



UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base
Corso di Laurea in Ingegneria Informatica

Elaborato in **Architettura dei Sistemi Digitali**

Elaborato finale

Anno Accademico 2024/25

Studenti

Filomena Vigliotti

matr. M63001734

Ciro Scognamilgio

matr. M6300

Antonio Sirignano

matr. M63001732

Indice

1	Problema 1.1	1
1.1	Multiplexer 16:1	1
1.1.1	Progetto e architettura	3
1.1.2	Implementazione	8
1.1.3	Simulazione	12
1.2	Rete di interconnessione a 16 ingressi e 4 uscite	15
1.2.1	Progettazione	15
1.2.2	Implementazione	15
1.2.3	Simulazione	15
1.3	Implementazione su board del punto precedente	15
2	Esercizio 2 - Sistema ROM+M	16
3	Title	17
3.1	Section	17
4	Title	21
4.1	Section	21
5	Conclusion	25

Bibliografia

26

Chapter 1

Problema 1.1

1.1 Multiplexer 16:1

Un multiplexer è una **macchina combinatoria**, ovvero una macchina la cui uscita in un determinato istante di tempo dipende solo dall'ingresso nel medesimo istante, e quindi realizza una funzione del tipo:

$$U = f(I)$$

dove I e U rappresentano rispettivamente gli insiemi limitati dei valori di ingresso e di uscita.

Il Multiplexer realizza una connessione $n:1$, ovvero connette n sorgenti a un'unica destinazione sulla base di segnali di selezione.

Un **Multiplexer lineare** è composto da n segnali in ingresso e n segnali di selezione. Tale dispositivo convoglia uno specifico segnale in ingresso verso l'uscita solo se il corrispondente segnale di selezione è

alto. Uno svantaggio di un dispositivo di questo tipo è il numero eccessivo di fili per i segnali di selezione. Per risolvere ciò si può aggiungere un **Decoder**, un altro dispositivo notevole, che riceve in ingresso una parola codice di n bit e presenta in uscita la sua rappresentazione decodificata di 2^n bit.

Unendo un Multiplexer lineare a un Decoder, l'architettura diventa quella in figura, e si ottiene un componente definito **Multiplexer indirizzabile**, che diversamente da quello lineare, prende solo 2 segnali di selezione in ingresso. Un MUX indirizzabile è a sua volta una macchina notevole, caratterizzata da 2^n ingressi, n ingressi di selezione e un'unica uscita.

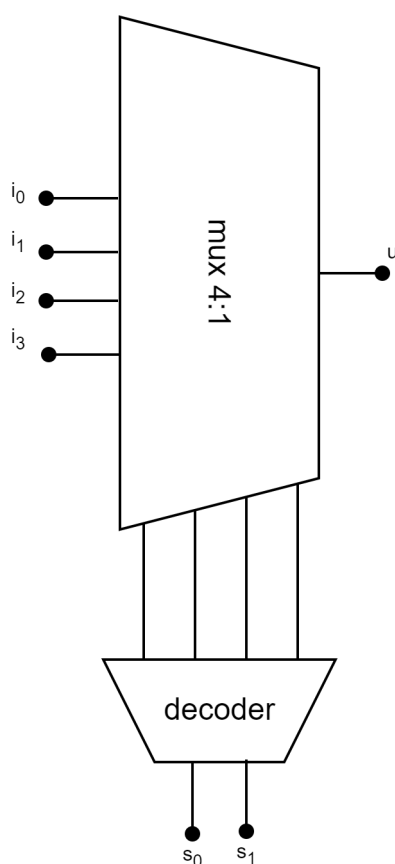


Figure 1.1: multiplexer indirizzabile

Si vuole ora progettare un multiplexer indirizzabile 16:1, utilizzando un approccio per composizione, a partire da multiplexer 4:1.

Tale multiplexer è rappresentato di seguito.

