Prova Finale
Progetto di Reti Logiche

Studente: Alessandro Messori

Codice Persona: 10598510

Numero Matricola: 889232

Professore: Fabio Salice

Politecnico di Milano
Corso di Reti Logiche

Indice

1 Introduzione

L’obiettivo di questo progetto è quello di realizzare tramite un componente VHDL un sistema di codifica in “Working Zone”, cioè un algoritmo che dato in ingresso un indirizzo ed una serie di working zones (cioè intervalli di indirizzi di dimensione fissa) restituisce in uscita l’indirizzo scritto con una specifica codifica se presente in una delle wz.

Nella nostra implementazione le wz e l’indirizzo sono numeri interi da 0 a 127 e sono memorizzate in una memoria RAM con indirizzi di 8 bit (il primo bit in ingresso è sempre posto a ‘0’).
Le 8 Working Zones hanno dimensione fissa di 4 elementi (compreso l’indirizzo di partenza) e sono memorizzate negli indirizzi della RAM che vanno da 0 a 7.
Gli indirizzi 8 e 9 sono riservati rispettivamente all’indirizzo da codificare e a quello che verrà codificato tramite l’esecuzione del nostro modulo.

Il modulo VHDL dovrà leggere dalla RAM le working zones e se l’indirizzo da codificare non rienta in nessuna di esse viene riscritto in uscita senza nessun cambiamento,in caso contrario viene salvato in memoria secondo la seguente codifica:

- il primo bit viene sempre posto a ‘1’
- i bit dal secondo al quarto sono usati per codificare l’indice della working zones di appartenenza in binario
- i bit dal quinto all ottavo sono usati per codificare in one hot la posizione dell’indirizzo nella working zones in questo modo:
  - posizione 1 → “0001”
  - posizione 2 → “0010”
  - posizione 3 → “0100”
  - posizione 4 → “1000”

Nella pagina seguente sono presenti due tabelle che rappresentano due possibili set di memoria dopo l’esecuzione del nostro modulo.

1.1 Tabelle di Esempio

Caso con valore presente in working zone 3 :

Indirizzo	Contenuto Indirizzo	Descrizione
0	65	WZ0
1	3	WZ1
2	20	WZ2
3	90	WZ3
4	10	WZ4
5	78	WZ5
6	32	WZ6
7	45	WZ7
8	91	Address da codificare
9	178	Address codificato (1 011 0010)

Caso con valore non presente in nessuna working zone :

Indirizzo	Contenuto Indirizzo	Descrizione
0	65	WZ0
1	3	WZ1
2	20	WZ2
3	78	WZ3
4	10	WZ4
5	90	WZ5
6	32	WZ6
7	45	WZ7
8	28	Address da codificare
9	28	Address codificato

2 Architettura

Ho optato per una macchina di Mealy sincronizzata per agevolare le fasi di design e sviluppo.
Il modulo è composto da vari segnali di input e output,alcuni usati per comunicare con la memoria RAM e altri per la gestione dell'esecuzione.
In particolare l'ingresso di reset viene usato per riportare la macchina nel suo stato iniziale,quello di start serve a far partire la codifica e l'uscita done viene usata dalla macchina per segnalare il termine del processo di scrittura in memoria dell'indirizzo codificato.
I segnali interni più significativi sono quelli di counter a addwz, usati rispettivamente per memorizzare il contatore dell'indirizzo di memoria corrente e l'indirizzo da codificare.

La macchina è composta da 7 stati:

- S0: Stato a cui si accede dopo un reset, vengono assegnati valori di default a tutte i segnali e variabili, si rimane in questo stato fino a che il segnale di start non viene posto a '1', a quel punto si passa allo sta to S1;

- S1: Lettura dell'indirizzo da codificare e inizializzazione del contatore di memoria a 0, lo stato successivo viene posto a S2;

- S2: Se il contatore di memoria è superiore a 7 significa che il l'indirizzo non fa parte di nessuna delle 8 working zones ,viene quindi posto lo stato successivo a S3; in caso contrario viene verificato se l'indirizo da codificare sia compreso nella working zone in memoria all'indirizzo corrente,e in caso positivo viene posto lo stato successivo a S4, in caso negativo viene incrementato il contatore e si rimane nello stato corrente;

- S3: Non essendo l'indirizzo parte di nessuna working zone,esso viene riscritto in memoria senza nessun cambiamento;lo stato successivo viene posto a S5;

