Politecnico di Milano

Progetto Finale di Reti Logiche

Codifica Working Zone

Nicolò Sonnino

supervisore
Prof. Gianluca Palermo

11 agosto 2020

Indice

1 Specifiche

1.1 Descrizione

Il progetto ha come scopo la realizzazione della codifica **working zone**, questa strategia viene utilizzata sopratutto nello sviluppo di microprocessori. Il consumo di energia dei pin è una parte integrante nella loro progettazione, per ridurlo, si suppone, che determinati programmi favoriscano poche zone all'interno del loro spazio di memoria in ogni istante.
Quindi vengono identificate tali zone e con un riferimento dell'indirizzo scelto viene inviato anche un offset codificato "one-hot" per trovare la sua posizione rispetto all'indirizzo base della working zone.

1.2 WZE (Working Zone Encoding)

Il progetto presenta una memoria RAM da 65536 indirizzi con valore base di 8 bit per ogni cella di memoria; di questi i primi otto sono riempiti con indirizzi base, mentre il nono contiene il valore da codificare e il decimo è quello riservato alla scrittura del risultato codificato.
A questo punto abbiamo due possibili casi: il valore rientra in una delle working zones, oppure non appartiene a nessuno dei sette indirizzi; nel primo caso si identifica la working zone (WZ_NUM) e l'offset rispetto ad essa (WZ_OFFSET), si pone WZ_BIT= 1 e si scrive nell'indirizzo di posizione 9 WZ_BIT & WZ_NUM & WZ_OFFSET[1], mentre nel secondo caso viene posto WZ_BIT= 0, il valore (ADDR) viene salvato e l'output risulta WZ_BIT & ADDR.
L'offset **one-hot** associa all'unico 1 il valore da rappresentare nel seguente modo:

- WZ_OFFSET = 0 => **"0001"**
- WZ_OFFSET = 1 => **"0010"**
- WZ_OFFSET = 2 => **"0100"**
- WZ_OFFSET = 3 => **"1000"**

[1] & simbolo di concatenazione

1.3 Esempi

WZ_BIT	WZ_NUM	WZ_OFFSET
1	3 bits	4 bits

1|010|0100

	n°indirizzo	Dmz
0	0000000000000000	17
1	⋮	9
2	⋮	**33**
3	⋮	42
4	⋮	54
5	⋮	88
6	⋮	69
7	⋮	180
8	⋮	**35**
9	0000000000001001	**164**

Figura 1: Valore contenuto in posizione 2

WZ_BIT	ADDR
0	7 bits

0|0110110

	n°indirizzo	Dmz
0	0000000000000000	17
1	⋮	9
2	⋮	23
3	⋮	42
4	⋮	115
5	⋮	88
6	⋮	69
7	⋮	180
8	⋮	**54**
9	0000000000001001	**54**

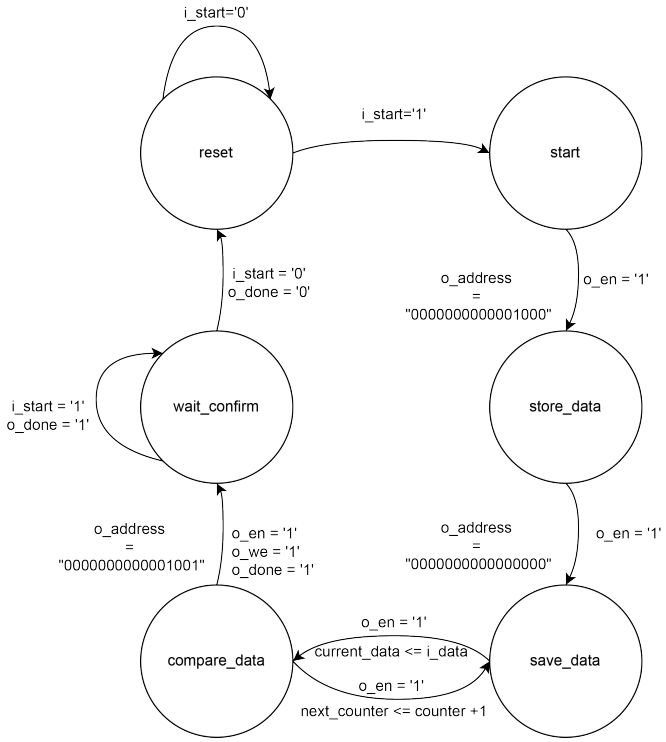
Figura 2: Valore non contenuto in nessuna working zone

