.1. Entradas

- CLK: Es una entrada que llega al transmisor de SPI desde el CPU con una frecuencia determinada. El flanco activo de CLK es el flanco creciente.
- Reset: Entrada de reinicio del transmisor. Si RESET=1 el transmisor funciona normalmente. En caso contrario, el transmisor vuelve a su estado inicial y todas las salidas toman el valor de cero.
- CS: Indicación de "Chip Select". Es una salida del transmisor de SPI, para indicar que va a enviar una transacción hacia el receptor. La señal CS es activa en bajo.
- SS: Indicación de "Slave Select". Es una entrada del receptor que le indica que el transmisor está enviándole una transacción. La señal SS es activa en bajo..
- CKP: La entrada de polaridad del reloj (clock polarity) determina el estado del reloj cuando el dispositivo está en reposo (idle), es decir, cuando no está realizando una transacción. Si es cero el reloj permanece en cero cuando no se está enviando una transacción y si es 1, deberá permanecer en 1.

- CPH: La entrada de fase del reloj (clock phase) determina en cuál flanco de SCK se realiza la transición. Si es cero se transmite en el flanco creciente y si es 1 en el decreciente.

2.2. Salidas

- Balance_actualizado: Salida de reloj para el transmisor de SPI. El flanco activo de la señal SCK es el flanco creciente. Observe que SCK es una salida del transmisor, que deberá tener una frecuencia del 25 % de la frecuencia de la entrada CLK. El transmisor debe generar SCK con la frecuencia correcta para cualquier posible valor de la frecuencia de entrada CLK. La señal de SCK es, a su vez, una entrada para el receptor, que deberá entregar y recibir un bit nuevo en las interfaces MISO y MOSI en cada ciclo de SCK. Adicionalmente, el comportamiento de SCK deberá ser consistente con las indicaciones de CKP y CPH como se describe más adelante.
- MOSI: Salida serial del transmisor, entrada serial para el receptor. En cada ciclo de SCK se espera que el transmisor envíe un nuevo bit de datos y que el receptor lo reciba a través de la entrada del mismo nombre.
- MISO: Entrada serial del transmisor, salida serial del receptor. En cada ciclo de SCK se espera que el receptor envíe un nuevo bit de datos y que el transmisor lo reciba a través de esta señal.

2.3. Estados

Cada modulo tiene sus estados segun sea el modulo transmisor o el receptor, las de transmisor son:

- RDY (Ready): En este estado, el transmisor está esperando una señal de reinicio (reset). Si se recibe un reinicio (reset=1), se pasa al estado START; de lo contrario, el transmisor permanece en el estado RDY y configura algunas salidas en valores predeterminados.
- START: En este estado, el transmisor configura la señal CS (Chip Select) en bajo y envía los datos a través de la señal MOSI (Master Out, Slave In). El transmisor transmite un bit de datos a la vez y disminuye el índice para acceder a los bits de datos siguientes. Luego, pasa al estado TRANSMIT.
- TRANSMIT: En este estado, el transmisor sigue enviando los datos restantes hasta que se alcance el último bit de datos. En cada ciclo de reloj, se envía un nuevo bit de datos a través de la señal MOSI y se almacena el bit recibido a través de la señal MISO (Master In, Slave Out) en el registro data_temp. El índice se decrementa hasta que alcanza el valor 0.
- STOP: En este estado, el transmisor completa la transmisión de datos. Se almacena el valor final de los datos en el registro data1 y se restablecen algunas variables y señales. Además, la señal CS se configura en alto para indicar el final de la transacción. Luego, el transmisor vuelve al estado RDY para esperar una nueva transacción.

Estados del receptor:

- RDY (Ready): En este estado, el receptor está esperando la activación de la señal SS (Slave Select) para comenzar a recibir datos. Cuando la señal SS está activa (ss=0), el receptor pasa al estado START.

- **START:** En este estado, el receptor configura la señal MISO en función del bit de datos actual almacenado en el registro carne y decrementa el índice. Luego, pasa al estado RECEIVE.
- **RECEIVE:** En este estado, el receptor sigue recibiendo los datos restantes hasta que se alcance el último bit de datos. En cada ciclo de reloj, se configura la señal MISO con el bit de datos correspondiente del registro carne y se almacena el bit recibido a través de la señal MOSI en el registro data_temp. El índice se decrementa hasta que alcanza el valor 0.
- **STOP:** En este estado, el receptor completa la recepción de datos. Se almacena el valor final de los datos en el registro data y se restablecen algunas variables y señales. Además, se actualiza el valor de carne para configurar el próximo bit de datos en la siguiente transacción. Luego, el receptor vuelve al estado RDY para esperar una nueva transacción.

