

# Progetto finale di Reti Logiche

Prof. Fornaciari, Prof. Palermo e Prof. Salice  
Anno Accademico 2022 - 2023

(AGGIORNATO AL 15 Dicembre 2022)

## Descrizione generale

La specifica della “*Prova Finale (Progetto di Reti Logiche)*” per l’Anno Accademico 2022/2023 chiede di implementare un modulo HW (descritto in VHDL) che si interfacci con una memoria e che rispetti le indicazioni riportate nella seguente specifica.

Ad elevato livello di astrazione, il sistema riceve indicazioni circa una locazione di memoria, il cui contenuto deve essere indirizzato verso un canale di uscita fra i quattro disponibili.

Le indicazioni circa il canale da utilizzare e l’indirizzo di memoria a cui accedere vengono forniti mediante un ingresso seriale da un bit, mentre le uscite del sistema, ovvero i succitati canali, forniscono tutti i bit della parola di memoria in parallelo.

## Interfacce

Il modulo da implementare ha **due ingressi primari da 1 bit (W e START)** e **5 uscite primarie**. Le uscite sono le seguenti: quattro da 8 bit (**Z0, Z1, Z2, Z3**) e una da 1 bit (**DONE**). Inoltre, il modulo ha un segnale di clock **CLK**, unico per tutto il sistema e un segnale di reset **RESET** anch’esso unico.

## Funzionamento

All’istante iniziale, quello relativo al **reset** del sistema, le uscite hanno i seguenti valori:

Z0, Z1, Z2 e Z3 sono 0000 0000, DONE è 0.

I dati in ingresso, ottenuti come sequenze sull'**ingresso primario seriale W** lette sul fronte di salita del clock, sono organizzati nel seguente modo:

- 2 bit di intestazione (i primi della sequenza) seguiti da
- N bit di indirizzo della memoria.

Gli N bit permettono di costruire un indirizzo di memoria (si legga qui di seguito la specifica per questi N bit). All’indirizzo di memoria è memorizzato il **messaggio da 8 bit** che deve essere indirizzato verso un canale di **uscita**.

I due bit di **intestazione** identificano il **canale d’uscita** (Z0, Z1, Z2 o Z3) sul quale deve essere indirizzato il messaggio. Il primo bit è il bit più significativo del canale di uscita, il secondo quello meno significativo, più in dettaglio:

00 identifica Z0, 01 identifica Z1, 10 identifica Z2 e, infine, 11 identifica Z3.

Gli **N** bit di indirizzo possono variare da 0 fino ad un **massimo di 16 bit**. Gli indirizzi di memoria sono tutti di 16 bit.

Se il numero di bit di N è inferiore a 16, l'indirizzo viene **esteso con 0 sui bit più significativi**. Ad esempio:

(N = 7)	1010111	-> 000000001010111
(N = 16)	1110000001010111	-> 1110000001010111
(N = 0)	0000000000000000	-> 0000000000000000

Tutti i bit su W devono essere letti sul fronte di salita del clock.

La sequenza di ingresso è valida quando il segnale START è alto (=1) e termina quando il segnale START è basso (=0).

Il segnale START rimane alto per almeno di 2 cicli di clock e non più di 18 cicli di clock (2 bit del canale e 16 bit per il massimo numero di bit per indirizzare la memoria). Si assume questa condizione sempre verificata (non è necessario gestire il caso in cui il segnale di START rimanga attivo meno di 2 cicli di clock o più di 18).

Le uscite Z0, Z1, Z2 e Z3 sono inizialmente 0. I valori rimangano inalterati eccetto il canale sul quale viene mandato il messaggio letto in memoria; i valori sono visibili solo quando il valore di DONE è 1.

Quando il segnale DONE è 0 tutti i canali Z0, Z1, Z2 e Z3 devono essere a zero (32 bit a 0). Contemporaneamente alla scrittura del messaggio sul canale, il segnale DONE passa da 0 passa a 1 e rimane attivo per un solo ciclo di clock (dopo 1 ciclo di clock DONE passa da 1 a 0). In pratica quando DONE=1 il canale associato al messaggio cambierà il suo valore, mentre gli altri canali mostreranno l'ultimo valore trasmesso derivato dai messaggi ad essi associati.

Il segnale START è garantito rimanere a 0 fino a che il segnale DONE non è tornato a 0.

