Progetto di Reti Logiche a.a. 2017/18

10522652 Baldasseroni Eva

II consegna

Progetto di Reti Logiche a.a. 2017/18

10522652 Baldasseroni Eva

Consegna

Si vuole implementare un componente HW descritto in VHDL che, data un'immagine in scala di grigi in un formato descritto successivamente (vedere documento delle specifiche), calcoli l'area del rettangolo minimo che circoscrive totalmente una figura di interesse presente nell'immagine stessa. Il termine circoscrive totalmente indica il fatto che il rettangolo deve essere il più piccolo tale che tutti i pixel facenti parte della figura di interesse siano interni o appartenenti al perimetro del rettangolo.

Implementazione

Introduzione

Per l'implementazione, la struttura del progetto è stata divisa in 5 parti, associate ciascuna ad un componente:

- **leggi_e_salva:** è il componente che si occupa della lettura dei dati in ingresso e del loro corretto salvataggio.
- **posizione_attuale:** si occupa di calcolare le coordinate (riga, colonna) della matrice che leggiamo in ingresso e di alzare il segnale di FINITO quando deve terminare la lettura.
- **analisi_x:** si occupa di effettuare le analisi che ci forniscono le coordinate delimitanti l'area del rettangolo da calcolare.
- **calcolo_risultato:** quando è finita la lettura, prende le coordinate fornite dal componente *analisi_x* e calcola il risultato.
- **caricamento_risultato:** quando il risultato è stato calcolato, lo carica correttamente in memoria.

Tra i componenti esistono dei segnali che ne regolano l'attività correttamente:

- **lettura** *(std_logic)*: coinvolge i componenti *leggi_e_salva* e *posizione_attuale*.
- **finito** *(std_logic)*: coinvolge i componenti *leggi_e_salva*, *posizione_attuale analisi_x* e *calcolo_risultato*.
- **carica** *(std_logic)*: coinvolge i componenti *calcolo_risultato* e *caricamento_risultato.*
- **zero** *(std_logic)*: coinvolge i componenti *leggi_e_salva, calcolo_risultato* e *posizione_attuale*.

Tutti i componenti lavorano sul fronte di discesa del clock.

Analisi generale

Il progetto in generale si può riassumere in una macchina a stati finiti, regolata dai segnali sopra descritti.

