



**Politecnico
di Torino**

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATIONS
Master degree in Electronic Engineering

Laboratorio di Sistemi Digitali Integrati

Laboratorio 2: SPI

Autore:

Roberta Leo, Manuel Cataldo, Lorenzo Leli

5 febbraio 2024

Indice

1	Descrizione progetto	2
2	Timing Diagram	2
3	Descrizione FSM	3
4	Descrizione Datapath	4
5	Testbench	6
6	Caricamento su scheda	7

1 Descrizione progetto

L'obiettivo di questo laboratorio è progettare un'interfaccia slave SPI che permetta ad un master SPI di effettuare operazioni di lettura e scrittura su dei registri.

Il blocco da progettare campiona i dati in ingresso sul fronte di discesa del clock Sck e invia dati al master ogni fronte di salita di quest'ultimo.

2 Timing Diagram

Per lo sviluppo del progetto dello Slave SPI, sono stati prima descritti i due cicli di scrittura e lettura riportati rispettivamente nelle figure 1 e 2.

Questi due timing diagram sono accomunati dallo stesso processo di lettura del CMD (comando da eseguire) e dell'ADDR (indirizzo), questi avvengono grazie a dei segnali di Shift Enable utili per leggere l'ingresso seriale del Master, entrambi sono arrestati da due segnali, rispettivamente TC8 e TC15. Questi segnali sono utili per leggere prima gli 8 bit del comando e poi gli altri 8 dell'indirizzo, dopo aver letto tutti questi, in base al comando da eseguire opereremo due diverse operazioni.

Nel caso della scrittura, continueremo a leggere l'ingresso seriale per poi generare un segnale di Write al termine della lettura. Qui per finire l'operazione utilizziamo sempre il segnale TC15, così da leggere i 16 bit di DIN (dati in ingresso).

Nel caso della lettura invece attiviamo l'uscita seriale diretta al Master per leggere il dato restituito dall'operazione di Read precedentemente richiesta. In questo caso non useremo lo stesso segnale TC15, ma il segnale TC0 per ritrovare il clock Sck sempre sul suo fronte di salita.

I TC sono stati creati in questo modo così da non dover mai resettare il counter se non quando nSS è disattivo.

Il segnale RST.M rappresenta il reset della memoria in blocco, utile solo nel caso di reset completo della macchina.

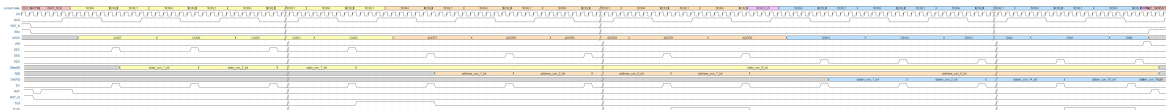


Figura 1: Ciclo di scrittura dello Slave SPI.

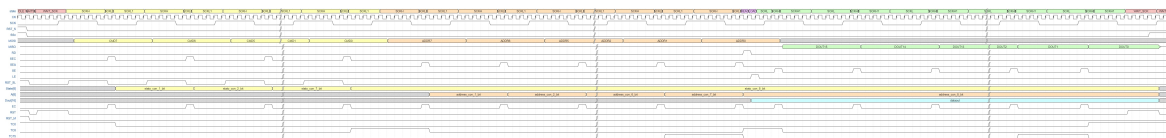


Figura 2: Ciclo di lettura dello Slave SPI.

3 Descrizione FSM

Dal timing diagram è stata ricavata la Finite State Machine per la realizzazione dell'interfaccia SPI, come si vede dalla figura 3.

