



POLITECNICO DI MILANO

Prova finale di Reti Logiche

Autori:

*Edoardo Gennaretti (CP: 10743751, NM: 955326),
Samuele Pietro Galli (CP: 10710025, NM: 955426)*

Supervisore:

Prof. Fabio SALICE

12 maggio 2023

Indice

1	Specifiche	2
1.1	Descrizione del progetto	2
1.2	Descrizione delle scelte progettuali	3
2	Architettura	4
2.1	Disegno FSM	5
2.2	Schematic dell'intero progetto	5
3	Risultati Sperimentali	6
3.1	Sintesi	6
4	Testing	7
4.1	Test 1	7
4.2	Test 2	8
4.3	Test 3	9
5	Conclusione	10

1 Specifiche

1.1 Descrizione del progetto

L'obiettivo del progetto è quello di realizzare un componente HW che, ricevuti in ingresso un indirizzo di memoria e l'informazione riguardo il canale di uscita richiesto, stampi il contenuto dell'indirizzo sul canale specificato.

Si presentano 7 interfacce, di cui 2 ingressi primari (W e START), entrambi da 1 bit, e 5 uscite (Z0, Z1, Z2, Z3, DONE), di cui, le prime 4 (da 8 bit), sulle quali andranno riportati tutti i bit della parola di memoria, e DONE da 1 bit. Sono inoltre presenti un segnale di reset (RESET) e un segnale di clock (CLK), unici per il componente.

La specifica chiede di implementare un modulo hardware in VHDL che si interfacci con una memoria e che riceve informazioni tramite un ingresso seriale a un bit riguardo ad una locazione di memoria il cui contenuto deve essere indirizzato verso uno dei quattro canali di uscita disponibili.

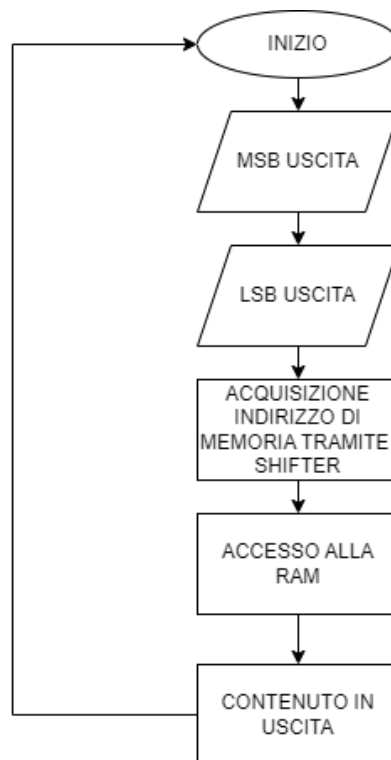
Il modulo deve essere in grado di leggere la sequenza sull'ingresso primario seriale W, quando il segnale START è '1'. La sequenza è organizzata in 2 bit di intestazione, seguiti da N bit (con $0 \leq N \leq 16$). I due bit di intestazione identificano il canale d'uscita sul quale deve essere indirizzato il messaggio. Gli N bit ricostruiscono l'indirizzo di memoria composto sempre da 16 bit. Il segnale START rimane attivo per almeno 2 cicli di clock e non più di 18 cicli di clock.

Le uscite Z0, Z1, Z2 e Z3 sono inizializzate a zero, e rimangono a tale valore fintanto che il segnale DONE rimane basso. Quando DONE diventa 1, il loro valore cambia, mostrando il dato appena letto dalla RAM sul canale associato, mentre su tutte le altre uscite si mantiene l'ultimo valore salvato. Il segnale DONE rimane attivo per un solo ciclo di clock, dopo il quale si abbassa nuovamente.

