

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0523 – Circuitos Digitales II

I ciclo 2024

Reporte Tarea 3

Protocolo SPI

Alexa López Marcos B94353

Profesor: Ana Eugenia Sánchez Villalobos.

9 de Junio del 2024

# 1. Ejecución

Para realizar una correcta ejecución del programa se facilita el siguiente archivo tipo makefile. Para poder ejecutar este archivo desde la terminal de Linux se debe de :

- Abrir la terminal de Linux.
- Posicionarse en la carpeta en la que se encuentra el archivo makefile con los archivos de los módulos del programa.
- Colocar en la terminal **make spi**

Este archivo tipo makefile se utilizara para automatizar la ejecución del programa y contiene las líneas necesarias para ejecutar el programa desde la terminal, se colocan dentro de una bandera del archivo tipo makefile, para que al ejecutar en la terminal se generen los archivos necesarios para ejecutar y abrir GTKWave para ver el comportamiento de las señales.

Dicho makefile contiene:

```
1 spi:
2     iverilog -o tb.vvp testbench.v
3     vvp tb.vvp
4     gtkwave tb.vcd
5     rm -rf tb.vvp tb.vcd
6
7 clean:
8     rm -rf tb.vvp tb.vcd
```

Con esto, ya no es necesario ejecutar una a una las líneas necesarias para ejecutar el programa, solamente se ejecutaría **make spi** desde la terminal para ejecutar el programa.

## 2. Señales utilizadas

### 2.1. MASTER

El archivo llamado **master.v** corresponde al módulo master del protocolo SPI.

#### 2.1.1. Entradas:

En este caso, este módulo contiene como entradas las señales:

- clk
- reset
- datain de 16 bits
- mensaje (añadida)
- MISO (añadida)

#### 2.1.2. Salidas:

En este caso, este módulo contiene como salidas las señales:

- spi\_cs1\_l
- spi\_cs2\_l (añadida)
- sclk
- spi\_data
- counter de 16 bits

La señal MISO fue agregada para recibir la señal MISO de salida de los slaves.

La señal mensaje fue añadida para recibir un 0 o un 1 del slave confirmando si MISO es igual que MOSI.

La señal spi\_cs2\_l fue añadida para el CS del slave 2.

### 2.2. Slaves

El archivo llamado **slave1.v** corresponde al módulo slave del protocolo SPI.

#### 2.2.1. Entradas:

En este caso, este módulo contiene como entradas las señales:

- reset
- sclk
- spi\_data\_in
- spi\_cs\_l

### 2.2.2. Salidas:

En este caso, este módulo contiene como salidas las señales:

- spi\_data\_out
- mensaje

## 2.3. TESTBENCH

El archivo llamado **testbench.v** corresponde al módulo TESTBENCH y TESTER juntos, en el que se instancian los módulos master y ambos slaves.

## 3. Conexión de los Slaves al Master

Para la conexión de los slaves al master se siguió la imagen de referencia proporcionada en el enunciado de la tarea. Se trató de ser lo más fiel a la imagen de referencia y al código base proporcionado para la tarea. La conexión de estos se realizó en el módulo de Testbench tal cual se muestra en la figura 1

