# Prova Finale Reti Logiche

A.A. 2022/2023

Marco Laurenzi
959690
10764667
Prof. Fabio Salice

POLITECNICO
MILANO 1863

# Introduzione

Il progetto prevede la realizzazione di un componente hardware capace di interagire con una memoria esterna. Il componente, visto più ad alto livello, realizza, in maniera semplificata, un sistema di puntatori in cui è possibile fornire in ingresso alla memoria un indirizzo e ricevere da essa il valore corrispondente a tale indirizzo.

Il componente prevede 7 canali primari, 2 di ingresso e 5 di uscita. Nel primo dei due canali di ingresso, I_W, è possibile ricevere in maniera seriale l'input, composto da un minimo di 2 bit ad un massimo di 18 bit, contenenti in ordine il canale di uscita selezionato e l'indirizzo di memoria richiesto; il secondo, I_START, permette di individuare le sequenze di ingresso valide, ovvero le sequenze che effettivamente rappresentano un'interrogazione della memoria.

Per quanto riguarda invece i 5 canali di uscita primari: sui primi 4, O_Z0, O_Z1, O_Z2 e O_Z3, si possono visualizzare, in parallelo, i valori contenuti nella memoria mentre sul quinto, O_DONE, è possibile, quando questo vale 1, leggere le uscite che rimangono invece mascherate quando questo vale 0.

Il componente prevede altri due canali aggiuntivi unici: I_RST e I_CLK.

# Premessa

- Ogni indirizzo di memoria è composto da 16 bit, i quali possono provenire tutti dal canale di ingresso, indicato con W, oppure ottenuti estendendo gli N bit ricevuti in ingresso con degli zeri più significativi come mostrato in figura 1.
- I quattro canali di uscita sono canali da 8 bit, sui quali verranno letti i valori in corrispondenza degli indirizzi della memoria.

- I canali di uscita, essendo 4 vengono codificati con due cifre binarie nella seguente maniera: 00 indica il canale Z0, 01 il canale Z1, 10 il canale Z2 e 11 il canale Z3.
- I restanti canali, W, START, CLK, RESET, DONE sono tutti canali da un unico bit.

```
(N = 7)    1010111          -> 0000000001010111
(N = 16)   1110000001010111 -> 1110000001010111
(N = 0)    0000000000000000 -> 0000000000000000
```

Figura 1: estensione indirizzi di memoria

# Ipotesi Progettuali

- Tutti i bit su W devono essere letti sul fronte di salita del clock.
- Il segnale di START è garantito rimanere alto per almeno 2 cicli di clock e al massimo per 18 cicli di clock, durante tutta la durata della sequenza di ingresso valida.
- I valori sulle uscite sono visibili solo quando il segnale DONE è 1.
- Le uscite sono capaci di memorizzare i valori presenti anche dopo essere state oscurate dal segnale DONE=0 e fintanto che non si presenti il segnale di RESET=1.