1.2 Descrizione delle scelte progettuali

La descrizione del componente HW prevede l'utilizzo di 14 segnali e il funzionamento si divide in tre fasi principali:

- **SELEZIONE DELL'USCITA:** Fase iniziale nella quale si acquisiscono i bit descrittivi del canale di uscita.
- **TRASCRIZIONE DELL'INDIRIZZO DI MEMORIA:** Seconda fase nella quale si ricevono gli N bit dell'indirizzo in un registro. La memorizzazione avviene con uno shift a sinistra di tutto l'indirizzo acquisito fino a quel momento, e del salvataggio del segnale W nella posizione meno significativa. In questa fase avviene anche la correzione della ricezione in quanto la sequenza di bit risulta "shiftata" di una posizione di troppo a sinistra.
- **CONCLUSIONE E PRESENTAZIONE DEL MESSAGGIO:** Fase conclusiva nella quale viene indirizzato e presentato il messaggio sull'uscita corrispondente a quella specificata.



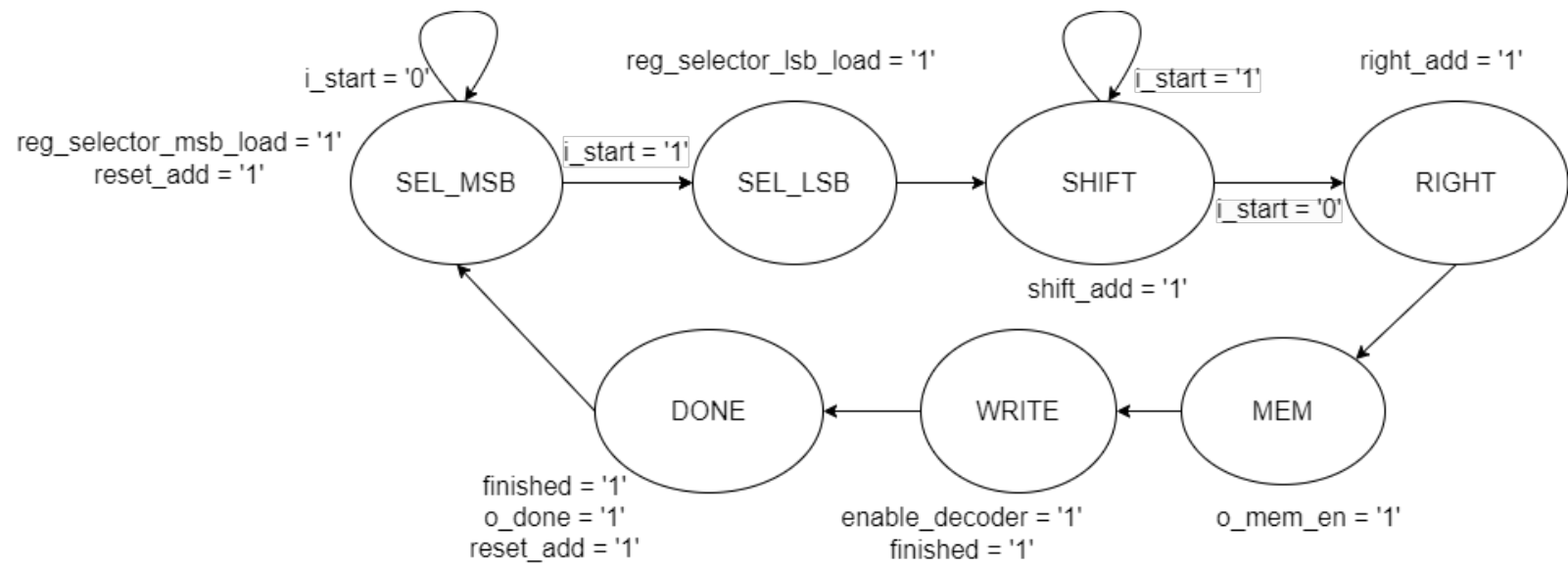
2 Architettura

Di seguito viene mostrata la FSM (Final State Machine) in versione semplificata, sono omessi tutti gli archi che da qualsiasi stato tornano a **SEL_MSB** quando **i_rst='1'**. Tutti i segnali non specificati sono posti a '0'.

