

# PROVA FINALE DI RETI LOGICHE

Filippo Calì (907675) - Cod.Persona: 10628126  
Giovanni Caleffi (907455) - Cod.Persona: 10665233

Prof. William Fornaciari - AA: 2020/2021

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del progetto . . . . .	1
1.2	Specifiche generali . . . . .	1
1.3	Interfaccia del componente . . . . .	2
1.4	Dati e descrizione memoria . . . . .	3
<b>2</b>	<b>Design</b>	<b>3</b>
2.1	Stati della macchina . . . . .	3
2.1.1	Processo gestione oaddress . . . . .	4
<b>3</b>	<b>Risultati dei test</b>	<b>4</b>
<b>4</b>	<b>Conclusioni</b>	<b>4</b>
4.1	Risultati della sintesi . . . . .	4
4.2	Ottimizzazioni . . . . .	4

# 1 Introduzione

## 1.1 Scopo del progetto

Lo scopo del progetto è la realizzazione di un componente hardware, scritto in VHDL. Esso riceve in ingresso un'immagine in scala di grigi a 256 livelli e, dopo aver applicato un algoritmo di equalizzazione a ciascun pixel, scrive in output l'immagine equalizzata.

Di seguito, un esempio di un'immagine 2x2 equalizzata (l'indirizzo dei dati in memoria verrà spiegato nel paragrafo 1.4).

0	1	2	3	4	5	6	7	8	9
2	2	46	131	62	89	0	255	64	172

## 1.2 Specifiche generali

L'algoritmo usato per l'equalizzazione delle immagini è una versione semplificata rispetto all'algoritmo standard. Esso può essere applicato solo a immagini in scala di grigi e per trasformare ogni pixel dell'immagine, esegue le seguenti operazioni:

```
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 rappresentano rispettivamente il massimo e il minimo valore dei pixel dell'immagine, CURRENT\_PIXEL\_VALUE rappresenta il valore del pixel da trasformare e NEW\_PIXEL\_VALUE rappresenta il valore del nuovo pixel in output.

Il componente hardware è inoltre progettato per poter codificare più immagini, una dopo l'altra. Prima di codificare l'immagine successiva, però, l'algoritmo di equalizzazione deve essere stato applicato prima a tutti i pixel dell'immagine precedente.

### 1.3 Interfaccia del componente

L'interfaccia del componente, così come presentata nelle specifiche, è la seguente:

```
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`: segnale di CLOCK in ingresso generato dal TestBench;
- `i_rst`: segnale di RESET che inizializza la macchina pronta per ricevere il primo segnale di START;
- `i_start`: segnale di START generato dal Test Bench;
- `i_data`: segnale (vettore) che arriva dalla memoria in seguito ad una richiesta di lettura;
- `o_address`: segnale (vettore) di uscita che manda l'indirizzo alla memoria;
- `o_done`: segnale di uscita che comunica la fine dell'elaborazione e il dato di uscita scritto in memoria;
- `o_en`: segnale di ENABLE da dover mandare alla memoria per poter comunicare (sia in lettura che in scrittura);
- `o_we`: segnale di WRITE ENABLE da dover mandare alla memoria (=1) per poter scriverci. Per leggere da memoria esso deve essere 0;
- `o_data`: segnale (vettore) di uscita dal componente verso la memoria.

## 1.4 Dati e descrizione memoria

Le dimensioni dell'immagine, ciascuna di dimensione di 8 bit, sono memorizzati in una memoria con indirizzamento al Byte:

- Nell'indirizzo 0 viene salvato il numero di colonne (N-COL) dell'immagine.
- Nell'indirizzo 1 viene salvato il numero di righe (N-RIG) dell'immagine.
- A partire dall'indirizzo 2 vengono memorizzati i pixel dell'immagine, ciascuno di 8 bit.
- A partire dall'indirizzo  $2+(N\_COL*N\_RIG)$  vengono memorizzati i pixel dell'immagine equalizzata.

N_COLONNE	Indirizzo 0
N_RIGHE	Indirizzo 1
PIXEL_1	Indirizzo 2
...	
PIXEL_N	
NEW_PIXEL_1	Indirizzo $2+(N\_COL*N\_RIGHE)$
...	
NEW_PIXEL_N	Indirizzo $1+2*(N\_COL*N\_RIGHE)$

La dimensione massima dell'immagine è 128x128 pixel.

## 2 Design

### 2.1 Stati della macchina

La macchina è composta da 17 stati,

**2.1.1** Processo gestione oaddress

## **3 Risultati dei test**

## **4 Conclusioni**

**4.1** Risultati della sintesi

**4.2** Ottimizzazioni

Particolare attenzione a non avere latch