

Professor William Fornaciari / Davide Zoni

Prova finale (Progetto di reti logiche)

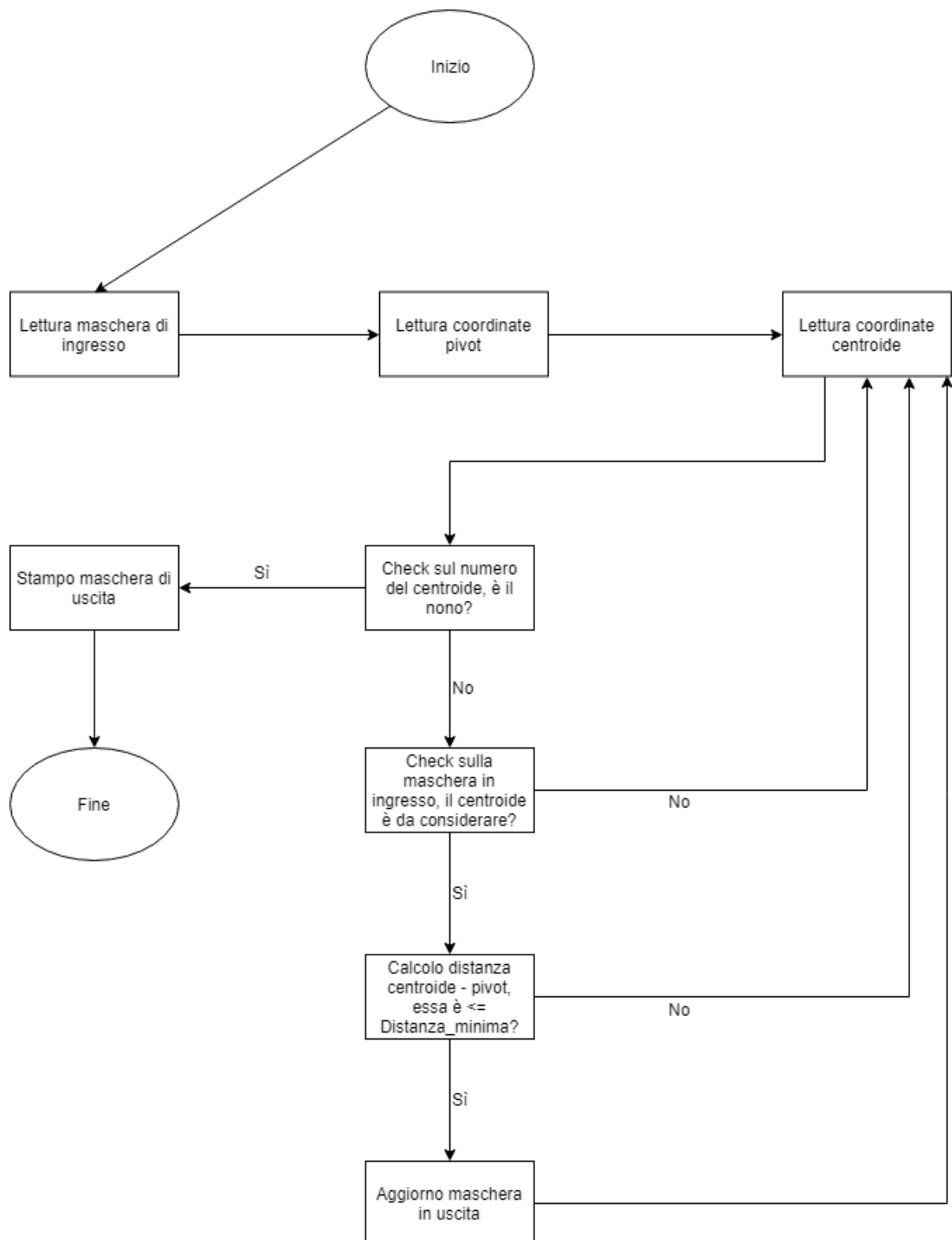
AA 2018/2019

Amedeo Carrioli
10568274-866256

INDICE

