

# Relazione di vhd1

Verdicchio Giacomo e Zarbo Nicola

24 marzo 2022

## 1 Introduzione

////////////////////////////////////da inserire  
fatto da Nicola Zarbo e Giacomo Verdicchio

## 2 Architettura

Architettura: L'obiettivo è quello di riportare un schema funzionale (lo schema in moduli... un bel disegno chiaro... i segnali i bus, il/i clock, reset... );

### 2.1 Modulo 1

—————disegno di alto livello del funzionamento—————

Il modulo gestisce la lettura e scrittura da memoria e i segnali del modulo 2 'codificatore convoluzionale'.

#### 2.1.1 Segnali

i.clk : segnale di clock  
i.rst : segnale di reset asincrono  
i.start : segnale di enable '1'=> operativo,'0' => fermo (quindi il modulo viene resettato)  
i.data : bus 8 bit, dati di lettura da ram  
o.address : bus 16 bit, comunica alla ram l'indirizzo su cui eseguire lettura/scrittura  
o.done : segnale di finita elaborazione '1' => flusso elaborato/scritto in memoria  
o.en : segnale di enable per ram  
o.we : segnale per comunicare alla ram quale operazione svolgere, '0'=> read, '1' => write  
o.data : bus 8 bit, dati in scrittura per ram

#### 2.1.2 Registri

stato.att, st.prox (8 bit): registri di stato per fsm  
in.value (4 bit): dove viene copiato la parola di 8 bit letta da i.data  
out.value.buffer (8 bit) : dove viene scritta la parola da scrivere, collegato a o.data  
in.addr (16 bit): per mantenere address per lettura e per controllo terminazione codifica  
in.a.prox (16 bit): per incrementare l'address per l'operazione di read  
out.addr, out.a.prox (16 bit):registri per mantenere e incrementare address per write  
nTerminazione (9 bit): usato per controllo terminazione, mantiene il valore della cella ram '0000' incrementato di 1

#### 2.1.3 Segnali per componente interno

fU : flusso di bit in lettura da codificare  
fY : flusso di 2 bit in uscita da codificatore  
stop.en : segnale per fermare la macchina a stati del codificatore al di fuori dei clock di codifica

### 2.1.4 FSM

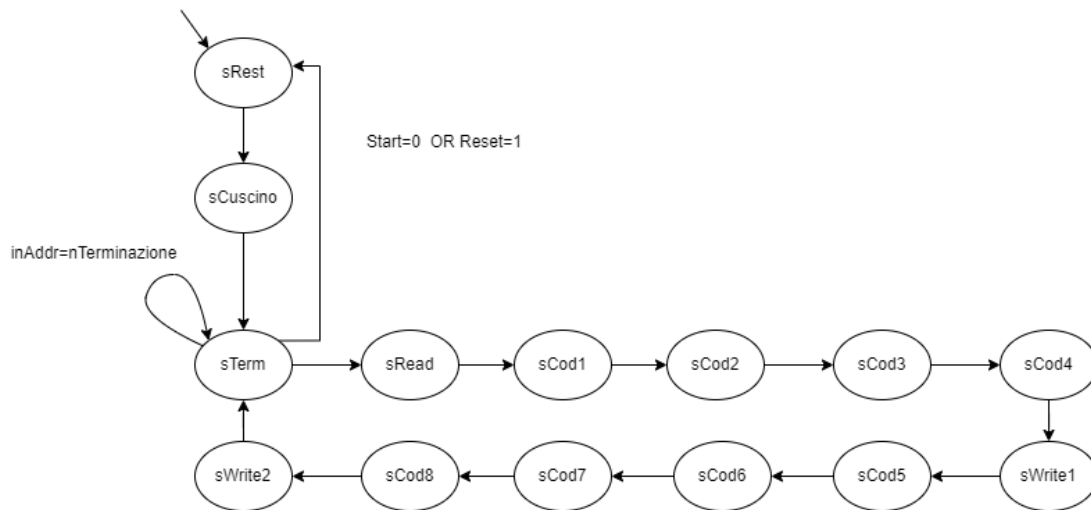


Figura 1: Disegno esplicativo della FSM

#### Descrizione stati:

sReStart : stato di reset, viene letto la cella all'indirizzo '0000' e il suo valore incrementato di uno viene salvato in nTerminazione

sCuscino : stato attraversato solo una volta in tutta l'operazione di codifica, serve per consentire funzionamento con n di parole nullo ( per ulteriori info vedere test seq\_min)

sTerm : controllo terminazione codifica, confronta il numero totale di parole da leggere (+ 1) con il prossim indirizzo di lettura

sRead : stato di lettura, fornisce alla ram i segnali per leggere la prossima parola da elaborare

sCod1 : codificatore in funzionamento, viene salvata la parola appena letta da i\_data nel registro in\_value, viene inserito in fU il primo bit dalla parola letta

sCod2 to sCod4 : inserito in fU il nuovo bit da leggere preso da in\_value, bit codificati da fY salvati in out\_value\_buffer nell'apposita posizione

sWrite1 : inseriti bit da fY negli ultimi due bit del registro

out\_value\_buffer, forniti segnali alla ram per scrivere la parola appena codificata, codificatore bloccato

sCod5 to sCod8 : codificatore in funzionamento, funzionamento equivalente a sCod1-sCod4, usando gli ultimi 4 bit di in\_value

sWrite2 : equivalente a sWrite1

## 2.2 Modulo 2 : Codificatore Convolutionale

Il codificatore convoluzionario è il modulo che si occupa dell'effettiva codifica dei dati in ingresso

