

UART_PO_PC

Corso di ASE anno 18/19

Gruppo 14

PREVITERA GABRIELE

PENNONE MIRKO

PENNA SIMONE

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	anodes_manager Entity Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Data Documentation	5
3.1.2.1	IEEE	6
3.1.2.2	STD_LOGIC_1164	6
3.2	cathodes_manager Entity Reference	6
3.2.1	Detailed Description	6
3.2.2	Member Data Documentation	7
3.2.2.1	STD_LOGIC_1164	7
3.3	clock_divisor Entity Reference	7
3.3.1	Detailed Description	8
3.3.2	Member Data Documentation	8
3.3.2.1	STD_LOGIC_1164	8
3.4	counter_UpMod2n_Re_Sr Entity Reference	8
3.4.1	Detailed Description	9
3.5	counter_UpN_Re_preset_Sr Entity Reference	9
3.6	counter_UpN_Re_Sr Entity Reference	10

3.7	display_7_segments Entity Reference	10
3.7.1	Detailed Description	11
3.8	flag_FF Entity Reference	11
3.8.1	Detailed Description	12
3.9	flipflop_d_risingEdge_asyncReset Entity Reference	12
3.9.1	Detailed Description	13
3.9.2	Member Data Documentation	13
3.9.2.1	IEEE	13
3.9.2.2	STD_LOGIC_1164	13
3.10	flipflopmux Entity Reference	13
3.10.1	Detailed Description	14
3.11	mux2_1 Entity Reference	14
3.11.1	Detailed Description	14
3.11.2	Member Data Documentation	15
3.11.2.1	IEEE	15
3.11.2.2	STD_LOGIC_1164	15
3.12	register_d_Re_Ar Entity Reference	15
3.12.1	Detailed Description	16
3.12.2	Member Data Documentation	16
3.12.2.1	STD_LOGIC_1164	16
3.13	scan_chain Entity Reference	16
3.13.1	Detailed Description	17
3.14	uart Entity Reference	17
3.14.1	Member Data Documentation	18
3.14.1.1	rd_uart	18
3.15	uart_onBoard Entity Reference	18
3.16	uart_rx Entity Reference	19
3.16.1	Detailed Description	20
3.17	uart_rx_cu Entity Reference	20
3.17.1	Detailed Description	21
3.18	uart_rx_po Entity Reference	21
3.19	uart_tx Entity Reference	22
3.20	uart_tx_cu Entity Reference	22
3.21	uart_tx_po Entity Reference	23

4 File Documentation	25
4.1 counter_UpMod2n_Re_Sr.vhd File Reference	25
4.1.1 Detailed Description	25
4.2 counter_UpN_preset_Re_Sr.vhd File Reference	25
4.2.1 Detailed Description	26
4.3 counter_UpN_Re_Sr.vhd File Reference	26
4.3.1 Detailed Description	26
4.4 display_7_segments.vhd File Reference	27
4.4.1 Detailed Description	27
4.5 flag_FF.vhd File Reference	27
4.5.1 Detailed Description	27
4.6 flipflopmux.vhd File Reference	28
4.6.1 Detailed Description	28
4.7 scan_chain.vhd File Reference	28
4.7.1 Detailed Description	28
4.8 uart.vhd File Reference	29
4.8.1 Detailed Description	29
4.9 uart_onBoard.vhd File Reference	29
4.9.1 Detailed Description	29
4.10 uart_rx.vhd File Reference	30
4.10.1 Detailed Description	30
4.11 uart_rx_cu.vhd File Reference	30
4.11.1 Detailed Description	30
4.12 uart_rx_po.vhd File Reference	31
4.12.1 Detailed Description	31
4.13 uart_tx.vhd File Reference	31
4.13.1 Detailed Description	31
4.14 uart_tx_cu.vhd File Reference	32
4.14.1 Detailed Description	32
4.15 uart_tx_po.vhd File Reference	32
4.15.1 Detailed Description	32
Index	33

