



# POLITECNICO MILANO 1863

## Progetto di Reti Logiche

William Zeni  
matricola 10613915

Cristina Urso  
matricola 10599689

4 maggio 2021

Progetto sostenuto presso il Politecnico di Milano dipartimento di Elettronica,  
Informazione e Bioingegneria, diretto dal professor Gianluca Palermo nell'anno  
2020/21.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Scopo del progetto . . . . .	3
1.2	Specifiche generali . . . . .	3
1.3	Interfaccia del componente . . . . .	3
1.4	Dati e Descrizione memoria . . . . .	3
<b>2</b>	<b>Desing Pattern</b>	<b>3</b>
2.1	Scelte Progettuali . . . . .	3
2.2	Elenco Stati . . . . .	3
2.2.1	START . . . . .	3
2.2.2	INIT . . . . .	3
2.2.3	ABILIT_READ . . . . .	3
2.2.4	ABILIT_WRITE . . . . .	3
2.2.5	WAIT_MEM . . . . .	4
2.2.6	GET_RC . . . . .	4
2.2.7	GET_DIM . . . . .	4
2.2.8	READ_PIXEL . . . . .	4
2.2.9	GET_MINMAX . . . . .	4
2.2.10	GET_DELTA . . . . .	4
2.2.11	CALC_SHIFT . . . . .	4
2.2.12	GET_PIXEL . . . . .	4
2.2.13	CALC_NEWPIXEL . . . . .	4
2.2.14	WRITE_PIXEL . . . . .	4
2.2.15	DONE . . . . .	4
2.2.16	WAITINGPIC . . . . .	5
<b>3</b>	<b>Risultati dei Test</b>	<b>5</b>
<b>4</b>	<b>Conclusioni</b>	<b>5</b>

# 1 Introduzione

## 1.1 Scopo del progetto

Write something here

## 1.2 Specifiche generali

Write something here

## 1.3 Interfaccia del componente

Write something here

## 1.4 Dati e Descrizione memoria

Write something here

# 2 Desing Pattern

## 2.1 Scelte Progettuali

Write something here

## 2.2 Elenco Stati

### 2.2.1 START

Lo stato di **START** è stato pensato come stato di attesa iniziale. Questo stato viene invocato in due situazioni differenti: se il segnale di **i\_rst** viene portato alto, oppure quando il segnale **i\_start** viene riportato basso dopo la computazione di un immagine. Lo stato **START** non cambia fino a quando il segnale **i\_start** non viene portato alto. In quel momento lo stato successivo viene impostato **INIT**.

### 2.2.2 INIT

Lo stato **INIT** è uno stato di transizione nel quale il processore si assicura che i segnali siano inizializzati con i valori opportuni. Successivamente imposta lo stato prossimo a **ABILIT\_READ**.

### 2.2.3 ABILIT\_READ

Lo stato **ABILIT\_READ** è lo stato attraverso il quale abilitiamo la memoria alla sola lettura. Viene richiamato in momenti diversi del progetto e, in base allo stato chiamante, instrada lo stato prossimo a quello opportuno.

### 2.2.4 ABILIT\_WRITE

Lo stato **ABILIT\_WRITE** abilita la memoria alla lettura e alla scrittura. Viene invocato subito dopo aver computato il valore del nuovo pixel e in nessun altro momento. Instrada poi lo stato prossimo a **WRITE\_PIXEL**.

### 2.2.5 WAIT\_MEM

Lo stato WAIT\_MEM è uno stato centrale durante la gestione del flusso di dati. Sostanzialmente "spreca" un ciclo di clock. Questo ci assicura sia in caso di scrittura, sia in caso di lettura, che i segnali in ingresso e in uscita siano letti o scritti correttamente. Nel caso specifico alla quale ci rifacciamo, alcune chiamate a questo stato potevano essere evitate. Questa informazione è emersa durante lo stress test a cui il processore è stato sottoposto. Tuttavia, abbiamo preferito lasciarle per mantenere la struttura del processore. Ciò, a nostro avviso, permette una maggior robustezza, sebbene un aumento nella latenza della computazione.

### 2.2.6 GET\_RC

Lo stato GET\_RC è uno stato in preparazione al calcolo della dimensione dell'immagine e dei punti in cui bisognerà scrivere all'interno della memoria. Lo stato GET\_RC viene invocato dopo l'abilitazione della memoria alla lettura. Questo stato si occupa del recuperare i valori dalla memoria e aggiornare i segnali `n_col` e `n_row`.

### 2.2.7 GET\_DIM

Lo stato GET\_DIM è lo stato che si preoccupa di aggiornare il segnale `dim_address` con il valore opportuno. Il calcolo  $n\_col \cdot n\_row + 2$  aggiorna il segnale al primo bit libero per la scrittura.

### 2.2.8 READ\_PIXEL

Lo stato READ\_PIXEL è uno stato strettamente accoppiato con lo stato GET\_MINMAX. Richiede alla memoria il valore del pixel e aggiorna il segnale `curr_address` a quello successivo. In questo modo il valore del pixel sarà disponibile sul segnale `i.data` al `rising-edge` successivo.

### 2.2.9 GET\_MINMAX

Write something here

### 2.2.10 GET\_DELTA

Write something here

### 2.2.11 CALC\_SHIFT

Write something here

### 2.2.12 GET\_PIXEL

Write something here

### 2.2.13 CALC\_NEWPIXEL

Write something here

### 2.2.14 WRITE\_PIXEL

Write something here

### 2.2.15 DONE

Write something here

#### **2.2.16 WAITINGPIC**

Write something here

### **3 Risultati dei Test**

Write something here!

### **4 Conclusioni**

Write something here