

scan\_chain\_on\_board

Corso di ASE anno 18/19

Gruppo 14  
PREVITERA GABRIELE  
PENNONE MIRKO  
PENNA SIMONE



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	anodes_manager Entity Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.2	cathodes_manager Entity Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.3	clock_divisor Entity Reference . . . . .	6
3.3.1	Detailed Description . . . . .	7
3.3.2	Member Data Documentation . . . . .	7
3.3.2.1	STD_LOGIC_1164 . . . . .	7
3.4	counter_UpMod2n_Re_Sr Entity Reference . . . . .	7
3.4.1	Detailed Description . . . . .	8
3.5	display_7_segmenti Entity Reference . . . . .	8
3.5.1	Detailed Description . . . . .	9
3.6	flipflop_d Entity Reference . . . . .	9
3.6.1	Detailed Description . . . . .	10
3.7	flipflop_mux Entity Reference . . . . .	10
3.7.1	Detailed Description . . . . .	11
3.8	mux2 Entity Reference . . . . .	11
3.8.1	Detailed Description . . . . .	11
3.9	register_d_Re_Ar Entity Reference . . . . .	11
3.9.1	Detailed Description . . . . .	12
3.9.2	Member Data Documentation . . . . .	12
3.9.2.1	STD_LOGIC_1164 . . . . .	12
3.10	scan_chain Entity Reference . . . . .	13
3.10.1	Detailed Description . . . . .	13
3.11	scan_chain_on_board Entity Reference . . . . .	14
3.11.1	Member Data Documentation . . . . .	14
3.11.1.1	STD_LOGIC_1164 . . . . .	14
3.12	tb_shifter Entity Reference . . . . .	14

<b>4 File Documentation</b>	<b>15</b>
4.1 anodes_manager.vhd File Reference . . . . .	15
4.1.1 Detailed Description . . . . .	15
4.2 cathodes_manager.vhd File Reference . . . . .	15
4.2.1 Detailed Description . . . . .	16
4.3 counter_UpMod2n_Re_Sr.vhd File Reference . . . . .	16
4.3.1 Detailed Description . . . . .	16
4.4 display_7_segmenti.vhd File Reference . . . . .	17
4.4.1 Detailed Description . . . . .	17
4.5 flipflop_d.vhd File Reference . . . . .	17
4.5.1 Detailed Description . . . . .	17
4.6 flipflop_mux.vhd File Reference . . . . .	18
4.6.1 Detailed Description . . . . .	18
4.7 mux2.vhd File Reference . . . . .	18
4.7.1 Detailed Description . . . . .	18
4.8 scan_chain.vhd File Reference . . . . .	19
4.8.1 Detailed Description . . . . .	19
<b>Index</b>	<b>21</b>

# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

entity <a href="#">anodes_manager</a>	
Permette di gestire gli anodi associati ad ogni cifra(digit) di un display a 7 segmenti.	
Per accendere la cifra giusta(digit) è necessario che l'anodo sia 0, poichè gli anodi sono pilotati da segnali 0-attivi	5
entity <a href="#">cathodes_manager</a>	6
entity <a href="#">clock_divisor</a>	
Filtra i fronti del clock ad una frequenza "clock_frequency_in" per averli ad una frequenza più bassa "clock_frequency_out"	6
entity <a href="#">counter_UpMod2n_Re_Sr</a>	7
entity <a href="#">display_7_segmenti</a>	8
entity <a href="#">flipflop_d</a>	
Flipflop implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono	9
entity <a href="#">flipflop_mux</a>	10
entity <a href="#">mux2</a>	11
entity <a href="#">register_d_Re_Ar</a>	
Registro di dimensione "dimension" che prende in ingresso un dato D e lo memorizza	11
entity <a href="#">scan_chain</a>	13
entity <a href="#">scan_chain_on_board</a>	14
entity <a href="#">tb_shifter</a>	14