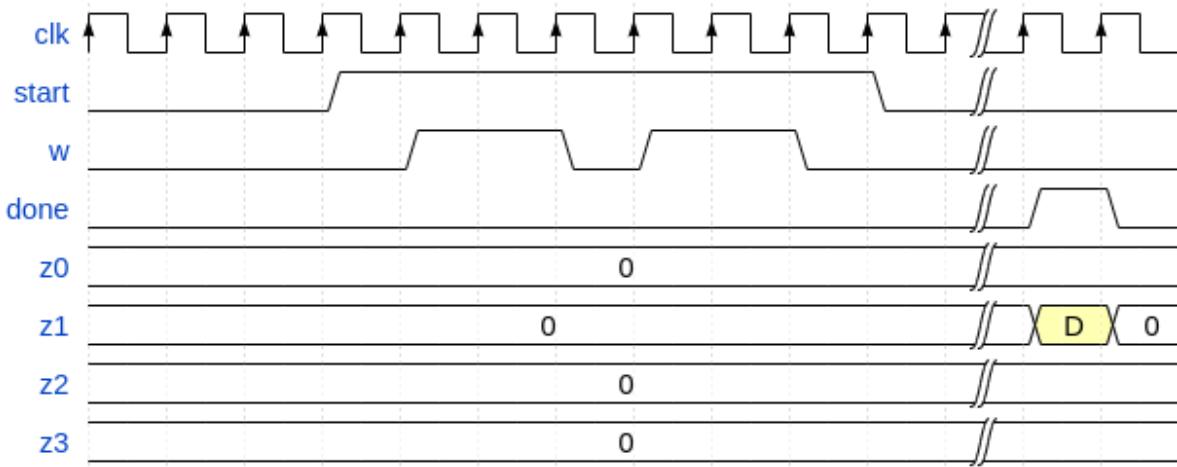
Il tempo massimo per produrre il risultato (ovvero il tempo trascorso tra START=0 e DONE=1) deve essere inferiore a 20 cicli di clock.

Il modulo deve essere progettato considerando che prima del primo START=1 (e prima di richiedere il corretto funzionamento del modulo) verrà **sempre** dato il RESET (RESET=1). Una seconda (o successiva) elaborazione con START=1 non dovrà invece attendere il reset del modulo. Ogni qual volta viene dato il segnale di RESET (RESET=1), il modulo viene re-inizializzato.

### Esempio (diagrammi temporali)

Lettura di un dato "D" dall'indirizzo di memoria 0000000000010110.

La scrittura del dato "D" avviene sull'uscita specificata Z1 (bit di intestazione "01")



### Interfaccia del Componente

Il componente da descrivere deve avere la seguente interfaccia.

```
entity project_reti_logiche is
    port (
        i_clk      : in std_logic;
        i_RST      : in std_logic;
        i_start    : in std_logic;
        i_w        : in std_logic;

        o_z0       : out std_logic_vector(7 downto 0);
        o_z1       : out std_logic_vector(7 downto 0);
        o_z2       : out std_logic_vector(7 downto 0);
        o_z3       : out std_logic_vector(7 downto 0);
        o_done     : out std_logic;

        o_mem_addr : out std_logic_vector(15 downto 0);
        i_mem_data : in std_logic_vector(7 downto 0);
        o_mem_we   : out std_logic;
        o_mem_en   : out std_logic
    );
end project_reti_logiche;
```

In particolare:

- il nome del modulo **deve essere** `project_reti_logiche` e deve essere presente **una sola architettura** per ogni entità; la violazione di queste indicazioni comporta l'impossibilità di eseguire il Test Bench e una conseguente valutazione di zero;
- `i_clk` è il segnale di CLOCK in ingresso generato dal Test Bench;
- `i_RST` è il segnale di RESET che inizializza la macchina pronta per ricevere il primo segnale di START;

- i\_start è il segnale di START generato dal Test Bench;
- i\_w è il segnale W precedentemente descritto e generato dal Test Bench;
- o\_z0, o\_z1, o\_z2, o\_z3 sono i quattro canali di uscita;
- o\_done è il segnale di uscita che comunica la fine dell'elaborazione;
- o\_mem\_addr è il segnale (vettore) di uscita che manda l'indirizzo alla memoria;
- i\_mem\_data è il segnale (vettore) che arriva dalla memoria in seguito ad una richiesta di lettura;
- o\_mem\_en è il segnale di ENABLE da dover mandare alla memoria per poter comunicare (sia in lettura che in scrittura);
- o\_mem\_we è il segnale di WRITE ENABLE da dover mandare alla memoria (=1) per poter scriverci. Per leggere da memoria esso deve essere 0.

