

PROGETTO DI RETI LOGICHE

Scaglione prof. Palermo

Martina Pegoraro

codice persona: 10584373

INDICE

1. Introduzione
2. Architettura
3. Risultati sperimentali e testbench
4. Conclusioni

INTRODUZIONE

La prova finale è ispirata al metodo di codifica a bassa dissipazione di potenza denominato “Working Zone”. Questo metodo è usato per trasformare degli indirizzi, tenendo conto della loro appartenenza ad una determinata Working Zone.

Le Working Zones sono gruppi di indirizzi, che partono da un indirizzo base (già scritto in memoria) e hanno dimensione fissa.

L'indirizzo da verificare viene confrontato con le Working Zones e, se appartiene ad una di esse, l'indirizzo viene trasformato in: numero della Working Zone di appartenenza seguito da un offset calcolato in base a quanto l'indirizzo è distante dalla base della WZ (tutto codificato in binario), il tutto anticipato da un bit aggiuntivo posto ad 1; se non appartiene a nessuna Working Zone, l'indirizzo è lasciato invariato ma viene aggiunto un bit aggiuntivo all'inizio posto a 0.

46

| | | | | | | | | |
|---|----|----|---|----|----|----|----|-----|
| 4 | 13 | 24 | X | 56 | 61 | 89 | 95 | 108 |
|---|----|----|---|----|----|----|----|-----|

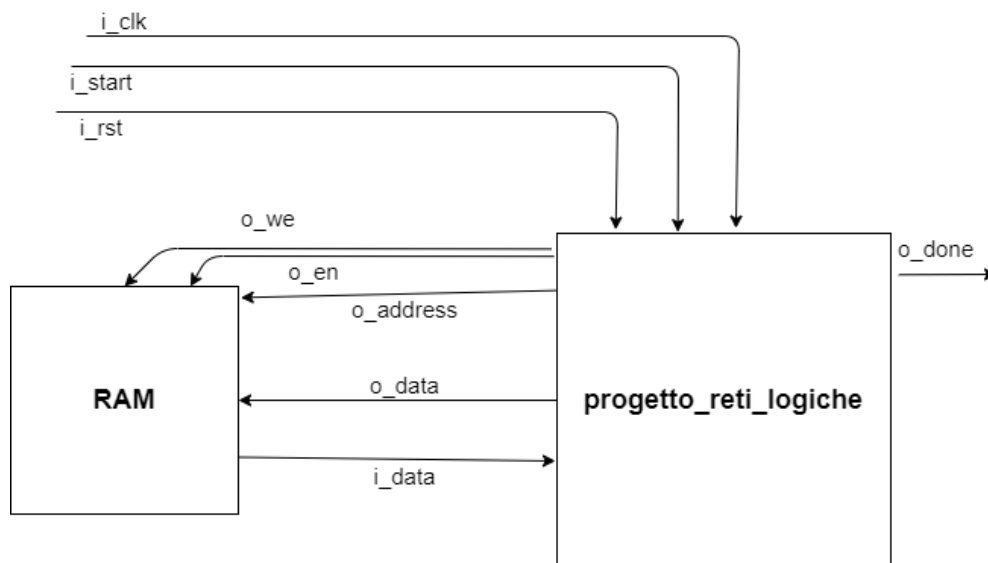
58

| | | | | | | | | |
|---|----|----|----|---|----|----|----|-----|
| 4 | 13 | 24 | 56 | ✓ | 61 | 89 | 95 | 108 |
|---|----|----|----|---|----|----|----|-----|

ARCHITETTURA

L'interfaccia del componente è stata donata nelle specifiche. Esso si deve interfacciare con un chip RAM contenente le Working Zones e l'indirizzo di partenza, su cui poi si scrive l'indirizzo in output.

Di seguito è riportato il modo in cui il componente si interfaccia con la RAM e con i segnali esterni:



Per il componente Hardware si è deciso di realizzare un unico processo, che implementa una macchina a stati finiti definendo gli stati e controlla il fronte di discesa del clock.

Il segnale di reset può alzarsi in qualsiasi momento, in qualunque stato la macchina si trovi, se il segnale di reset si alza, essa ricomincia da capo.