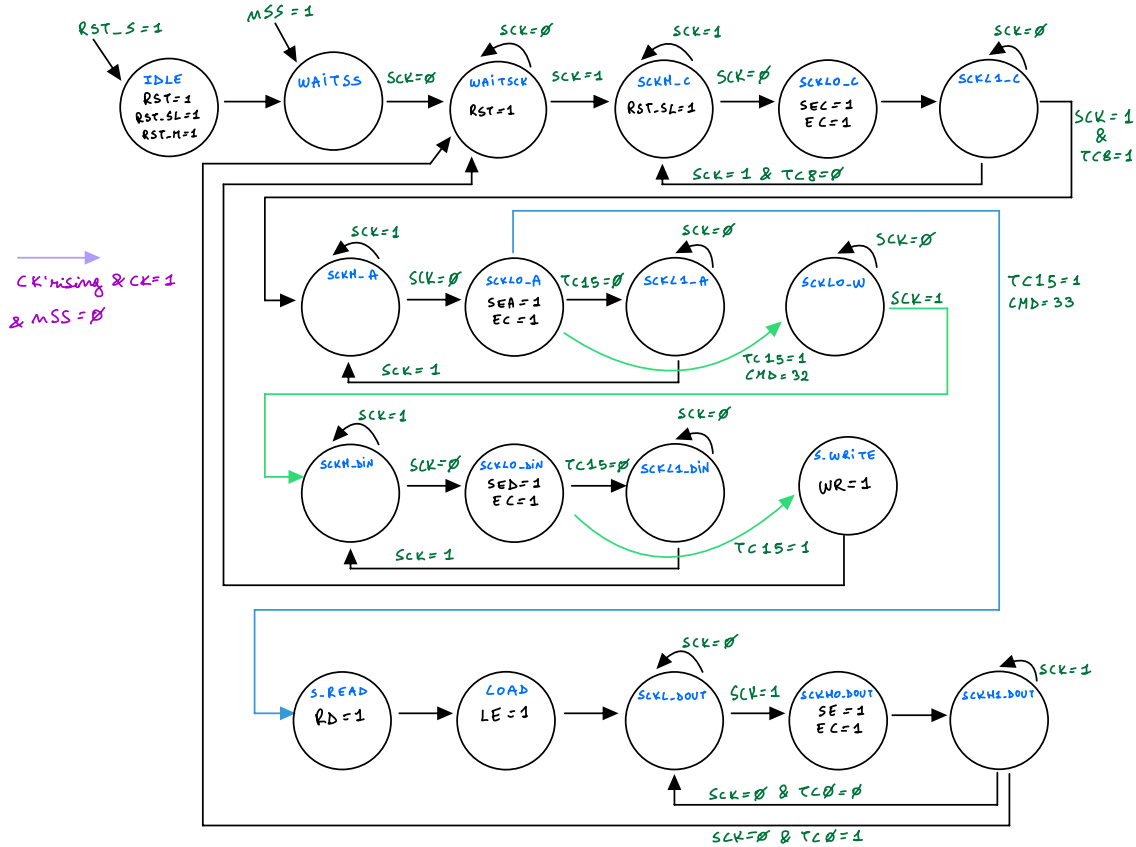


Figura 3: FSM.

Il passaggio allo stato successivo si ha al fronte attivo del clock dello slave e con Slave Select attivo. Se quest'ultimo segnale viene disattivato, si ritorna nello stato di IDLE.

1. **IDLE** : Si entra in questo stato quando il reset esterno asincrono viene attivato. Qui vengono attivati tutti i segnali che effettuano il reset completo della macchina. Quando il segnale di reset esterno viene disasserto, si passa allo stato di WAITSS.
2. **WAITSS** : E' lo stato di attesa in cui la macchina non effettua nessuna operazione. Si rimane in questo stato finché il segnale NSS è disasserto oppure $Sck=0$.
3. **WAITSCK** : Si rimane in questo stato se $Sck=0$. Quando questo segnale va al livello logico 1 si passa allo stato SCKH_C.
4. **SCKH_C** : Effettua il reset sul blocco SHIFT REGISTER with LOAD per portare il MISO in alta impedenza. Si rimane in questo stato finché $Sck=1$, in caso contrario si passa allo stato SCKL0_C.
5. **SCKL0_C** : Qui viene attivato lo Shift Enable del comando e l'Enable del counter.

