

Politecnico di Milano

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
Laurea Triennale – Ingegneria Informatica

Progetto finale di reti logiche

Luca Vecchio
888901
10565156

Professore: **Palermo Gianluca**

Anno Accademico 2019 – 2020

1 Introduzione

Una parte sostanziale del consumo energetico di un processore è dovuta alla comunicazione input-output tra il processore e le altre componenti del calcolatore. Per ridurre questo consumo, è stata proposta una soluzione che mira ad abbassare il numero di segnali passanti sul Bus indirizzi tra il processore e la memoria, introducendo un metodo di codifica degli indirizzi basato su Working Zone. Questo metodo si basa sulla congettura che i processi eseguiti dal calcolatore accedono spesso ad un numero limitato di zone di lavoro del loro spazio di indirizzi riservato in memoria. Il metodo Working Zone consente di trasformare il valore di un indirizzo quando questo viene trasmesso, se appartiene a certi intervalli, detti working-zone (WZ). Una working-zone è definita come un intervallo di indirizzi di dimensione fissa (D_{WZ}) che parte da un indirizzo base. All'interno dello schema di codifica possono esistere multiple working-zone. Una volta impostato l'indirizzo di riferimento di una working-zone e la sua dimensione, il processore può accedere agli indirizzi in memoria utilizzando solamente l'offset rispetto all'indirizzo di riferimento, che vuol dire un utilizzo minore del Bus indirizzi, che porta a sua volta ad un minor consumo di energia.

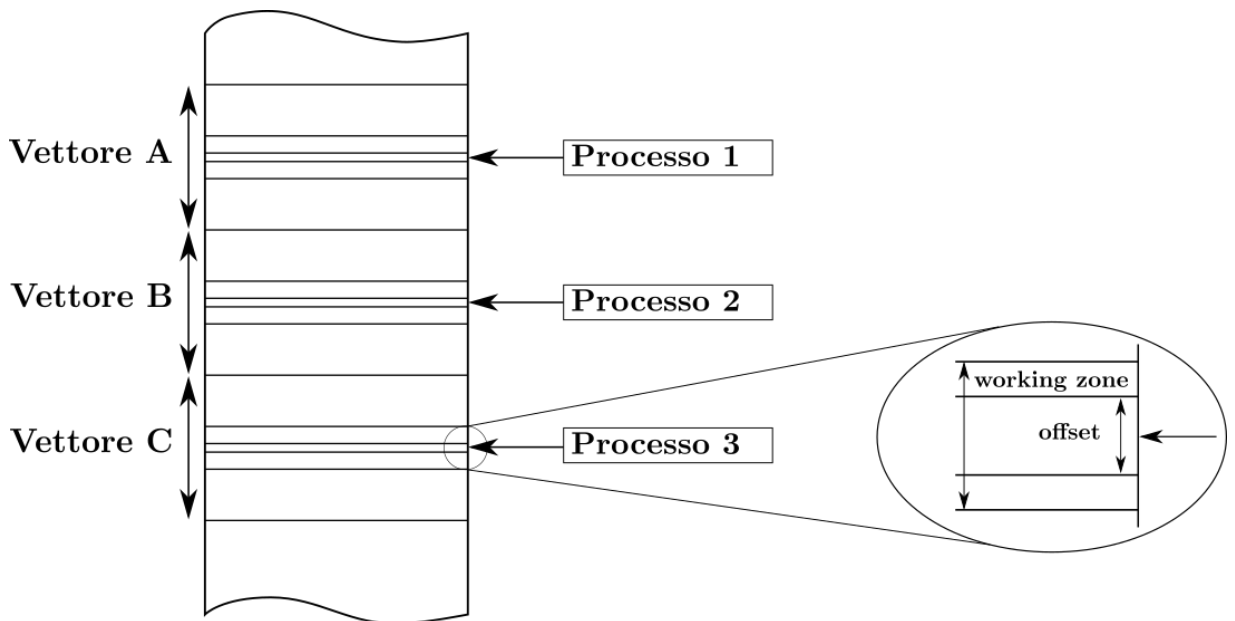


Figura 1. Spazio di indirizzamento con tre processi.