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">anodes_manager.vhd</a>	
Anodees_manager per il display a 7 segmenti . . . . .	15
<a href="#">cathodes_manager.vhd</a>	
Cathodes_manager per il display a 7 segmenti . . . . .	15
<a href="#">counter_UpMod2n_Re_Sr.vhd</a>	
Contatore modulo 2 alla N . . . . .	16
<a href="#">display_7_segmenti.vhd</a>	
Componente che permette di pilotare fino a 4 digit ricevendo il valore da mostrare sul display come sequenza di bit . . . . .	17
<a href="#">flipflop.d.vhd</a>	
Flipflop.d implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono . . . . .	17
<a href="#">flipflop.mux.vhd</a>	
Flipflop.mux è un flip flop D con multiplexer . . . . .	18
<a href="#">mux2.vhd</a>	
Implementazione dataflow di un mux 2 . . . . .	18
<a href="#">scan_chain.vhd</a>	
Scan chain è un registro di width flipflop D multiplexati . . . . .	19





## Chapter 3

# Class Documentation

### 3.1 anodes\_manager Entity Reference

Permette di gestire gli anodi associati ad ogni cifra(digit) di un display a 7 segmenti.

Per accendere la cifra giusta(digit) è necessario che l'anodo sia 0, poichè gli anodi sono pilotati da segnali 0-attivi.

#### Libraries

- [IEEE](#)

#### Use Clauses

- [STD\\_LOGIC\\_1164](#)

#### Ports

- [select\\_digit](#) in **STD\_LOGIC\_VECTOR( 2 downto 0 )**  
*anodes\_manager input: seleziona digit*
- [enable\\_digit](#) in **STD\_LOGIC\_VECTOR( 7 downto 0 )**  
*anodes\_manager input: abilita digit*
- [anodes](#) out **STD\_LOGIC\_VECTOR( 7 downto 0 )**  
*anodes\_manager output: digit da accendere*

#### 3.1.1 Detailed Description

Permette di gestire gli anodi associati ad ogni cifra(digit) di un display a 7 segmenti.

Per accendere la cifra giusta(digit) è necessario che l'anodo sia 0, poichè gli anodi sono pilotati da segnali 0-attivi.

The documentation for this class was generated from the following file:

- [anodes\\_manager.vhd](#)

## 3.2 cathodes\_manager Entity Reference

### Libraries

- [IEEE](#)

*architecture dataflow of [anodes\\_manager](#) end*

### Use Clauses

- [STD\\_LOGIC\\_1164](#)
- [NUMERIC\\_STD](#)

### Ports

- [select\\_digit](#) in [STD\\_LOGIC\\_VECTOR](#)( [2](#) downto [0](#) )  
*[cathodes\\_manager](#) input: seleziona digit su cui mostrare la cifra*
- [values](#) in [STD\\_LOGIC\\_VECTOR](#)( [31](#) downto [0](#) )  
*[cathodes\\_manager](#) input: valore da mostrare (codifica esadecimale)*
- [dots](#) in [STD\\_LOGIC\\_VECTOR](#)( [7](#) downto [0](#) )  
*[cathodes\\_manager](#) input: punto da accendere per la parte decimale*
- [cathodes](#) out [STD\\_LOGIC\\_VECTOR](#)( [7](#) downto [0](#) )  
*[cathodes\\_manager](#) output: catodo da accendere*

### 3.2.1 Detailed Description

Permette di gestire l'abilitazione dei catodi associati ad ogni segmento omologo di ogni cifra(digit) di un display a 7 segmenti.

Per accendere il giusto segmento è necessario che il catodo sia 0, poichè i catodi sono pilotati da segnali 0-attivi.

The documentation for this class was generated from the following file:

- [cathodes\\_manager.vhd](#)

## 3.3 clock\_divisor Entity Reference

Filtra i fronti del clock ad una frequenza "clock\_frequency\_in" per averli ad una frequenza più bassa "clock\_frequency\_out".