6. **SCKL1_C** : Si rimane in questo stato finché Sck=0. Quando questo segnale passa al livello logico 1, se TC8=0 si ritorna in SCKH_C, altrimenti se TC8=1, si passa allo stato SCKH_A in cui si procede a caricare l'indirizzo.
7. **SCKH_A** : Si rimane in questo stato finché Sck=1 altrimenti si passa allo stato SCKL0_A.
8. **SCKL0_A** : Qui viene attivato lo Shift Enable dell'indirizzo e l'Enable del counter.
Quando il segnale TC15 è attivo, se il comando, caricato nei cicli precedenti, è uguale a "32", si avrà una scrittura e quindi lo stato successivo sarà SCKL0_W, mentre, se il comando è 33, si passerà allo stato S_READ.
Se il counter non è ancora arrivato a contare 15, si andrà nello stato SCKL1_A.
9. **SCKL1_A** : Si rimane in questo stato finché Sck=0 altrimenti si passa allo stato SCKH_A.
10. **SCKL0_W** : Si rimane in questo stato finché Sck=0 altrimenti si passa allo stato SCKH_DIN.
11. **SCKH_DIN** : Si rimane in questo stato finché Sck=1 altrimenti si passa allo stato SCKL0_DIN.
12. **SCKL0_DIN** : Qui viene attivato lo Shift Enable del dato in entrata da scrivere e l'Enable del counter.
Se il segnale TC15=0 si passa allo stato SCKL1_DIN, altrimenti si passa allo stato S_WRITE.
13. **SCKL1_DIN** : Si rimane in questo stato finché Sck=0 altrimenti si passa allo stato SCKH_DIN.
14. **S_WRITE** : Il segnale di Write è attivato e successivamente si torna allo stato WAIT_SCK.
15. **S_READ** : Il segnale di Read è attivato.
16. **LOAD** : Il segnale di Load Enable è attivato quindi i bit inizieranno ad essere caricati sul MISO.
17. **SCKL_DOUT** : Si rimane in questo stato finché Sck=0 altrimenti si passa allo stato SCKH0_DOUT.
18. **SCKH0_DOUT** : Qui viene attivato lo Shift Enable del dato in uscita che si sta leggendo e l'Enable del counter.
19. **SCKH1_DOUT** : Si rimane in questo stato finché Sck=1. Quando questo segnale passa al livello logico 0, se TC0=0 si ritorna in SCKL_DOUT, altrimenti se TC0=1, si passa allo stato WAIT_SCK.

Il vantaggio operativo della macchina appena descritta, è che il master può comunicare con lo slave SPI effettuando operazioni (che siano di lettura o scrittura) in sequenza senza che il segnale Slave Select debba essere disassorbito e riassorbito tra due operazioni consecutive. Questo comporta una maggior velocità di comunicazione tra i due dispositivi in quanto non è presente latenza tra un'operazione e la successiva effettuata con lo stesso slave.

4 Descrizione Datapath

Descrizione dei componenti usati per la realizzazione dell'interfaccia SPI:

1. **FSM**: macchina a stati per il controllo dei segnali che permettono di far funzionare l'interfaccia SPI.
2. **SHIFT REGISTER**: nel progetto vengono utilizzati tre shift register: uno da 8 bit per i bit di stato, un altro sempre da 8 bit per i bit di indirizzo e infine l'ultimo da 16 bit per il dato.
Essi vengono abilitati uno per volta tramite i corrispettivi segnali di Enable e possono essere resettati tramite il segnale di RST.

Gli shift register permettono di memorizzare uno a uno i bit trasmessi dal Master sulla linea MOSI (input seriale) e di fornire in uscita il vettore degli ultimi 8 o 16 bit memorizzati (output parallelo).

3. **SHIFT REGISTER with LOAD**: questo componente viene controllato dalla FSM tramite i segnali di RST_SL (reset), SE (shift enable) e LE (load enable).
Il componente ha un input parallelo su 16 bit, i quali vengono caricati tramite il segnale di LE, e un output seriale, il quale trasmette i dati in uscita sulla linea MOSI quando il segnale SE diventa attivo.
4. **Counter**: contatore controllato dalla FSM tramite i segnali di CE (counter enable) e RST.
Esso viene utilizzato per monitorare il numero di dati mandati o ricevuti tramite le corrispondenti linee di MISO o MOSI.
5. **TC (terminal counter)**: componente che genera un segnale in ingresso alla FSM il quale permette di far sapere a quest'ultima quando il contatore raggiunge un numero specificato.
Nel progetto sono stati implementati tre TC chiamati TC0, TC8, TC15 che rilevano quando il contatore raggiunge i corrispondenti valori di 0, 8 e 15.

Il disegno del Datapath è riportato in figura 4.