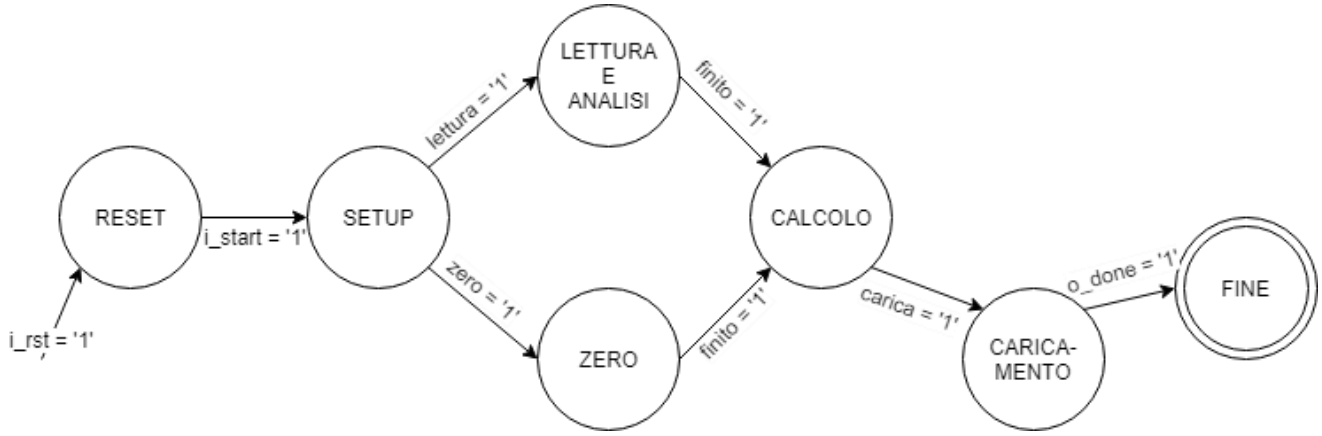


Fig.1

- **Stato di RESET:** il segnale di i_rst può essere asserito in qualsiasi momento, anche nel mezzo della computazione. Riporta i componenti in uno stato di default. Per semplicità sono state omesse le transizioni di stato con i_rst = '1', che vanno da ogni stato a quello di reset.
- **Stato di SETUP:** occupa in genere 4 cicli di clock. È dedicato a salvare *N_COLONNE*, *N_RIGHE* e *SOGLIA*. In questo momento è attivo solo il componente *leggi_e_salva*.
- **Stato di LETTURA E ANALISI:** quando inizia la lettura dei valori dell'immagine, i componenti *posizione_attuale* e *analisi_x* analizzano i dati in ingresso.
- **Stato di ZERO:** in caso *N_COLONNE* o *N_RIGHE* siano zero, questo stato imposta i giusti valori da fornire allo stato di calcolo.
- **Stato di CALCOLO:** occupa in genere 3 cicli di clock, in cui il componente *calcola_risultato* calcola il risultato.
- **Stato di CARICAMENTO:** occupa 4 cicli di clock, in cui il componente *carica_risultato* aspetta l'elaborazione del risultato, carica le due parti del risultato e alza il segnale o_done.
- **Stato di FINE:** termine della computazione.

