**Prova Finale**

**(Progetto di Reti Logiche)**

Professor Fabio Salice

Anno Accademico 2020-2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Huang**  Noemi | **Codice**  **Persona** | 10608004 | **Matricola** | 910420 |
| **Iuga**  Alexandra | **Codice**  **Persona** | 10623368 | **Matricola** | 908723 |

**Introduzione**

Il progetto prevede di realizzare in VHDL un componente hardware che riesce ad effettuare delle elaborazioni digitali di immagini (max. 128x128 pixel) secondo il metodo di equalizzazione dell’istogramma.



Il metodo ricalibra il contrasto di un’immagine che abbia intervallo di valori di intensità molto vicini, distribuisce in modo opportuno le intensità sull’istogramma aumentando il contrasto dell’immagine.

**TOOL**

E' stato utilizzato il software 'xilinx vivado webpack per l'implementazione del componente.

**Architettura**

Per la realizzazione del componente richiesto dalle specifiche, si è scelto di utilizzare un componente che implementi una macchina a stati finiti e abbia due processi di cui:

* Processo A, sensibile ai segnali i\_clk e i\_rst, effettua l’operazione di passaggio di stati della FSA.
* Processo B, sensibile ai segnali i\_clk, i\_start e CURRENT\_STATE, effettua tutte le operazioni della FSA a seconda dello stato attuale.

La macchina (Figura 3) è composta da 16 stati:



* Stato RESET: stato in cui vengono settati tutti le variabili al valore iniziale, si torna a questo stato ogni volta che incontra un segnale i\_rst=‘1’, o quando termina i\_start viene riportato a 0;
* Stato START: stato in cui invia la richiesta di lettura del primo byte;
* Stato READ\_COLUMN: stato in cui memorizza il numero di colonne dell’immagine da analizzare, se è uguale a 0 termina e passa allo stato DONE.
* Stato ROW\_ADDRESS: stato in cui invia la richiesta di lettura del secondo byte;
* Stato READ\_ROW: stato in cui memorizza il numero di righe dell’immagine, se è uguale a 0 viene alzato il cont, se diventa >0 prima che cont arriva ad un numero prestabilito passa allo stato successivo, altrimenti termina e passa allo stato DONE;
* Stato READ\_PIXEL: stato in cui vengono inviati la richiesta di lettura dei vari pixel per n\_pixel volte;
* Stato CALCULATE\_MAX\_MIN: stato dove analizza il pixel ricevuto, e imposta i valori massimi e minimi dell’immagine;
* Stato INCREMENT\_ADDRESS: stato in cui viene modificato l’indirizzo da leggere aggiungendo di 1;
* Stato SET\_FOR\_CALCULUS: stato in cui viene resettato l’indirizzo da leggere;
* Stato DELTA\_VALUE: stato in cui viene calcolato il delta\_value;
* Stato SHIFT\_LEVEL: stato in cui viene calcolato il valore shift\_level;
* Stato READ\_PIXEL\_AGAIN: stato in cui calcola il valore di temp\_pixel dopo la seconda lettura dei pixel dall’inizio sempre per n\_pixel volte;
* Stato CALCULATE\_NEW\_VALUE: stato in cui viene calcolato il valore definitivo del pixel da assegnare alla variabile new\_pixel;
* Stato WRITE\_OUTPUT: stato in cui scrive il valore nuovo dei pixel nell’indirizzo di memoria opportuno;
* Stato RESET\_ADDRESS: stato in cui reimposta l’indirizzo del pixel da leggere;
* Stato DONE: stato in cui alza il segnale o\_done a ‘1’ e attende che la memoria abbassi i\_start per abbassare o\_done, e ritornare allo stato iniziale RESET.

**Risultati Sperimentali**

Il componente è stato sottoposto a molti test bench tra cui alcuni generati casualmente. E' stata testata la risposta sia a un input con immagini di varia dimensione, sia con vari valori dei pixel stessi.

Tutti le simulazioni sono stati eseguite sia in behavioural, sia in post-sintesi functional e timing.

Riteniamo opportuno presentare i risultati ai test nei seguenti casi limite:

**TEST BENCH ZERO PIXELS:**

Il test bench verifica la risposta del circuito nel caso gli venga sottoposta un'immagine che abbia una dimensione di zero pixel sia in lunghezza che larghezza. In questo caso il circuito può direttamente passare agli ultimi stati e settare o\_done a uno.





**TEST BENCH UN PIXEL:**

Il rispettivo test verifica il comportamento del componente nel caso in cui l'immagine sia composta da un solo pixel. Questo infatti sottopone la macchina a vari comportamenti, tra cui i più interessanti: il massimo e il minimo coincidono e lo shift level risulta massimo.



**TEST BENCH 128 PIXEL:**

Questo test bench verifica il funzionamento del componente nel caso venga sottoposta un' immagine della dimensione massima richiesta nella specifica. In questo modo si può osservare anche i la memoria e il tempo massimi per quanto riguarda questa specifica. Il nostro progetto non è infatti vincolato al massimo di 128 pixel e sono stati effettuati test bench per verificare che possa funzionare anche con immagini più grandi.



**Simulazioni**

TODO

**Conclusione**

TODO