

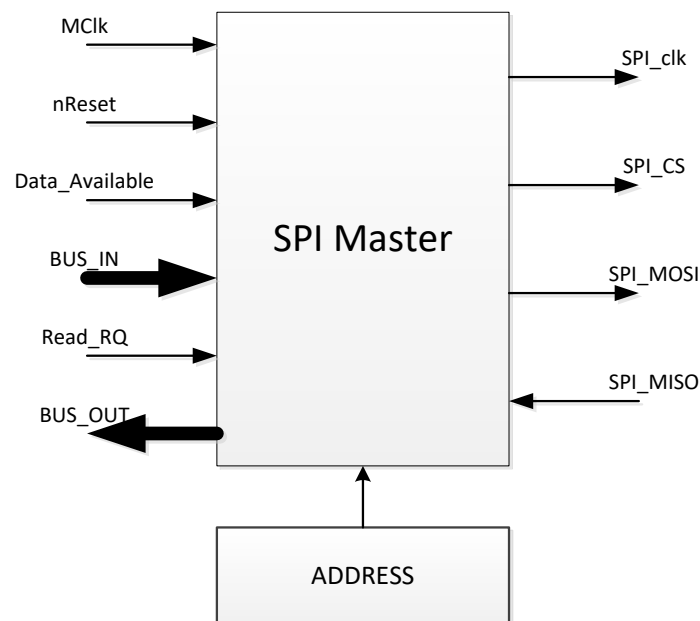
Segundo Proyecto (grupal)

Valor 15%

Fecha de entrega: 24 de junio, 2017

Descripción:

Implementar un núcleo SPI master. Las señales en el módulo TOP se muestran en la figura:



Protocolo:

La comunicación entre este SPI maestro y un esclavo comienza cuando la señal SPI_CS cambia a activa (nivel 0). En el momento en que SPI_CS es activo, el reloj SPI se habilita a la salida. El bloque que se transmite contiene inicialmente un byte de dirección (se envía por medio de la línea MOSI). Dependiendo si la operación es escritura o lectura, se continua enviando datos por el MOSI o se leen datos por el MISO.

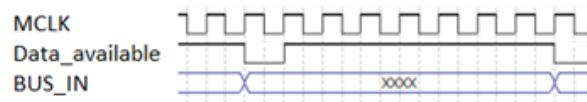
Las siguientes son consideraciones a tomar en cuenta:

1. Los bloques de datos transmitidos o recibidos son de 15bytes
2. El módulo contiene dos buffers FIFO de 15 bytes. Estos buffers son usados para transmisión y recepción de datos.
3. El byte de dirección se divide en: 1 bit de control para escritura/lectura (MSB) y los siguientes 7 bits contienen la dirección.

4. La dirección se toma de un bloque externo. Para el proyecto asuma la dirección 0x5A
5. Tomar en cuenta que mientras se está transmitiendo, hay probabilidades de que lleguen nuevos bytes al BUS_IN (que serán parte del siguiente frame una vez que se complete otra vez la cuenta de 15)
6. La transmisión y recepción **no** se dan al mismo tiempo
7. El bus de entrada BUS_IN es un bus paralelo de 8bits
8. El bus de salida BUS_OUT es un bus paralelo de 15x8bits
9. MISO, MOSI, SPI_CS, SPI_clk son señales de 1 bit.
10. La implementación del FIFO de entrada y salida debe realizarlo utilizando el IP Core de altera.

Transmisión de datos

Los datos en el bus BUS_IN pueden ser registrados en el FIFO en el flanco positivo de la señal Data_available. La señal Data_available se ejecuta 512 veces más lenta que la señal MCLK. (la señal data_available normalmente está en alto y cuando se escribe un dato se presenta como un pulso corto)



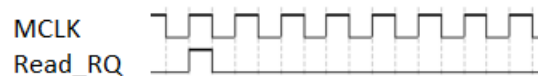
La transmisión se realiza cada vez el bloque ha recibido 15 bytes en el FIFO de transmisión.



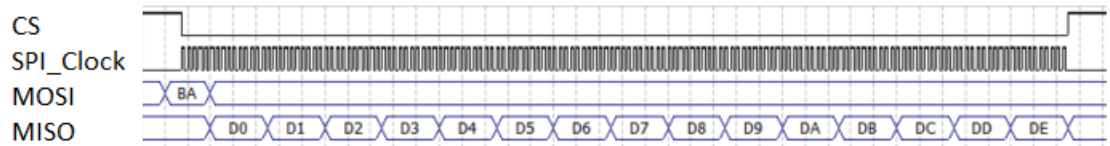
Cuando se da la transmisión, el primer byte corresponde a la dirección, y el bit de control se coloca en 0 para indicar que se enviarán datos al esclavo. Luego de este byte de dirección se envían de forma serial los 15 bytes de información. La transmisión debe ser lo suficientemente rápida para que se pueda transmitir al menos un byte de información antes de que llegue un nuevo byte al FIFO causando overflow.

Recepción de datos

La recepción se inicia cuando se detecta un flanco positivo en la entrada "Read_RQ".



La recepción de datos inicia enviando por el MOSI el byte de dirección con el bit de control en 1 para indicar al esclavo que se va a realizar una lectura. Luego de enviar este byte de dirección se debe leer el MISO en cada flanco negativo del reloj SPI.



Una vez que el FIFO de entrada está lleno (15bytes), los datos se colocan en la interfaz paralela BUS_OUT.

Especificaciones

El reloj Mclk es un reloj de 25MHz, mientras que el reloj SPI es un reloj de 3,125MHz. EL reset (nReset) es asíncrono y activo en bajo.

El SPI_CLK debe mantenerse en 0 mientras está inactivo. Los datos de salida se escriben en el flanco positivo del reloj (el primer flanco debe contener el primer bit válido).

Análisis:

1. Inicie generando un documento con las especificaciones del proyecto (por funcionalidad). Revise el formato sugerido en las diapositivas de “development cycle”. Recuerde incluir su plan de pruebas.
2. Genere un diagrama de bloques a alto nivel donde detalle la arquitectura de su diseño.
 - a. ¿Cuáles son los sub-bloques que va a requerir?
 - b. ¿Cómo va a realizar la implementación de cada bloque? (e.g.: IP Core, Máquina de estados, Verilog RTL)
 - c. Estime de forma gruesa los recursos a requerir para su diseño (elementos de memoria, registros).
 - i. Si necesita bloques de memoria recuerde que cada MLAB provee 640 bits en memoria distribuida, y cada M10K provee 10Kbits de memoria block.
3. Escriba el código RTL para el bloque SPI. También escriba el testbench.
4. Incluya en sus resultados especificaciones de consumo de recursos y además timing para su implementación (la herramienta le proporciona estos datos en el reporte de TimeQuest al ejecutar place and route)
5. La implementación debe ser compatible con diseño para FPGAs. En el caso de que este bloque fuera a ser implementado en un ASIC; que cambios agregaría la código de tal forma que tenga un solo fuente para ambas implementaciones? (puede asumir que existe un BLACK-BOX llamado SPI_CM, el cual se encarga de generar el reloj de 20MHz y el de 3,125MHz)
6. Presente sus resultados en un documento a dos columnas de no más de 3 páginas (simulaciones, proceso de diseño, diagramas, etc), además agregue como apéndice el documento de especificaciones y el documento de arquitectura. Además recuerde comparar la estimación inicial que hizo de recursos con los recursos finales que el sistema ocupa.