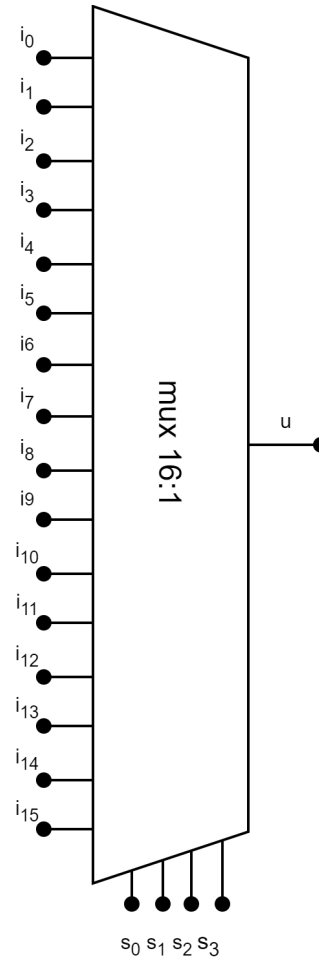


Figure 1.2: multiplexer 16:1

1.1.1 Progetto e architettura

Dapprima si utilizza un approccio per composizione per realizzare un multiplexer 4:1 con multiplexer 2:1.

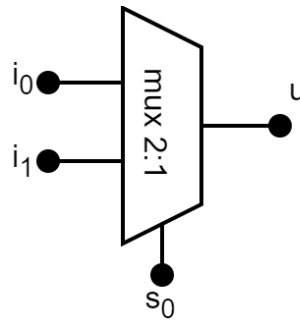


Figure 1.3: multiplexer 2:1

Il primo componente che si realizza è un multiplexer 2:1, caratterizzato dalla seguente tabella di verità:

s_0	i_1	i_0	u
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Table 1.1: Tabella di verità di un Mux 2:1

da cui si ottiene l'equazione:

$$y = (i_0 \text{ AND } \bar{s}_0) \text{ OR } (i_1 \text{ AND } s_0)$$

Il successivo componente da costruire è un multiplexer 4:1.

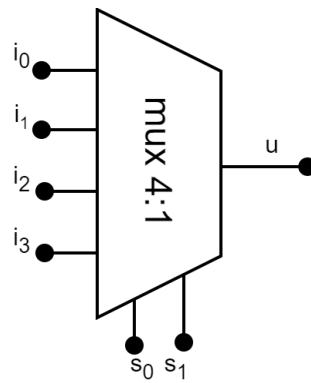


Figure 1.4: multiplexer 4:1

Per composizione, a partire da 3 multiplexer 2:1, si può ottenere un multiplexer 4:1

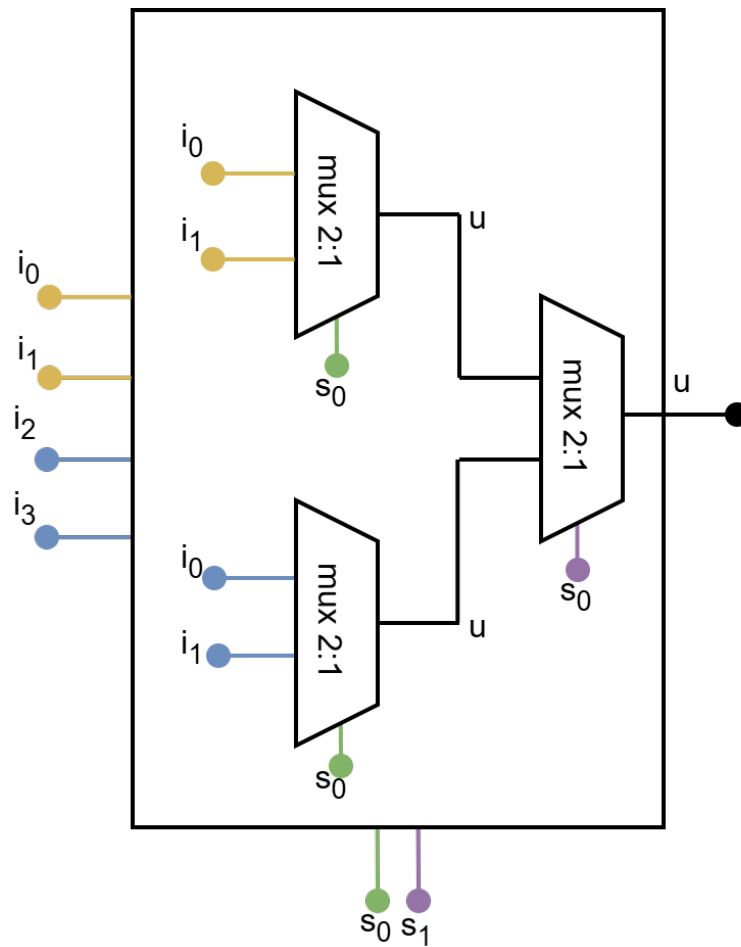


Figure 1.5: multiplexer 4:1 per composizione di multiplexer 2:1

I 4 ingressi entrano in due multiplexer 2:1, che prendono due ingressi e producono un'uscita ciascuno; tali uscite vengono immesse nel terzo multiplexer, che produrrà l'unico output finale. Concettualmente, si divide la selezione in ingresso al multiplexer esterno in due parti:

- la parte meno significativa (indicata dal colore verde) s_0 , viene posta in ingresso ai multiplexer del primo stadio e seleziona per ciascuno un filo in uscita;
- la parte più significativa (indicata dal colore viola) s_1 entra nel multiplexer del secondo stadio e decide quale dei due fili, provenienti dai due blocchi precedenti, sarà immessa in uscita.

In maniera analoga si procede con la progettazione del multiplexer 16:1.

Anche in questo caso, sono stati usati dei colori per identificare i collegamenti tra le componenti.

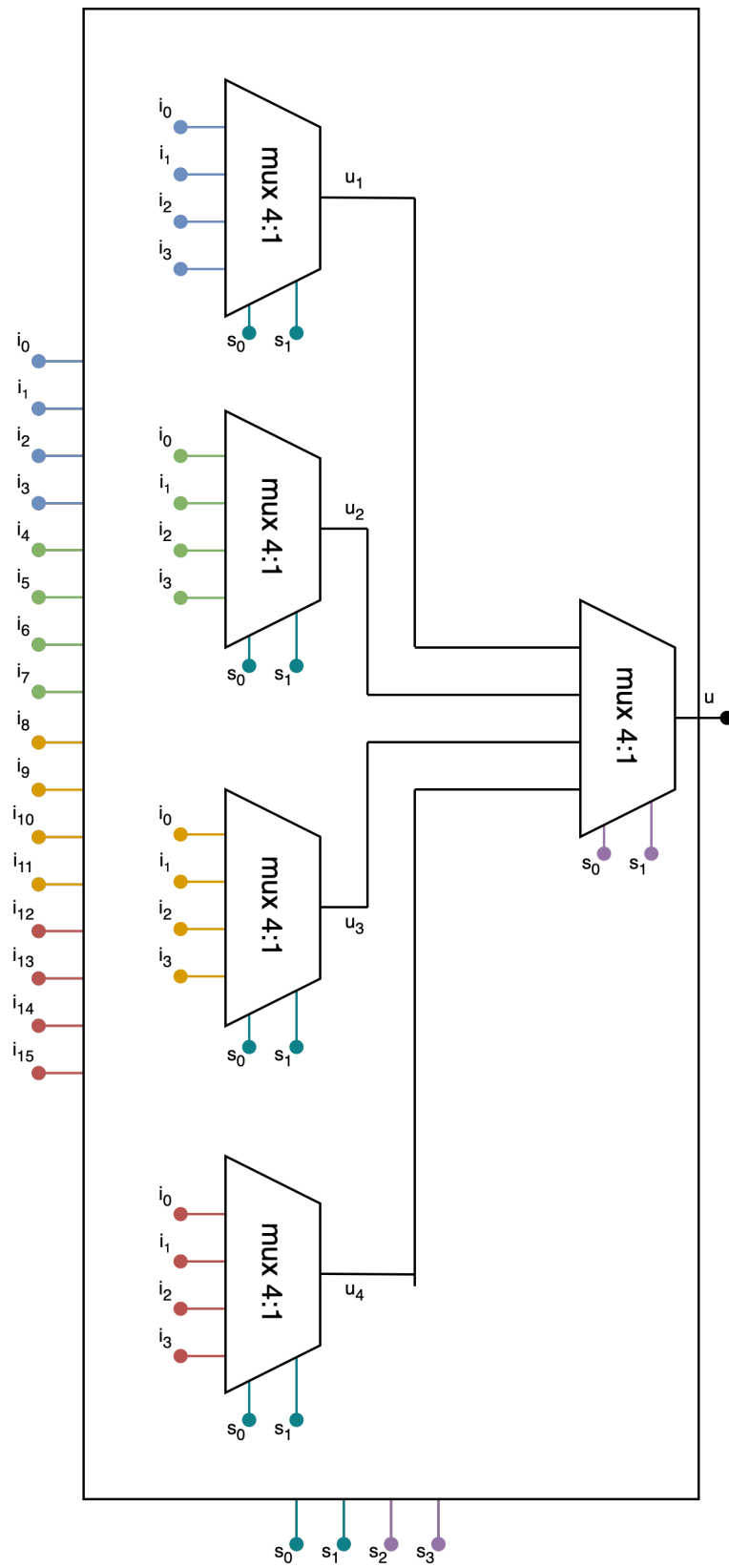


Figure 1.6: multiplexer 16:1 per composizione di multiplexer 4:1

1.1.2 Implementazione

Per l'implementazione si procede con un approccio di tipo strutturale, iniziando quindi dalla codifica del multiplexer 2:1, e, a partire da questo si compongono dispositivi sempre più complessi fino ad arrivare all'obiettivo del multiplexer 16:1.

Mux 2:1 Di seguito il codice riguardante il Mux 2:1.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.all;
3
4  entity mux_21 is
5      port
6      (
7          i0, i1: in std_logic;
8          s0: in std_logic;
9          u: out std_logic
10     );
11 end mux_21;
12
13 architecture rtl of mux_21 is
14 begin
15     u <= i0 when s0='0' else
16         i1 when s0='1' else
17         '-';
18 end rtl;
```

Code 1.1: Multiplexer 2:1 in VHDL

L'interfaccia del componente ha come ingressi *i0* ed *i1*, come selezione *s0* e come uscita *u*.

Al seguito della definizione dell'interfaccia, si definisce il comportamento dell'entità, che risponde alla tabella della verità 1.1.

Mux 4:1 Si prosegue con il Mux 4:1.

Come anticipato, viene costruito a partire da tre mux 2:1.

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity mux_41 is
5      port
6      (
7          i: in std_logic_vector(0 to 3);
8          s: in std_logic_vector(1 downto 0);
9          u: out std_logic
10     );
11 end mux_41;
12
13 architecture structural of mux_41 is
14     signal u_mid: std_logic_vector(0 to 1);
15
16     component mux_21 is
17         port (
18             i0, i1: in std_logic;
19             s0: in std_logic;
20             u: out std_logic
21         );
22     end component;
23
24     begin
25         mux0to1: FOR k IN 0 TO 1 GENERATE
26             m: mux_21