2.4. Señales internas

- **Carné:** Un registro de 8 bits que almacena el número del carné que se esta transmitiendo o recibiendo.
- **data_temp:** Registros de 8 bits que almacena los dígitos recibidos por mosi o miso.
- **index:** Un registro entero que lleva la cuenta del dígito que se esta recibiendo en mosi o miso.

3. Plan de Pruebas

- **Prueba 1 MODO 0 (ckp=0,cph=0),** se establecen ckp y cph y se comprueba si se leen y escriben correctamente los números del carne, además de comprobar que el comportamiento del reloj sck sea el correcto.
- **Prueba 2 MODO 1 (ckp=0,cph=1),** se establecen ckp y cph y se comprueba si se leen y escriben correctamente los números del carne, además de comprobar que el comportamiento del reloj sck sea el correcto.
- **Prueba 3 MODO 2 (ckp=1,cph=0),** se establecen ckp y cph y se comprueba si se leen y escriben correctamente los números del carne, además de comprobar que el comportamiento del reloj sck sea el correcto.
- **Prueba 4 MODO 3 (ckp=1,cph=1),** se establecen ckp y cph y se comprueba si se leen y escriben correctamente los números del carne, además de comprobar que el comportamiento del reloj sck sea el correcto.

4. Instrucciones de utilización de la simulación

Esta tarea fue desarrollada en el entorno de EDAPlayground por lo que basta con ingresar al siguiente link y correr la simulación.

- <https://www.edaplayground.com/x/LBf8>

5. Resultados

- **Prueba 1:** Se establece el MODO 0, con ckp y cph en 0, sin embargo estas no causan cambios en el comportamiento de sck, por lo que se tiene algún error en esta parte del código, sin embargo esta señal si cumple con ser el 25 % de clk, con respecto a la transacción de escritura se puede ver en **received_data_slave** que el receptor recibe el numero de manera correcta, tanto el 8 como el 2 del carné. Sin embargo para el caso de lectura en **received_data_master** solo se obtiene el 6 del del carné y se pierde el 3. También se puede ver que con la señal de reset en 0 se sigue con el funcionamiento normal, por lo que tiene otro fallo en esta parte.

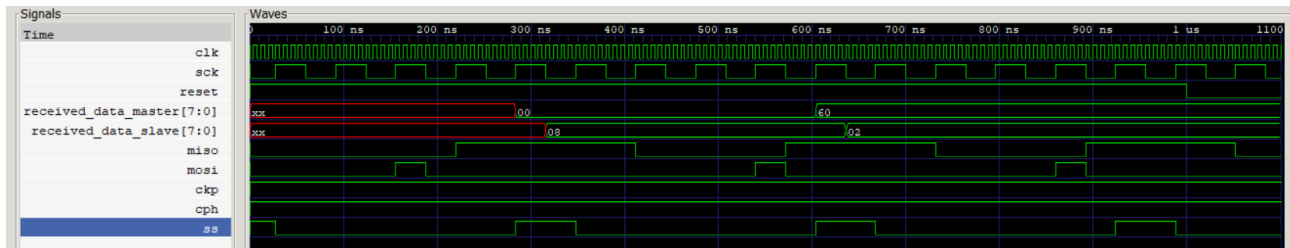


Figura 2: Prueba 1. Diagramas de tiempo.

Para las pruebas de los demás modos se tienen los mismos resultados.

6. Conclusiones

- Se logro de manera correcta la transacción de escritura del 8 y el 2 del carné.
- En código tiene dos principales problemas, el primero siendo que no se logro incluir el comportamiento de sck ante los cambios de cph y ckp. Por lo que no se pueden analizar los distintos modos del protocolo SPI, El segundo es con respecto a la lectura de datos, ya que se logro leer el 6 del carné pero no el 3.
- Se tiene un tercer problema con respecto a la señal de reset, donde esta cambia a 0 se sigue con el comportamiento normal de los módulos, esto a pesar de que se puede comprobar que esta cambia.

Referencias

- [1] I. Motorola, "Spi block guide v04.01," *Freescale Semiconductor, Inc.*, 2004.
- [2] Anusha. (2017) <https://www.electronicshub.org/basics-serial-peripheral-interface-spi/>.