## APPENDICE: Descrizione Memoria

**NOTA:** La memoria è già istanziata all'interno del Test Bench e non va sintetizzata

La memoria e il suo protocollo può essere estratto dalla seguente descrizione VHDL che fa parte del test bench e che è derivata dalla User guide di VIVADO disponibile al seguente link:

[https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2017\\_3/ug901-vivado-synthesis.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_3/ug901-vivado-synthesis.pdf)

```
-- Single-Port Block RAM Write-First Mode (recommended template)
--
-- File: rams_02.vhd
--

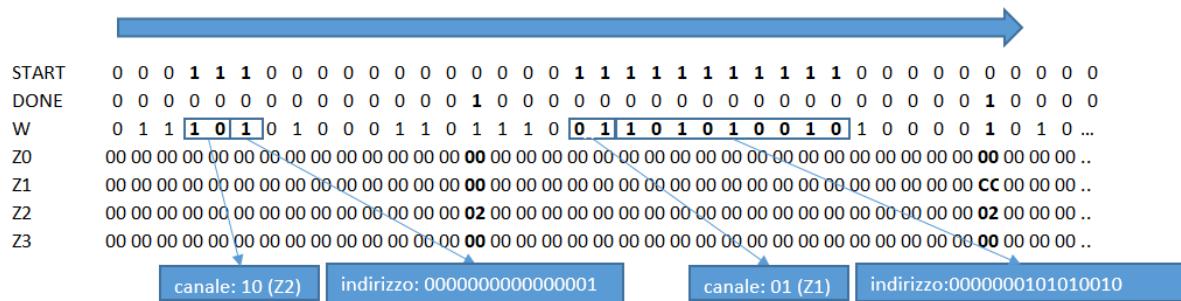
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity rams_sp_wf is
port(
    clk  : in std_logic;
    we   : in std_logic;
    en   : in std_logic;
    addr : in std_logic_vector(15 downto 0);
    di   : in std_logic_vector(7 downto 0);
    do   : out std_logic_vector(7 downto 0)
);
end rams_sp_wf;

architecture syn of rams_sp_wf is
type ram_type is array (65535 downto 0) of std_logic_vector(7 downto 0);
signal RAM : ram_type;
begin
process(clk)
begin
if clk'event and clk = '1' then
    if en = '1' then
        if we = '1' then
            RAM(conv_integer(addr)) <= di;
            do                      <= di after 2 ns;
        else
            do <= RAM(conv_integer(addr)) after 2 ns;
        end if;
    end if;
end if;
end process;
end syn;
```

## ESEMPI

Gli esempi qui di seguito hanno lo scopo di esemplificare la relazione tra la stringa di ingresso utile (indicata come W utile), cioè durante il periodo nel quale START assume valore 1, il dato (colonna DATO) presente nella locazione di memoria richiesta (colonna INDIRIZZO) e la configurazione assunta dalle uscite nel solo momento utile, cioè durante il periodo nel quale DONE assume valore 1.

## Esempio 1



## Esempio 2

		RESET	00	00	00	00	W utile
CANALE	INDIRIZZO	DATO	Z0	Z1	Z2	Z3	
03	EA60	04	00	00	00	04	000001100101011111
03	EBC3	59	00	00	00	59	110000111101011111
03	EBC9	66	00	00	00	66	100100111101011111
02	ECEB	EE	00	00	EE	66	110101110011011101
02	F063	2E	00	00	2E	66	110001100000111101
03	F391	94	00	00	2E	94	10001001100111111
03	F700	3A	00	00	2E	3A	00000000110111111
01	FAA0	52	00	52	2E	3A	00000101010111110
03	FE1D	14	00	52	2E	14	101110000111111111
02	FB78	F4	00	52	F4	14	00011101101111101
01	FD5F	43	00	43	F4	14	11110101011111110
02	FEA1	1F	00	43	1F	14	100001010111111101
03	FEB7	E6	00	43	1F	E6	111011010111111111
03	FFA2	86	00	43	1F	86	010001011111111111
03	FC4F	E4	00	43	1F	E4	111100100011111111
02	FEBC	31	00	43	31	E4	001111010111111101
...	...	...	...	...	...	...	...