27             port map
28             (
29                 i(k*2),
30                 i(k*2 + 1),
31                 s(0),
32                 u_mid(k)
33             );
34         end GENERATE;
35
36         mux_2: mux_21
37         port map
38         (
39             u_mid(0),
40             u_mid(1),
41             s(1),
```

```
42         u
43     );
44
45 end structural;
```

Code 1.2: Multiplexer 4:1 in VHDL

In quest'entità, l'interfaccia è dichiarata come segue:

- Il parametro `i` vettore di 4 elementi, ognuno corrispondente ad un ingresso del mux 4:1.
- Il parametro `s` vettore di 2 elementi, ognuno corrispondente ad un ingresso di selezione.
- Il parametro `u` corrispondente all'uscita del multiplexer.

A seguire si definisce la struttura del mux 4:1, utilizzando mux 2:1 come componenti.

Con il ciclo `for`, vengono stanziati i primi due mux 2:1, i quali riceveranno in ingresso rispettivamente, gli ingressi del mux 4:1 e la loro uscita è il vettore d'appoggio `u_mid`, il quale è talvolta l'ingresso del terzo mux 2:1.

Mux 16:1 In maniera analoga si procede con la costruzione del mux 16:1. Il codice è il seguente:

```
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity mux_161 is
```

```
5      port
6      (
7          i: in std_logic_vector(0 to 15);
8          s: in std_logic_vector(3 downto 0);
9          u: out std_logic
10     );
11 end mux_161;
12
13 architecture structural of mux_161 is
14
15     signal u_mid: std_logic_vector(0 to 3);
16
17     component mux_41 is
18     port
19     (
20         i: in std_logic_vector(0 to 3);
21         s: in std_logic_vector(0 to 1);
22         u: out std_logic
23     );
24 end component;
25
26 begin
27     mux0to3: FOR k IN 0 TO 3 GENERATE
28         m: mux_41
29         port map
30         (
31             i => i((k*4) to (k*4 + 3)),
32             s => s(1 downto 0),
33             u => u_mid(k)
34         );
35     end GENERATE;
36
37     mux_2: mux_41
38     port map
39     (
40         i => u_mid,
41         s => s(3 downto 2),
42         u => u
43     );
44
45 end structural;
```

Code 1.3: Multiplexer 16:1 in VHDL

1.1.3 Simulazione

Per la simulazione, vi è la necessità di un testbench, il quale generiamo in maniera automatica tramite software appositi.