Chapter 1

Class Index

1.1 Class List

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

entity anodes_manager	
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 cathodes_manager	6
entity clock_divisor	
Filtra i fronti del clock ad una frequenza "clock_frequency_in" per averli ad una frequenza più bassa "clock_frequency_out"	7
entity counter_UpMod2n_Re_Sr	8
entity counter_UpN_Re_preset_Sr	9
entity counter_UpN_Re_Sr	10
entity display_7_segments	10
entity flag_FF	11
entity flipflop_d_risingEdge_asyncReset	
Flipflop_d_risingEdge_asyncReset implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono	12
entity flipflopmux	13
entity mux2_1	
Definisco il componente e la sua interfaccia	14
entity register_d_Re_Ar	
Registro di dimensione "dimension" che prende in ingresso un dato D e lo memorizza	15
entity scan_chain	16
entity uart	17
entity uart_onBoard	18
entity uart_rx	19
entity uart_rx_cu	20
entity uart_rx_po	21
entity uart_tx	22
entity uart_tx_cu	22
entity uart_tx_po	23

Chapter 2

File Index

2.1 File List

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

counter_UpMod2n_Re_Sr.vhd	
Contatore modulo 2 alla N	25
counter_UpN_preset_Re_Sr.vhd	
Contatore modulo N con preset	25
counter_UpN_Re_Sr.vhd	
Contatore modulo N	26
display_7_segments.vhd	
Componente che permette di pilotare le digit di un display a 7 segmenti	27
flag_FF.vhd	
Flip Flop che segnala lo stato dell'UART (buffer pieno o vuoto)	27
flipflopmux.vhd	
Flip Flop D con multiplexer	28
scan_chain.vhd	
Registro di N flip flop D multiplexati	28
uart.vhd	
Componente che permette la comunicazione asincrona tra 2 device	29
uart_onBoard.vhd	
Componente di alto livello per implementare l'UART su board	29
uart_rx.vhd	
Parte di ricezione dell'UART	30
uart_rx_cu.vhd	
Unità di controllo della parte di ricezione dell'UART	30
uart_rx_po.vhd	
Parte operativa della parte di ricezione dell'UART	31
uart_tx.vhd	
Parte di trasmissione dell'UART	31
uart_tx_cu.vhd	
Unità di controllo della parte di trasmissione dell'UART	32
uart_tx_po.vhd	
Parte operativa della parte di trasmissione dell'UART	32

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.

3.1.2 Member Data Documentation

3.1.2.1 IEEE

[IEEE](#) [Library]

FEDERICO II , CORSO DI ASE 18/19, Gruppo 14 –

3.1.2.2 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:

- `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.

3.2.2 Member Data Documentation

3.2.2.1 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:

- [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 counter_UpN_Re_preset_Sr Entity Reference

Libraries

- [IEEE](#)
architecture behavioral of counter_UpMod2n_Re_Sr end

Use Clauses

- [STD_LOGIC_1164](#)
- [numeric_std](#)
- [math_real](#)

Generics

- [n](#) **NATURAL** := **4**
- [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
- [load](#) in **STD_LOGIC**
- [preset](#) in **STD_LOGIC_VECTOR**((integer(ceil(log2(real(n))))- 1)downto 0)
- [count_hit](#) out **STD_LOGIC**
count_hit output
- [COUNTS](#) out **STD_LOGIC_VECTOR**((integer(ceil(log2(real(n))))- 1)downto 0)
COUNT output.

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

- [counter_UpN_preset_Re_Sr.vhd](#)

3.6 counter_UpN_Re_Sr Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [numeric_std](#)
- [math_real](#)

Generics

- [n](#) **NATURAL** := [2](#)
- [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**((integer(ceil(log2(real(n))))- [1](#))downto [0](#))
COUNT output.

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

- [counter_UpN_Re_Sr.vhd](#)

3.7 display_7_segments Entity Reference

Libraries

- [IEEE](#)

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.

3.7.1 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_segments.vhd](#)

3.8 flag_FF Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)

Ports

- **clock** in STD_LOGIC
- **reset** in STD_LOGIC
- **clr_flag** in STD_LOGIC
setto lo stato del buffer come vuoto
- **set_flag** in STD_LOGIC
setta lo stato del buffer come pieno
- **flag** out STD_LOGIC
segnala lo stato del buffer

3.8.1 Detailed Description

buffer per l'UART che segnala il suo stato: che può essere vuoto o pieno flag : 0 il registro è vuoto ci si può scrivere all'interno

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

- [flag_FF.vhd](#)