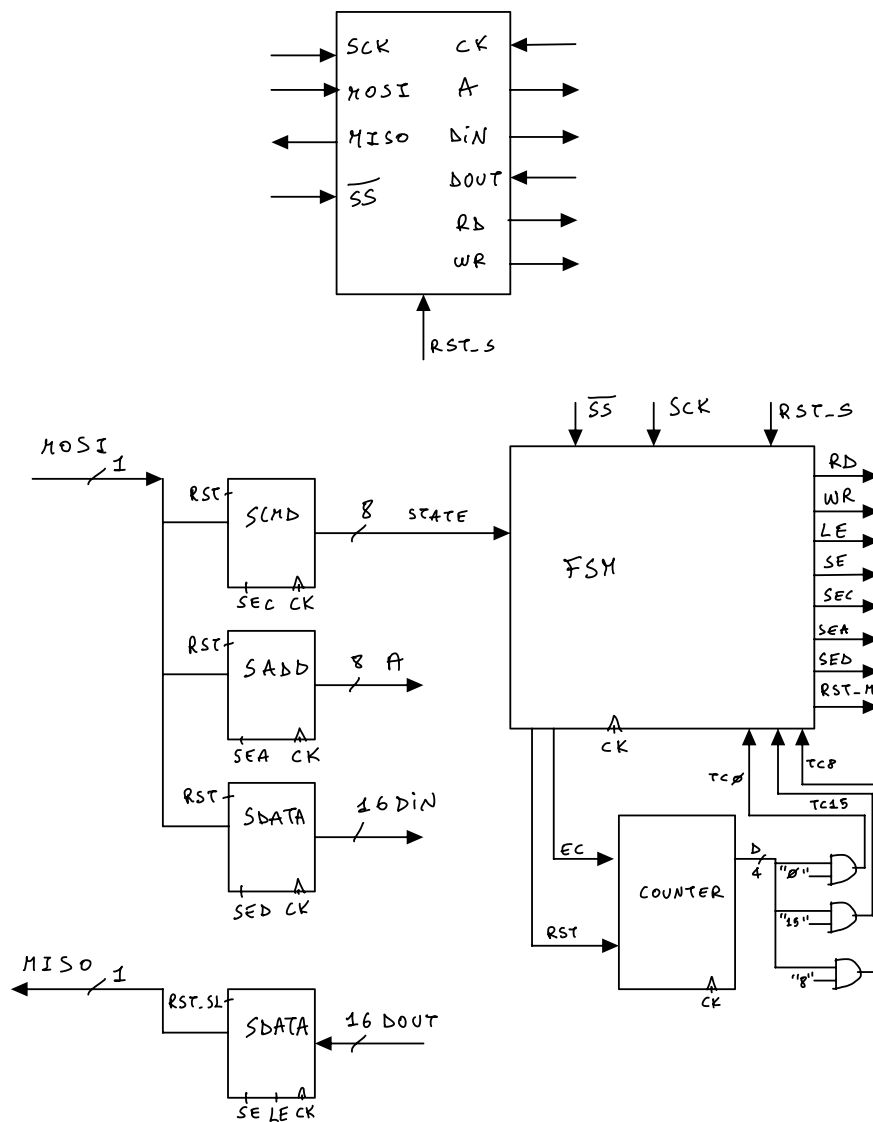


Figura 4: Datapath.

5 Testbench

Per validare il progetto è stata utilizzata una generica memoria e sono state svolte due verifiche, una per il comando di scrittura e una per il comando di lettura, rispettivamente raffigurate nelle figure 5 e 6. Si nota come queste figure e quelle ideate nella sezione 2 non sono dissimili tra loro.

