

Progetto di Reti Logiche

William Zeni
matricola 10613915

Cristina Urso
matricola 10599689

3 maggio 2021

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



POLITECNICO
MILANO 1863

Indice

1	Introduzione	3
1.1	Scopo del progetto	3
1.2	Specifiche generali	3
1.3	Interfaccia del componente	3
1.4	Dati e Descrizione memoria	3
2	Desing Pattern	3
2.1	Scelte Progettuali	3
2.1.1	START	3
2.1.2	INIT	3
2.1.3	ABILIT_READ	3
2.1.4	ABILIT_WRITE	3
2.1.5	WAIT_MEM	3
2.1.6	GET_RC	4
2.1.7	GET_DIM	4
2.1.8	READ_PIXEL	4
2.1.9	GET_MINMAX	4
2.1.10	GET_DELTA	4
2.1.11	CALC_SHIFT	4
2.1.12	GET_PIXEL	4
2.1.13	CALC_NEWPIXEL	4
2.1.14	WRITE_PIXEL	4
2.1.15	DONE	4
2.1.16	WAITINGPIC	4
3	Risultati dei Test	4
4	Conclusioni	4

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

2.1.1 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** rimane tale fino a quando il segnale **i_start** non venga posto alto. In quel momento lo stato successivo viene impostato **INIT**.

2.1.2 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**.

2.1.3 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 ed in base allo stato chiamante, instrada lo stato prossimo a quello opportuno.

2.1.4 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**.

2.1.5 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 una aumento nella latenza della computazione.

2.1.6 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**.

2.1.7 GET_DIM

Write something here

2.1.8 READ_PIXEL

Write something here

2.1.9 GET_MINMAX

Write something here

2.1.10 GET_DELTA

Write something here

2.1.11 CALC_SHIFT

Write something here

2.1.12 GET_PIXEL

Write something here

2.1.13 CALC_NEWPIXEL

Write something here

2.1.14 WRITE_PIXEL

Write something here

2.1.15 DONE

Write something here

2.1.16 WAITINGPIC

Write something here

3 Risultati dei Test

Write something here!

4 Conclusioni

Write something here