2 Architettura

Di seguito viene mostrata la FSM (Final State Machine) in versione semplificata, sono omessi tutti gli anelli che da qualsiasi stato tornano a **reset** quando **i_rst='1'**.

Descrizione di ogni stato:

- **reset**: stato di inizializzazione, se il segnale **i_rst** viene alzato si ritorna a questo stato. Fintanto che **i_start** rimane basso si ritorna a reset, se viene alzato si passa a start.
- **start**: stato di preparazione, i segnali **o_we, wz_bit** vengono abbassati, l'**offset** viene posto a 0000; per permettere la lettura dell'indirizzo in posizione 8, **o_en** viene alzato a 1 e si assegna 0000000000001000 a **o_address**.
- **store_data**: stato di memorizzazione del valore da codificare, **i_data** letto precedentemente viene salvato nel registro **data** e viene inizializzato il valore **next_counter**, responsabile di iterare gli indirizzi di memoria.



- **save_data**: stato di salvataggio, il valore di **i_data** viene salvato per essere utilizzato per confrontarlo successivamente.
- **compare_data**: stato di confronto dei valori, attraverso un contatore interno allo stato si confronta il valore di **data** con **current_data** con sommato il contatore; se risultano uguali viene abilitata la scrittura tramite l'assegnazione di **o_en** e **o_we** a 1 e **o_address** viene posto a 9, infine **o_data** viene codificato come discusso nelle specifiche e **o_done** viene alzato a 1. Nel caso in cui non appartenga alla working

zone, se non è al ottavo indirizzo, il contatore viene incrementato e si ritorna allo stato precedente, altrimenti viene abilitata la scrittura del valore codificato e si alza **o_done**.

- **wait_confirm**: stato di attesa, se allo stato precendente è stato alzato **o_done** si arriva a quest'ultimo; vengono quindi azzerati i segnali di scrittura e lettura, **o_done** rimane alzato fino a quando non viene ricevuto un segnale **i_start** = 0 e allora in quel caso **o_done** viene abbassato e si ritorna a reset.

3 Risultati Sperimentali

3.1 Sintesi

Il progetto è stato testato sulla versione di **Vivado 2016.4**, utilizzando come FPGA target **xc7a200tfbg484-1**, e ha generato il seguente report di sintesi:

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	46	0	134600	0.03
LUT as Logic	46	0	134600	0.03
LUT as Memory	0	0	46200	0.00
Slice Registers	19	0	269200	<0.01
Register as Flip Flop	5	0	269200	<0.01
Register as Latch	14	0	269200	<0.01
F7 Muxes	0	0	67300	0.00
F8 Muxes	0	0	33650	0.00

Per gli stati e il consumo invece:

State	New Enc.	Previous Enc.
reset	000	000
start	001	001
store_data	010	010
save_data	011	011
compare_data	100	100
wait_confirm	101	101

Type	Power	Util%
Signals	0.448 W	16%
Logic	0.285 W	10%
I/O	2.059 W	74%
Device Static	0.142 W	5%