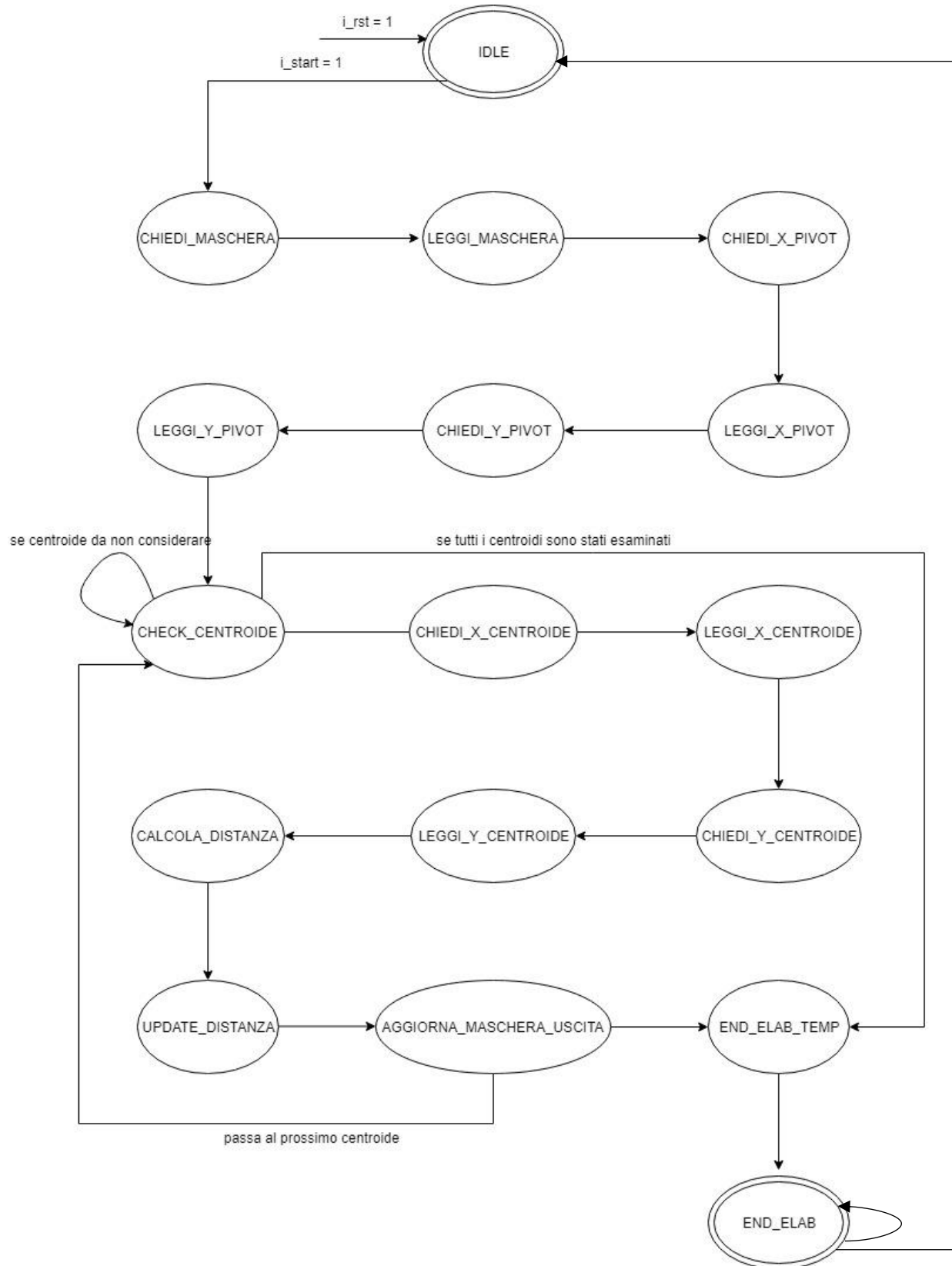
- RAPPRESENTAZIONE ALGORITMO	2
- RAPPRESENTAZIONE MACCHINA A STATI	2
- DESCRIZIONE MACCHINA A STATI E IMPLEMENTAZIONE VHDL	4
- TEST	6

