Prova Finale (Progetto di reti logiche)

Politecnico di Milano
Anno scolastico 2018-2019

Documentazione

Alessia Paccagnella
Matricola: -
Codice persona: -
Professor G.Palermo, Tutor D.Conficconi

Indice

1 Specifiche del progetto

1.1 Obbiettivo

E' dato uno spazio bidimensionale quadrato (256x256), le coordinate di un punto in tale spazio, un insieme di altri N = 8 punti (detti *centroidi*) nello stesso spazio e una maschera di N bit rappresentante i centroidi da considerare. Ogni bit della maschera assume il valore '1' se il centroide corrispondente a quella posizione è da considerare per la valutazione, '0' altrimenti. Le coordinate nello spazio dei centroidi e del punto da valutare sono memorizzate in una memoria già implementata.
L'obbiettivo del progetto è l'implementazione un componente hardware descritto in VHDL che restituisca in uscita una maschera rappresentante i centroidi validi che si trovano a distanza (Manhattan distance) minima dal punto di inizio.

1.2 Interfaccia del componente

Il componente ha la seguente intefaccia:

```
entity project_reti_logiche is
Port (
    i_clk         : in  std_logic;
    i_start       : in  std_logic;
    i_rst         : in  std_logic;
    i_data        : in  std_logic_vector(7 downto 0);
    o_address     : out std_logic_vector(15 downto 0);
    o_done        : out std_logic;
    o_en          : out std_logic;
    o_we          : out std_logic;
    o_data        : out std_logic_vector(7 downto 0)
);
end project_reti_logiche;
```

Prima che la computazione del componente cominci, la RAM viene inizializzata con il suo contenuto e il segnale i_start viene portato alto.
Si può leggere dalla RAM scrivendo l'indirizzo di lettura in o_address, portando alto o_en e abbassando o_we. Il dato viene scritto su i_data dopo il successivo fronte di salita del clock.
Si può scrivere nella RAM portando alto il segnale o_we e scrivendo il dato su o_data.
Il segnale o_done viene alzato solo al termine della computazione, in seguito il segnale i_start viene portato basso.

2 Strumenti di sintesi utilizzate

Per la sintesi del componente da realizzare sono stati utilizzati:
- XILINX VIVADO WEBPACK;
- Target FPGA xc7a200tfbg484-1.

3 Soluzione e scelte implementative

Per la progettazione del componente è stata scelta l'implementazione tramite macchina a stati finiti (FSM). Essa descrive il funzionamento dell'algoritmo, a partire dall'inizializzazione dei valori delle variabili utilizzate, passando per la lettura della memoria, per le valutazioni opportune e per il

calcolo dei risultati, arrivando infine alla scrittura in memoria.

Sono stati utilizzati tre processi:
1. ***MAIN_PROCESS***, che contiene le istruzioni da eseguire in corrispondenza del fronte di discesa del clock in ogni stato e ad ogni esecuzione;
2. ***state_sequence***, che, se viene rilevato un segnale di reset in qualsiasi istante dell'esecuzione permette il ritorno allo stato RESET, altrimenti aggiorna lo stato corrente al successivo per ogni ciclo di clock;
3. ***transitions***, che indica la sequenza di stati da seguire, assegnando per ciascuno il **next_state**.