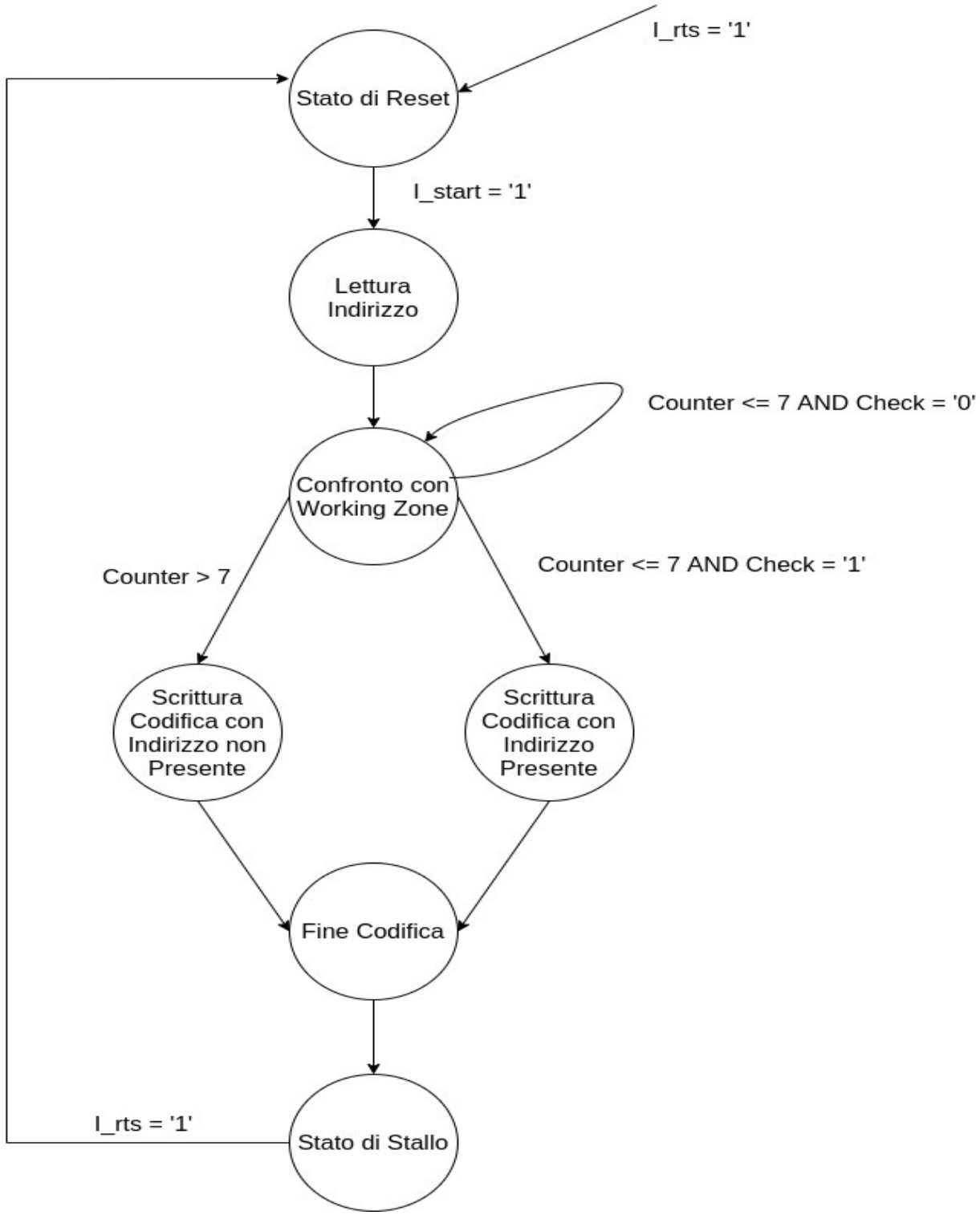
- S4: L'indirizzo fa parte della working zone nell'indirizzo corrente,viene calcolata e scitta in memoria la codifica corrispondente,lo stato successivo viene posto a S5;

- S5: La codifica è terminata,il segnale di done viene posto a '1' e si passa allo stato S6;

- S6: Stato di stallo,si attende che che il segnale di start venga riposto a '0', a quel punto il segnale di done viene abbassato e si ritorna allo stato di reset S0.

