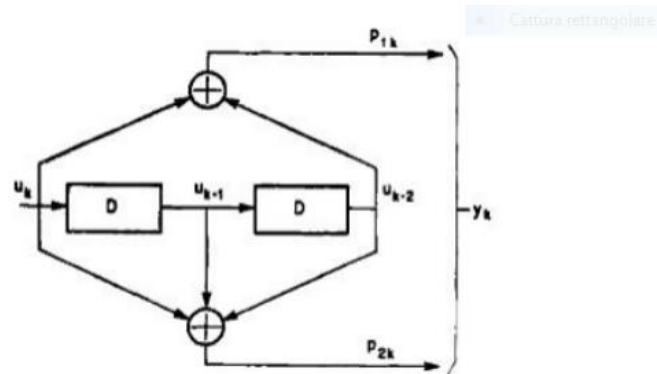


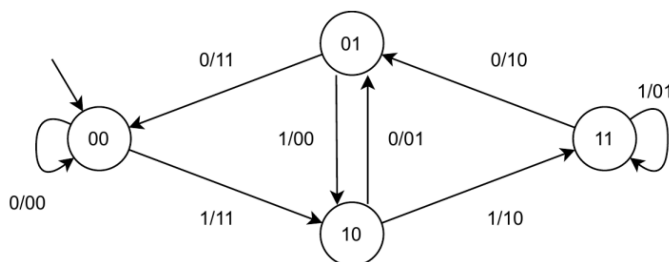
1) specifiche di progetto

La specifica della “Prova Finale (Progetto di Reti Logiche)” 2021/2022 chiede di implementare un modulo HW (descritto in VHDL) che si interfacci con una memoria e che segua la seguente specifica.



Codificatore convoluzionale con tasso di trasmissione $\frac{1}{2}$.

Il modulo riceve in ingresso una sequenza continua di W parole, ognuna di 8 bit, e restituisce in uscita una sequenza continua di Z parole, ognuna da 8 bit. Ognuna delle parole di ingresso viene serializzata; in questo modo viene generato un flusso continuo U da 1 bit. Su questo flusso viene applicato il codice convoluzionale $\frac{1}{2}$ (ogni bit viene codificato con 2 bit) secondo lo schema riportato in figura; questa operazione genera in uscita un flusso continuo Y . Il flusso Y è ottenuto come concatenamento alternato dei due bit di uscita. Utilizzando la notazione riportata in figura, il bit u_k genera i bit p_{1k} e p_{2k} che sono poi concatenati per generare un flusso continuo y_k (flusso da 1 bit). La sequenza d'uscita Z è la parallelizzazione, su 8 bit, del flusso continuo y_k .