4 Registri utilizzati

L'algoritmo implementato utilizza diversi registri:
1. **addr**: viene utilizzato per accedere alla memoria, a fini di lettura o scrittura. E' stata definita una regola all'inizio del codice percui o_address <= std_logic_vector(addr).
2. **xcentroide**: memorizza l'ascissa, letta dalla memoria, del centroide corrente.
3. **ycentroide**: memorizza l'ordinata, letta dalla memoria, del centroide corrente.
4. **xpunto**: memorizza l'ascissa, letta dalla memoria, del punto di riferimento.
5. **ypunto**: memorizza l'ordinata, letta dalla memoria, del punto di riferimento.
6. **maschera**: memorizza la maschera, letta dalla memoria, da tenere come riferimento.
7. **contatore**: viene utilizzato per tener traccia del centroide che si sta valutando: esso viene incrementato e quando arriva a "00000000" indica che tutti i centroidi sono stati valutati.
8. **distanza**: memorizza la distanza calcolata corrente.
9. **distanzaminima**: memorizza la distanza minima trovata fino a quel punto di esecuzione. Può essere aggiornata nel caso se ne trovi una minore.
10. **mascherauscita**: memorizza la maschera di uscita che rappresenta i centroidi a distanza di Manhattan minima dal punto in considerazione.

5 FSM

5.1 Descrizione generale

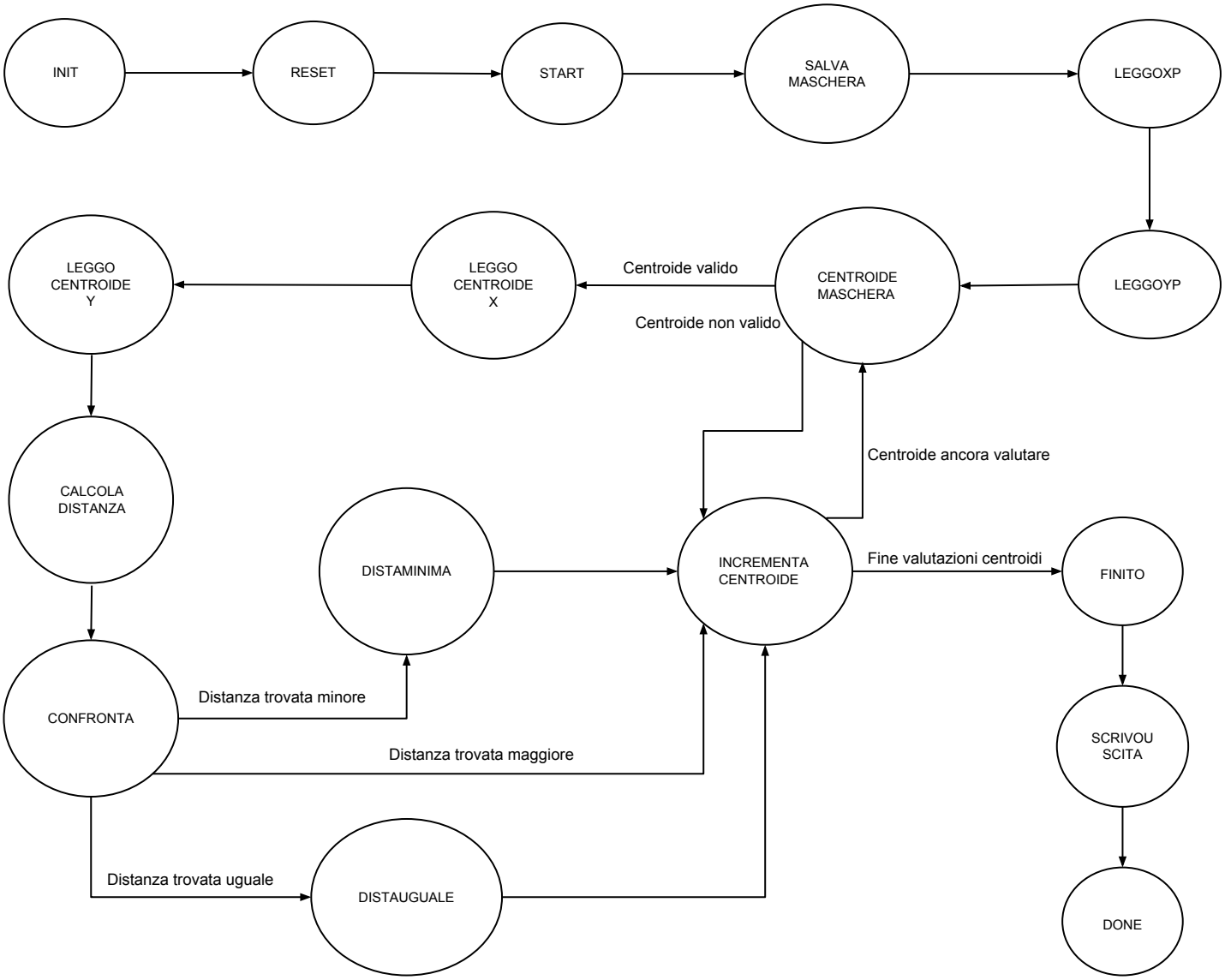
Il primo passo è il salvataggio della maschera d'ingresso nel registro *maschera*. In seguito vengono salvate, in *xpunto* e *ypunto*, le coordinate del punto da cui calcolare le distanze. Il registro *contatore* tiene traccia di quale sia il centroide in fase di valutazione, a partire da "00000001". Ad ogni iterazione il contatore viene fatto scorrere a sinistra di un bit, in modo da arrivare a considerare tutti gli 8 centroidi. Nel momento in cui diventa "00000000" significa che ha valutato anche l'ottavo centroide e quindi ha finito, per cui si procede alla scrittura in memoria del contenuto del registro *mascherauscita*. Per determinare se il centroide i-esimo è valido, si esegue per ogni iterazione l'AND bit a bit tra il contatore e la maschera d'ingresso. Se il risultato è diverso da "00000000" allora il centroide è da valutare, quindi si procede a calcolare la distanza, altrimenti si procede alla valutazione del prossimo centroide, se esiste. Se la nuova distanza è minore della distanza minima precedente, il registro *mascherauscita* viene aggiornato con tutti i bit a 0 eccetto il bit in posizione i-esima che viene posto a 1 e *distanzaminima* viene aggiornato col nuovo valore. Se la nuova distanza è uguale a quella precedente, all'output verra aggiunto un 1 in posizione i-esima. Se invece la nuova distanza è maggiore della precedente l'output non cambia e si procede alla valutazione del prossimo centroide, se esiste.