# Architettura

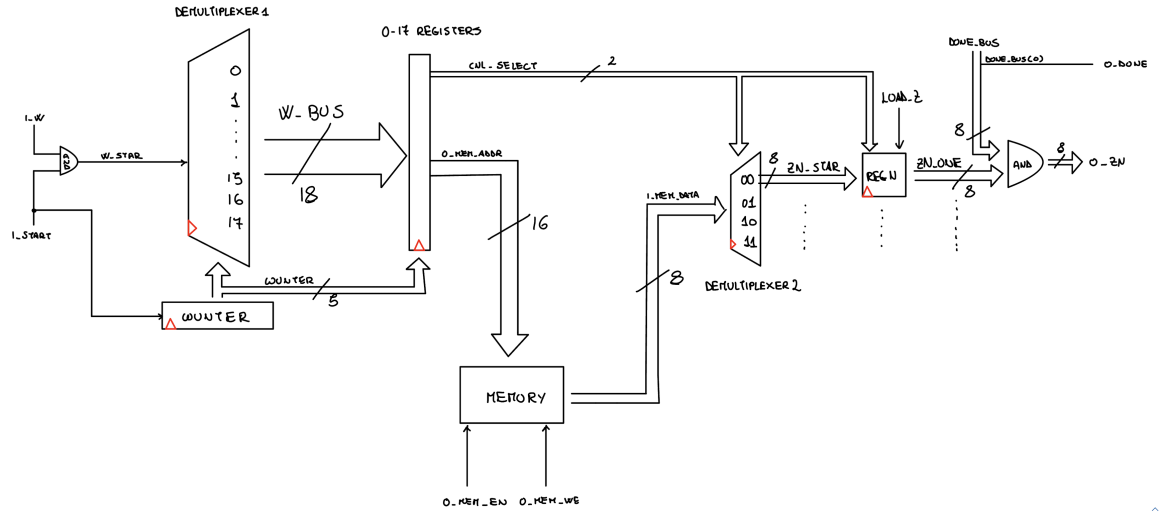


Figura 2: rappresentazione semplificata del componente hardware

Per alleggerire la rappresentazione sono stati omessi alcuni segnali come I_CLK, da considerarsi collegato a tutte le componenti marchiate con il triangolo rosso in figura 2.
Per ottenere una maggiore leggibilità solo il primo canale di uscita è stato rappresentato interamente, essendo gli altri tre analoghi con i rispettivi segnali. Da notare invece come il segnale LOAD_Z sia lo stesso per tutti e 4 i canali.

## Descrizione ad alto livello

L'architettura di per sé si basa su elementi combinatori molto semplici, abbiamo infatti registri di vario tipo, un contatore, demultiplexer e porte logiche elementari.
L'idea dietro questa realizzazione circuitale è quella di parallelizzare il segnale in ingresso memorizzandolo su un bus. Il segnale viene poi inoltrato alla memoria dalla quale viene successivamente letto il valore corrispondente a tale indirizzo nel momento opportuno e inoltrato sui canali di uscita.
Nello specifico:

1. Il segnale di ingresso I_W viene messo in AND con il segnale I_START in maniera coerente con le ipotesi progettuali precedentemente elencate.

2. Il DEMULTPLEXER1, comandato dal COUNTER, propaga in uscita su W_BUS gli N ($2 <= N <= 18$) bit ricevuti fintanto che I_START = 1.
3. Il registro indicato con 0-17 REGISTERS è in realtà un semplice registro a due bit per i bit 0-TO-2 di W_BUS che si riflettono nei bit 1-DOWNTO-0 di CNL_SELECT in cui verrà memorizzato il canale di uscita scelto. Per quanto riguarda i bit 2-17 di W_BUS, si tratta di uno shift-register che in uscita si riflette nei bit 0-TO-15 di O_MEM_ADDR; riceve ad ogni ciclo di clock il nuovo bit in posizione 0 e sposta i bit dalla posizione 2-TO-(k+1) di W_BUS alla posizione k-DOWNTO-1 di O_MEM_ADDR [k=2, 3, 4, …, 15].
4. O_MEM_ADDR contiene l'indirizzo di memoria opportunamente salvato e, tramite due segnali da un bit, O_MEM_EN e O_MEM_WE, si interagisce con la memoria e si legge sul segnale I_MEM_DATA il valore presente in corrispondenza di tale indirizzo nel ciclo successivo a quello in cui O_MEM_EN viene alzato per garantire la lettura corretta. In particolare O_MEM_WE non viene mai utilizzato e dunque tenuto basso, non dovendo mai scrivere in memoria ma solo leggere da essa.
5. Il valore di 8 bit presente in I_MEM_DATA viene, tramite il DEMULTIPLEXER2 e il segnale da 2 bit CNL_SELECT, memorizzato in un registro chiamato REGN corrispondente al canale (N) in questione.
6. Il valore presente nel REGN viene poi, tramite l'opportuna sincronizzazione dei segnali CNL_SELECT, I_CLK e LOAD_Z propagato sull'uscita corrispondente nel momento opportuno.
7. Tutte le uscite principali, O_Z1, O_Z2, O_Z3 e O_Z4, sono in AND bit-a-bit con un bus di 8 bit chiamato DONE_BUS che non è altro che l'estensione ad 8 bit di O_DONE (O_DONE = 0, DONE_BUS = 00000000; O_DONE = 1, DONE_BUS = 11111111).