### Esempio 3

			RESET	00	00	00	00	
CANALE	INDIRIZZO	DATO	Z0	Z1	Z2	Z3		W utile
00	03E8	16	16	00	00	00		000101111100
03	2176	DA	16	00	00	DA		0110111010000111
02	3873	EB	16	00	EB	DA		1100111000011101
02	495D	E3	16	00	E3	DA		10111010100100101
01	6CDA	89	16	89	E3	DA		01011011001101110
03	8C28	24	16	89	E3	24		000101000011000111
02	9A3F	4D	16	89	4D	24		11111000101100101
00	B25C	4B	4B	89	4D	24		001110100100110100
00	CBAD	BB	BB	89	4D	24		101101011101001100
03	D202	5D	BB	89	4D	5D		010000000100101111
00	D322	73	73	89	4D	5D		010001001100101100
03	E536	DF	73	89	4D	DF		011011001010011111
03	F680	C7	73	89	4D	C7		000000010110111111
01	EB9E	C9	73	C9	4D	C7		011110011101011110
01	C98F	81	73	81	4D	C7		111100011001001110
00	CF7A	0B	0B	81	4D	C7		010111101111001100
...	...	...	...	...	...	...	...	...

#### Esempio 4

		RESET	00	00	00	00	
CANALE	INDIRIZZO	DATO	Z0	Z1	Z2	Z3	W utile
03	03E8	29	00	00	00	29	000101111111
01	A2E5	1D	00	1D	00	29	101001110100010110
00	5CA4	39	39	1D	00	29	00100101001110100
03	8EED	1A	39	1D	00	1A	101101110111000111
02	6616	15	39	1D	15	1A	01101000011001101
01	F177	07	39	07	15	1A	111011101000111110
03	5680	27	39	07	15	27	00000001011010111
02	7E09	E9	39	07	E9	27	10010000011111101
02	1585	7C	39	07	7C	27	101000011010101
03	CC42	05	39	07	7C	05	010000100011001111
02	3ACD	8F	39	07	8F	05	1011001101011101
02	170F	BC	39	07	BC	05	111100001110101
00	B73E	CA	CA	07	BC	05	01111001110110100
03	3B89	FE	CA	07	BC	FE	1001000111011111
02	09FD	FB	CA	07	FB	FE	1011111100101
01	4630	82	CA	82	FB	FE	00001100011000110
...	...	...	...	...	...	...	...

### Esempio 5

			RESET	00	00	00	00	
CANALE	INDIRIZZO	DATO	Z0	Z1	Z2	Z3		W utile
02	03E8	D6	00	00	D6	00		000101111101
01	2FCD	A6	00	A6	D6	00		1011001111110110
01	3E1C	1C	00	1C	D6	00		0011100001111110
01	891B	DE	00	DE	D6	00		110110001001000110
00	ACD7	14	14	DE	D6	00		111010110011010100
02	BEBD	3B	14	DE	3B	00		101111010111110101
01	020E	AB	14	AB	3B	00		011100000110
00	20BB	43	43	AB	3B	00		1101110100000100
02	4E5F	FF	43	AB	FF	00		11111010011100101
01	7B5F	DA	43	DA	FF	00		11111010110111110
03	9CCA	15	43	DA	FF	15		010100110011100111
00	D717	DA	DA	DA	FF	15		111010001110101100
01	0633	7E	DA	7E	FF	15		1100110001110
02	29C1	C2	DA	7E	C2	15		1000001110010101
02	591B	B9	DA	7E	B9	15		11011000100110101
03	89CB	EB	DA	7E	B9	EB		110100111001000111
...	...	...	...	...	...	...	...	...

## Esempio 6

		RESET	00	00	00	00	W utile
CANALE	INDIRIZZO	DATO	Z0	Z1	Z2	Z3	
01	000A	20	00	20	00	00	010110
02	000F	07	00	20	07	00	111101
03	0012	1D	00	20	07	1D	0100111
00	0015	F7	F7	20	07	1D	1010100
02	001E	26	F7	20	26	1D	0111101
01	0023	6D	F7	6D	26	1D	11000110
03	0025	C6	F7	6D	26	C6	10100111
03	002F	CA	F7	6D	26	CA	11110111
01	0036	35	F7	35	26	CA	01101110
00	003F	03	03	35	26	CA	11111100
00	0043	E9	E9	35	26	CA	110000100
00	0049	65	65	35	26	CA	100100100
01	0050	40	65	40	26	CA	000010110
02	005A	6E	65	40	6E	CA	010110101
02	005F	F6	65	40	F6	CA	111110101
00	0068	26	26	40	F6	CA	000101100
...	...	...	...	...	...	...	...