3.9 flipflop_d_risingEdge_asyncReset Entity Reference

[flipflop_d_risingEdge_asyncReset](#) 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_d_risingEdge_asyncReset input : segnale di clock per sincronizzare
- [enable](#) **in** **STD_LOGIC**
flipflop_d_risingEdge_asyncReset input : segnale enable
- [reset](#) **in** **STD_LOGIC**
flipflop_d_risingEdge_asyncReset input : segnale reset
- [d](#) **in** **STD_LOGIC**
flipflop_d_risingEdge_asyncReset input : input data
- [q](#) **out** **STD_LOGIC**
flipflop_d_risingEdge_asyncReset output : output data

3.9.1 Detailed Description

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

3.9.2 Member Data Documentation

3.9.2.1 IEEE

[IEEE](#) [Library]

FEDERICO II , CORSO DI ASE 18/19, Gruppo 14 –

3.9.2.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

last changes: <14/11/2018> <13/11/2018> <log> create

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

- [flipflop_d_risingEdge_asyncReset.vhd](#)

3.10 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

3.10.1 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](#)

3.11 mux2_1 Entity Reference

definisco il componente e la sua interfaccia

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)

Generics

- **width natural:= 1**
parallelismo dell' I/O del multiplexer

Ports

- **SEL in STD_LOGIC**
mux2_1 input: selezione
- **A in STD_LOGIC_VECTOR((width - 1)downto 0)**
mux2_1 input: A
- **B in STD_LOGIC_VECTOR((width - 1)downto 0)**
mux2_1 input: B
- **X out STD_LOGIC_VECTOR((width - 1)downto 0)**
mux2_1 output: X

3.11.1 Detailed Description

definisco il componente e la sua interfaccia

Descrizione Quando l'ingresso SEL è basso, l'uscita assume il valore del segnale A, altrimenti quando il segnale SEL è alto l'uscita assume il valore del segnale B.

3.11.2 Member Data Documentation

3.11.2.1 IEEE

[IEEE](#) [Library]

FEDERICO II , CORSO DI ASE 18/19, Gruppo 14 –

3.11.2.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

last changes: <14/11/2018> <13/11/2018> <log> create

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

- mux2_1.vhd

3.12 register_d_Re_Ar Entity Reference

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

Libraries

- [IEEE](#)
architecture dataflow of [mux2_1](#) end

Use Clauses

- [STD_LOGIC_1164](#)

Generics

- [dimension](#) **NATURAL** := 8
definisce il parallelismo del registro
- [reset_level](#) **STD_LOGIC** := ' 0 '
definisce il livello reset
- [enable_level](#) **STD_LOGIC** := ' 1 '
definisce il livello enable

Ports

- **clock** in STD_LOGIC
register_d_Re_Ar input : segnale di clock per sincronizzare
- **enable** 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

3.12.1 Detailed Description

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

3.12.2 Member Data Documentation

3.12.2.1 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

3.13 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

3.13.1 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](#)

3.14 uart Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [numeric_std](#)
- [math_real](#)

Generics

- `data_bits` **NATURAL:= 8**

Ports

- `clock` in **STD_LOGIC**
- `reset` in **STD_LOGIC**
- `rx` in **STD_LOGIC**
- `rd_uart` in **STD_LOGIC**
- `wr_uart` in **STD_LOGIC**
 - se alto segnala al buffer in ingresso che il dato è pronto per essere inviato*
- `din` in **STD_LOGIC_VECTOR(data_bits- 1 downto 0)**
 - byte da inviare*
- `tx out` **STD_LOGIC**
- `rx_empty` out **STD_LOGIC**
 - se è il buffer in uscita è vuoto vale 1*
- `tx_full` out **STD_LOGIC**
 - se è il buffer in ingresso è pieno vale 1*
- `dout` out **STD_LOGIC_VECTOR(data_bits- 1 downto 0)**

3.14.1 Member Data Documentation

3.14.1.1 rd_uart

`rd_uart` in **STD_LOGIC** [Port]

segnale se alto segnala al buffer in uscita che il dato è stato consumato

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

- `uart.vhd`

3.15 uart_onBoard Entity Reference

Libraries

- **IEEE**

Use Clauses

- **STD_LOGIC_1164**
- **numeric_std**
- **math_real**

Ports

- `clock` in STD_LOGIC
- `rx` in STD_LOGIC
- `rx_empty` out STD_LOGIC
se è vuoto vale 1
- `tx` out STD_LOGIC
- `tx_full` out STD_LOGIC
- `wr_uart` in STD_LOGIC
- `din` in STD_LOGIC_VECTOR(7 downto 0)
- `anodes` out STD_LOGIC_VECTOR(7 downto 0)
- `cathodes` out STD_LOGIC_VECTOR(7 downto 0)