Analisi dei componenti

leggi_e_salva

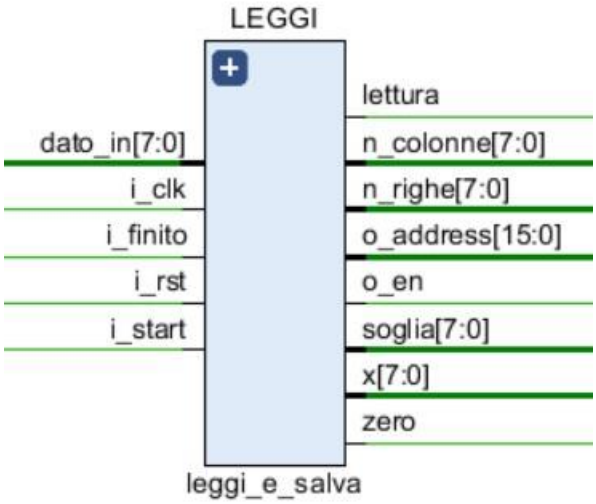


Fig.2

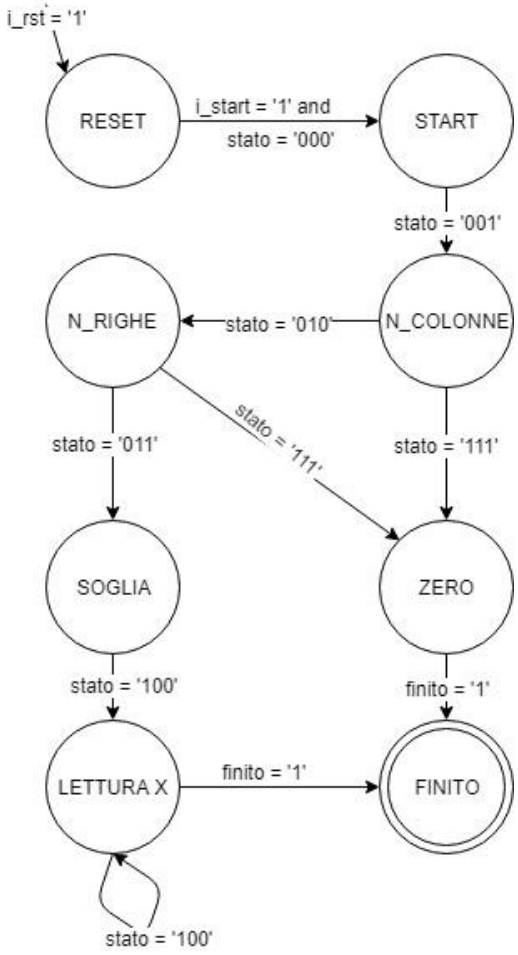


Fig.3

Il componente *leggi_e_salva* si occupa di prendere i dati in ingresso e iniziare a leggerli correttamente dopo un segnale di reset, a cui segua uno di start.
Leggere correttamente significa salvare i valori agli indirizzi in memoria 2, 3, 4, rispettivamente in *n_colonne*, *n_righe*, *soglia*. Agli indirizzi successivi a questi il componente aggiorna il valore di *x* a ogni ciclo di clock e incrementa di uno l'indirizzo da leggere al ciclo successivo. Questo avviene finché il segnale di *finito* non diventa 1. A quel punto vuol dire che non è rimasto niente da leggere in memoria.

Lo stato è dettato dal segnale interno **stato** *(std_logic_vector(2 downto 0))*.

La macchina a stati finiti è composta da:

- **RESET:** tutti i segnali vengono posti a zero
- **START:** se lo *stato* è '000' (cioè se c'è stato un reset prima), pone lo *stato* a '001' e *o_en* a 1

- **N_COLONNE:** salva *dato_in* in *n_colonne*, incrementa di 1 *o_address*
- **N_RIGHE:** salva *dato_in* in *n_righe*, incrementa di 1 *o_address*
- **SOGLIA:** salva *dato_in* in *soglia*, incrementa di 1 *o_address*
- **LETTURA X:** salva *dato_in* in *x*, incrementa di 1 *o_address*
- **ZERO:** se il *dato_in* è 0 in N_COLONNE o N_RIGHE, pone il segnale *zero* a 1
- **FINITO:** quando arriva il segnale di *finito* termina la lettura

Lo stato ZERO cattura le eccezioni nel caso in cui n_colonne o n_righe siano 0.

posizione_attuale

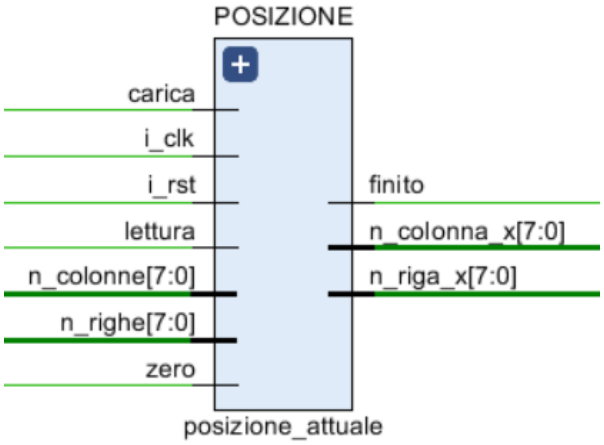


Fig.4

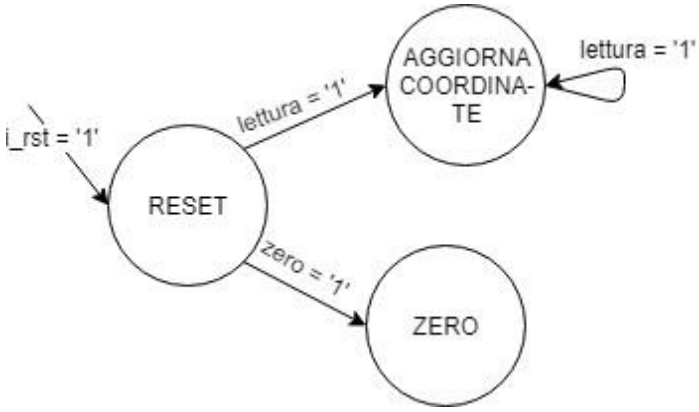


Fig.5

Il componente *posizione_attuale* restituisce la colonna e la riga a cui si trova l'elemento *x* che stiamo leggendo in quel ciclo di clock e alza a 1 il segnale di *finito* quando arriva alla fine delle colonne e delle righe o quando il segnale *zero* è alto.