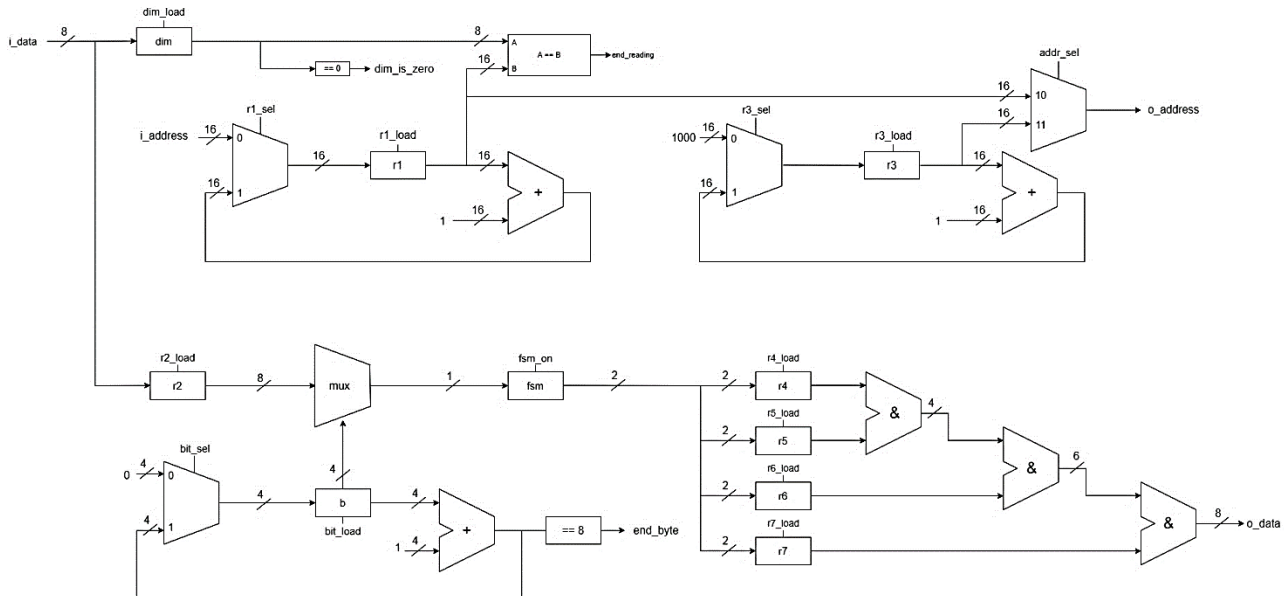
Dati

Il modulo da implementare deve leggere la sequenza da codificare da una memoria con indirizzamento al Byte in cui è memorizzato; ogni singola parola di memoria è un byte. La sequenza di byte è trasformata nella sequenza di bit U da elaborare. La quantità di parole W da codificare è memorizzata nell'indirizzo 0; il primo byte della sequenza W è memorizzato all'indirizzo 1. Lo stream di uscita Z deve essere memorizzato a partire dall'indirizzo 1000 (mille). La dimensione massima della sequenza di ingresso è 255 byte.

2) scelte progettuali

La descrizione VHDL del componente si compone di due moduli principali: un modulo datapath e un modulo di controllo basato sulla macchina a stati.

DATAPATH



La parte superiore del datapath serve per la lettura del numero di byte da elaborare e per il calcolo degli indirizzi.

- Nel registro DIM verrà letto il numero di byte da elaborare.
- Nel registro R1 vengono memorizzati gli indirizzi delle celle di memoria contenenti i byte da leggere ed elaborare.
- Nel registro R3 vengono memorizzati gli indirizzi dove andare a scrivere i byte elaborati nella memoria.

Il segnale `addr_sel` controlla un multiplexer che può selezionare in uscita su `o_address`:

- Il valore 10: per la lettura della quantità di parole (ogni parola di memoria è un byte) e delle singole parole di memoria da elaborare;
- Il valore 11 per la scrittura delle parole in memoria .

Il registro R1 parte da 0 e venendo incrementato man mano di 1 permette di generare gli indirizzi da 0 fino a `dim` (ovvero il numero dei byte da elaborare) necessari per la lettura all'ingresso 10 del mux.

Il registro R3 parte da 1000 e genera gli indirizzi fino a $(1000 + \text{dim} * 2) - 1$ necessari per la scrittura all'ingresso 11 del mux.

La parte inferiore del datapath effettua l'elaborazione dei byte vera e propria.

- Nel registro R2 viene letto dalla memoria il byte corrente da elaborare.
- Il registro B rappresenta l'indice di selezione degli otto bit del byte e dovrà contenere valori che vanno da 0 a 7.
- I registri R4, R5, R6 e R7 memorizzano ciascuno due bit che verranno concatenati per formare un byte.

Viene eseguita quindi un'operazione di scorrimento del byte memorizzato nel registro r2: il multiplexer, sulla base del valore contenuto nel registro B seleziona quale bit del byte deve essere dato in pasto alla macchina a stati (si tratta di una macchina di Mealy in cui l'uscita dipende sia dallo stato corrente che dall'ingresso attuale).

La computazione procede iterando la somma del valore contenuto nel registro B con 1 e selezionando i singoli bit che entrano nella macchina. Questi ultimi danno origine a due bit ciascuno che man mano vengono memorizzati rispettivamente nei registri R4, R5, R6 e R7 che hanno il compito di eseguire la concatenazione dei bit fino a formare il nuovo byte da caricare in memoria.