5.2 Descrizione dei 17 stati

1	INIT	E' lo stato di inizializzazione delle variabili. Una volta ricevuto il primo segnale di reset non è più possibile tornare a questo stato.
2	RESET	Inizializza tutti i segnali ai valori necessari perchè l'algoritmo implementato funzioni correttamente.
3	START	Questo stato fa partire l'esecuzione. Abilita quindi la lettura della memoria tramite l'assegnamento o_en <= '1'.
4	SALVOMASCHERA	Legge e salva la maschera all'indirizzo addr = "0000000000000000". Si porta poi all'indirizzo "0000000000010001" per permettere la prossima lettura.
5	LEGGOXP	Legge e salva l'ascissa del punto in questione all'indirizzo addr. Incrementa addr.
6	LEGGOYP	Legge e salva l'ordinata del punto in questione all'indirizzo addr. Inizializza il contatore a "00000001" per la valutazione del primo centroide. Riporta addr all'indirizzo "00000001", dove si troverà la coordinata del primo centroide da valutare.
7	CENTROIDEMASCHERA	Esegue l'AND bit a bit tra il contatore (che rappresenta il centroide in analisi) e la maschera. Se l'AND è costituito da un vettore di soli 0, allora il centroide non sarà da valutare. Altrimenti sarà da valutare in termini di distanza.
8	LEGGOCENTROIDEX	Legge e salva l'ascissa del centroide in fase di valutazione. Incrementa addr.
9	LEGGOCENTROIDEY	Legge e salva l'ordinata del centroide in fase di valutazione. Incrementa addr.
10	CALCOLADISTANZA	Calcola la distanza di Manhattan tra il punto in fase di valutazione e il punto di riferimento iniziale.
11	CONFRONTA	Confronta la distanza calcolata nello stato precedente con quella minima calcolata dalla macchina fino a questo punto dell'esecuzione. In base al confronto, decide lo stato prossimo.
12	DISTAMINIMA	Questo stato indica che la distanza calcolata è minore della distanza minima calcolata precedentemente. Per questo, lo stato aggiorna la maschera d'uscita, facendo l'OR tra "00000000" e il contatore (che indica il centroide in considerazione).
13	DISTAUGUALE	Questo stato indica che la distanza calcolata è uguale alla distanza minima calcolata precedentemente. Per questo, lo stato aggiorna la maschera d'uscita, facendo l'OR tra la vecchia maschera d'uscita e il contatore (che rappresenta il centroide in considerazione).
14	INCREMENTOCENTROIDE	Questo stato ha il ruolo di incrementare il contatore che rappresenta il centroide in considerazione, al fine di procedere alla valutazione di quelli successivi. Nel momento in cui il contatore arriva a "00000000" significa che tutti i centroidi sono stati valutati, quindi non continua il processo di valutazione.
15	FINITO	Questo stato attiva il segnale o_we di scrittura e modifica l'indirizzo addr a "0000000000010011", dove si dovrà poi scrivere in memoria.
16	SCRIVOUSCITA	Questo stato scrive in memoria il valore della maschera d'uscita trovata.
17	DONE	Stato finale.

