POLITECNICO DI MILANO

Progetto di Reti Logiche

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Ingegneria Informatica

Sezione Prof. Salice

A.A. 2020/2021

Studenti: Sofia Martellozzo, Ilaria Muratori

Matricole: 910488, 911815

Codice Persona: 10623060, 10677812

Indice

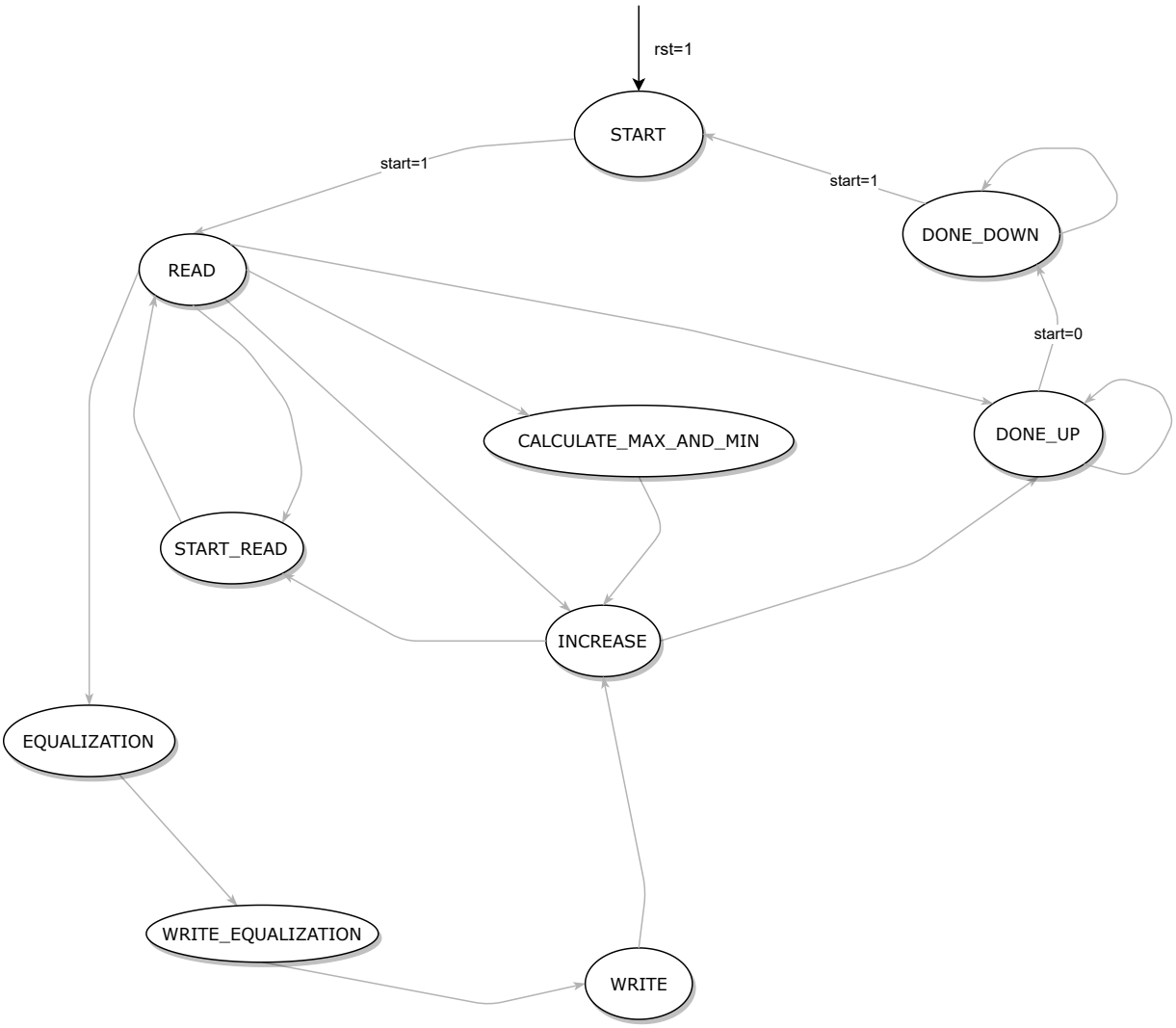
Introduzione

La seguente documentazione vuole descrivere in maniera dettagliata il processo volto alla risoluzione del problema proposto in forma di Prova Finale per l'Anno Accademico 2020/2021 dell'insegnamento di Reti Logiche.
La specifica della Prova finale (Progetto di Reti Logiche) 2020 è ispirata al metodo di equalizzazione dell'istogramma di una immagine.
Il metodo di equalizzazione dell'istogramma di una immagine è un metodo pensato per ricalibrare il contrasto di una immagine quando l'intervallo dei valori di intensità sono molto vicini effettuandone una distribuzione su tutto l'intervallo di intensità, al fine di incrementare il contrasto.