MACCHINA A STATI

Il componente è stato implementato progettando una macchina di Moore a stati finiti composta da 15 stati. Di seguito viene fornita una descrizione di questi ultimi:

S0: in questo stato il componente si limita ad attendere la ricezione di un segnale `i_start` per dare inizio alla computazione;

S1: questo stato è usato per fornire alla memoria l'indirizzo della cella contenente il numero di byte da elaborare (cella di memoria di indice 0);

S2: questo stato serve per fare in modo che il componente possa caricare nel registro `dim` il valore che indica il numero dei byte da elaborare;

S3: questo stato viene usato per incrementare il valore dell'indirizzo memorizzato nel registro `r1` (a tale indirizzo si troverà il primo byte da elaborare);

S4: questo stato si occupa di richiedere dalla memoria il byte che deve essere caricato nel registro `r2` per poter essere successivamente elaborato;

S5: in questo stato il byte viene memorizzato nel registro `r2` e viene posto a 0 il valore del registro `b` che svolge la funzione di indice che seleziona uno ad uno i bit del byte

S6: in questo stato inizia l'elaborazione del byte: viene processato il primo bit della prima/seconda metà del byte corrente. Questo dà origine a due bit da caricare nel registro `r4`;

S7: in questo stato avviene l'elaborazione del secondo bit della prima/seconda metà del byte e la conseguente produzione dei due bit da caricare nel registro `r5`;

S8: in questo stato avviene l'elaborazione del terzo bit della prima/seconda metà del byte e la conseguente produzione dei due bit da caricare nel registro `r6`;

S9: in questo stato avviene l'elaborazione del quarto bit della prima/seconda metà del byte e la conseguente produzione dei due bit da caricare nel registro `r7`;

S10: in questo stato il nuovo byte prodotto viene memorizzato nella cella di memoria il cui indirizzo è contenuto nel registro `r3`.

A questo punto se il segnale `end_byte` è basso si passa allo stato `S11`.

S11: questo stato viene raggiunto quando è terminata l'elaborazione della prima metà del byte e bisogna quindi passare ai restati quattro bit che costituiscono la seconda metà.

Vengono incrementati il valore dell'indirizzo a cui verrà salvato il prossimo nuovo byte (nel registro `r3`) e il valore dell'indice che scorre il byte (nel registro `b`). Si prosegue quindi ripercorrendo gli stati `s6`, `s7`, `s8`, `s9`, `s10` per produrre i nuovi bit da concatenare.

Se invece il segnale `end_byte` è alto e il segnale `end_reading` è basso si va nello stato `S12`.

Gli stati **S12**, **S13**, **S14** vengono eseguiti nel caso in cui in memoria ci siano ancora dei byte da elaborare.

Vengono incrementati i valori dei registri che contengono gli indirizzi di lettura del prossimo byte e di scrittura del byte prodotto (registri `r1` e `r3`) e viene richiesto dalla memoria il prossimo byte da elaborare.

