



POLITECNICO
MILANO 1863

Prova Finale

(Progetto di Reti Logiche)

W. Fornaciari D. Zoni

Dispoto
Giovanni

Matricola 892576

Codice
Persona 10540027

Colangelo
Ilaria

Matricola 845975

Codice
Persona 10503306

AA 2018/2019

INDICE

1. Descrizione dello schema funzionale.....	2
2. Diagramma della FSM.....	3
3. Test	4
4. Architettura e scelte progettuali.....	5

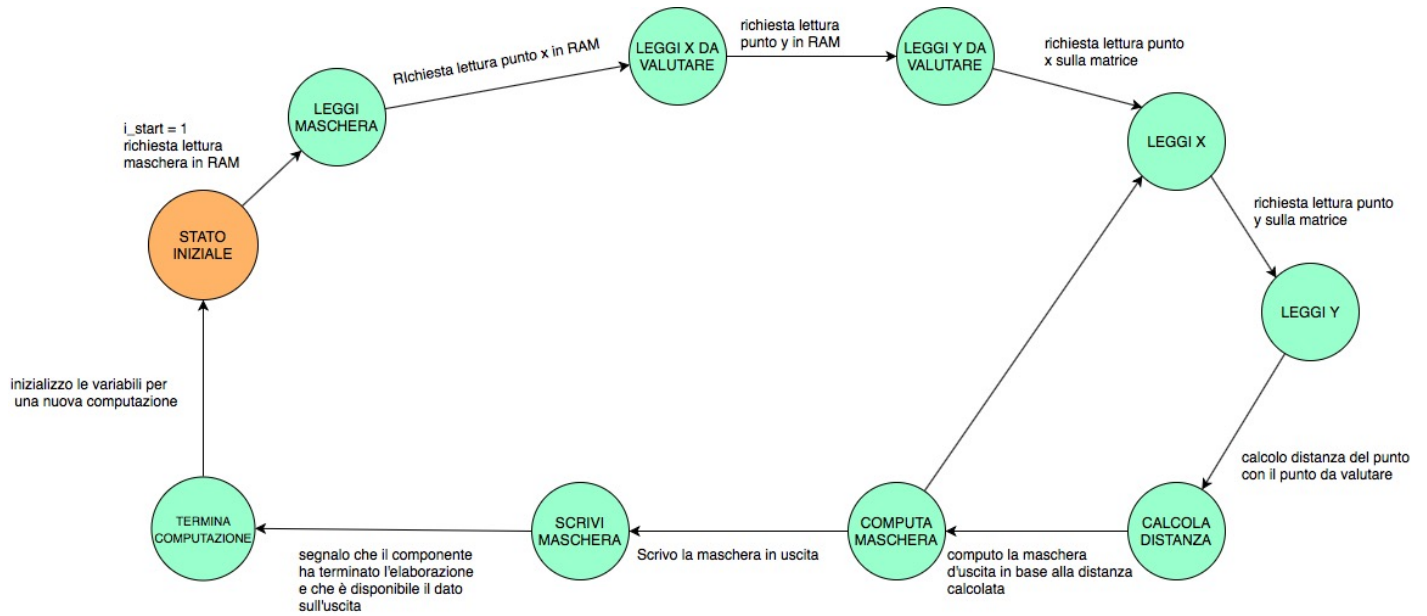
1. DESCRIZIONE DELLO SCHEMA FUNZIONALE

L'implementazione realizzata sfrutta una Final State Machine composta da 10 stati:

Nello stato iniziale il componente effettua un set up di tutti i segnali necessari al funzionamento e rimane in attesa del segnale "start". Una volta avvenuto ciò, cambia lo stato e passa alla lettura della maschera input contenente la lista di centroidi attivi. Nei due stati seguire vengono considerate le coordinate rispettivamente X e Y del punto da valutare e procede entrando in un loop composto di quattro stati che rispettivamente legge la X e la Y di un centroide e calcola la distanza del punto da valutare con il rispettivo centroide preso in esame. Il loop termina con l'esame dell'ultimo centroide fornito. Al termine della computazione della maschera il componente salta nel penultimo stato che scrive effettivamente in memoria la maschera ed infine si arriva nell'ultimo stato che segnala per un clock la terminazione e riporta il componente allo stato iniziale, settando tutte le variabili per una nuova computazione.

Nella pagina successiva abbiamo inserito la rappresentazione grafica della FSM.

2. DIAGRAMMA DELLA FSM



3. TEST

Per testare il funzionamento della nostra macchina a stati, sono stati generati all'incirca 100 testbench randomicamente da un software pre-esistente (black box). Inoltre vi è la presenza di ulteriori 7 test ideati e mirati a toccare i casi limite possibili (white box) che elenchiamo qui di seguito in tabella.

CASI DI TEST MIRATI	
TEST 1	Punto da valutare generico e centroidi tutti attivi e coincidenti in un punto generico diverso dal punto da valutare.
TEST 2	Punto da valutare generico e centroidi tutti disattivi e coincidenti in un punto generico diverso dal punto da valutare.
TEST 3	Punto da valutare generico e centroidi tutti disattivi e coincidenti in un punto generico coincidente al punto da valutare.
TEST 4	Punto da valutare generico e centroidi tutti attivi e coincidenti in un punto generico coincidente al punto da valutare.
TEST 5	Punto da valutare generico e centroidi posizionati sui bordi della matrice
TEST 6	Punto da valutare generico ed i centroidi attivi sono tutti equidistanti dal punto da valutare ma non coincidenti
TEST 7	Punto da valutare in (0,0) e presenza tra i centroidi attivi di uno posizionato all'estremo opposto della tabella (255,255).

4. ARCHITETTURA E SCELTE PROGETTUALI

Il progetto si basa su una Final State Machine composta da 10 stati ed organizzata in un processo combinatorio ed in un processo sincrono.

Inizialmente l'idea di progetto prevedeva una FSM ad un singolo processo che seppur con codice molto più semplificato non permetteva di allontanarsi del tutto da una logica sequenziale non proprio applicabile a questo progetto. Motivo per cui la scelta è stata quella di dividere il tutto in una parte combinatoria e una sincrona.

Nel primo processo presente nel codice vengono gestiti i segnali asincroni e lo stato in cui si trova il componente. Osservando `i_clk`, `i_rst` e `i_start` viene stabilito se ci si trova nello stato iniziale ed inoltre, se necessario si attribuisce ad ogni segnale il suo successivo (es. `state <= next_state`)

Il secondo processo si occupa di quella che è l'effettiva logica della FSM che, come già anticipato, è composta di 10 stati, 4 dei quali coinvolti in un loop. Il loop è stato strutturato in modo da leggere prima tutti i centroidi, indipendentemente dal fatto che siano attivi o meno. Solo dopo, nel momento in cui il componente andrà a valutare la distanza per modificare - se necessario - la maschera di uscita, sarà verificato se gli effettivi centroidi a distanza minima trovata siano da considerare o meno. Questa logica ha permesso di mantenere molto semplice la struttura del grafo con il compromesso di perdere cicli di clock per leggere e processare dati effettivamente non utilizzati.

In ogni stato vengono preparati le uscite dei segnali che interagiscono con la RAM in modo tale da avere nel clock successivo i valori utili per la computazione. Per la maschera di uscita si è optato di calcolare la maschera secondo il minimo temporaneamente trovato e, nel momento in cui si fosse trovato un altro minimo inferiore al primo, di resettare la maschera e procedere con il calcolo in funzione di tale punto.