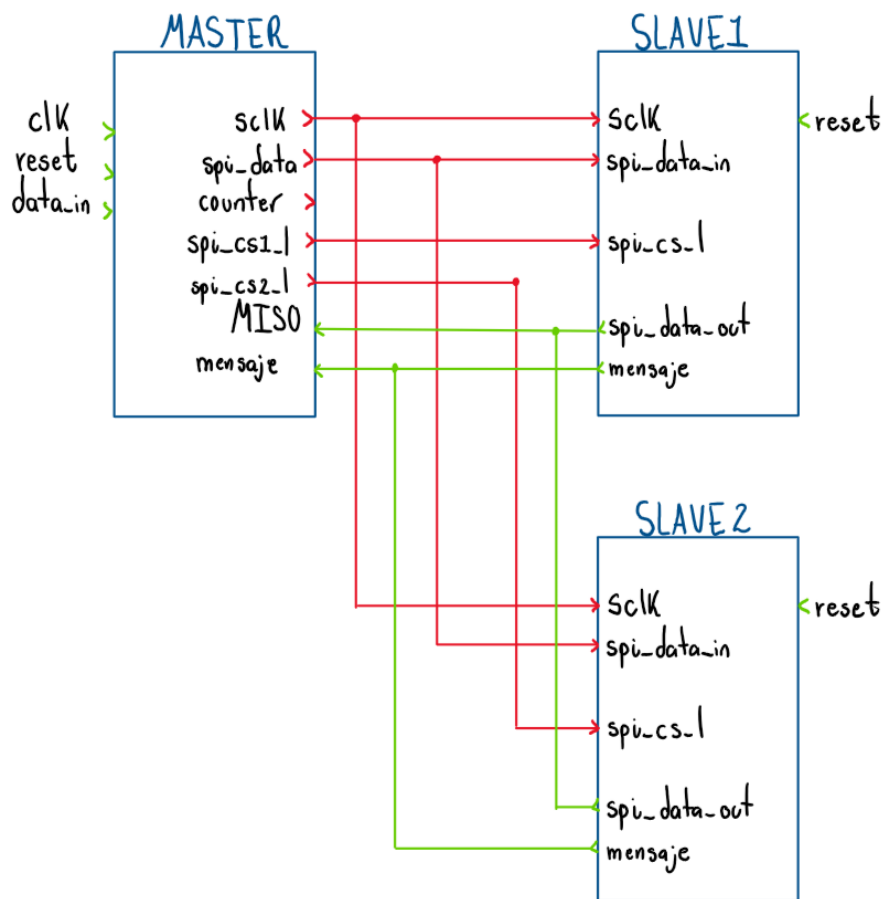


Figura 1: Imagen de las conexiones de los slaves al master.

## 4. WAVEFORMS

Para el módulo de MASTER se inicia con el reset en alto y mientras este esté en alto las señales de CS1 y CS2 también estarán en alto. Cuando CS1 y CS2 están en bajo ingresan los datos al master desde el testbench. Conforme estos datos se van enviando bit por bit a los slaves se va reduciendo el contador, cuando el contador está en 0 los slaves han terminado de recibir la información y se pasa al estado '3' en el que el master espera hasta que los slaves terminen de enviar información y los slaves manden en alto la señal de mensaje, esta señal de mensaje hace que el slave sepa que ya se terminó de recibir la información del MISO y solo así se activen las señales CS. Todo esto mostrado en la figura 2.

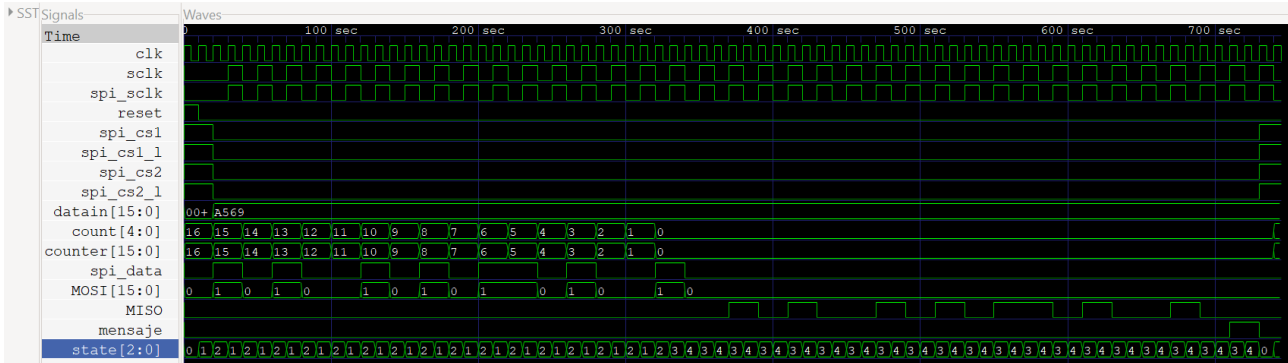


Figura 2: Imagen de las señales y registros del MASTER

Para los módulos de SLAVES, estos se rigen por la señal de entrada CS, mientras CS esté en alto no pueden operar. El recibe la información MOSI por medio de spi\_data.in del MASTER y la almacena en el registro MOSI, el contador interno va aumentando cada vez que se almacena un bit en MOSI, lo que permite que cuando el contador sea 16 se pueda disminuir para enviar los bits por medio de la señal del registro MISO asociado a la salida spi\_data.out. En el código hay una condición de que la señal en alto del mensaje no se envíe hasta que se compruebe que el registro MISO es igual a la señal de entrada spi\_data.in. Mostrado en las figuras 3 y 4.