In tale progetto la generazione viene effettuata tramite ChatGPT ed il codice è il seguente:

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.numeric_std.all; -- Libreria necessaria per `to_unsigned`
4
5  entity tb_mux_161 is
6  end tb_mux_161;
7
8  architecture behavior of tb_mux_161 is
9      -- Component declaration
10     component mux_161
11     port (
12         i: in std_logic_vector(0 to 15);
13         s: in std_logic_vector(3 downto 0);
14         u: out std_logic
15     );
16     end component;
17
18     -- Signals for testing
19     signal i: std_logic_vector(0 to 15);
20     signal s: std_logic_vector(3 downto 0);
21     signal u: std_logic;
22
23 begin
24     -- Instantiate the unit under test (UUT)
25     uut: mux_161
26     port map (
27         i => i,
28         s => s,
29         u => u
30     );
31
32     -- Test process
33     stim_proc: process
34         variable expected_output: std_logic; -- Variabile per il
           ↪ controllo

```

```

35  begin
36      -- Initialize inputs
37      i <= (others => '0');
38      s <= "0000";
39      wait for 10 ns;
40
41      -- Apply test cases
42      for sel in 0 to 15 loop
43          -- Set the ith bit of i to '1'
44          i <= (others => '0');
45          i(sel) <= '1';
46
47          -- Set the selector
48          s <= std_logic_vector(to_unsigned(sel, 4));
49
50          -- Aspetta che l'uscita si stabilizzi
51          wait for 10 ns;
52
53          -- Calcola l'uscita attesa
54          expected_output := i(sel);
55
56          -- Controlla se l'uscita è corretta
57          if u = expected_output then
58              report "Test passed for s = " & integer'image(sel) &
59                  ", u = " & std_logic'image(u);
60          else
61              report "Test failed for s = " & integer'image(sel) &
62                  ": expected = " &
63                  ⇨ std_logic'image(expected_output) &
64                  ", got = " & std_logic'image(u)
65                  severity error;
66          end if;
67      end loop;
68
69      -- Fine simulazione
70      report "All tests completed";
71      wait;
72  end process;
73  end behavior;

```

Code 1.4: Testbench multiplexer 16:1 in VHDL

Una volta generato ciò, utilizzando i software GHDL e GTKWAVE,

vengono eseguiti i seguenti comandi:

```
antoniosirignano@Antonios-MacBook-Pro Problema_1 % ghdl -a mux_2_1.vhdl
antoniosirignano@Antonios-MacBook-Pro Problema_1 % ghdl -a mux_4_1.vhdl
antoniosirignano@Antonios-MacBook-Pro Problema_1 % ghdl -a mux_16_1.vhdl
antoniosirignano@Antonios-MacBook-Pro Problema_1 % ghdl -a tb_mux_16_1.vhdl
antoniosirignano@Antonios-MacBook-Pro Problema_1 % ghdl -e tb_mux_161
antoniosirignano@Antonios-MacBook-Pro Problema_1 % ghdl -r tb_mux_161 --wave=mux_16_1.ghw

tb_mux_16_1.vhdl:58:17:@20ns:(report note): Test passed for s = 0, u = '1'
tb_mux_16_1.vhdl:58:17:@30ns:(report note): Test passed for s = 1, u = '1'
tb_mux_16_1.vhdl:58:17:@40ns:(report note): Test passed for s = 2, u = '1'
tb_mux_16_1.vhdl:58:17:@50ns:(report note): Test passed for s = 3, u = '1'
tb_mux_16_1.vhdl:58:17:@60ns:(report note): Test passed for s = 4, u = '1'
tb_mux_16_1.vhdl:58:17:@70ns:(report note): Test passed for s = 5, u = '1'
tb_mux_16_1.vhdl:58:17:@80ns:(report note): Test passed for s = 6, u = '1'
tb_mux_16_1.vhdl:58:17:@90ns:(report note): Test passed for s = 7, u = '1'
tb_mux_16_1.vhdl:58:17:@100ns:(report note): Test passed for s = 8, u = '1'
tb_mux_16_1.vhdl:58:17:@110ns:(report note): Test passed for s = 9, u = '1'
tb_mux_16_1.vhdl:58:17:@120ns:(report note): Test passed for s = 10, u = '1'
tb_mux_16_1.vhdl:58:17:@130ns:(report note): Test passed for s = 11, u = '1'
tb_mux_16_1.vhdl:58:17:@140ns:(report note): Test passed for s = 12, u = '1'
tb_mux_16_1.vhdl:58:17:@150ns:(report note): Test passed for s = 13, u = '1'
tb_mux_16_1.vhdl:58:17:@160ns:(report note): Test passed for s = 14, u = '1'
tb_mux_16_1.vhdl:58:17:@170ns:(report note): Test passed for s = 15, u = '1'
tb_mux_16_1.vhdl:69:9:@170ns:(report note): All tests completed
antoniosirignano@Antonios-MacBook-Pro Problema_1 % gtkwave mux_16_1.ghw