### Libraries

- [IEEE](#)

*architecture behavioral of [cathodes\\_manager](#) end*

### Use Clauses

- [STD\\_LOGIC\\_1164](#)

## Generics

- `clock_frequency_in` integer:= 100000000  
*frequenza del clock in ingresso*
- `clock_frequency_out` integer:= 1000  
*frequenza del clock in uscita*

## Ports

- `enable` in STD\_LOGIC  
*clock\_divisor input: segnale enable*
- `reset_n` in STD\_LOGIC  
*clock\_divisor input: segnale reset*
- `clock_freq_in` in STD\_LOGIC  
*clock\_divisor input: segnale di clock in ingresso*
- `clock_freq_out` out STD\_LOGIC  
*clock\_divisor output: segnale di clock in uscita*

### 3.3.1 Detailed Description

Filtra i fronti del clock ad una frequenza "clock\_frequency\_in" per averli ad una frequenza più bassa "clock\_frequency\_out".

### 3.3.2 Member Data Documentation

#### 3.3.2.1 STD\_LOGIC\_1164

STD\_LOGIC\_1164 [Package]

last changes: <11/11/2018> <15/10/2018> <log> Aggiunta doc doxygen

The documentation for this class was generated from the following file:

- clock\_divisor.vhd

## 3.4 counter\_UpMod2n\_Re\_Sr Entity Reference

### Libraries

- IEEE  
*architecture behavioral of clock\_divisor end*

## Use Clauses

- [STD\\_LOGIC\\_1164](#)
- [numeric\\_std](#)

## Generics

- [n](#) **NATURAL** := **1**
- [enable\\_level](#) **STD\_LOGIC** := '1'

## Ports

- [enable](#) in **STD\_LOGIC**  
*enable input*
- [reset\\_n](#) in **STD\_LOGIC**  
*reset input*
- [clock](#) in **STD\_LOGIC**  
*clock input*
- [count\\_hit](#) out **STD\_LOGIC**  
*count\_hit output*
- [COUNTS](#) out **STD\_LOGIC\_VECTOR**((**n-1**)downto **0**)  
*COUNT output.*

### 3.4.1 Detailed Description

Contatore modulo 2 alla N. Il conteggio viene effettuato sul fronte di salita del clock e il reset è sincrono.

The documentation for this class was generated from the following file:

- [counter\\_UpMod2n\\_Re\\_Sr.vhd](#)

## 3.5 display\_7\_segmenti Entity Reference

### Libraries

- [IEEE](#)  
*architecture behavioral of [counter\\_UpMod2n\\_Re\\_Sr](#) end*

### Use Clauses

- [STD\\_LOGIC\\_1164](#)

## Ports

- **enable** in STD\_LOGIC  
*enable del componente*
- **clock** in STD\_LOGIC  
*clock*
- **reset** in STD\_LOGIC  
*reset 1-attivo*
- **values** in STD\_LOGIC\_VECTOR( 31 downto 0 )  
*Stringa di bit del valore da mostrare.*
- **dots** in STD\_LOGIC\_VECTOR( 7 downto 0 )  
*Segnali che permette di pilotare i punti.*
- **enable\_digit** in STD\_LOGIC\_VECTOR( 7 downto 0 )  
*Segnali che attiva le digit.*
- **anodes** out STD\_LOGIC\_VECTOR( 7 downto 0 )  
*Uscita che pilota gli anodi.*
- **cathodes** out STD\_LOGIC\_VECTOR( 7 downto 0 )  
*Uscita che pilota i catodi.*

### 3.5.1 Detailed Description

Componente che permette di pilotare fino a 4 digit ricevendo il valore da mostrare sul display come sequenza di bit

The documentation for this class was generated from the following file:

- [display\\_7\\_segmenti.vhd](#)

## 3.6 flipflop Entity Reference

flipflop implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono.