## Segnali interni utilizzati

- W_STAR: rappresenta l'input ricevuto da I_W in AND con il segnale I_START
- COUNTER: un contatore a 5 bit che permette di parallelizzare i bit in ingresso su un vettore a 18 bit

- W_BUS: memorizza i 18 bit correttamente estesi se necessario
- CNL_SELECT: il canale di uscita selezionato
- ZN_STAR: segnali in uscita al DEMULTIPLEXER2, indirizzano l'uscita verso il registro corretto
- ZN_ONE: i valori in uscita al componente prima di essere mascherati
- LOAD_Z: il segnale singolo che comanda la propagazione da ZN_STAR a ZN_ONE
- DONE_BUS: bus di 8 bit utilizzato per mascherare le uscite
- CURR_STATE: lo stato corrente della FSM
- NEXT_STATE: lo stato prossimo della FSM

# Logica Combinatoria

La logica combinatoria è gestita dalla macchina di Moore descritta di seguito:

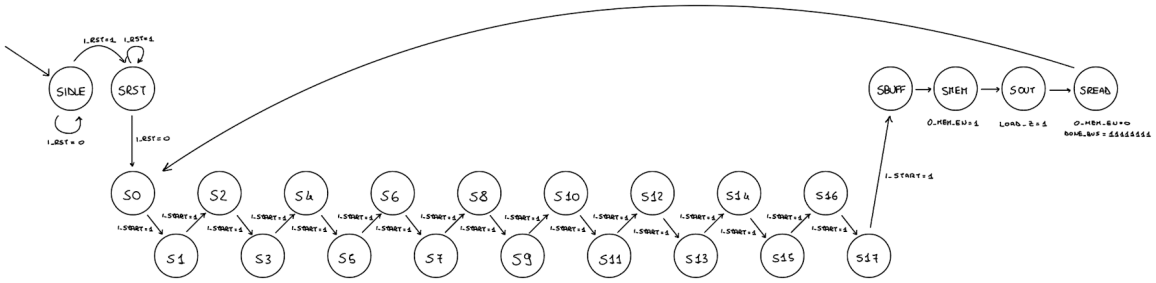


Figura 3: macchina di Moore

Per comodità di rappresentazione sono stati omesse alcune transizioni descritte nel codice e di seguito:

- **S0**: se I_RST=1 NEXT_STATE=SRST; se I_START=0 NEXT_STATE=S0, else NEXT_STATE=SMEM.
- **SN** (N=1, 2, …, 17): se I_RST=1 NEXT_STATE=SRST; else NEXT_STATE=SMEM.

### Descrizione singoli stati

- **SIDLE:** questo stato è lo stato in cui mi trovo inizialmente, prima ancora di ricevere il segnale I_RESET per la prima volta.
- **SRST:** lo stato in cui vado, e rimango, finché I_RST=1. E' lo stato in cui il componente circuitali resetta le sue uscite.
- **SN:** [N=0, 1, …, 17] 18 stati che rappresentano i 18 bit massimi ricevuti in ingresso.
- **SBUFF:** uno stato aggiuntivo utilizzato nel caso in cui l'ingresso sia esattamente di 18 bit per finire di caricare correttamente l'ultimo bit.
- **SMEM:** lo stato in cui viene attivata la memoria tramite il segnale O_MEM_EN.
- **SOUT:** lo stato in cui carico il valore uscito dalla memoria al ciclo precedente nel bus ZN_ONE tramite il segnale LOAD_Z.
- **SREAD:** lo stato in cui permetto la lettura dei 4 canali di uscita, l'unico stato in cui O_DONE sarà alto.

### Note

Sebbene l'utilizzo di 18 stati diversi corrispondenti ai 18 bit dell'input possa sembrare una complicazione del circuito, la scelta di optare per questa soluzione e non quella di usare un contatore è stata fatta ai fini della leggibilità delle forme d'onda, in quanto leggere uno stato chiamato SN risulta molto più chiaro rispetto alla lettura e decodifica di un contatore.

# Risultati Sperimentali

**Synthesis tool:** XILINX VIVADO WEBPACK

**FPGA target:** Artix-7 FPGA xc7a200tfbg484-1

**Behavioral Simulation**

| Test | Time | Description |
|---|---|---|
| tb_1 | 33700 ns | 352 bit input |
| tb_2 | 32400 ns | 339 bit input |
| tb_3 | 57900 ns | 594 bit input |
| tb_4 | 12500 ns | 140 bit input |
| tb_5 | 62300 ns | 638 bit input |
| tb_6 | 3300 ns | 48 bit input |
| tb_7 | 1700 ns | 32 bit input |

**Post-Synthesis Simulation**

| Test | Time |
|---|---|
| tb_1 | 33700100 ps |
| tb_2 | 32400100 ps |
| tb_3 | 57900100 ps |
| tb_4 | 12500100 ps |
| tb_5 | 62300100 ps |
| tb_6 | 3300100 ps |
| tb_7 | 1700100 ps |

**Slack (MET) :** 5.941ns

**Data Path Delay:** 3.677ns

logic 1.025ns (27.876%)

route 2.652ns (72.124%)