GTKWave Analyzer v3.4.0 (w)1999-2022 BSI

[0] start time.
[170000000] end time.
2024-11-25 18:54:06.970 gtkwave[78165:3049537] +[IMKClient subclass]: chose IMKClient_Modern
2024-11-25 18:54:06.970 gtkwave[78165:3049537] +[IMKInputSession subclass]: chose IMKInputSession_Modern
```

Figure 1.7: Comandi per la simulazione

Con l'esecuzione dell'ultimo comando, vi si apre una nuova finestra che permette la visualizzazione delle onde:

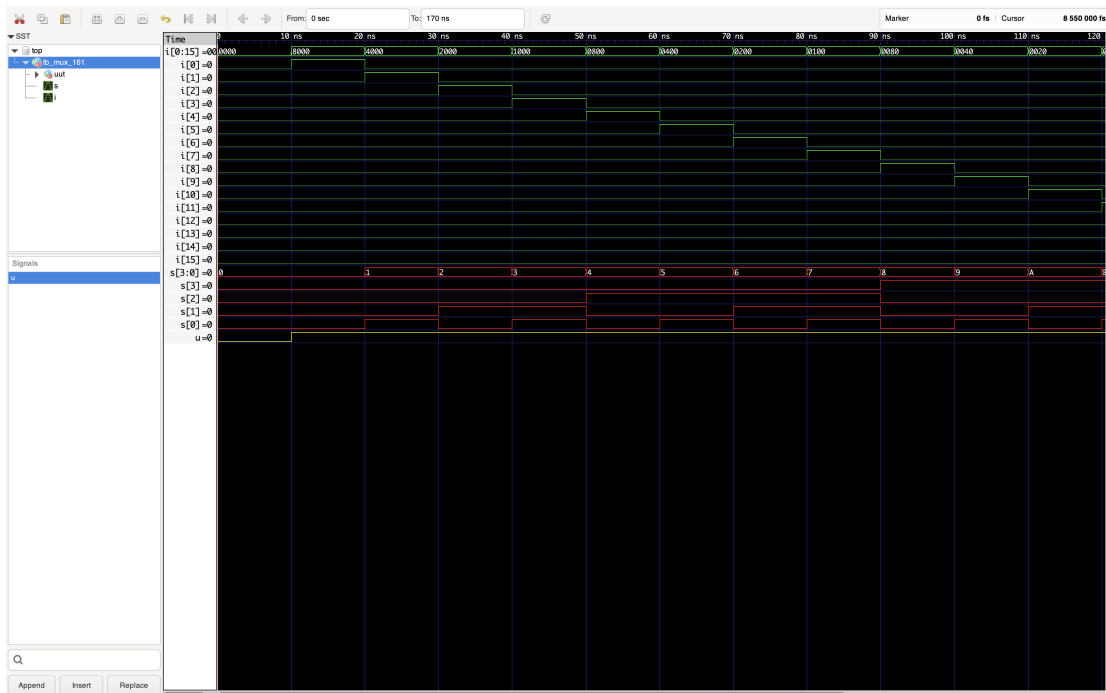


Figure 1.8: Comandi per la simulazione

1.2 Rete di interconnessione a 16 ingressi e 4 uscite

1.2.1 Progettazione

1.2.2 Implementazione

1.2.3 Simulazione

1.3 Implementazione su board del punto precedente

Chapter 2

Esercizio 2 - Sistema

ROM+M

Chapter 3

Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

3.1 Section ...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis,

viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate

metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros,

malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Chapter 4

Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

4.1 Section ...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis,

viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate

metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros,

malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Chapter 5

Conclusion

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Bibliografia