La macchina a stati finiti del componente è fatta da:

- **RESET:** il segnale *finito* è posto a 0, *n_colonna_x* e *n_riga_x* a 1
- **AGGIORNA COORDINATE:** incrementa *n_colonna_x* di 1 se può, altrimenti va a capo. Se non può candare a capo perché siamo all'ultima riga, imposta il segnale *finito* a 1
- **ZERO:** imposta il segnale di *finito* a 1

analisi_x

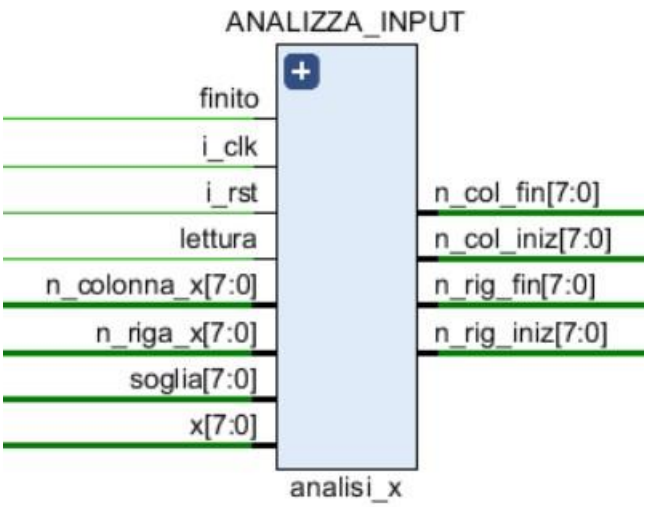


Fig.6

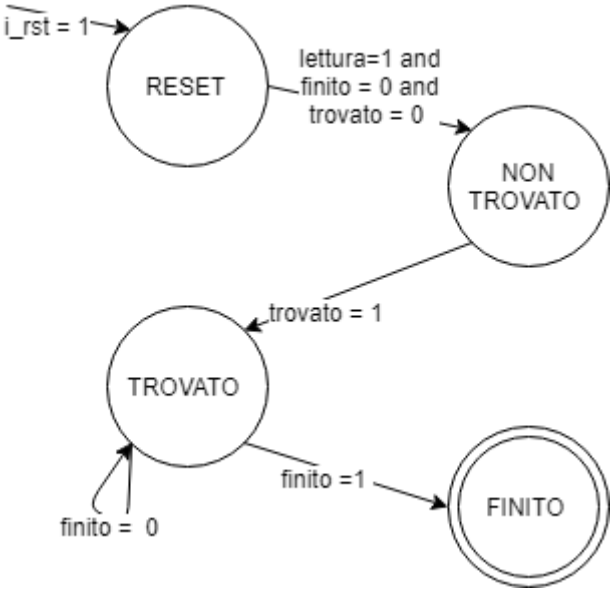


Fig.7

Il componente *analisi_x* è, nella fase di lettura, il componente che valuta i limiti del rettangolo che racchiude i valori maggiori uguali della soglia.

La macchina a stati finiti del componente è fatta da:

- **RESET:** i segnali uscenti sono posti a 0
- **NON TROVATO:** finché non ho trovato il primo valore sopra la soglia, sono in questo stato. Se trovo un valore maggiore uguale della soglia, imposto i suoi *n_colonna_x* e *n_riga_x* come *n_col_fin*, *n_col_iniz*, *n_rig_iniz* e *n_rig_fin*.
- **TROVATO:** confronto *n_colonna_x* e *n_riga_x* con i valori di *n_col_fin*, *n_col_iniz* e *n_rig_fin* salvati e valuto se sostituire questi ultimi con le coordinate dell'x attuale oppure no.
- **FINITO:** termina la computazione.