5.3 Rappresentazione grafica della macchina a stati

E' dato per scontato, quindi non è stato rappresentato nello schema, che ogni stato, in caso di ricezione di un segnale di reset, ritorni allo stato RESET.



6 Testing

Il codice è stato testato sia con il testbench fornito su beep, sia con test realizzati da me per ricoprire le casistiche limite dell'implementazione. Oltre ciò, l'implementazione è anche stata sottoposta ad un certo numero di test casuali.
Il componente passa correttamente tutti i test a livello di:
1. Behavioural.
2. Post-Synthesis Functional.
3. Post-Synthesis Timing.

Nelle tabelle che seguono le maschere di ingresso e uscita sono scritte con il bit piu significativo a destra, invertite rispetto alle maschere della memoria, al fine di renderle più leggibili.
I test realizzati da me ricoprono le seguenti condizioni:

6.1 Ricezione di un segnale di reset random durante l'esecuzione

Il componente torna nello stato RESET e riesegue per intero la computazione.

Maschera ingresso	1	1	0	0	0	0	1	0
Punto	Centroide 1	Centroide 2	Centroide 3	Centroide 4	Centroide 5	Centroide 6	Centroide 7	Centroide 8
131 199	147 209	151 193	138 180	156 200	161 161	107 197	117 187	126 178
Distanza	26	26	26	26	68	26	26	26
Maschera uscita	1	1	0	0	0	0	1	0

Tipo di test	Tempo di esecuzione	Esito
Behavioural	4900ns	Corretto
Post-Synthesis Functional	4900100ps	Corretto
Post-synthesis Timing	4903001ps	Corretto

6.2 Maschera di ingresso "11111111"

In questo test tutti i centroidi son da valutare. Le coordinate dei punti sono casuali.

Maschera ingresso	1	1	1	1	1	1	1	1
Punto	Centroide 1	Centroide 2	Centroide 3	Centroide 4	Centroide 5	Centroide 6	Centroide 7	Centroide 8
218 68	213 68	216 71	220 71	220 65	213 68	146 32	220 65	214 67
Distanza	5	5	5	5	5	108	5	5
Maschera uscita	1	1	1	1	1	0	1	1

Tipo di test	Tempo di esecuzione	Esito
Behavioural	6700ns	Corretto
Post-Synthesis Functional	6700100ps	Corretto
Post-synthesis Timing	6703001ps	Corretto

6.3 Maschera di ingresso "00000000"

In questo test nessun centroide è considerato da valutare.