3.2 Simulation

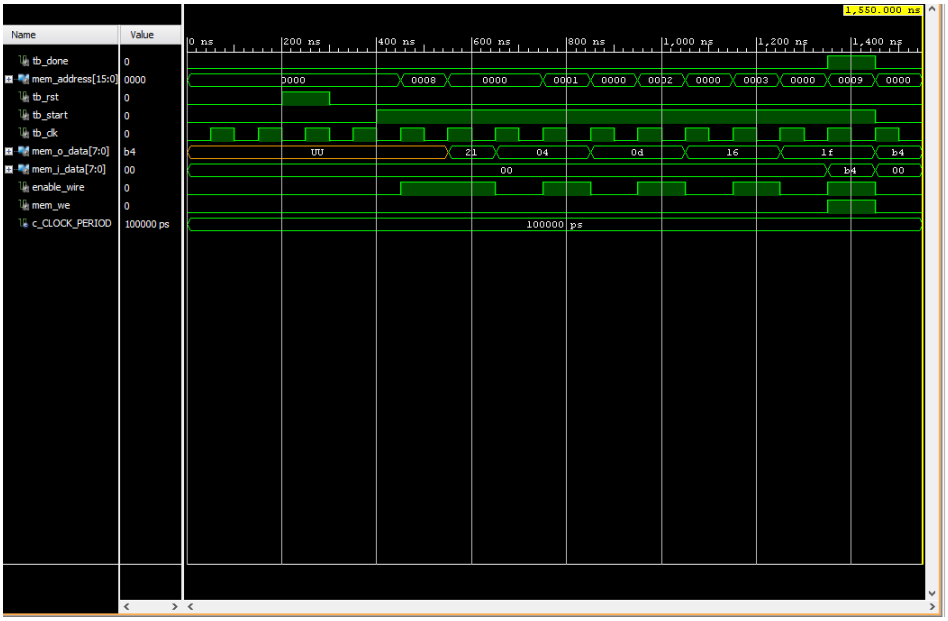


Figura 3: tb_pfrl_2020_in_wz

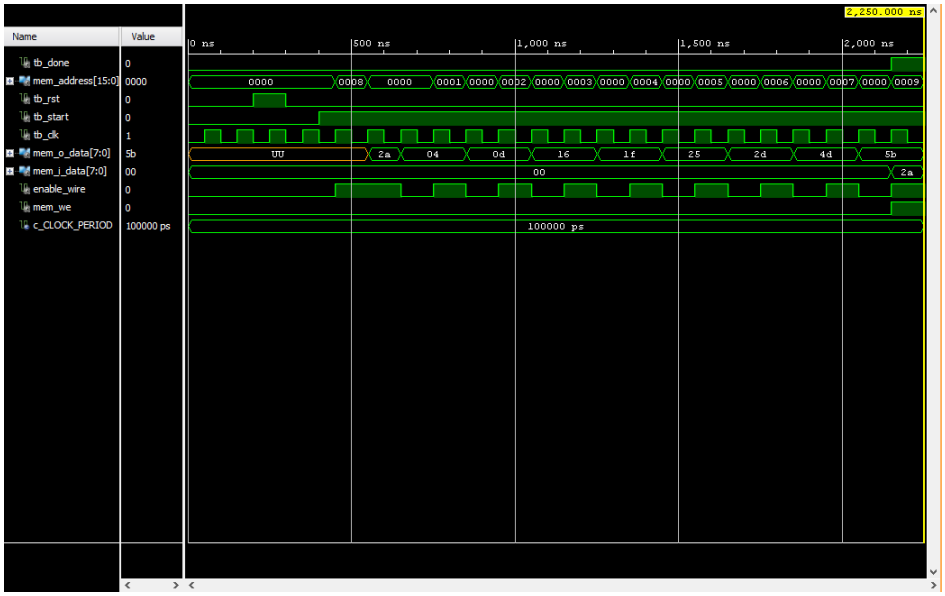
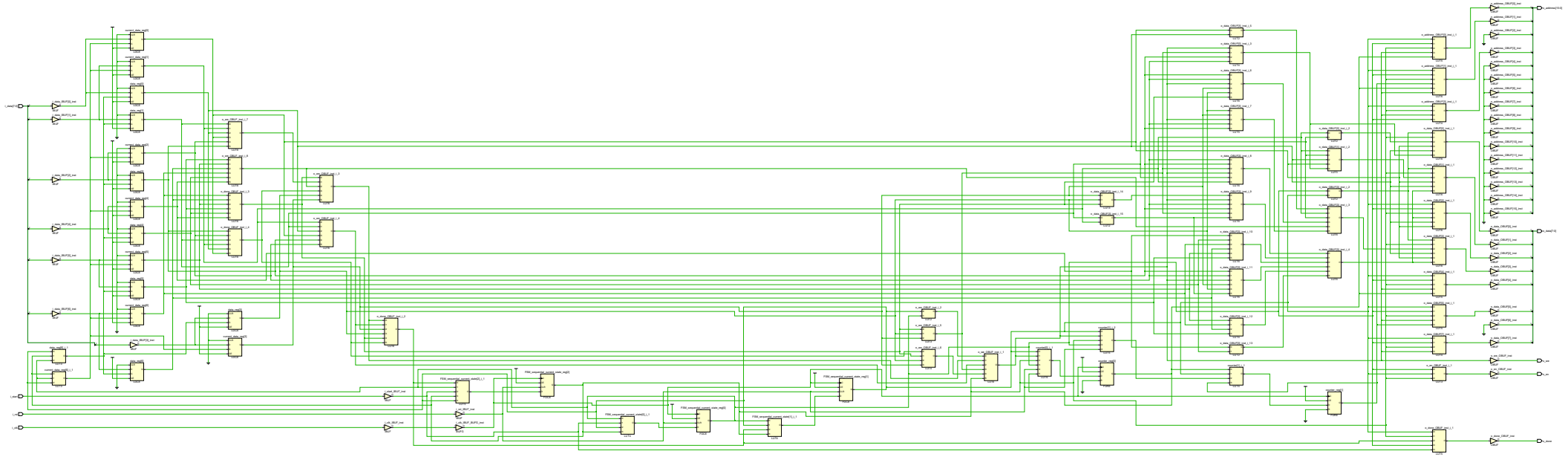


Figura 4: tb_pfrl_2020_no_wz

3.3 Schematic



4 Test Benches

4.1 tb_pfrl_2020_in_wz

In questo test bench si vuole verificare la correttezza del progetto nel caso di un valore appartenente a una working zone, in particolare il valore da codificare è 33. La posizione 3 di memoria ha il valore 31, quindi l'output corretto per passare questo testbench è 1 (WZ_BIT) & 011 (WZ_NUM) & 0100(OFFSET).

4.2 tb_pfrl_2020_no_wz

Lo scopo di questo test bench è verificare che nel caso di un valore non appartenente a una working zone, l'output risulti corretto. Poichè il valore da codificare è 42 e la working zone più vicina ad esso è 37, quindi non esiste $0 \leq$offset≤ 3 tale da raggiungere il valore, l'output risulta 0 (WZ_BIT) & 0101010 (42).

4.3 tb_casi_limite

Sono stati testati i casi limite quali:

- valore vicino a una working zone per un offset negativo;
- valore appartenente alla prima working zone;
- valore appartenente all'ultima working zone;
- multipli segnali di reset;
- attesa di segnale o_done = 0.

Per tutti questi casi il progetto ha ottenuto risultati positivi.

5 Conclusioni

Dai risultati sperimentali e dalla sintesi si può notare che si ha studiato la codifica working zone con offset one-hot nella sua interezza: testando casi limite, superando le due test benches forniteci, sintetizzando il progetto e analizzandone i risultati. Una possibile ottimizzazione potrebbe essere gestire

il valore di o_data ad ogni stato con un registro apposito per evitare di avere valori undefined, anche quando non ci interessa osservarlo, e quindi abbassare il numero di LUT ed eliminare un inferring latch.