calcolo_risultato

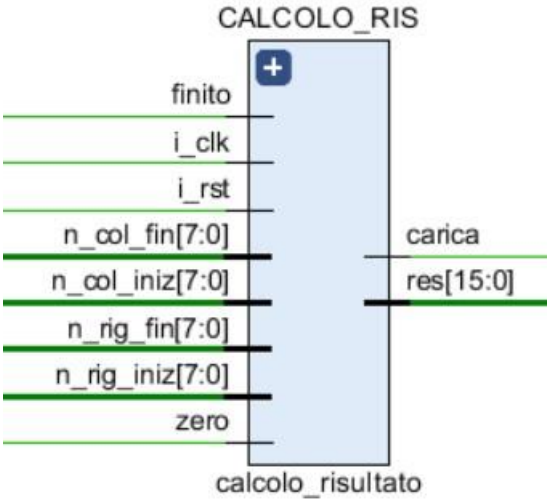


Fig.8

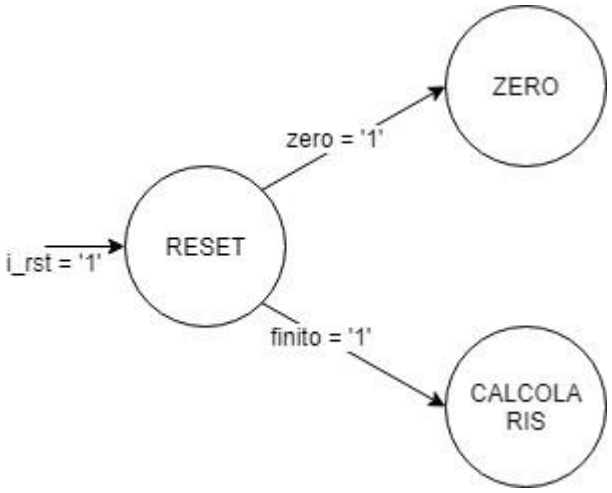


Fig.9

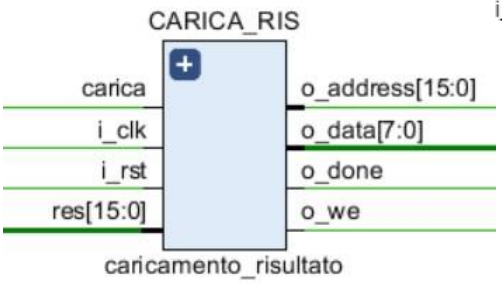
Il componente *calcolo_risultato* riceve i risultati dell'elaborazione del componente *analisi_x* e esegue il calcolo del risultato quando il segnale di finito è a 1. Se il segnale di zero è a 1, non esegue operazioni perché il valore che imposta lo stato di reset come risultato è adeguato.

La macchina a stati finiti del componente è fatta da:

- **RESET:** i segnali uscenti sono posti a 0
- **CALCOLA RIS:** occupa 2 cicli di clock per l'operazione di moltiplicazione
- **ZERO:** non fa niente, perché il risultato impostato dallo stato di reset è già quello corretto

In ogni caso, sia dopo lo stato ZERO, che dopo lo stato CALCOLA RIS, il segnale di carica è posto a uno.

carica_risultato



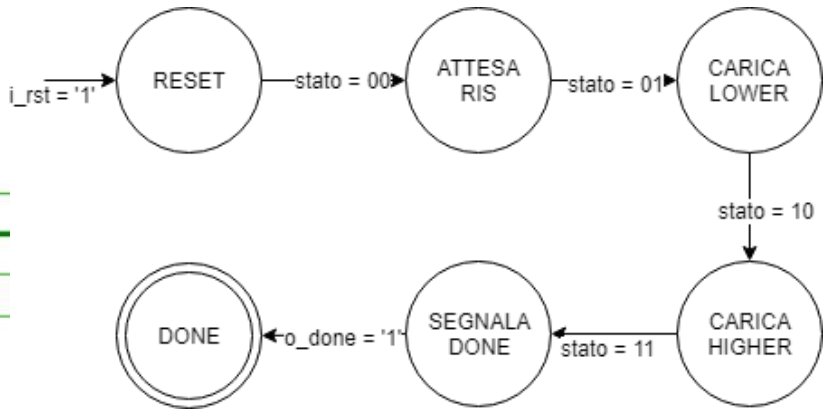


Fig.10 *Fig.11*

L'ultimo componente si occupa solo di caricare il risultato ricevuto in ingresso quando si alza il segnale di carica. Attraversa diversi stati per caricare il risultato secondo le specifiche e alza il segnale di o_done.