Oltre ai test bench elencati in tabella sono state testate anche altre funzionalità:

- Vi è completa uguaglianza fra il ricevere in input 2 soli bit validi rappresentanti il canale di uscita (I_W = 110, I_START = 110) e ricevere 3 bit validi indicanti l'indirizzo di memoria 0 (I_W = 1100, I_START = 1110). Questo test garantisce come i bus O_MEM_ADDR e W_BUS vengano correttamente inizializzati. [figura 7]
- Il segnale O_DONE, come da specifica, rimane alto per un unico ciclo di clock per ogni sequenza di input valido. Questo test è stato eseguito implementando un contatore la cui uscita viene portata a 0 se il segnale di I_RST=1 mentre ogni volta che I_START=1 viene incrementato il contatore di 1 se O_DONE=1. Così facendo si tiene conto di quante volte per ogni sequenza di input valida il segnale O_DONE sia stato alto. Questa funzionalità è stata abbondantemente testata sui test forniti e su numerosi altri test creati appositamente.
- Il caso limite in cui il segnale di I_START diventi alto nel momento in cui il segnale di I_RST diventi 0. Come si vede in figura 8 questo non crea alcun problema al componente.
- Il caso limite in cui l'input sia di dimensione massima, corrispondente a 18 bit, è stato abbondantemente e correttamente testato nel tb_2 come si può notare in figura 4.
- Il segnale di RESET porta nello stato SRST in qualunque momento arrivi, indipendentemente dal fatto che si stia leggendo un input valido (START=1). Questa funzionalità sebbene non rappresenti un caso critico è stata testata per completezza. [figura 6]