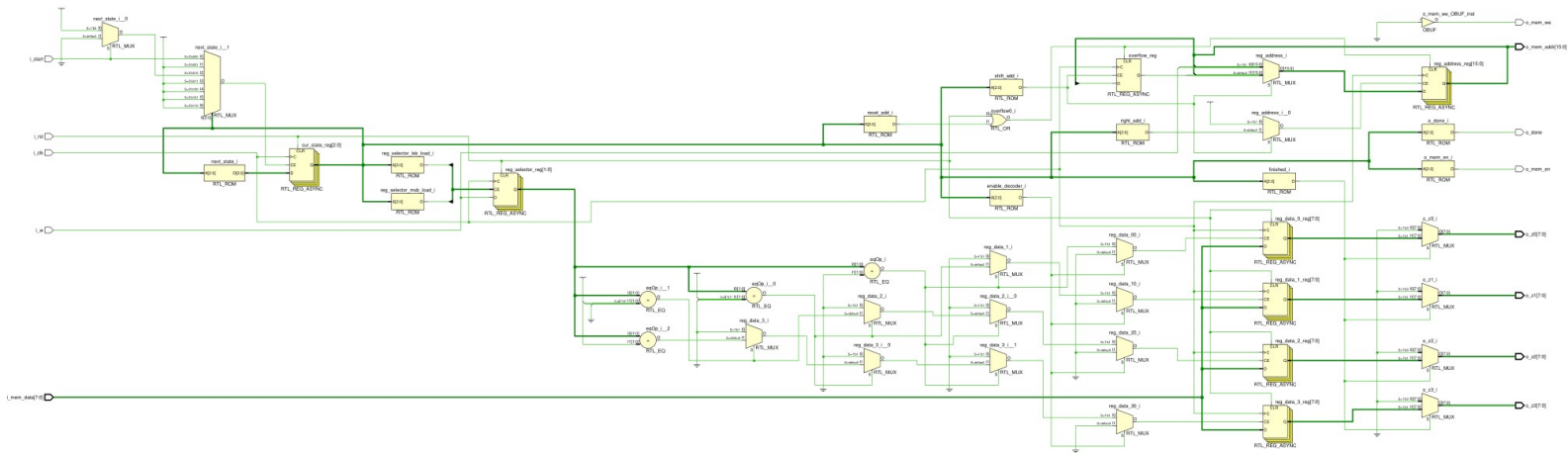
Descrizione di ogni stato:

- **SEL_MSB**: stato di inizializzazione dei registri nel quale viene acquisito il most significant bit dell'uscita. Nel momento in cui **i_start** diventa alto si passa allo stato **SEL_LSB**, altrimenti nulla accade. Il segnale **reg_selector_msb_load** viene posto a 1.
- **SEL_LSB**: stato di acquisizione del least significant bit dell'uscita. Il segnale **reg_selector_lsb_load** viene posto a 1.
- **SHIFT**: stato di memorizzazione dell'indirizzo di memoria mediante l'utilizzo del registro **reg_address**. Si rimane in questo stato fino a quando **i_start** è alto, nel momento in cui si abbassa si passa allo stato **RIGHT**. Il segnale **shift_add** viene posto a 1.
- **RIGHT**: stato di ultimazione dell'indirizzo acquisito, nel quale si shifta di una posizione a destra **reg_address**, inserendo nella posizione più significativa il registro **overflow**.
- **MEM**: stato in cui viene salvato l'indirizzo di memoria completato e si aspetta il dato dalla RAM. Il segnale **o_mem_en** viene posto a 1, in modo da abilitare la lettura nella RAM.
- **WRITE**: stato di scrittura del dato ricevuto dalla RAM nell'opportuna uscita. **enable_decoder** e **finished** vengono posti a '1', il primo per salvare il dato della RAM sull'uscita corretta, il secondo per mostrare ciò che è salvato nei registri di uscita.
- **DONE**: stato finale nel quale il segnale **finished** rimane alto. I segnali **o_done** e **reset_add** vengono posti a '1', il primo per mostrare che si è completato l'intero processo, il secondo per inizializzare nuovamente i registri **reg_address** e **overflow**.