2 Codifica Working Zone

Nel contesto di questo progetto considereremo il numero di bit dell'indirizzo da codificare a 7 e il numero di working-zone fisso a 8, ciascuna di dimensione $D_{WZ} = 4$. Le posizioni in memoria da 0 a 7 sono usati per memorizzare gli otto indirizzi base delle working-zone, mentre l'ottavo elemento contiene l'indirizzo da codificare e la nona posizione è dedicata al valore codificato.

La codifica avviene tramite il componente in esame per questo progetto, che verifica se l'indirizzo appartiene o no ad una working-zone:

- Se l'indirizzo non appartiene ad una working-zone il dato in uscita sarà codificato come segue:



- Se l'indirizzo è contenuto all'interno di una working-zone il componente restituirà alla memoria il dato con la codifica seguente, dove WZ_OFFSET è codificato one-hot col bit 0 considerato come il meno significativo:

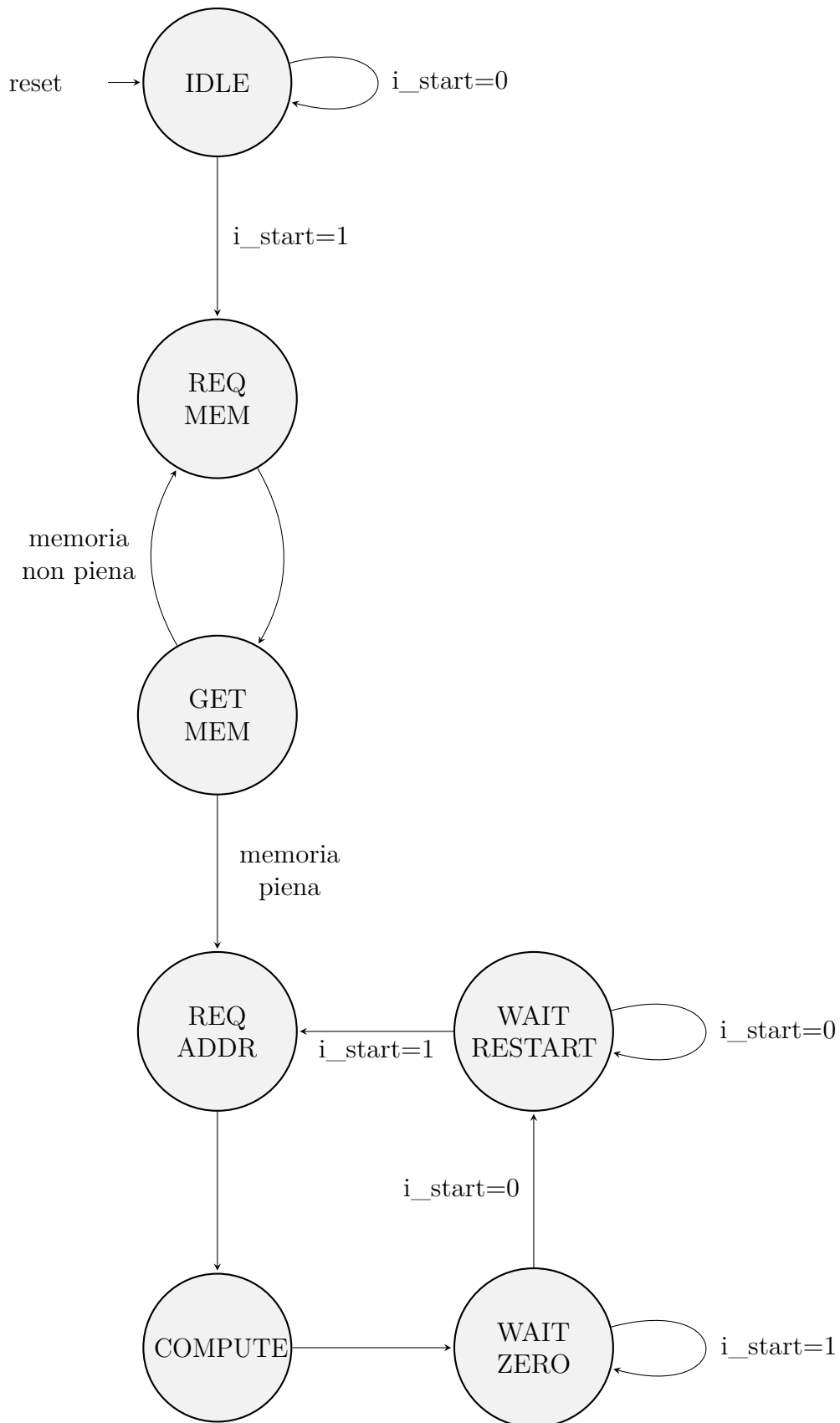


3 Architettura del componente

Il progetto prevede l'utilizzo della **FPGA xc7a200tfbg484-1**, con frequenza di clock pari a 10MHz. È stato scelto di descrivere il componente in modo puramente comportamentale (behavioral).

Essendo la codifica Working Zone basata sulla congettura che un programma predilige un intervallo di indirizzi in memoria, il componente fa uso di una memoria interna di 64 bit ad accesso diretto nella quale vengono memorizzati i riferimenti alle working-zone, in modo da ridurre il numero di interazioni con la memoria esterna e il tempo di esecuzione di codifiche ripetute, con una conseguente riduzione ulteriore di potenza elettrica dissipata. In particolare, dopo la ricezione di un segnale di *reset*, il modulo impiega 64 cicli di clock per memorizzare gli indirizzi di riferimento delle working-zone, per poi impiegare solamente 8 cicli per ogni codifica, fino alla ricezione di un altro segnale di *reset*. Il funzionamento del componente può essere formalizzato attraverso una *macchina a stati finiti*.

