Prova Finale (Progetto di Reti logiche)

Prof. Gianluca Palermo

Descrizione generale

- SPECIFICHE E REGOLE del progetto sono disponibili su WEBEEP
  - Nella sezione forum c'è un thread di discussione dedicato alle specifiche e uno dedicato alle regole.

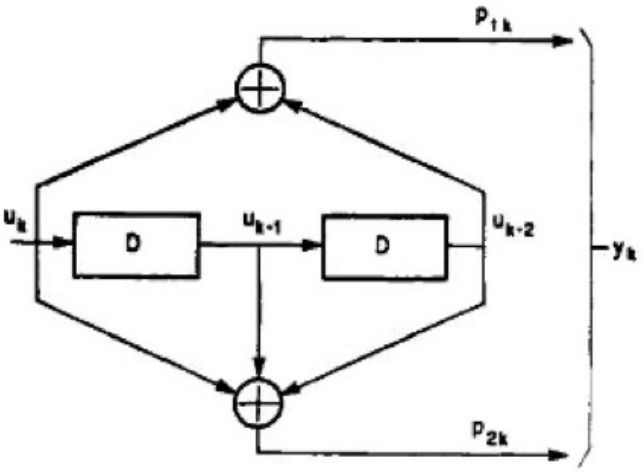
- La specifica della Prova Finale (Progetto di Reti Logiche) 2021-2022 è ispirata alle codifiche convoluzionali
- Le codifiche convoluzionali sono usate nelle telecomunicazioni come codici di correzione di errore e si utilizzano in numerose applicazioni allo scopo di ottenere un trasferimento di dati affidabile
- Nel progetto si vuole sviluppare un codice convoluzionale 1/2 , cioè un codice in cui per ogni bit in ingresso ne vengono generati 2 in uscita.

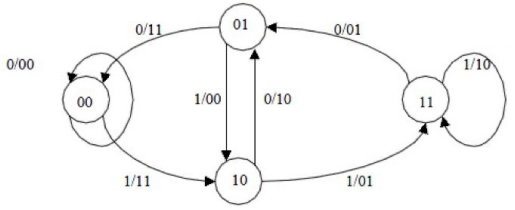
https://it.wikipedia.org/wiki/Codice_convoluzionale

DETTAGLI IMPLEMENTATIVI

- Il modulo da implementare
  - P1(t) = U(t) xor U(t-2)
  - P2(t) = U(t) xor U(t-1) xor U(t-2)
  - Y(t) è la concatenazione tra P1(t) e P2(t)
- Esempio
  - U = 10100010

T	0	1	2	3	4	5	6	7
Uk	1	0	1	0	0	0	1	0
P1k	1	0	0	0	1	0	1	0
P2k	1	1	0	1	1	0	1	1





DATI

- Il modulo da implementare dovrà leggere lo stream da una memoria con indirizzamento al byte dove 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;
  - *Dimensione massima dello stream 255 byte*
- 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).

Esempio1: (Sequenza lunghezza 2)
W: 10100010 01001011
Z: 11010001 11001101 11110111 11010010

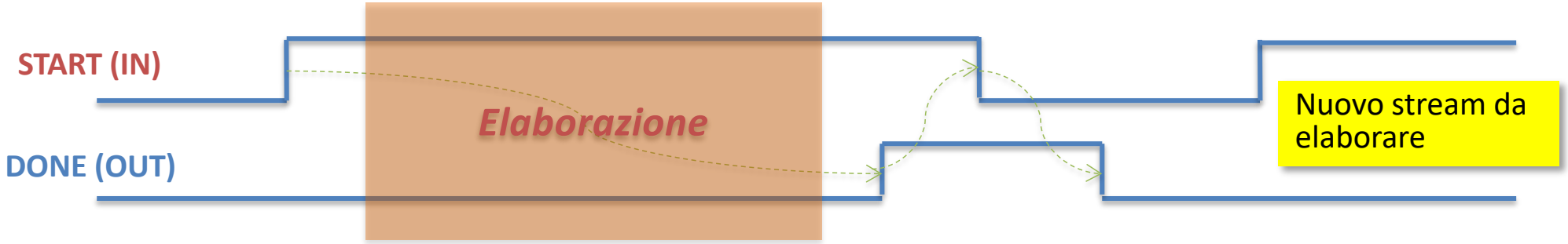
INDIRIZZO MEMORIA	VALORE	COMMENTO
0	2	\\ Byte lunghezza sequenza di ingresso
1	162	\\ primo Byte sequenza da codificare
2	75	
[...]		
1000	209	\\ primo Byte sequenza di uscita
1001	205	
1002	247	
1003	210	

Esempio3: (Sequenza lunghezza 3)
W: 01110000 10100100 00101101
Z: 00111001 10110000 11010001 11110111 00001101 00101000

INDIRIZZO MEMORIA	VALORE	COMMENTO
0	3	\\ Byte lunghezza sequenza di ingresso
1	112	\\ primo Byte sequenza da codificare
2	164	
3	45	
[...]		
1000	57	\\ primo Byte sequenza di uscita
1001	176	
1002	209	
1003	247	
1004	13	
1005	40	

Note ulteriori sulla specifica

- Il modulo deve essere progettato considerando che prima della prima codifica verrà SEMPRE dato il reset al modulo.
- Il modulo deve essere progettato per poter codificare più sequenze
- Una seconda elaborazione non dovrà attendere il reset del modulo, ma si deve rispettare il seguente protocollo



- L'immagine di ingresso può essere cambiata solo quando il segnale START è basso
- Il TB rispetterà SEMPRE questo protocollo

Interfaccia del Componente

```
entity project_reti_logiche is
      port (
            i_clk             : in  std_logic;
            i_start           : in  std_logic;
            i_rst             : in  std_logic;
            i_data            : in  std_logic_vector(7 downto 0);
            o_address         : out std_logic_vector(15 downto 0);
            o_done            : out std_logic;
            o_en              : out std_logic;
            o_we              : out std_logic;
            o_data            : out std_logic_vector (7 downto 0)
      );
end project_reti_logiche;
```