## Libraries

- [IEEE](#)

## Use Clauses

- [STD\\_LOGIC\\_1164](#)

## Generics

- **init\_value** STD\_LOGIC:= ' 0 '  
*definisce il livello iniziale del flipflop*
- **reset\_level** STD\_LOGIC:= ' 0 '  
*definisce il livello reset*
- **enable\_level** STD\_LOGIC:= ' 1 '  
*definisce il livello enable*

## Ports

- [clock](#) in **STD\_LOGIC**  
*flipflop input : segnale di clock per sincronizzare*
- [enable](#) in **STD\_LOGIC**  
*flipflop input : segnale enable*
- [reset](#) in **STD\_LOGIC**  
*flipflop input : segnale reset*
- [d](#) in **STD\_LOGIC**  
*flipflop input : input data*
- [q](#) out **STD\_LOGIC**  
*flipflop output : output data*

### 3.6.1 Detailed Description

flipflop implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono.

The documentation for this class was generated from the following file:

- [flipflop.vhd](#)

## 3.7 flipflopmux Entity Reference

### Libraries

- [IEEE](#)  
*architecture behavioural end*

### Use Clauses

- [STD\\_LOGIC\\_1164](#)

## Ports

- [clock](#) in **STD\_LOGIC**  
*clock*
- [en](#) in **STD\_LOGIC**  
*enable*
- [reset\\_n](#) in **STD\_LOGIC**  
*reset*
- [scan\\_en](#) in **STD\_LOGIC**  
*segnale di selezione del multiplexer per modalità (0 = normale, 1 = controllo)*
- [d](#) in **STD\_LOGIC**  
*ingresso del flipflop in modalità normale*
- [scan\\_in](#) in **STD\_LOGIC**  
*ingresso del flipflop in modalità controllo*
- [q](#) out **STD\_LOGIC**  
*uscita del flipflop*

### 3.7.1 Detailed Description

flipflopmux è un flip flop D con multiplexer: scan\_en è il segnale di controllo del multiplexer, se scan\_en = 0 l'ingresso è d, se scan\_en = 1 l'ingresso è scan\_in.

The documentation for this class was generated from the following file:

- [flipflopmux.vhd](#)

## 3.8 mux2 Entity Reference

### Libraries

- [IEEE](#)

### Use Clauses

- [STD\\_LOGIC\\_1164](#)

### Ports

- [in0](#) in **STD\_LOGIC**
- [in1](#) in **STD\_LOGIC**
- [sel](#) in **STD\_LOGIC**
- [o](#) out **STD\_LOGIC**

### 3.8.1 Detailed Description

multiplexer: l'uscita o è in0 se sel = 0, in1 se sel = 1 se sel assume altri valori, lo riporto in uscita per evitare fault masking

The documentation for this class was generated from the following file:

- [mux2.vhd](#)

## 3.9 register\_d\_Re\_Ar Entity Reference

Registro di dimensione "dimension" che prende in ingresso un dato D e lo memorizza.

### Libraries

- [IEEE](#)

## Use Clauses

- [STD\\_LOGIC\\_1164](#)

## Generics

- **dimension** **NATURAL** := **8**  
*definisce il parallelismo del registro*
- **reset\_level** **STD\_LOGIC** := **' 1 '**  
*definisce il livello reset*
- **load\_level** **STD\_LOGIC** := **' 1 '**  
*definisce il livello enable*

## Ports

- **clock** in **STD\_LOGIC**  
*register\_d\_Re\_Ar input : segnale di clock per sincronizzare*
- **load** in **STD\_LOGIC**  
*register\_d\_Re\_Ar input : segnale enable*
- **reset** in **STD\_LOGIC**  
*register\_d\_Re\_Ar input : segnale reset*
- **d** in **STD\_LOGIC\_VECTOR**(**dimension** - **1** downto **0**)  
*register\_d\_Re\_Ar input : inpput data*
- **q** out **STD\_LOGIC\_VECTOR**(**dimension** - **1** downto **0**)  
*register\_d\_Re\_Ar input : output data*

### 3.9.1 Detailed Description

Registro di dimensione "dimension" che prende in ingresso un dato D e lo memorizza.