RAPPRESENTAZIONE ALGORITMO



RAPPRESENTAZIONE MACCHINA A STATI

Dopo aver terminato la fase di progettazione dell'algoritmo, si procede con la vera e propria realizzazione del componente VHDL. Come prima cosa quindi si sintetizza l'algoritmo in una macchina a stati finiti:



DESCRIZIONE MACCHINA A STATI E IMPLEMENTAZIONE VHDL

Una volta posto il segnale `i_rst = 1`, il componente entra nel primo stato, `IDLE`, in cui setta tutti i segnali a valori iniziali, per esempio viene impostato l'indirizzo in memoria a cui andare a leggere allo stato successivo `NEXT_address <= "0000000000000000"`, la distanza minima che poi sarà oggetto di confronto con le distanze centroide-pivot `NEXT_distanza_minima <= "111111111"` e le coordinate del pivot e del centroide che leggerò successivamente, inizialmente settate a `"00000000"`. In questo stato vengono anche toccati i segnali di uscita `o_address`, `o_done`, `o_en`, `o_we` e `o_data`, tutti impostati a valori iniziali nulli.

Quando `i_start` è alto, il componente passa allo stato successivo, `CHIEDI_MASCHERA`, che si occupa di abilitare la lettura in memoria e imposta l'indirizzo di memoria al quale si andrà a leggere la maschera di ingresso, allo stato successivo, `LEGGI_MASCHERA`. In questo stato viene abbassato il segnale per leggere in memoria e viene assegnato al segnale maschera il valore effettivo (`i_data`) che si è andato a leggere all'indirizzo precedentemente accennato (`"00000000"`).

Lo stesso procedimento viene effettuato per la lettura delle coordinate del pivot, viene quindi alzato il segnale per la lettura in memoria, impostato l'indirizzo di memoria a cui andare a leggere e, allo stato successivo, viene letto e salvato il valore salvato in un segnale e abbassato il segnale di lettura in memoria.

Una volta acquisita maschera e coordinate del pivot, il componente passa allo stato `CHECK_CENTROIDE`, in cui viene preso in considerazione il numero del centroide che verrà analizzato negli stati successivi, se `numero_centroide = 8`, significa che il componente ha finito l'analisi di tutti i centroidi, quindi salta ad `END_ELAB_TEMP`, stato in cui vengono impostati correttamente i segnali di uscita, ovvero `o_address <= "000000000010011"`, `o_we <= '1'`, `o_en <= '1'` e `o_data <= maschera_uscita`.

Successivamente si passa al vero proprio stato di terminazione dei processi, `END_ELAB`, in cui vengono abbassati `o_we` e `o_en` e, finché `i_start` è alto, si rimane nello stato `END_ELAB`, ma quando `i_start` si abbassa, `o_done` viene alzato e si ritorna al primo stato, `IDLE`.

Tornando allo stato `CHECK_CENTROIDE`, nel caso in cui il numero del centroide da analizzare sia valido (≥ 0 e ≤ 7), si fa un rapido check alla maschera in ingresso, per vedere se il centroide è da considerare, nel caso lo sia, si passa a calcolare la sua distanza dal pivot ed eventualmente aggiornare distanza minima e maschera di uscita. Invece, nel caso non si debba analizzare, viene semplicemente incrementato di uno il numero del centroide e di due l'indirizzo a cui chiedere le coordinate del prossimo centroide, quindi il prossimo stato sarà ancora `CHECK_CENTROIDE`.

Se il centroide è da analizzare viene quindi impostato l'indirizzo di memoria al quale accedere in lettura e si passa allo stato `CHIEDI_X_CENTROIDE`, in cui ancora una volta viene alzato il segnale per consentire lettura in memoria. Dopo di che si passa a `LEGGI_X_CENTROIDE`, in cui avviene la lettura e viene salvato il valore nel segnale `NEXT_X_CENTROIDE`. Lo stesso avviene per la coordinata `y` del centroide.

Si passa poi al vero e proprio calcolo della distanza dal centroide al pivot, siamo quindi nello stato `CALCOLA_DISTANZA`. Qui vengono salvati in due segnali, `distanza_lungo_x` e `distanza_lungo_y`, i valori della distanza che, sommati (nello stato successivo `UPDATE_DISTANZA`), daranno la distanza Manhattan centroide-pivot.