Maschera ingresso	0	0	0	0	0	0	0	0
Punto	Centroide 1	Centroide 2	Centroide 3	Centroide 4	Centroide 5	Centroide 6	Centroide 7	Centroide 8
31 99	95 50	10 197	30 99	20 9	30 99	241 192	62 29	199 229
Distanza	113	119	1	101	1	303	101	298
Maschera uscita	0	0	0	0	0	0	0	0

Tipo di test	Tempo di esecuzione	Esito
Behavioural	2800ns	Corretto
Post-Synthesis Functional	2800100ps	Corretto
Post-synthesis Timing	2803001ps	Corretto

6.4 Centroidi coincidenti col punto in questione e tutti uguali tra di loro

In questo caso la distanza dal punto è 0 per tutti i centroidi.

Maschera ingresso	0	0	1	1	1	1	0	1
Punto	Centroide 1	Centroide 2	Centroide 3	Centroide 4	Centroide 5	Centroide 6	Centroide 7	Centroide 8
91 177	91 177	91 177	91 177	91 177	91 177	91 177	91 177	91 177
Distanza	0	0	0	0	0	0	0	0
Maschera uscita	0	0	1	1	1	1	0	1

Tipo di test	Tempo di esecuzione	Esito
Behavioural	5300ns	Corretto
Post-Synthesis Functional	5300100ps	Corretto
Post-synthesis Timing	5303001ps	Corretto

6.5 Tutti i centroidi diversi tra di loro

Maschera ingresso	0	1	1	0	0	1	1	1
Punto	Centroide 1	Centroide 2	Centroide 3	Centroide 4	Centroide 5	Centroide 6	Centroide 7	Centroide 8
196 211	196 225	10 171	207 208	178 14	197 224	201 202	193 222	166 225
Distanza	14	226	14	215	14	14	14	44
Maschera uscita	0	0	1	0	0	1	1	0

Tipo di test	Tempo di esecuzione	Esito
Behavioural	5200ns	Corretto
Post-Synthesis Functional	5200100ps	Corretto
Post-synthesis Timing	5203001ps	Corretto

6.6 Presenza di un centroide a massima distanza dal punto in questione

Questo test vuole rilevare un possibile overflow.

Maschera ingresso	1	0	1	0	1	1	1	0
Punto	Centroide 1	Centroide 2	Centroide 3	Centroide 4	Centroide 5	Centroide 6	Centroide 7	Centroide 8
85 198	0 0	255 255	255 255	67 17	71 163	60 222	254 19	13 207
Distanza	283	227	227	199	49	49	348	81
Maschera uscita	0	0	0	0	1	1	0	0

Tipo di test	Tempo di esecuzione	Esito
Behavioural	5200ns	Corretto
Post-Synthesis Functional	5200100ps	Corretto
Post-synthesis Timing	5203001ps	Corretto

6.7 Dopo aver terminato, l'esecuzione riparte tramite un riassegmanento della RAM

Si vuole verificare che la computazione ricominci in seguito a un riassegnamento della RAM.

Maschera ingresso	0	0	0	0	0	0	1	1
Punto	Centroide 1	Centroide 2	Centroide 3	Centroide 4	Centroide 5	Centroide 6	Centroide 7	Centroide 8
60 60	100 0	75 85	60 70	1 230	86 135	50 60	215 78	60 50
Distanza	100	10	10	229	101	10	173	10
Maschera uscita	0	0	0	0	0	0	0	1

Tipo di test	Tempo di esecuzione	Esito
Behavioural	5750ns	Corretto
Post-Synthesis Functional	5750ns	Corretto
Post-synthesis Timing	5750ns	Corretto

7 Conclusione

Il componente realizzato rispetta tutte le caratteristiche della specifica ed è stato sottoposto ad una grande quantità di test. Sarebbe stata possibile un'ottimizzazione tramite l'accorpamento di vari stati assieme ma non è stato fatto per avere una suddivisione logica e funzionale della macchina.