3.1 Macchina a stati



- **IDLE:** Stato raggiunto subito dopo l'accensione del dispositivo, oppure esclusivamente in seguito alla ricezione del segnale di *reset*. Ogni segnale interno ed esterno viene inizializzato. La macchina avanza allo stato *REQ MEM* non appena il segnale in ingresso *i_start* passa al valore 1.
- **REQ MEM:** Viene richiesto il dato in posizione k , dove k è un contatore con valori compresi tra 0 (memoria interna vuota) e 7 (memoria interna piena). Si avanza allo stato *GET MEM*.
- **GET MEM:** Viene ricevuto e memorizzato il dato richiesto dallo stato precedente, inoltre viene incrementato il valore del contatore k . Se $k = 7$, ovvero la memoria interna è piena, la macchina avanza allo stato *REQ ADDR*, altrimenti torna allo stato *REQ MEM*.
- **REQ ADDR:** Viene richiesto alla memoria il dato in posizione 8, ovvero l'indirizzo da codificare. Si avanza allo stato *COMPUTE*.
- **COMPUTE:** In questo stato viene ricevuto il dato richiesto dallo stato precedente, che viene quindi codificato e inviato alla memoria esterna all'indirizzo 9. La macchina avanza allo stato *WAIT ZERO*.
- **WAIT ZERO:** Si aspetta che il segnale *i_start* passi a 0: in caso affermativo si passa a *WAIT RESTART*, altrimenti la macchina rimane allo stato corrente.
- **WAIT RESTART:** Si aspetta una nuova codifica, quindi la macchina resta allo stato corrente finché *i_start*=0, altrimenti avanza allo stato *REQ ADDR*.

4 Risultati sperimentali

I seguenti dati sono relativi alla **FPGA xc7a200tfbg484-1**.

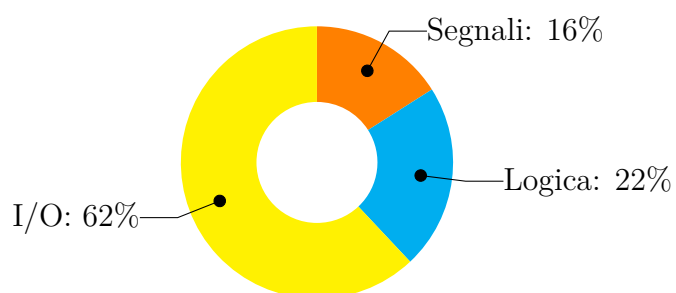
4.1 Sintesi

Risorsa	Utilizzo	Disponibilità	Utilizzo %
LUT	303	134600	0.23
FF	98	269200	0.04
I/O	38	285	13.33

4.2 Potenza dissipata

Essendo la potenza dissipata al centro della logica del metodo working-zone è stato fatto un resoconto energetico del modulo con parametri industriali e carico massimo.