Il componente salta poi ad AGGIORNA_MASCHERA_USCITA, che si occupa di confrontare la distanza centroide-pivot in questione con la distanza centroide-pivot minima fino ad ora calcolata. Se essa è minore viene posta lei come distanza minima, settata a zero la maschera di uscita
NEXT_maschera_uscita <= "00000000", e viene alzato solo il bit corrispondente al centroide appena analizzato, NEXT_maschera_uscita (to_integer(unsigned(numero_centroide - "0001"))) <= '1'.

Invece, nel caso in cui la distanza centroide-pivot in questione si uguale alla distanza minima, viene semplicemente alzato il bit corrispondente al centroide appena analizzato, senza prima porre a zero la maschera di uscita.

Dopo di che viene controllato il numero del centroide, se numero_centroide = 8, il prossimo stato sarà END_ELAB_TEMP, altrimenti si torna a CHECK_CENTROIDE, per chiedere ed analizzare il prossimo centroide.

All'interno del codice sono presenti due process. Uno, chiamato delta_lambda, è combinatorio ed è sensibile a tutti i segnali usati (sia present che next). E' il processo che gestisce gli stati tramite un case. All'interno avvengono quindi le transizioni tra stati, vengono letti gli input, impostato l'output e tutto ciò che è necessario per il corretto funzionamento del componente.

Il secondo process, state, è invece sequenziale ed è sensibile solo al reset e al clock (i_rst, i_clk). In questo processo ad ogni evento di clock viene assegnato ad ogni segnale presente il corrispondente segnale next, ricavato nel process combinatorio al ciclo di clock precedente, e allo stato presente lo stato successivo. Viene inoltre trattato il caso in cui i_rst è alto, in cui il prossimo stato diventa IDLE e tutti i segnali vengono settati a valori di default.

Nel processo combinatorio vengono toccati solo i segnali next, mentre in quello sequenziale solamente i present.

TEST

I test fatti sono stati divisi in base alla parte di codice e quindi di componente sollecitato

Maschera ingresso	Coordinate centroide 1	2	3	4	5	6	7	8	Coordinate pivot	Maschera uscita
-------------------	------------------------	---	---	---	---	---	---	---	------------------	-----------------

Questi tests sono stati fatti per dimostrare il corretto funzionamento della lettura della maschera di ingresso per controllare se il centroide è da considerare:

00000000	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	00000000
00100000	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	00100000
00100010	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	00100010
10101010	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	10101010
01010101	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	01010101

I seguenti test sollecitano invece la parte del calcolo della distanza e aggiornamento maschera di uscita:

11111111	1, 1	1, 2	1, 3	1, 4	1, 5	1, 6	1, 7	1, 8	1, 1	00000001
11111111	1, 1	1, 2	1, 3	1, 4	1, 5	1, 6	1, 7	1, 8	1, 9	10000000
11111111	1, 1	1, 2	1, 3	1, 4	1, 5	1, 6	1, 7	1, 8	2, 5	00010000
11111111	1, 1	124, 124	124, 124	124, 124	124, 124	124, 124	124, 124	124, 124	1, 2	00000001
11111111	1, 1	124, 124	124, 124	124, 124	124, 124	124, 124	124, 124	1, 1	1, 2	10000001
11111111	0, 0	0, 0	0, 255	0, 255	255, 0	255, 0	255, 255	255, 255	127, 127	00000011
11111111	0, 0	0, 0	0, 255	0, 255	255, 0	255, 0	255, 255	255, 255	128, 128	11000000
11111111	0, 0	0, 0	0, 255	0, 255	255, 0	255, 0	255, 255	255, 255	127, 128	00001100
11111111	0, 0	0, 0	0, 255	0, 255	255, 0	255, 0	255, 255	255, 255	128, 127	00110000

Altri tests:

000000001	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	255, 255	00000001
100000000	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	255, 255	10000000
100000001	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	255, 255	10000001
011111110	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	255, 255	011111110
111100000	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	255, 255	111100000
000011111	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	255, 255	000011111

Dopo di che ho testato il componente con centroidi in posizioni casuali e maschera di ingresso casuale.