Architettura

Descrizione macchina a stati

Abbiamo scelto un implementazione a singolo processo sviluppando una MSF con 10 stati, a cui abbiamo aggiunto un 11esimo (DELAY) per permettere il caricamento dei valori dei segnali e di lettura e scrittura in memoria.



Stato	Descrizione
START	Stato di inizio processo e equalizzazione di un immagine a seguito di un segnale di *reset* alto per la prima immagine e un segnale alto di *start*. Vengono inizializzate tutte le variabili e i segnali al valore di default 0.
START_READ	Stato per inizio lettura da memoria impostando il segnale di *enable* a 1 e l'*o_address* per indicare l'indirizzo di memoria da cui leggere pari al segnale *current_address*

Stato	Descrizione
READ	Stato in cui in base a indirizzo di memoria a cui mi trovo, il numero letto sarà: • memorizzato nel segnale che indica il numero di colonne dell'immagine (se prima lettura ovvero all'indirizzo 0) • memorizzato nel segnale che indica il numero di righe dell'immagine (se seconda lettura ovvero all'indirizzo 1) • il pixel dell'immagine letto, usato per confrontarlo con tutti gli altri allo scopo di trovare il massimo e il minimo (operazione effettivamente svolta nello stato apposito) • se ho finito la lettura dei pixel dell'immagine originale (ovvero mi trovo all'indirizzo = riga x colonna +1) carico nell'*o_address* l'indirizzo del relativo pixel da equalizzare, il quale sarà poi memorizzato dopo l'elaborazione nell'attuale indirizzo. Inoltre calcolo il *delta_value* avendo già a disposizione il minimo e massimo valore tra i pixel dell'immagine letta tutta precedentemente.
CALCULATE_MAX_AND_MIN	Stato in cui effettuo il confronto effettivo con l'attuale pixel letto e il massimo e minimo attuali, eventualmente aggiornati.
INCREASE	Stato in cui incremento (di 1) sia il contatore *step* che mi tiene traccia dell'evoluzione del processo di equalizzazione corrente, sia del segnale per l'indirizzo corrente di memoria a cui si trova l'elaborazione a seguito di una fine lettura del pixel dell'immagine o dell'equalizzazione di un pixel stesso.
EQUALIZATION	Stato in cui inizia effettivamente l'equalizzazione con il calcolo dello *shift level* da applicare all'algoritmo, con un controllo di doglia del delta value calcolato nello stato di READ. Infine viene caricato nell'apposito segnale il valore del pixel da equalizzare.
WRITE_EQUALIZATION	Stato in cui, scaricato il valore del pixel da elaborare dalla memoria, procedo con l'algoritmo di equalizzazione: applico lo shift a sinistra di tante posizioni quante indicate dallo *shift_level* e tale risultato lo confronto con 255. Il risultato dell'operazione viene salvato nella variabile *new_pixel_value* e viene riportato l'*o_address* all'indirizzo in cui scriveremo il pixel equalizzato.
WRITE	Stato dove alzo il valore del segnale di *write enable* per poter scrivere in memoria (nel primo indirizzo libero in memoria pari all'indirizzo del pixel equalizzato + (riga x colonna)) il *new_pixel_value*
DONE_UP	Stato che mi segnala la fine elaborazione portando il segnale di *done* alto, tenendolo inoltre tale finché non diventa basso il segnale di *start*

Stato	Descrizione
DONE_DOWN	Stato successivo a DONE_UP a seguito di un abbassamento del segnale di *start* in cui viene portato il segnale *done* basso e rimane in attesa di una nuova elaborazione segnalata con un alzamento del segnale di *start*
DELAY	