2.1 Disegno FSM



2.2 Schematic dell'intero progetto



3 Risultati Sperimentali

3.1 Sintesi

Il progetto è stato realizzato e testato sulla versione di **Vivado 2019.1**, specificando come FPGA target Artix-7 xc7a200tfbg484-1. È stato generato il seguente report di sintesi:

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	37	0	134600	0.03
LUT as Logic	37	0	134600	0.03
LUT as Memory	0	0	46200	0.00
Slice Registers	54	0	269200	0.02
Register as Flip Flop	54	0	269200	0.02
Register as Latch	0	0	269200	0.00
F7 Muxes	0	0	67300	0.00
F8 Muxes	0	0	33650	0.00

Per gli stati e il consumo invece:

Type	Power	Util%
Signals	0.448 W	16%
Logic	0.285 W	10%
I/O	2.059 W	74%
Device Static	0.142 W	5%

4 Testing

Conclusa la fase di descrizione e sintesi del componente, si è passati alla fase di test, nella quale il componente ha risposto correttamente a tutti i testbench a cui è stato sottoposto. Si riportano di seguito alcune simulazioni Behavioural, che mostrano in particolare casi limite.

4.1 Test 1

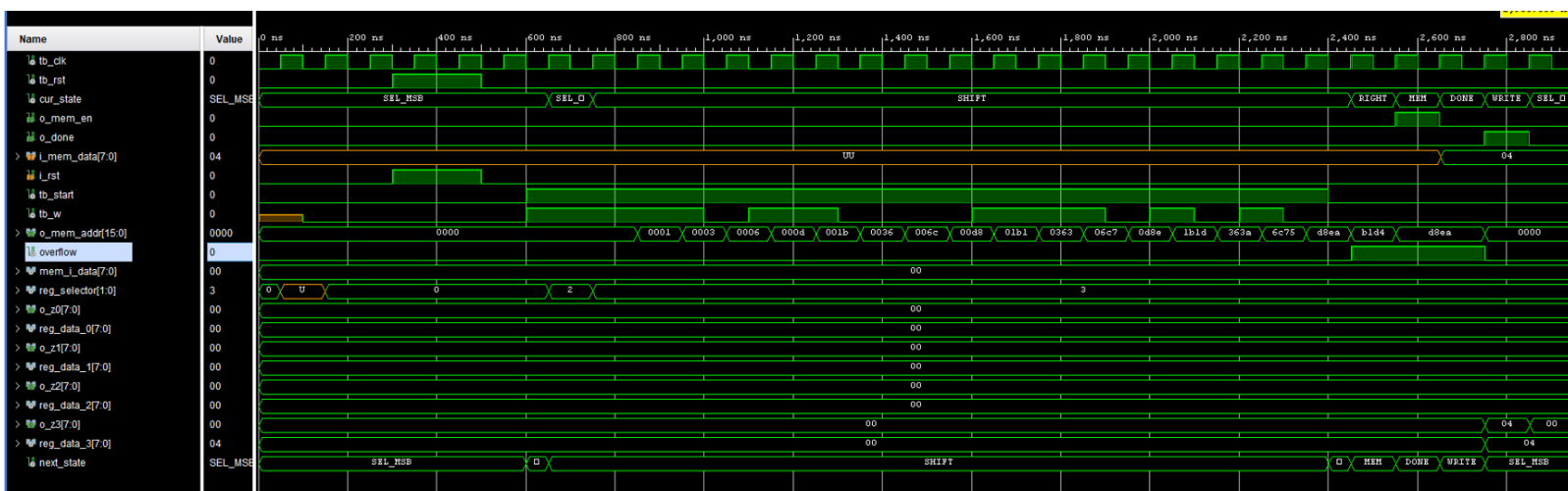


Figura 1: Simulazione Behavioural del primo test.

Questo test è stato riportato in quanto vengono utilizzati tutti i 16 bit dell'indirizzo, e viene posto ad '1' il registro di overflow, mostrandone l'utilità.