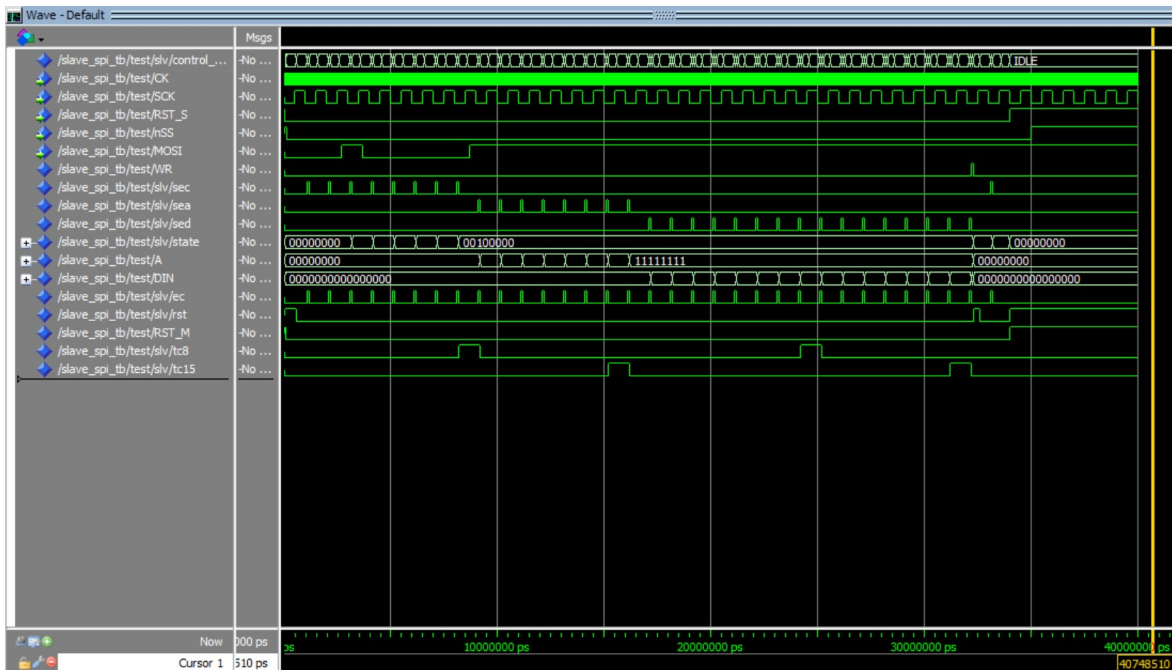


Figura 5: Testbench nel caso della scrittura.

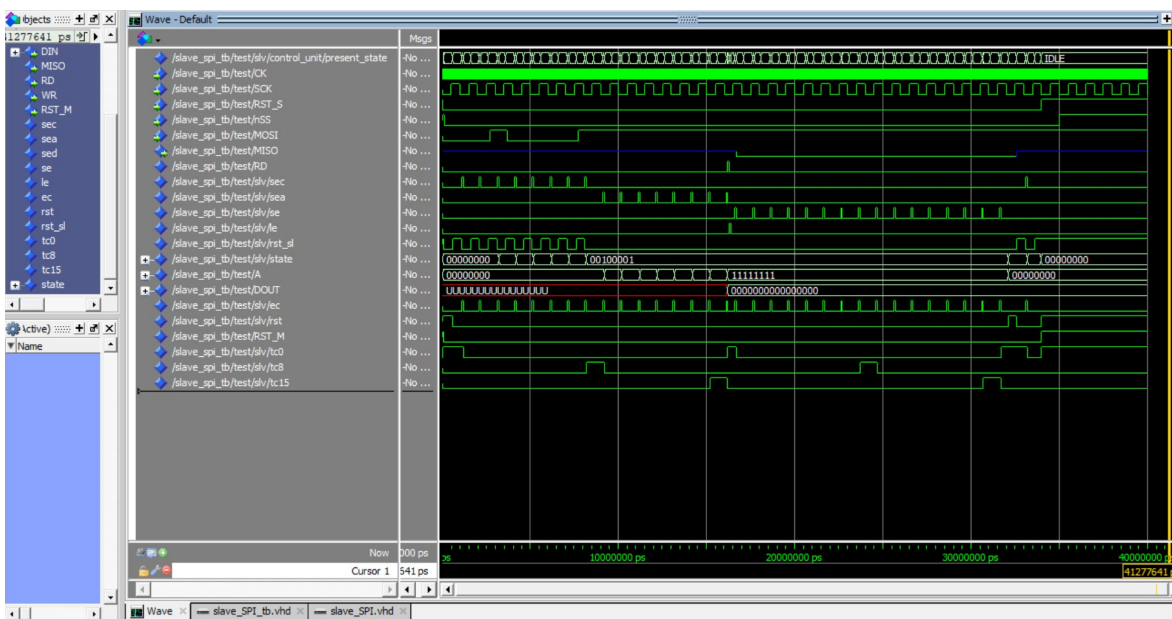


Figura 6: Testbench nel caso della lettura.

6 Caricamento su scheda

Mediante un emulatore di terminale seriale, mandando il comando di write ad alcuni registri, nella forma "Wrddddd" e leggendoli successivamente attraverso il comando "Rrrr", sono state svolte alcune verifiche del funzionamento del programma.