La macchina a stati finiti del componente è fatta da:

- **RESET:** i segnali uscenti sono posti a 0
- **ATTESA RIS:** attende per un ciclo di clock l'elaborazione del risultato
- **CARICA LOWER:** carica in memoria all'indirizzo 0 il byte meno significativo
- **CARICA HIGHER:** carica in memoria all'indirizzo 1 il byte più significativo
- **SEGNALA DONE:** alza il segnale di o_done
- **DONE:** stato finale

Casi di test

Il progetto è stato testato con tutti i testbench proposti. È stato inoltre verificato il corretto funzionamento anche con i casi limite di:

- N_COLONNE = 0 *(risultato 0)*
- N_RIGHE = 0 *(risultato 0)*
- SOGLIA = 0 *(risultato N_COLONNE * N_RIGHE)*
- Valori dei pixel solo inferiori alla soglia *(risultato 0)*

Conclusioni

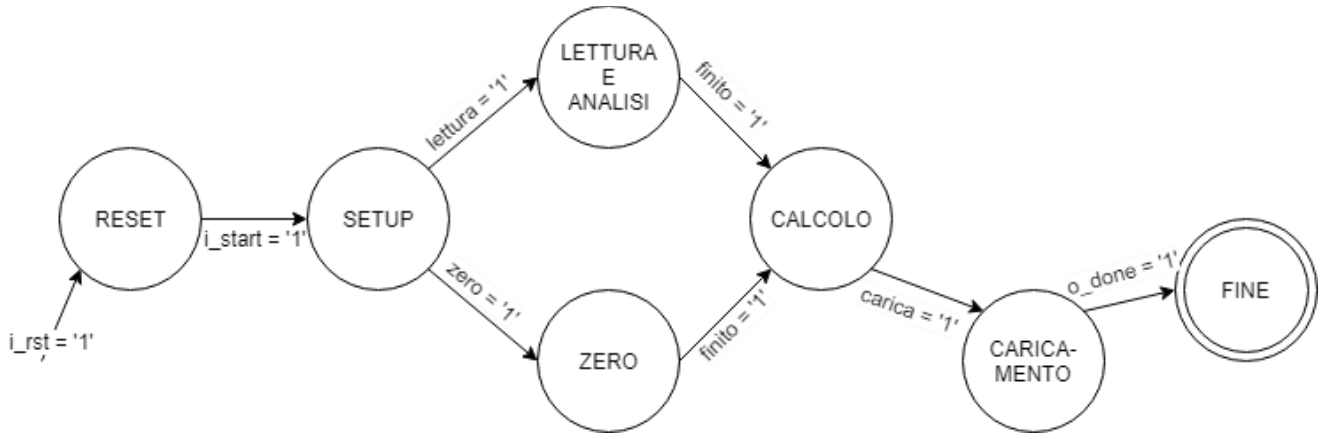


Fig.1

Cicli di clock impiegati in ogni stato a partire dal segnale di *i_start* e sotto l'ipotesi che N_COLONNE e N_RIGHE siano diverse da 0:

- **3** per lo stato di *SETUP*
- **N** per lo stato di *LETTURA E ANALISI*, dove N = N_COLONNE * N_RIGHE (ossia, leggo ogni pixel dell'immagine)

- **1** per lo stato di *CALCOLO*
- **4** per lo stato di *CARICAMENTO* (dove 1 ciclo si perde nell'attesa del calcolo del risultato) Totale = 8 + N cicli di clock

Nel caso di N_COLONNE = 0:

- **2** : SETUP
- **1** : ZERO
- **0** : CALCOLO (perché si passa direttamente il risultato impostato dal reset)
- **4** : CARICAMENTO

Totale = 7 cicli di clock

Nel caso di N_RIGHE = 0:

- **3** : SETUP
- **1** : ZERO
- **0** : CALCOLO (perché si passa direttamente il risultato impostato dal reset)
- **4** : CARICAMENTO

Totale = 8 cicli di clock