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

- `uart_onBoard.vhd`

3.16 uart_rx Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164
- NUMERIC_STD
- STD_LOGIC_ARITH
- STD_LOGIC_UNSIGNED

Generics

- `data_bits` NATURAL:= 8
Numero di bit dati da ricevere.
- `ticks` NATURAL:= 16
numero di bit in cui divido l'intervallo di ricezione
- `start_Ticks` NATURAL:= 7
numero di tick per posizionarmi al centro dell'intervallo di trasmissione
- `stop_Ticks` NATURAL:= 16
Numero di conteggi per determinare la fine della trasmissione.
- `n` NATURAL:= 651

Ports

- [clock](#) in STD_LOGIC
- [enable](#) in STD_LOGIC
enable 1-attivo
- [reset](#) in STD_LOGIC
reset 1-attivo
- [rx](#) in STD_LOGIC
bit ricevuto
- [rd_uart](#) in STD_LOGIC
lettura del byte ricevuto
- [rx_empty](#) out STD_LOGIC
va basso quando è stato ricevuto un byte
- [dout](#) out STD_LOGIC_VECTOR([data_bits](#) - 1 downto 0)
byte ricevuto

3.16.1 Detailed Description

parte di ricezione dell'UART PC e PO unico blocco versione digiland/ libro

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

- [uart_rx.vhd](#)

3.17 uart_rx_cu Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [NUMERIC_STD](#)
- [STD_LOGIC_ARITH](#)
- [STD_LOGIC_UNSIGNED](#)

Ports

- [clock](#) in STD_LOGIC
- [reset](#) in STD_LOGIC
- [tick_hit](#) in STD_LOGIC
segnale che indica il raggiungimento dei 16 conteggi del baud gen, indica che si deve campionare
- [stop_hit](#) in STD_LOGIC
segnale che indica il raggiungimento dei numero di tick di stop che indica la fine della trasmissione
- [received_byte_hit](#) in STD_LOGIC
- [rx](#) in STD_LOGIC
- [rx_done](#) out STD_LOGIC
- [increase_n_bits](#) out STD_LOGIC
- [load_tick](#) out STD_LOGIC
- [read_rx](#) out STD_LOGIC
collegato a scan_en della scan chain
- [enable_stop](#) out STD_LOGIC
- [reset_tick](#) out STD_LOGIC
- [reset_out](#) out STD_LOGIC

3.17.1 Detailed Description

parte di ricezione dell'UART PC e PO unico blocco versione digiland/ libro

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

- [uart_rx_cu.vhd](#)

3.18 uart_rx_po Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [numeric_std](#)
- [math_real](#)

Generics

- [data_bits](#) **NATURAL:= 8**
Numero di bit dati.
- [ticks](#) **NATURAL:= 16**
- [start_Ticks](#) **NATURAL:= 7**
- [stop_Ticks](#) **NATURAL:= 16**
Numero di conteggi per determinare la fine della trasmissione.
- [N](#) **NATURAL:= 651**

Ports

- [clock](#) in STD_LOGIC
- [reset](#) in STD_LOGIC
- [load_tick](#) in STD_LOGIC
- [enable](#) in STD_LOGIC
- [enable_stop](#) in STD_LOGIC
- [received_bit](#) in STD_LOGIC
- [read_rx](#) in STD_LOGIC
- [rx](#) in STD_LOGIC
- [rx_done](#) in STD_LOGIC
- [rd_uart](#) in STD_LOGIC
- [reset_tick](#) in STD_LOGIC
- [rx_out](#) out STD_LOGIC
- [rx_empty](#) out STD_LOGIC
- [received_byte_hit](#) out STD_LOGIC
- [tick_hit](#) out STD_LOGIC
- [stop_hit](#) out STD_LOGIC
- [received_byte](#) out STD_LOGIC_VECTOR(([data_bits](#) - 1)downto 0)

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

- [uart_rx_po.vhd](#)

3.19 uart_tx Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [NUMERIC_STD](#)
- [STD_LOGIC_ARITH](#)
- [STD_LOGIC_UNSIGNED](#)

Generics

- [data_bits](#) **NATURAL:= 8**
Numero di bit dati.
- [ticks](#) **NATURAL:= 16**
- [start_Ticks](#) **NATURAL:= 7**
- [stop_Ticks](#) **NATURAL:= 16**
Numero di conteggi per determinare la fine della trasmissione.

