

Prova Finale di Reti Logiche

Truong Kien Tuong

May 1st, 2020

Matricola: 887907
Codice Persona: 10582491
Docente: Gianluca Palermo

1 Introduzione

La prova prevede l'implementazione in VHDL di una macchina che opera su una memoria e svolge la seguente operazione.

La macchina deve per prima cosa leggere il primo byte dalla memoria che identifica il numero di parole che sono state fornite come input, questa informazione è importante per capire quando la macchina deve terminare la lettura.

Dopodiché ogni parola successiva viene tradotta in due parole di memoria che vengono scritte progressivamente in un'altra parte della memoria.

Le parole di memoria sono interpretate con un flusso continuo e l'uscita è un flusso di lunghezza doppia. Visto che i byte vengono tradotti come un unico flusso l'output di una certa parola dipende anche da quelle precedenti e non possono essere processate separatamente.

1.1 Esempio

Indirizzo	Valore	Codifica binaria
0	2	0000 0010
1	35	0010 0011
2	161	1010 0001

Questo stato della memoria si traduce nell'input [35, 161] e in questo caso la lunghezza dell'input $W=2$ quindi mi aspetto una lunghezza dell'output $Z=4$.

Indirizzo	Valore	Codifica binaria
1000	13	0000 1101
1001	206	1100 1110
1002	97	0110 0001
1003	195	1100 0011

Rappresenta l'output [13, 206, 97, 195] dove la codifica dei numeri [13, 206]

è l'output prodotto dai bit dal 35 in ingresso. Mentre [97, 195] sono il risultato dei bit presenti nella la codifica di 161.

1.2 Ipotesi Progettuali

- Si utilizza la scheda Artix-7 FPGA xc7a200tfbg484-1
- Ogni byte può contenere numeri da 0 a 255.
- La quantità di numeri in ingresso (W) è contenuta in una parola da un byte quindi anche il numero massimo di parole da tradurre è 255.
- Dato che l'input occupa al massimo 256 byte posso scrivere sui byte successivi quindi l'output parte sempre dal millesimo indirizzo di memoria che sicuramente non contiene l'input

2 Architettura

L'implementazione è stata realizzata in un solo modulo per semplicità, si possono distinguere due parti all'interno della macchina che cooperano per raggiungere il risultato. È presente una macchina che simula il funzionamento dell'automa a stati finiti fornito nella specifica, questa è responsabile della traduzione uno a due dei bit. Inoltre è presente una seconda macchina principale che si occupa della lettura e scrittura dei flussi di bit e tratta la prima come una scatola nera. Ciascuna macchina ha una variabile separata che contiene il suo stato.

2.1 Descrizione ad alto livello

Questa è la descrizione a parole delle fasi che l'esecuzione segue. All'interno del programma ciascuna di queste diverse fasi corrisponde ad uno stato (o a più stati).

1. Inizio:
 - (a) Legge il primo byte di memoria.
 - (b) Inizializza il punto di lettura al secondo byte e quello di scrittura al millesimo.
2. Traduzione:
 - (a) Legge un byte e sposta il punto di lettura.
 - (b) Simula un passo dell'automa dando in ingresso un bit dal byte appena letto. (Ripetuto 4 volte)
 - (c) L'output dell'automa ora contiene 8 bit quindi viene salvato in una parola di memoria e si incrementa l'indirizzo di scrittura.
 - (d) Se sono rimasti bit da tradurre nel byte letto torna al punto 2.b

- (e) Altrimenti se sono rimasti byte da leggere torna al punto 2.a
- (f) Altrimenti termina.

Ogni operazione di lettura o scrittura nella memoria si scompone in più stati nel programma. Ad esempio nel caso della lettura: richiesta della lettura, attesa della fine dell'operazione, utilizzo del dato letto o salvataggio in una variabile.

2.2 Diagramma degli stati

2.3 Macchina a stati finiti?

3 Risultati sperimentali

Risultati sperimentali

4 Conclusioni

Conclusioni