4.2 Test 2

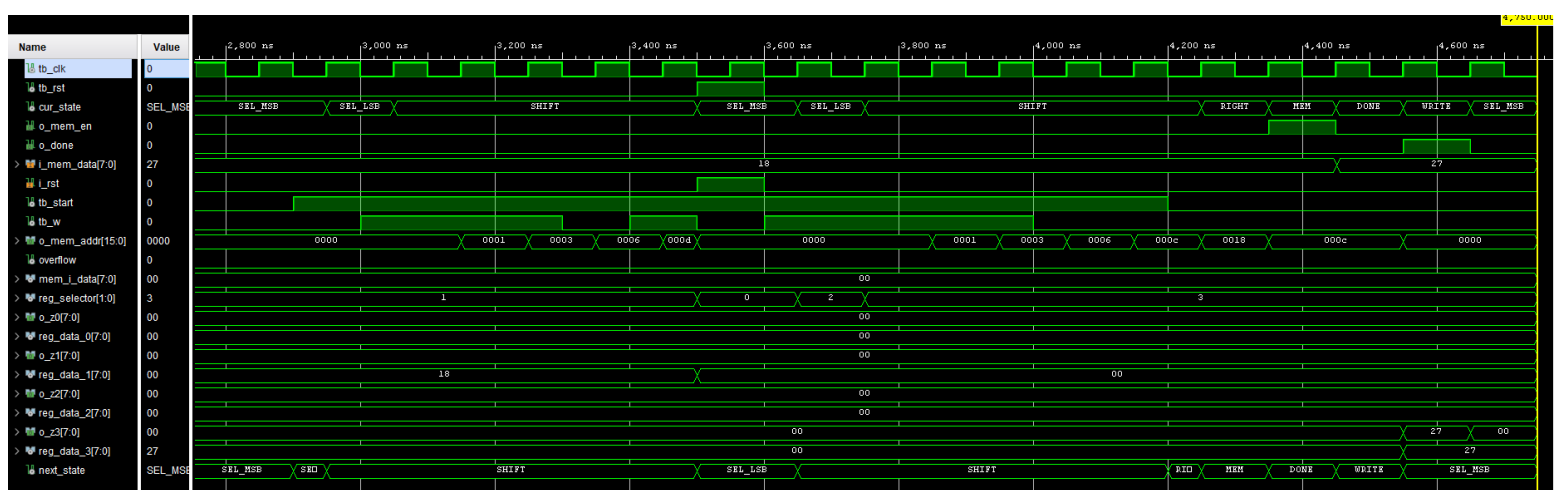


Figura 2: Simulazione Behavioural del secondo test.

La simulazione di questo test è stata riportata in quanto prevede un caso di reset che viene posto a '1', mentre il componente è in lettura.

