

Informe Laboratori 3: Comunicació SPI

Introducció

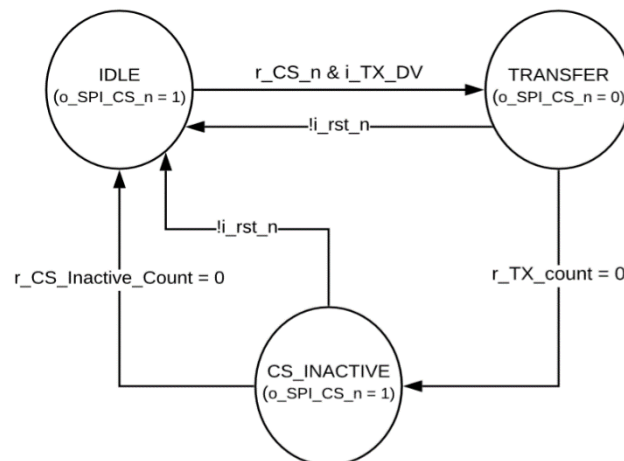
La finalitat d'aquesta pràctica és la implementació i configuració d'una comunicació SPI (Serial Peripheral Interface), la qual treballa amb un rellotge propi (o_SPI_clk), una senyal Master Output Slave Input (o_SPI_MOSI), una senyal Master Input Slave Output (o_SPI_MISO) i una senyal Chip Select (o_SPI_CS_n). L'objectiu d'aquesta pràctica és la correcta configuració d'un Master SPI (*), la realització d'un test mitjançant el programa de llenguatge convencional Verilog i la correcta síntesi d'aquest mitjançant el programa Quartus II. Per fer-ho m'he basat en el projecte realitzat sobre la comunicació SPI de la pàgina Nandland el qual podem trobar a la següent pàgina : <https://github.com/nandland/spi-master>.

(*): Adjunt a aquest document, hauria de trobar-se un altre document "...\\Lorenzo_SPI\\doc\\Datasheet_SPI_MLF.pdf" on detallem la configuració d'un mòdul Màster SPI.

Enviar dades per MOSI

Per enviar dades per la senyal MOSI de forma correcta hem realitzat una màquina d'estats de Mealy, la qual té com a sortida el valor de la senyal Chip Select i conté tres estats diferents:

- **IDLE (00):** Aquest estat el considerarem l'estat inicial, tots els registres passen al seu valor inicial, ens trobarem en aquest estat un cop iniciem el sistema, fem un reset asíncron o acabem d'enviar una trama, un cop ens trobem en aquest estat, el Màster indicarà que està preparat per rebre una trama.
- **TRANSFER (01):** Aquest estat és l'estat de transferència, entrarem en aquest estat un cop ens trobem al estat IDLE i tinguem unes dades vàlides a enviar, un cop ens trobem en aquest estat començarem a enviar dades per la senyal MOSI.
- **CS_INACTIVE (10):** Aquest estat és l'estat d'inactivitat de la senyal Chip Select, entrarem en aquest estat un cop haguem acabat d'enviar totes les dades que volíem enviar, mentre ens trobem en aquest estat simplement esperarem un període de temps el qual haurem configurat prèviament, d'aquesta manera tenim versatilitat alhora d'utilitzar dispositius amb comunicació SPI que requereixin un petit delay entre trames, no entrarem en aquest estat en el cas que estiguem enviant més d'un byte i no haguem acabat d'enviar-los tots.



Il·lustració 1: Diagrama d'estats per poder enviar dades per MOSI: i_rst_n (reset asíncron negat), o_SPI_CS_n (senyal Chip Select negada), r_CS_n (registre Chip Select negat), i_TX_DV (Senyal que ens indica que tenim dades a enviar), r_TX_count (registre que conta els bytes que queden per enviar en la transferència), r_CS_Inactive_Counts (Registre que compta el delay necessari pel CS)

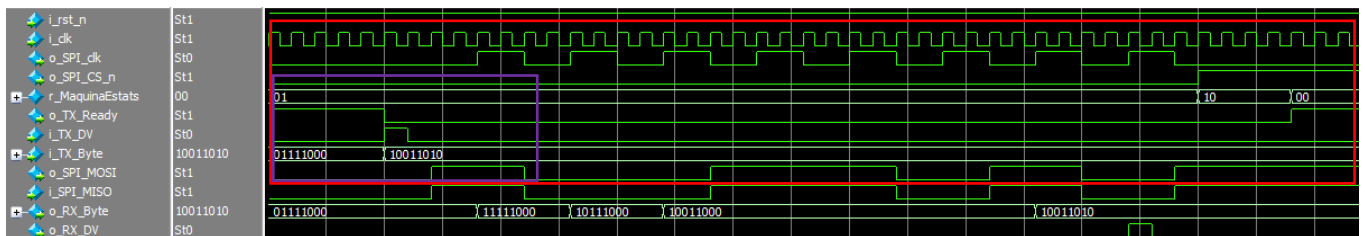
Rebre dades

Per rebre les dades de la senyal MISO simplement cridem al codi a més baix nivell perquè gestioni tota la arribada de dades, en el nostre model TOP a més hem realitzat un curtcircuit entre els pins MISO i MOSI, per poder realitzar un eco i demostrar la correcta implementació d'aquest protocol de comunicació, el qual hauria de poder rebre i enviar dades alhora, el que si que tindrem en compte serà quants bytes hem rebut mitjançant un comptador el qual anirà contant de forma ascendent i tornarà al seu valor inicial un cop acabi de rebre dades.