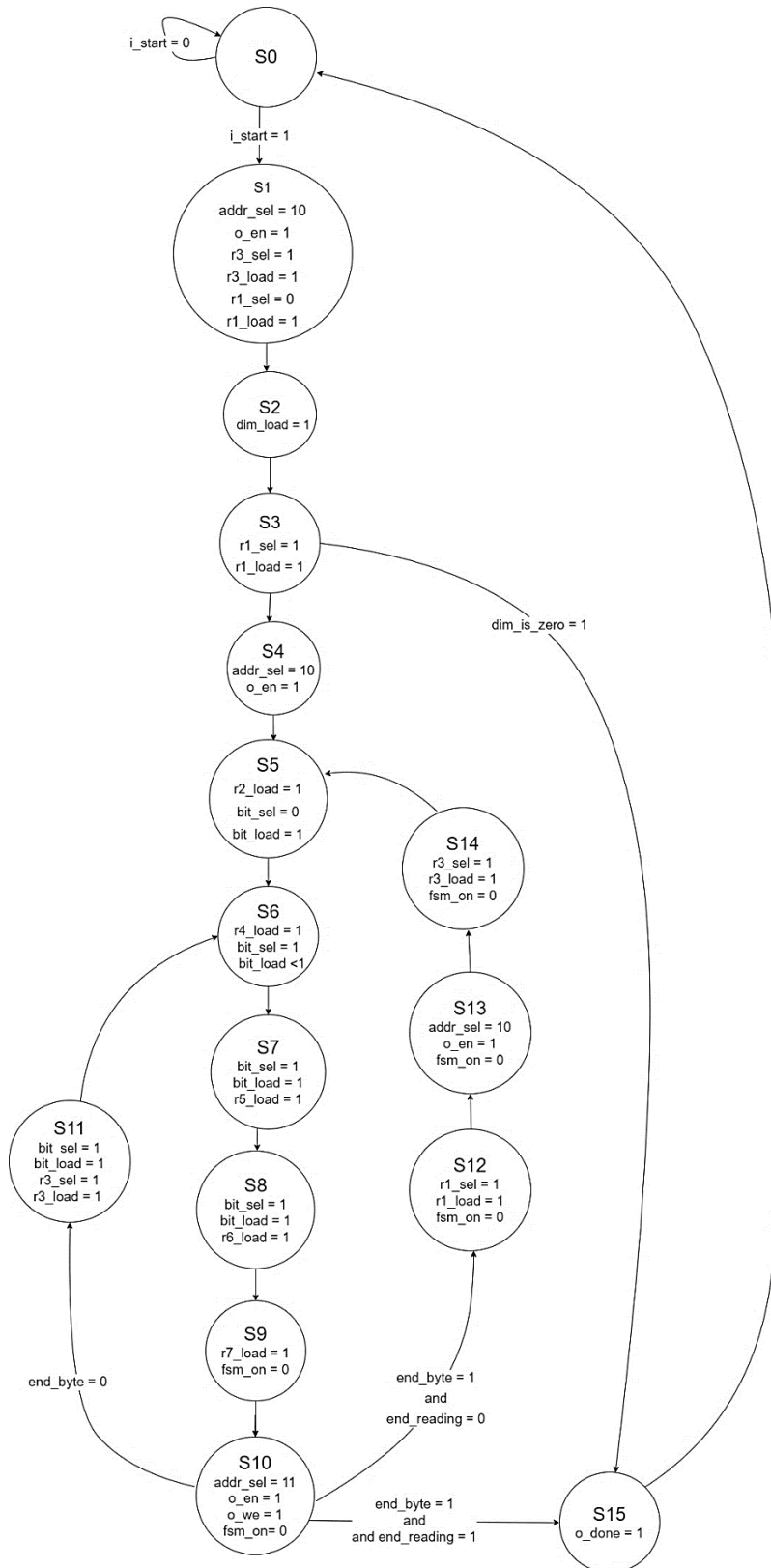
La computazione finisce in 2 casi:

- Se nello stato `S3` è alto il segnale `dim_is_zero`. Ciò si verifica se il numero dei byte da elaborare è uguale a 0.

- Quando nello stato S10 sono alti i segnali end_byte e end_reading, ovvero se è terminata l'elaborazione dell'ultimo byte disponibile in memoria.

S15: rappresenta lo stato finale.

Raggiunto lo stato 15 si alza il segnale o_done e si aspetta in questo stato fino a quando i_start è alto. Quando questo segnale viene abbassato si ritorna allo stato S0 e si riabbassa o_done. Una nuova esecuzione partirà quando viene alzato di nuovo il segnale i_start.



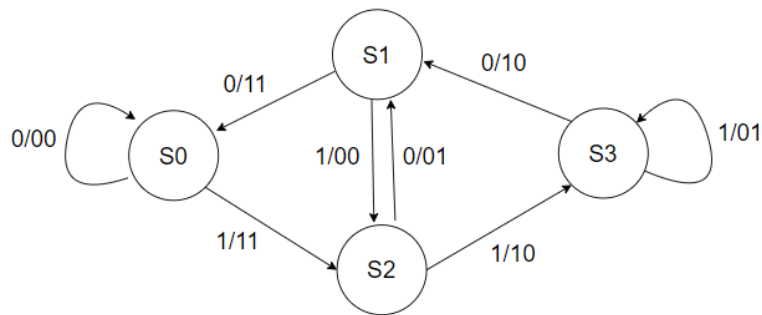
3) test

Per verificare il corretto comportamento del componente è stato eseguito un test con i seguenti valori (che qui sono rappresentati in decimale).