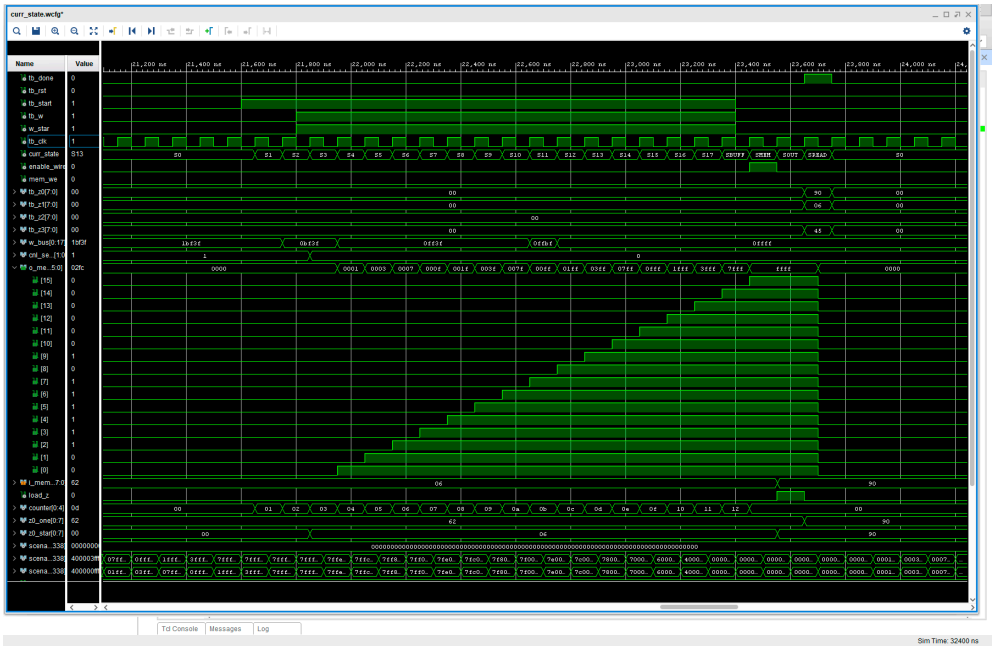

Figura 4: forme d'onda tb_2

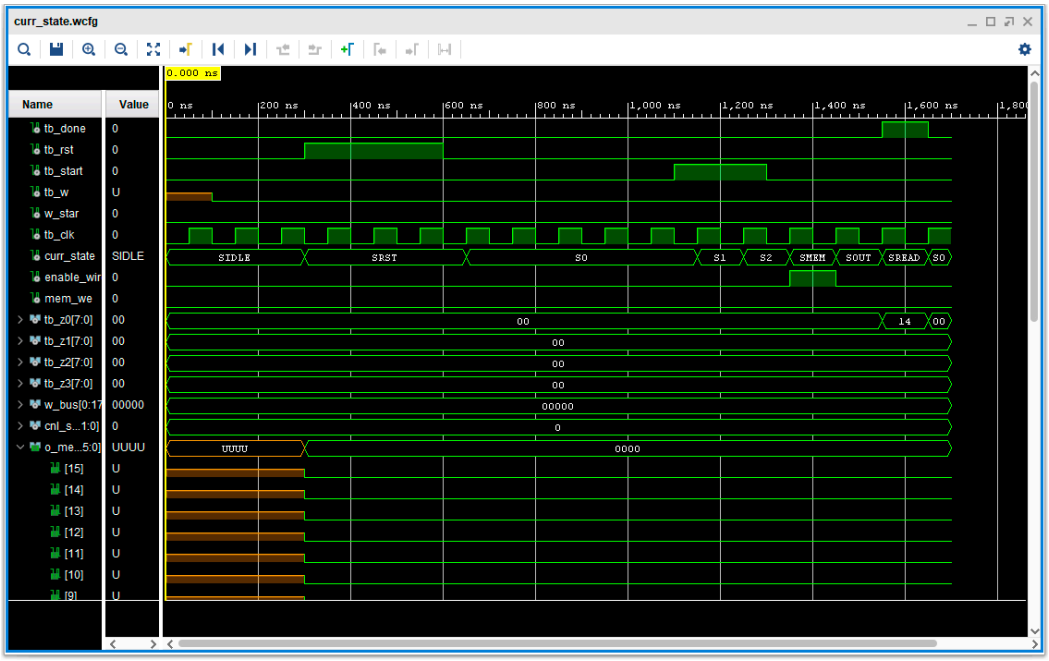


Figura 5: forme d'onda tb_7

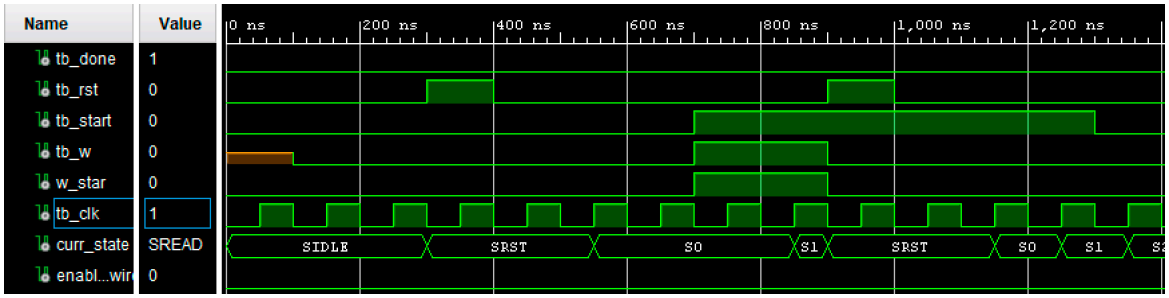


Figura 6

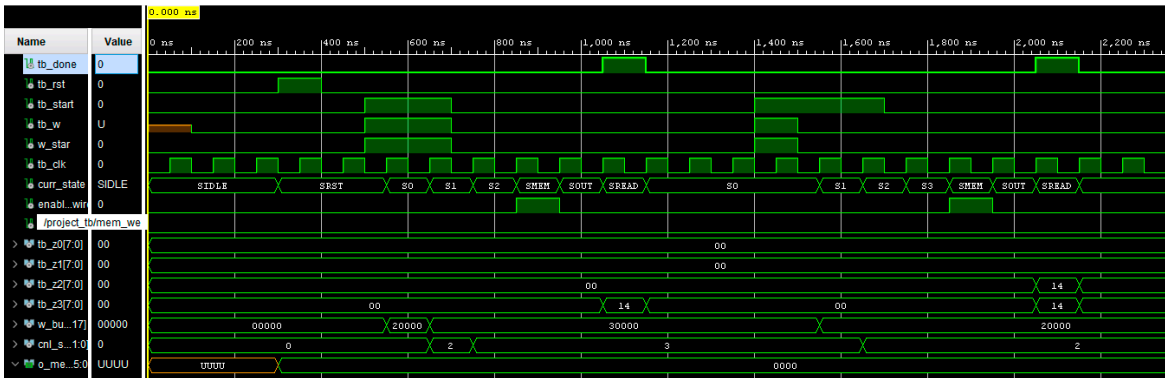


Figura 7

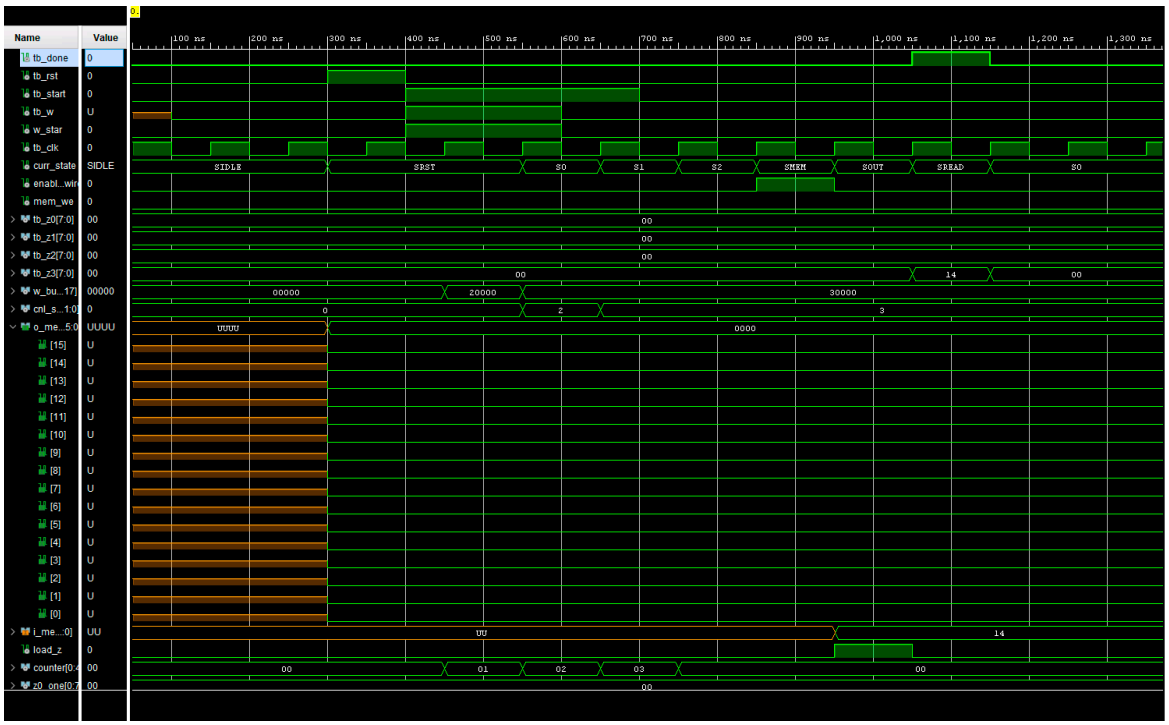


Figura 8

# Utilization Report

| Site Time | Used | Fixed | Available | %Util |
|---|---|---|---|---|
| Slide LUTs | 141 | 0 | 134600 | 0.10 |
| as Logic | 141 | 0 | 134600 | 0.10 |
| as Memory | 0 | 0 | 46200 | 0.00 |
| Slice Registers | 96 | 0 | 269200 | 0.04 |
| as FF | 96 | 0 | 269200 | 0.04 |
| as Latch | 0 | 0 | 269200 | 0.00 |
| F7 Muxes | 0 | 0 | 67300 | 0.00 |
| F8 Muxes | 0 | 0 | 33650 | 0.00 |

# Conclusioni

Sul componente qui descritto sono state, durante il corso della progettazione, effettuate alcune ottimizzazioni. Inizialmente il modulo era stato pensato con 4 segnali di LOAD, uno per ognuno dei canali di uscita ZN, ma, in seguito, per cercare di riutilizzare i segnali già istanziati e per evitare di complicare ulteriormente il componente si è deciso di riutilizzare il bus a 2 bit contenente il canale di uscita CNL_SELECT, per poter incanalare sin da subito il valore proveniente dalla memoria nel giusto canale di uscita e, al momento della sua scrittura in uscita, il registro REGN propaga l'ingresso in uscita solamente se CNL_SELECT=N (e.g. se CNL_SELECT = 10 il registro REG2 sarà l'unico a propagare l'ingresso in uscita e sarà anche l'unico ad aver ricevuto dal MULTIPLEXER2 il valore corretto in ingresso).

In conclusione il componente Hardware progettato svolge correttamente, rispettando i constraints imposti e la specifica fornita, nonché supera correttamente tutti i test forniti ed alcuni test realizzati appositamente per alcune funzioni specifiche.