Prova Finale (Progetto di Reti Logiche)

Prof. Fabio Salice – Anno Accademico 2020/2021

Claudio Galimberti (Codice Persona 10610720 – Matricola 911834)

Pietro Marco Gallo (Codice Persona 10665739 – Matricola 910579)

**1 Introduzione**

**1.1 Descrizione generale**

La Prova Finale (Progetto di Reti Logiche) 2020/2021 è finalizzata all’implementazione di un circuito hardware in una FPGA, che abbia lo scopo di ricalibrare e incrementare il contrasto di un’immagine in input quando i suoi valori d’intensità sono contenuti in un intervallo molto ristretto. Come risultato, i pixel dell’immagine sono distribuiti su tutto l’intervallo di intensità, con valori da 0 a 255. Per questo scopo, viene sfruttato l’algoritmo di equalizzazione di un’immagine, del quale il circuito hardware implementa una versione semplificata rispetto a quella originale. Le immagini coinvolte sono solo quelle in scala di grigi a 256 livelli.

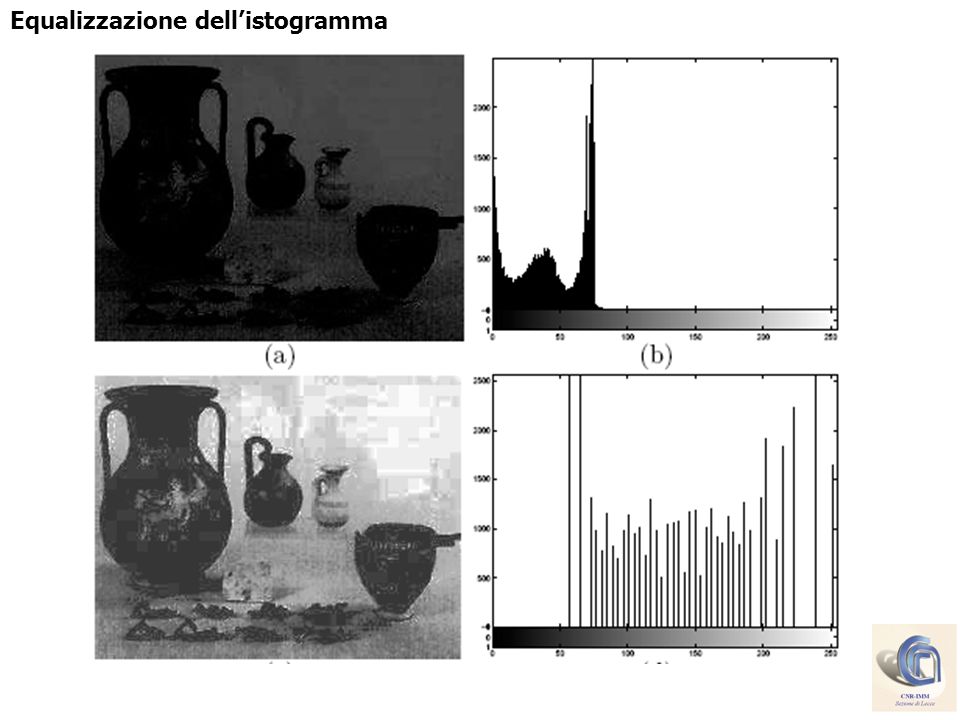


Figura 1 – figure di immagini equalizzate e non, con relativi istogrammi (sorgente: <https://slideplayer.it/slide/596329/>)

**1.2 Descrizione della memoria**

L’immagine è contenuta sequenzialmente in una memoria RAM con indirizzamento al byte e con una capienza massima di 65535 byte. Ogni pixel è memorizzato all’interno di un byte a partire dall’indirizzo 2. Nei primi due indirizzi sono memorizzati rispettivamente l’indice di colonna (indirizzo 0) e quello di riga (indirizzo 1) che definiscono la dimensione dell’immagine. Ogni immagine ha una dimensione massima di 128x128 pixel e viene letta riga per riga. L’immagine equalizzata viene salvata sequenzialmente a partire dall’indirizzo di memoria contiguo a quello in cui è memorizzato l’ultimo pixel dell’immagine non equalizzata.

È possibile equalizzare più immagini nell’intero processo, purché siano elaborate una alla volta. In caso di più di un’immagine da equalizzare, ciascuna risulta salvata in una memoria RAM diversa.

|  |
| --- |
| Indice di colonna (n\_col) |
| Indice di riga (n\_rig) |
| Pixel 1 non equalizzato |
| Pixel 2 non equalizzato |
| … |
| Pixel n\_col \* n\_rig non equalizzato |
| Pixel 1 equalizzato |
| Pixel 2 equalizzato |
| … |
| Pixel n\_col \* n\_rig equalizzato |

**1.3 Specifiche del progetto**

Ogni pixel dell’immagine non equalizzata viene sottoposto al seguente processo:

DELTA\_VALUE = MAX\_PIXEL\_VALUE – MIN\_PIXEL\_VALUE

SHIFT\_LEVEL = (8 – FLOOR(LOG2(DELTA\_VALUE + 1)))

TEMP\_PIXEL = (CURRENT\_PIXEL\_VALUE – MIN\_PIXEL\_VALUE) << SHIFT\_LEVEL

NEW\_PIXEL\_VALUE = MIN(255, TEMP\_PIXEL)

MAX\_PIXEL\_VALUE e MIN\_PIXEL\_VALUE sono rispettivamente il massimo e minimo valore dei pixel dell’immagine, CURRENT\_PIXEL\_VALUE è il valore del pixel attualmente da trasformare, NEW\_PIXEL\_VALUE è il valore assunto da CURRENT\_PIXEL\_VALUE dopo il processo di equalizzazione.

**1.4 Interfaccia del componente**

Il componente da descrivere presenta la seguente interfaccia:

entity project\_reti\_logiche is port (

i\_clk : in std\_logic;

i\_rst : in std\_logic;

i\_start : 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;

In particolare:

● i\_clk è il segnale di CLOCK in ingresso generato dal TestBench;

● i\_rst è il segnale di RESET che inizializza la macchina pronta per ricevere il primo segnale di START;

● i\_start è il segnale di START generato dal Test Bench;

● i\_data è il segnale (vettore) che arriva dalla memoria in seguito ad una richiesta di lettura;

● o\_address è il segnale (vettore) di uscita che manda l’indirizzo alla memoria;

● o\_done è il segnale di uscita che comunica la fine dell’elaborazione e il dato di uscita scritto in memoria;

● o\_en è il segnale di ENABLE da dover mandare alla memoria per poter comunicare (sia in lettura che in scrittura);

● o\_we è il segnale di WRITE ENABLE da dover mandare alla memoria (= 1) per poter scriverci. Per leggere da memoria esso deve essere 0;

● o\_data è il segnale (vettore) di uscita dal componente verso la memoria.

Il segnale i\_start viene posto a 1 dal test bench dopo un ciclo del clock nel quale è stato alto il segnale i\_rst. L’innalzamento del segnale i\_start comporta l’inizio della lettura in memoria. La fine dell’equalizzazione di un’immagine è scandita dall’innalzamento del segnale o\_done da parte del circuito, al quale il test bench risponde con l’abbassamento del segnale i\_start. Dopo un ciclo di clock, il circuito abbassa anche il segnale o\_done ed è pronto a leggere una nuova immagine quando i\_start viene nuovamente alzato dal test bench. Il segnale i\_rst è asincrono e può essere posto a 1 non solo all’inizio della computazione della prima immagine, ma anche durante l’elaborazione di un’immagine qualunque.