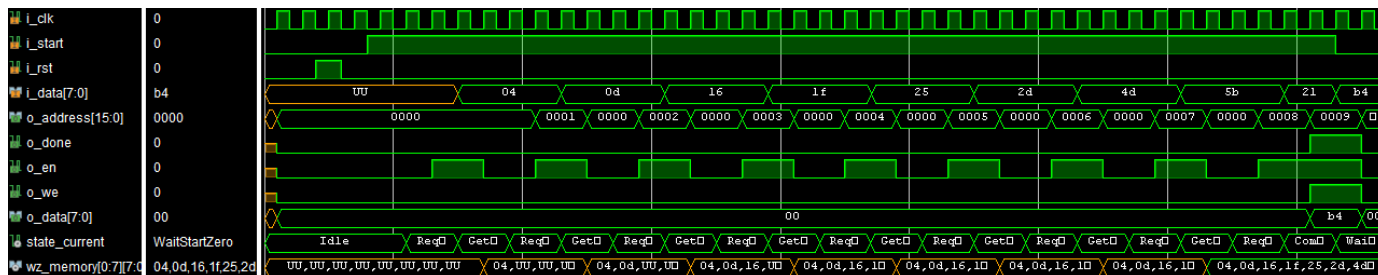
Risorsa	Potenza	Potenza %
Segnali	0.566 W	16%
Logica	0.786 W	22%
I/O	2.155 W	62%
Totale	3.643 W	100%



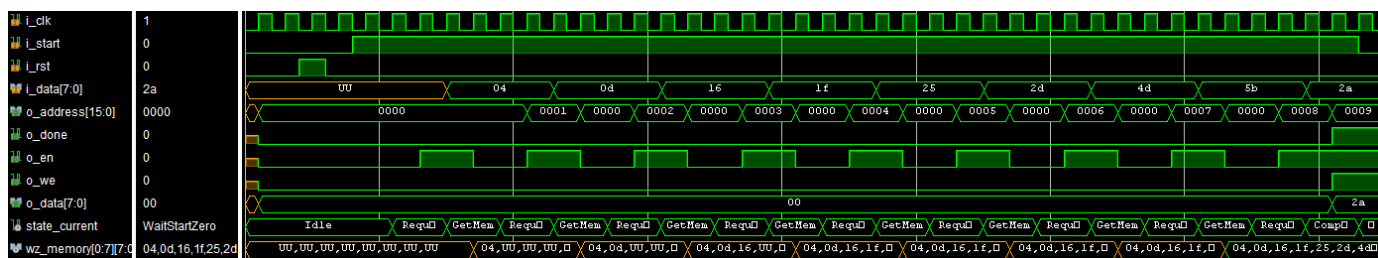
5 Testing

Per verificare il corretto funzionamento del componente sono stati effettuati i seguenti test:

- Test di funzionamento base, in cui il dato da codificare appartiene alla working-zone:



- Test di funzionamento base, in cui il dato da codificare non appartiene alla working-zone:



- Multipli segnali di *reset* durante l'esecuzione, con conseguenti variazioni dei valori all'interno della memoria esterna, che simulano un cambio di contesto all'interno dell'esecuzione di uno o più programmi;
- Multiple codifiche all'interno dello stesso contesto, ovvero l'utilizzo degli stessi valori presenti nella memoria esterna per più codifiche consecutive;
- Combinazioni dei due test sopra elencati, che simulano quindi esecuzioni prolungate di un numero di contesti, ovvero variazioni dei valori all'interno della memoria esterna ad ogni segnale di *reset*, e un numero variabile di codifiche consecutive tra i segnali di *reset*.

6 Conclusione

Il modulo progettato consente al calcolatore di ridurre il consumo energetico dovuto all'utilizzo del Bus indirizzi, riducendo il numero di segnali scambiati tra il processore e la memoria.

Il componente si rivela tanto più efficiente in termini di potenza dissipata e tempo medio impiegato per le codifiche quanto più è alto il numero di codifiche richieste prima di un cambio di contesto, come ad esempio in applicazioni Internet Of Things, per le quali l'efficienza energetica è di vitale importanza e il contesto varia raramente.