### 2.2.1 Segnali

i\_U : flusso di bit in ingresso

i\_start : segnale di enable del componente, se off il componente viene resettato

i\_rst : segnale di reset asincrono

stop\_en : segnale di enable del componente, se off viene mantenuto in standby

o\_Y : flusso di 2 bit da codifica in uscita

### 2.2.2 Registri

st\_att, s\_pros : registri per stati della fsm di mealy

### 2.2.3 FSM

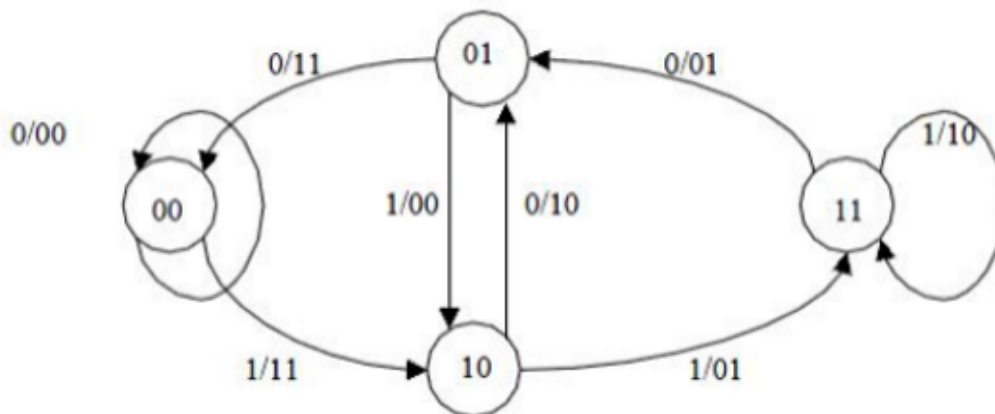


Figura 2: Disegno esplicativo del codificatore convoluzionario

## 3 Risultati sperimentali

### 3.1 Sintesi

(report di sintesi)

### 3.2 Simulazioni

Le simulazioni che abbiamo creato servono a testare i casi limite che potrebbero mandare in loop, blocco o crash la macchina a stati da noi sviluppati

#### 3.2.1 Simulazione max

test bench 1 (cosa fa e perchè lo fa e cosa verifica; per esempio, controlla una condizione limite)

#### 3.2.2 Simulazione min

test bench 2 (cosa fa e perchè lo fa e cosa verifica; per esempio, controlla una condizione limite)

### **3.2.3 Simulazione reset multipli**

### **3.2.4 Simulazione start multipli**

## **4 Conclusione**

leggibilità over velocità

tutto facilmente modificabile e leggibile

### **4.1 note particolari modulo 1**

La fsm usa molti stati equivalenti, scelta adottata per favorire la comprensibilità usando un approccio simile agli automi a stati finiti, evitando quindi il più possibile dei controlli per la scelta del prossimo stato, tranne ovviamente in sTerm, dove viene verificata la terminazione della codifica.

### **4.2 note particolari modulo 2**

La codifica avviene tramite questo componente, pensato per rispecchiare il più possibile quello descritto dalla specifica e per essere facilmente utilizzabile in contesti diversi senza bisogno del Modulo 1. Questo tipo di codificatori sono usati nelle telecomunicazioni, quindi potrebbe essere utilizzato collegando l'uscita ad un componente per trasmettere direttamente il segnale codificato invece di salvarlo su una memoria.



## 4.7 How to write Mathematics

L<sup>A</sup>T<sub>E</sub>X is great at typesetting mathematics. Let  $X_1, X_2, \dots, X_n$  be a sequence of independent and identically distributed random variables with  $E[X_i] = \mu$  and  $\text{Var}[X_i] = \sigma^2 < \infty$ , and let

$$S_n = \frac{X_1 + X_2 + \dots + X_n}{n} = \frac{1}{n} \sum_i^n X_i$$

denote their mean. Then as  $n$  approaches infinity, the random variables  $\sqrt{n}(S_n - \mu)$  converge in distribution to a normal  $\mathcal{N}(0, \sigma^2)$ .

## 4.8 How to change the margins and paper size

Usually the template you're using will have the page margins and paper size set correctly for that use-case. For example, if you're using a journal article template provided by the journal publisher, that template will be formatted according to their requirements. In these cases, it's best not to alter the margins directly.

If however you're using a more general template, such as this one, and would like to alter the margins, a common way to do so is via the `geometry` package. You can find the `geometry` package loaded in the preamble at the top of this example file, and if you'd like to learn more about how to adjust the settings, please visit this help article on [page size and margins](#).

## 4.9 How to change the document language and spell check settings

Overleaf supports many different languages, including multiple different languages within one document.

To configure the document language, simply edit the option provided to the `babel` package in the preamble at the top of this example project. To learn more about the different options, please visit this help article on [international language support](#).

To change the spell check language, simply open the Overleaf menu at the top left of the editor window, scroll down to the spell check setting, and adjust accordingly.

## 4.10 How to add Citations and a References List

You can simply upload a `.bib` file containing your BibTeX entries, created with a tool such as JabRef. You can then cite entries from it, like this: [\[Gre93\]](#). Just remember to specify a bibliography style, as well as the filename of the `.bib`. You can find a [video tutorial here](#) to learn more about BibTeX.

If you have an [upgraded account](#), you can also import your Mendeley or Zotero library directly as a `.bib` file, via the upload menu in the file-tree.

## 4.11 Good luck!

We hope you find Overleaf useful, and do take a look at our [help library](#) for more tutorials and user guides! Please also let us know if you have any feedback using the Contact Us link at the bottom of the Overleaf menu — or use the contact form at <https://www.overleaf.com/contact>.

## Riferimenti bibliografici

[Gre93] George D. Greenwade. The Comprehensive Tex Archive Network (CTAN). *TUGBoat*, 14(3):342–351, 1993.