### 3.9.2 Member Data Documentation

#### 3.9.2.1 STD\_LOGIC\_1164

[STD\\_LOGIC\\_1164](#) [Package]

last changes: <16/11/2018> <16/11/2018> <log> create

The documentation for this class was generated from the following file:

- register\_d\_Re\_Ar.vhd



## 3.10 scan\_chain Entity Reference

### Libraries

- [IEEE](#)  
*architecture behavioral of [register\\_d\\_Re\\_Ar](#) end*

### Use Clauses

- [STD\\_LOGIC\\_1164](#)

### Generics

- **width integer := 8**  
*dimensione del registro*
- **shift\_direction std\_logic := '1'**  
*shift a sinistra*

### Ports

- **clock in STD\_LOGIC**  
*segnale clock di temporizzazione*
- **en in STD\_LOGIC**  
*segnale di abilitazione 1-attivo*
- **reset\_n in STD\_LOGIC**  
*segnale di reset 0-attivo*
- **scan\_en in STD\_LOGIC**  
*segnale di selezione modalità (0 = normale, 1 = controllo)*
- **scan\_in in STD\_LOGIC**  
*primo valore scan-in*
- **d\_reg in STD\_LOGIC\_VECTOR(width - 1 downto 0)**  
*valore in ingresso nel registro*
- **scan\_out out STD\_LOGIC**  
*ultimo valore scan-out*
- **q\_reg out STD\_LOGIC\_VECTOR(width - 1 downto 0)**  
*valore in uscita del registro*

### 3.10.1 Detailed Description

Scan chain è un registro di width flipflop D multiplexati. Quando scan\_en = 0, il componente si comporta come un normale registro. Quando scan\_en = 1, diventa uno shift register che shifta ad ogni colpo di clock. La direzione dello shift è regolata dal generic shift\_direction (0 = right, 1 = left)

The documentation for this class was generated from the following file:

- [scan\\_chain.vhd](#)

### 3.11 scan\_chain\_on\_board Entity Reference

#### Libraries

- [IEEE](#)

#### Use Clauses

- [STD\\_LOGIC\\_1164](#)

#### Ports

- [clock](#) in STD\_LOGIC
- [scan\\_in](#) in STD\_LOGIC
- [scan\\_clk](#) in STD\_LOGIC
- [scan\\_en](#) in STD\_LOGIC
- [scan\\_out](#) out STD\_LOGIC
- [anodes](#) out STD\_LOGIC\_VECTOR( 7 downto 0 )
- [cathodes](#) out STD\_LOGIC\_VECTOR( 7 downto 0 )

#### 3.11.1 Member Data Documentation

##### 3.11.1.1 STD\_LOGIC\_1164

[STD\\_LOGIC\\_1164](#) [Package]

last changes: <16/11/2018> <16/11/2018> <log> create

The documentation for this class was generated from the following file:

- scan\_chain\_on\_board.vhd

### 3.12 tb\_shifter Entity Reference

#### Libraries

- [ieee](#)

#### Use Clauses

- [std\\_logic\\_1164](#)

The documentation for this class was generated from the following file:

- tb\_shifter.vhd

## Chapter 4

# File Documentation

### 4.1 `anodes_manager.vhd` File Reference

`anodees_manager` per il display a 7 segmenti

#### Entities

- `anodes_manager` entity

*Permette di gestire gli anodi associati ad ogni cifra(digit) di un display a 7 segmenti.*

*Per accendere la cifra giusta(digit) è necessario che l'anodo sia 0, poichè gli anodi sono pilotati da segnali 0-attivi.*

#### 4.1.1 Detailed Description

`anodees_manager` per il display a 7 segmenti

#### Author

Gabriele Previtera, Mirko Pennone, Simone Penna

#### Date

04/03/2019

#### Version

0.2

#### Dependencies:

Nothings

### 4.2 `cathodes_manager.vhd` File Reference

`cathodes_manager` per il display a 7 segmenti

## Entities

- [cathodes\\_manager](#) entity

### 4.2.1 Detailed Description

[cathodes\\_manager](#) per il display a 7 segmenti

#### Author

Gabriele Previtera, Mirko Pennone, Simone Penna

#### Date

04/03/2019

#### Version

0.2

#### Dependencies:

Nothings

## 4.3 counter\_UpMod2n\_Re\_Sr.vhd File Reference

Contatore modulo 2 alla N.