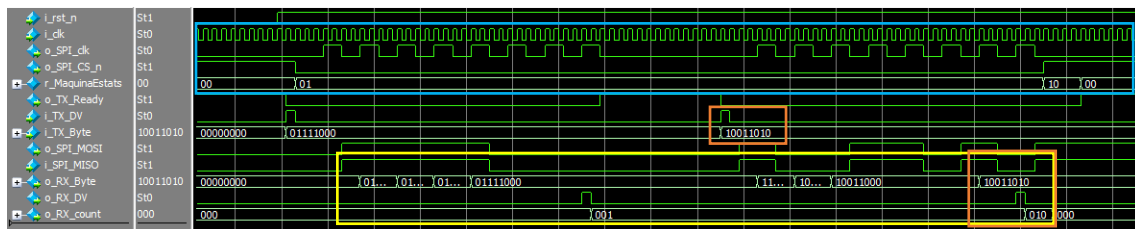
Test

Per realitzar el test de la comunicació SPI de forma correcta i veure que tots els canals funcionaven correctament, he realitzat un curtcircuit entre els pins MOSI i MISO, a continuació he generat una tasca la qual em permet enviar qualsevol byte en format hexadecimal i l'utilitzaré per enviar dos bytes diferents 0x78 i 0x9A, amb un delay entre ells per veure analitzar de forma correcta totes les senyals en aquest informe, un cop he enviat aquest byte, el rebo i per consola podem veure un missatge el qual ens indica quin byte hem enviat i la resposta que hem obtingut, si el test és satisfactori aquests dos bytes haurien de coincidir.

A continuació, presentaré dues imatges (**Realitzades amb configuració SPI 0**), en una d'elles podrem veure el correcte funcionament per enviar dades per MOSI i en l'altra podrem veure com aquestes dades es reben per MISO, com es gestiona el Chip Select junt amb la màquina d'estats i per acabar, la correcta generació del rellotge SPI.



Il·lustració 2: Test on podem veure com s'envia un byte per la senyal MOSI i rebem l'eco per MISO. Vermell = [1] Lila = [2]



Il·lustració 3: Test on podem veure com s'envien dos bytes per la senyal MOSI i es reben per la senyal MISO.

Groc = [3] Taronja = [4] Blau = [5]

Enviar dades

En la Il·lustració 2 podem veure com enviem el segon byte d'una trama per MOSI, el byte es carrega primer en paral·lel i, seguint el rellotge de la comunicació SPI el va passant en sèrie per MISO [1], també podem veure com el Màster dona la senyal que està preparat per enviar, però s'espera fins que el codi a més baix nivell doni la senyal per començar a enviar el nou byte [2].

Rebre dades

En la Il·lustració 3 podem veure com rebem dos bytes diferents en sèrie per MISO i els convertim en paral·lel, un cop s'ha convertit tot el byte, donem una senyal que el byte ha arribat correctament i incrementem el compte de bytes que hem rebut [3] podem veure el correcte funcionament del eco que hem generat [4].

Generació SPI_clk i màquina d'estats

En la *Il·lustració 3* podem veure la escalada del rellotge del sistema per enviar i rebre correctament els bytes mitjançant el rellotge de la nostra interfície SPI (Més detalls al Datasheet adjunt ^(*)) a més a més, també podem veure el correcte funcionament de la màquina d'estats junt amb el Chip Select, el qual només s'activa alhora de transmetre dades i s'espera un petit delay quan ha acabat la transmissió [5].

TOP

A continuació presentarem una taula on podem veure totes les d'entrades i sortides de la nostra interfície SPI, junt amb quin cable els connectem per realitzar correctament el test.

Pin	I/O	Wire
Senyals de control		
i_rst_n	I	r_rst_n
i_clk	I	r_clk
Senyals de TX (MOSI)		
i_TX_counter	I	r_Master_TX_count
i_TX_Byte	I	r_Master_TX_Byte
i_TX_DV	I	r_Master_TX_DV
o_TX_Ready	O	w_Master_TX_ready
Senyals de RX (MISO)		
o_RX_count	O	w_Master_RX_count
o_RX_DV	O	w_Master_RX_DV
o_RX_Byte	O	w_Master_RX_Byte
Interfície SPI		
o_SPI_clk	O	w_SPI_clk
i_SPI_MISO	I	w_SPI_MOSI
o_SPI_MOSI	O	w_SPI_MOSI
o_SPI_CS_n	O	w_SPI_CS_n

Conclusions

Com a conclusions podem dir que el test del mòdul "SPI_Master_MLF.v" ⁽⁺⁾ s'ha realitzat amb èxit i sense cap dificultat, hem pogut veure que el nostre SPI Màster ens permet enviar una trama de més d'un byte, mantenint el CS actiu mitjançant l'aplicació de forma correcta d'una màquina d'estats. Aquest, també controla la sortida i entrada de les senyals MISO i MOSI, les quals es gestionen mitjançant senyals i registres interns, que ens permeten poder interpretar dades en paral·lel i convertir-les en sèrie o viceversa.

(+): Adjunt a aquest document, hauria de trobar-se un altre document "...\\Lorenzo_SPI\\rtl\\SPI_Master_MLF.v" on trobem la configuració del Màster per la nostra comunicació SPI.