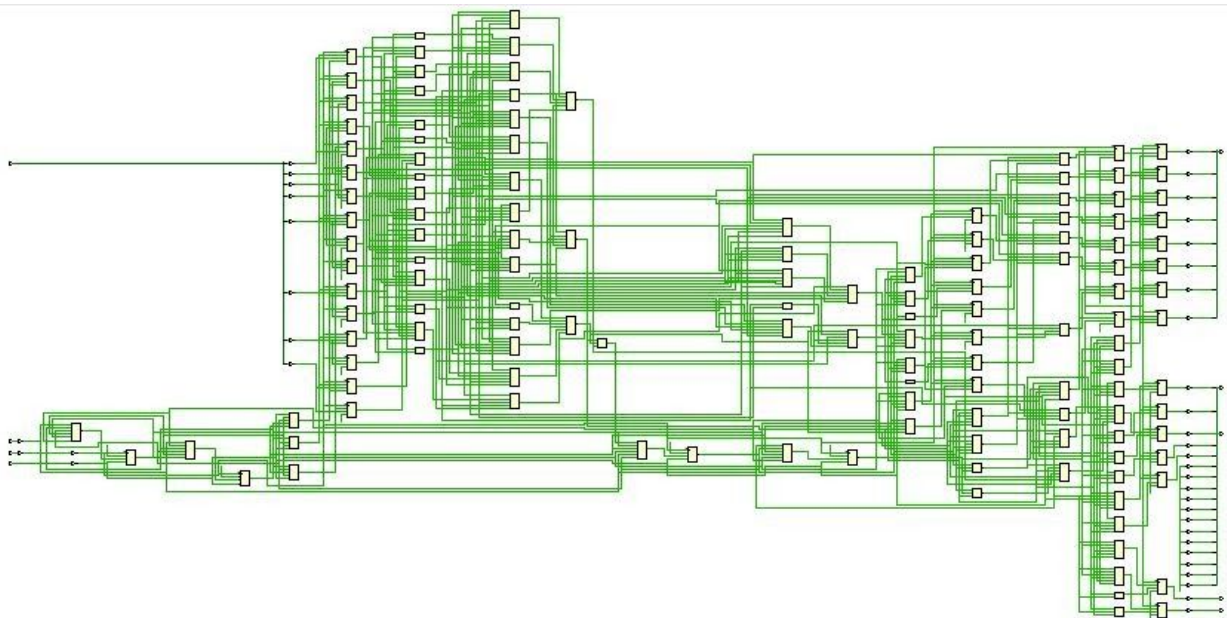
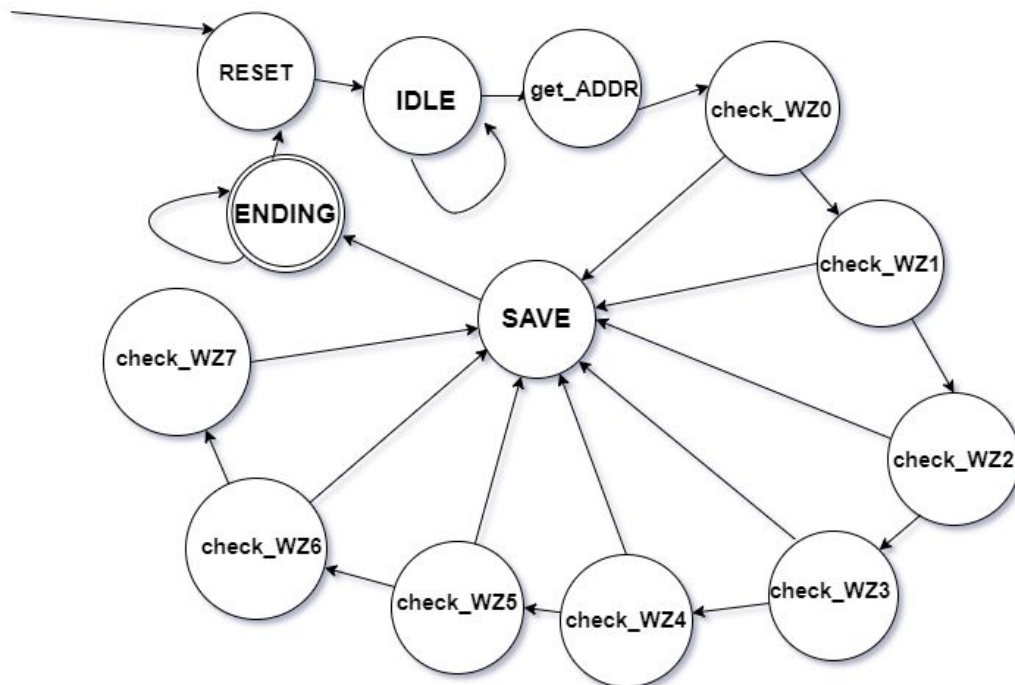
L'algoritmo viene eseguito così:

1. Inizialmente si parte dallo stato "reset", dove si azzerano tutte le variabili.
2. Subito dopo si passa allo stato "idle", che è uno stato di attesa e che, quando il segnale di start diventa uguale a 1, si prepara a leggere l'indirizzo da confrontare con la WZ dalla RAM che viene successivamente salvato.
3. In seguito, viene confrontato l'indirizzo di partenza con ciascuna WZ e se esso appartiene ad una di queste, viene calcolato l'output e la FSM salta allo stato

“save”.

4. Se l'algoritmo arriva alla fine delle WZ possibili e non ha trovato nessun match, passa comunque allo stato “save” che si occuperà di scrivere il nuovo indirizzo nella RAM e passerà allo stato “ending”.
5. Lo stato “ending” rimane in loop finchè start è uguale a 1 o non arriva un segnale di reset.

Di seguito vengono riportati lo schema della FSM e lo schema del design elaborato da Vivado.

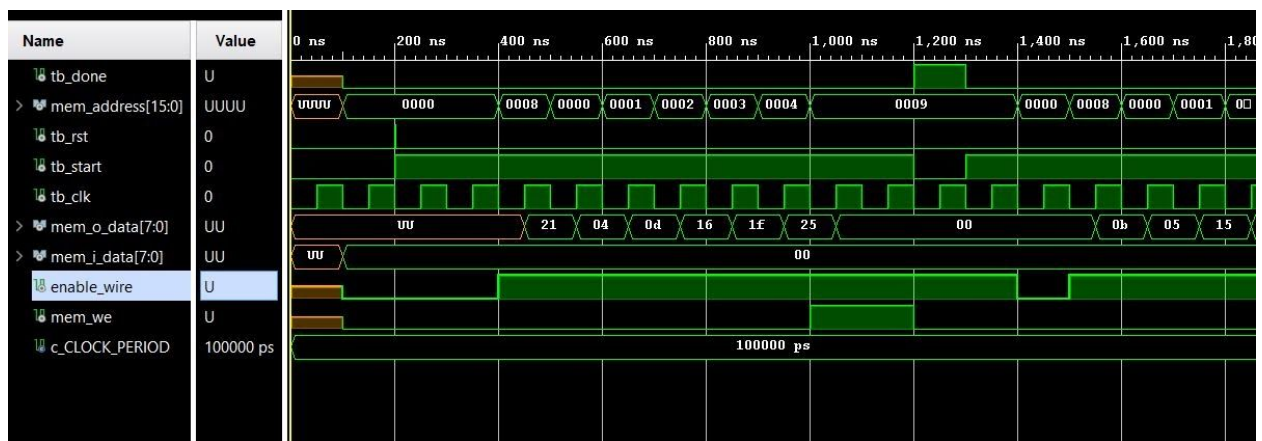


RISULTATI SPERIMENTALI E TESTBENCH

Oltre ai testbench forniti, sono stati eseguiti altri test che hanno testato:

- tutte le WZ possibili
- per ogni WZ, tutti gli offset possibili
- il valore dell'indirizzo più alto possibile (256)
- il valore dell'indirizzo più basso possibile(0)

Di seguito è riportato lo schema di uno dei test eseguiti.



CONCLUSIONI

Il componente ha superato con successo tutti i test eseguiti, sia nelle Behavioral Simulations sia nelle Post-Synthesis Functional Simulations, quindi si ritiene sia funzionante e rispetti tutte le caratteristiche richieste.