## Entities

- [counter\\_UpMod2n\\_Re\\_Sr](#) entity

### 4.3.1 Detailed Description

Contatore modulo 2 alla N.

#### Author

Gabriele Previtera, Mirko Pennone, Simone Penna

#### Date

04/03/2019

#### Version

0.2

#### Dependencies:

Nothings

## 4.4 display\_7\_segmenti.vhd File Reference

Componente che permette di pilotare fino a 4 digit ricevendo il valore da mostrare sul display come sequenza di bit.

### Entities

- [display\\_7\\_segmenti](#) entity

### 4.4.1 Detailed Description

Componente che permette di pilotare fino a 4 digit ricevendo il valore da mostrare sul display come sequenza di bit.

#### Author

Gabriele Previtera, Mirko Pennone, Simone Penna

#### Date

04/03/2019

#### Version

0.2

#### Dependencies:

Nothings

## 4.5 flipflop.vhd File Reference

flipflop implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono.

### Entities

- [flipflop](#) entity  
*flipflop implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono.*

### 4.5.1 Detailed Description

flipflop implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono.

#### Author

Gabriele Previtera, Mirko Pennone, Simone Penna

#### Date

04/03/2019

#### Version

0.2

#### Dependencies:

Nothings

## 4.6 flipflopmux.vhd File Reference

flipflopmux è un flip flop D con multiplexer

### Entities

- [flipflopmux](#) entity

### 4.6.1 Detailed Description

flipflopmux è un flip flop D con multiplexer

#### Author

Gabriele Previtera, Mirko Pennone, Simone Penna

#### Date

04/03/2019

#### Version

0.2

#### Dependencies:

Nothings

## 4.7 mux2.vhd File Reference

implementazione dataflow di un mux 2

### Entities

- [mux2](#) entity

### 4.7.1 Detailed Description

implementazione dataflow di un mux 2

#### Author

Gabriele Previtera, Mirko Pennone, Simone Penna

#### Date

04/03/2019

#### Version

0.2

#### Dependencies:

Nothings

## 4.8 scan\_chain.vhd File Reference

Scan chain è un registro di width flipflop D multiplexati.

### Entities

- [scan\\_chain](#) entity

### 4.8.1 Detailed Description

Scan chain è un registro di width flipflop D multiplexati.

#### Author

Gabriele Previtera, Mirko Pennone, Simone Penna

#### Date

04/03/2019

#### Version

0.2

#### Dependencies:

Nothings





# Index

anodes\_manager, [5](#)  
anodes\_manager.vhd, [15](#)

cathodes\_manager, [6](#)  
cathodes\_manager.vhd, [15](#)  
clock\_divisor, [6](#)  
    STD\_LOGIC\_1164, [7](#)  
counter\_UpMod2n\_Re\_Sr, [7](#)  
counter\_UpMod2n\_Re\_Sr.vhd, [16](#)

display\_7\_segmenti, [8](#)  
display\_7\_segmenti.vhd, [17](#)

flipflop, [9](#)  
flipflop.vhd, [17](#)  
flipflopmux, [10](#)  
flipflopmux.vhd, [18](#)

mux2, [11](#)  
mux2.vhd, [18](#)

register\_d\_Re\_Ar, [11](#)  
    STD\_LOGIC\_1164, [12](#)

STD\_LOGIC\_1164  
    clock\_divisor, [7](#)  
    register\_d\_Re\_Ar, [12](#)  
    scan\_chain\_on\_board, [14](#)  
scan\_chain, [13](#)  
scan\_chain.vhd, [19](#)  
scan\_chain\_on\_board, [14](#)  
    STD\_LOGIC\_1164, [14](#)

tb\_shifter, [14](#)