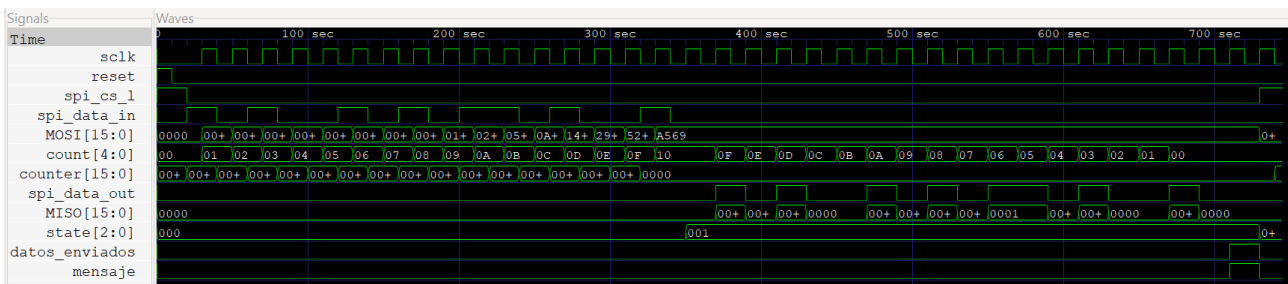


Figura 3: Imagen de las señales y registros del SLAVE1

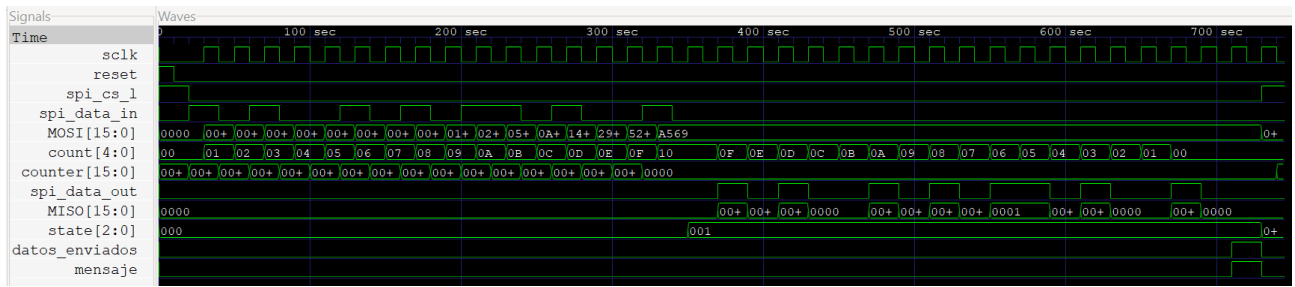


Figura 4: Imagen de las señales y registros del SLAVE2

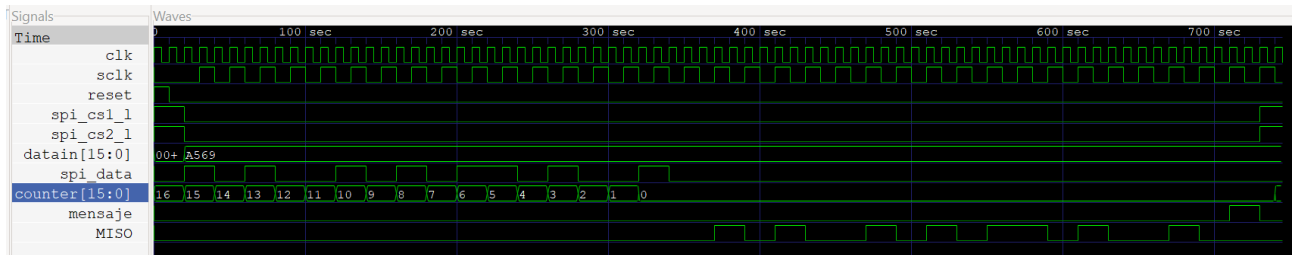


Figura 5: Imagen de las señales y registros del TESTBENCH