In questa implementazione le working zones vengono rilette ad ogni nuova esecuzione anche in assenza di un segnale di reset.
Questo tipo di architettura può avere lo svantaggio di avere tempi di esecuzione più lunghi rispetto al caso in cui le working zones vengano lette solo una volta e poi memorizzate, ma presenta i vantaggi di avere un numero di stati e segnali interni abbastanza limitato, che equivale ad avere quindi un consumo di area contenuto.
Nella pagina successiva viene presentato un diagramma sintetico della macchina a stati.

2.1 Diagramma della Macchina a Stati



N.B. Il segnale Check nel diagramma rappresenta il controllo sulla presenza dell'indirizzo da codificare nella working zone che sta venendo letta da memoria in quel ciclo.

3 Risultati Sperimentali

Come previsto nella fase di progettazione, il report di sintesi mostra un'implementazione del circuito che utilizza un numero abbastanza limitato di componenti e registri.
Nelle foto sottostanti è possibile osservare la descrizione fatta dal report dei componenti e delle celle utilizzate per sintetizzare il progetto.

```
Start RTL Component Statistics
---------------------------------------------
Detailed RTL Component Info :
+---Adders :
       2 Input     16 Bit       Adders := 1
       2 Input      8 Bit       Adders := 4
+---Registers :
                   16 Bit    Registers := 1
                    8 Bit    Registers := 1
+---Muxes :
       2 Input     16 Bit        Muxes := 4
       7 Input     16 Bit        Muxes := 2
       7 Input      8 Bit        Muxes := 2
       2 Input      4 Bit        Muxes := 1
       2 Input      3 Bit        Muxes := 4
       7 Input      3 Bit        Muxes := 1
       2 Input      2 Bit        Muxes := 1
       2 Input      1 Bit        Muxes := 1
       7 Input      1 Bit        Muxes := 5
---------------------------------------------
```

```
Report Cell Usage:
+------+-------+------+
|      |Cell   |Count |
+------+-------+------+
|1     |BUFG   |     1|
|2     |CARRY4 |     6|
|3     |LUT1   |     1|
|4     |LUT2   |     2|
|5     |LUT3   |    23|
|6     |LUT4   |    22|
|7     |LUT5   |    20|
|8     |LUT6   |    38|
|9     |FDCE   |     3|
|10    |FDRE   |    24|
|11    |IBUF   |    11|
|12    |OBUF   |    27|
+------+-------+------+
```

Questo è invece il messaggio di completamento della sintesi:

```
INFO: [Common 17-83] Releasing license: Synthesis
15 Infos, 0 Warnings, 0 Critical Warnings and 0 Errors encountered.
synth_design completed successfully
synth_design: Time (s): cpu = 00:00:07 ; elapsed = 00:00:08 . Memory (MB): peak = 1729.387 ;
gain = 429.211 ; free physical = 179 ; free virtual = 3486
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 .
Memory (MB): peak = 1729.387 ; gain = 0.000 ; free physical = 179 ; free virtual = 3486
```

Durante la sintesi sono state utilizzate 78 LUT e 27 FF.

4 Simulazioni

Dopo aver passato i test bench forniti dal professore, ho testato il corretto funzionamento del mio modulo con test di progressiva difficoltà e in più condizioni differenti possibili, correggendo gli errori che venivano evidenziati durante il testing.
I casi di test considerati sono stati i seguenti:

- Indirizzo non presente in nessuna working zone
- Indirizzo presente in working zone ad ogni indice possibile (da 0 a 7)
- Indirizzo presente in working zone ad ogni posizione possibile (da 0 a 3)
- Indirizzo presente in working zone con valori limite (0 e 124)
- Ripetizione dei test precedenti con più esecuzioni consecutive senza segnale di reset
- Ripetizione dei test precedenti con più esecuzioni consecutive con segnale reset inserito in modo casuale.

Tutti i test sopracitati sono stati ripetuti molte volte cambiando il contenuto della RAM in modo automatico usando un programma di generazione dei casi di test.
Il componente VHDL finale supera correttamente tutti i test bench sia in pre-sintesi che in post-sintesi.

5 Conclusioni

Il prodotto finale è quindi un modulo VHDL sintetizzabile,funzionante con un periodo di clock di 100 ns, in grado di svolgere la specifica correttamente sotto uno spettro di casi di test molto ampio, progettato per minimizzare la dimensione del circuito finale ed il numero di registri utilizzati. Questo tipo di design penalizza leggermente i tempi di esecuzione, ma che rimangono comunque buoni e valgono per la singola esecuzione nel caso ottimo (indirizzo presente nella working zone in posizione 0) 1100 ns e in quello pessimo (indirizzo non presente in nessuna working zone) 1900 ns.