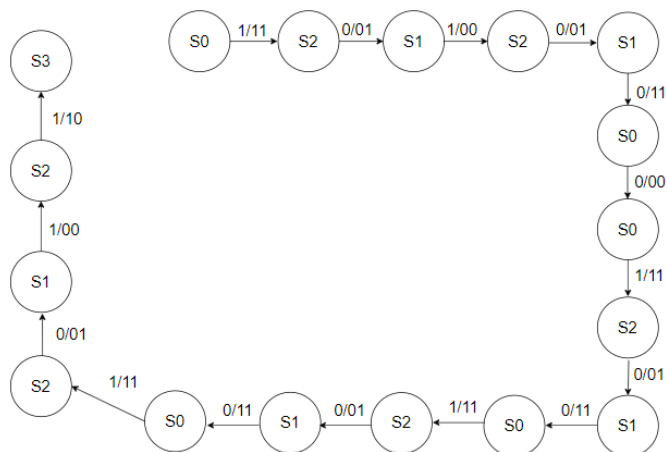
W: 10100010 01001011

Z: 11010001 11001101 11110111 11010010

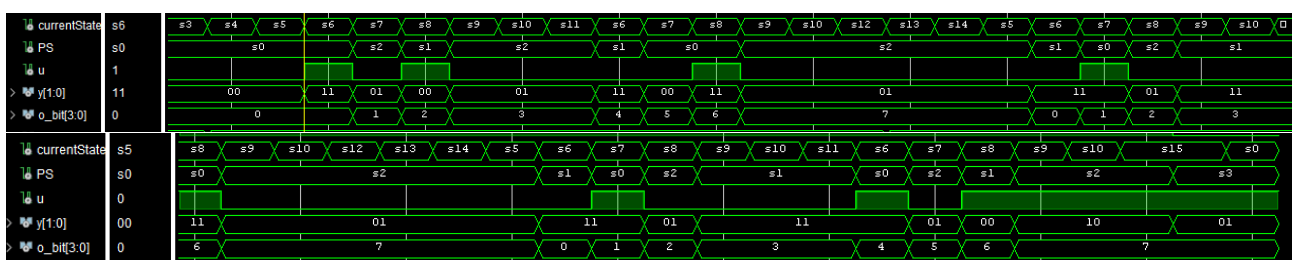
INDIRIZZO	MEMORIA	VALORE	COMMENTO
0		2	lunghezza sequenza di ingresso
1		162	primo Byte sequenza da codificare
2		75	
1000		209	primo Byte sequenza di uscita
1001		205	
1002		247	
1003		210	



LA SEQUENZA DEGLI STATI DA PERCORRERE dato l'input del test:



RISULTATO DELLA SIMULAZIONE:



Nella simulazione:

- u rappresenta il bit che entra nella macchina a stati per poter essere elaborato;
- y rappresenta i due bit in uscita dalla macchina a stati;
- o_bit è la posizione del bit elaborato all'interno del byte(va da 0 a 7);
- PS è lo stato corrente della macchina di Mealy.

4) risultati della sintesi

Il progetto è stato realizzato utilizzando come FPGA target xc7a200tfbg484-1 e ha generato il seguente report di sintesi:

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	74	0	134600	0.05
LUT as Logic	74	0	134600	0.05
LUT as Memory	0	0	46200	0.00
Slice Registers	78	0	269200	0.03
Register as Flip Flop	78	0	269200	0.03
Register as Latch	0	0	269200	0.00
F7 Muxes	1	0	67300	<0.01
F8 Muxes	0	0	33650	0.00

Da cui si può notare l'assenza dei latch (Register as Latch 0).