4.3 Test 3

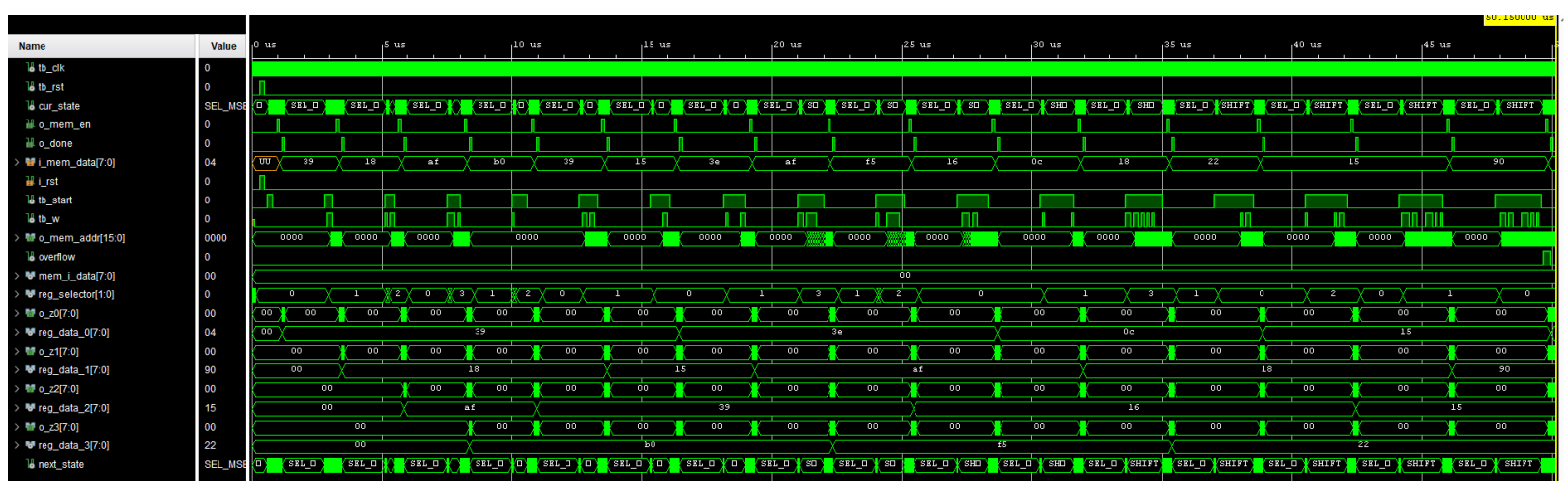


Figura 3: Simulazione Behavioural del terzo test.

La simulazione di questo test è stata riportata in quanto può essere considerata una simulazione 'stress' con funzionamenti ripetuti.

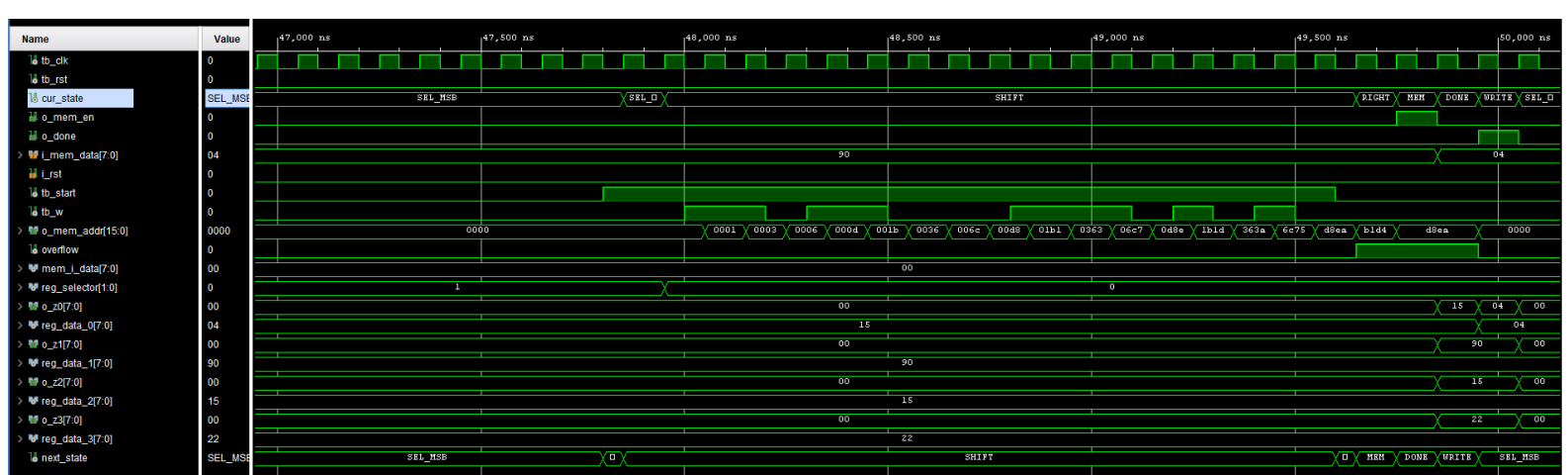


Figura 4: Particolare della simulazione del terzo test riportata in quanto viene attivato il registro overflow.

5 Conclusione

Studiare e realizzare questo progetto ci ha permesso di comprendere il funzionamento dell'indirizzamento della memoria ad un livello astrattivo molto basso, andando ad approfondire un aspetto largamente usato in moltissimi linguaggi di programmazione.

Ci è stato possibile inoltre analizzare e gestire casi diversi, cercando di ottenere una copertura totale per tutti i casi estremi. La collaborazione è stato un aspetto molto importante in quanto ha portato idee differenti sia per individuare una soluzione sia per trovare casi che potessero bloccare il corretto funzionamento del componente.