Ports

- [clock](#) in **STD_LOGIC**
- [reset](#) in **STD_LOGIC**
- [enable](#) in **STD_LOGIC**
- [wr_uart](#) in **STD_LOGIC**
- [din](#) in **STD_LOGIC_VECTOR(data_bits - 1 downto 0)**
- [tx_full](#) out **STD_LOGIC**
- [tx](#) out **STD_LOGIC**

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

- [uart_tx.vhd](#)

3.20 uart_tx_cu Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [NUMERIC_STD](#)
- [STD_LOGIC_ARITH](#)
- [STD_LOGIC_UNSIGNED](#)

Ports

- [clock](#) in STD_LOGIC
- [reset](#) in STD_LOGIC
- [tx_start](#) in STD_LOGIC
- [tick_hit](#) in STD_LOGIC
segnale dal bound rate gen
- [stop_hit](#) in STD_LOGIC
- [sended_byte_hit](#) in STD_LOGIC
- [bit_to_send](#) in STD_LOGIC
- [increase_n_bits](#) out STD_LOGIC
- [send_tx](#) out STD_LOGIC
- [load_din](#) out STD_LOGIC
0 attivo carica nella scan chain din
- [reset_out](#) out STD_LOGIC
- [tx_out](#) out STD_LOGIC
- [reset_tick](#) out STD_LOGIC
- [tx_done](#) out STD_LOGIC

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

- [uart_tx_cu.vhd](#)

3.21 uart_tx_po Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [numeric_std](#)
- [math_real](#)

Generics

- [data_bits](#) **NATURAL:= 8**
Numero di bit dati.
- [ticks](#) **NATURAL:= 16**
- [start_Ticks](#) **NATURAL:= 7**
- [stop_Ticks](#) **NATURAL:= 16**
Numero di conteggi per determinare la fine della trasmissione.

Ports

- `clock` in STD_LOGIC
- `reset` in STD_LOGIC
- `enable` in STD_LOGIC
- `sended_bit` in STD_LOGIC
- `send_tx` in STD_LOGIC
- `tx_done` in STD_LOGIC
- `wr_uart` in STD_LOGIC
- `load_din` in STD_LOGIC
- `tx_in` in STD_LOGIC
- `reset_tick` in STD_LOGIC
- `byte_to_send` in STD_LOGIC_VECTOR((`data_bits` - 1) downto 0)
- `bit_to_send` out STD_LOGIC
- `tx` out STD_LOGIC
- `tx_full` out STD_LOGIC
- `sended_byte_hit` out STD_LOGIC
- `tick_hit` out STD_LOGIC
- `stop_hit` out STD_LOGIC

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

- `uart_tx_po.vhd`

Chapter 4

File Documentation

4.1 counter_UpMod2n_Re_Sr.vhd File Reference

Contatore modulo 2 alla N.

Entities

- [counter_UpMod2n_Re_Sr](#) entity

4.1.1 Detailed Description

Contatore modulo 2 alla N.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.2 counter_UpN_preset_Re_Sr.vhd File Reference

Contatore modulo N con preset.

Entities

- [counter_UpN_Re_preset_Sr](#) entity

4.2.1 Detailed Description

Contatore modulo N con preset.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.3 counter_UpN_Re_Sr.vhd File Reference

Contatore modulo N.

Entities

- [counter_UpN_Re_Sr](#) entity

4.3.1 Detailed Description

Contatore modulo N.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.4 display_7_segments.vhd File Reference

Componente che permette di pilotare le digit di un display a 7 segmenti.

Entities

- [display_7_segments](#) entity

4.4.1 Detailed Description

Componente che permette di pilotare le digit di un display a 7 segmenti.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.5 flag_FF.vhd File Reference

Flip Flop che segnala lo stato dell'UART (buffer pieno o vuoto)

Entities

- [flag_FF](#) entity

4.5.1 Detailed Description

Flip Flop che segnala lo stato dell'UART (buffer pieno o vuoto)

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.6 flipflopmux.vhd File Reference

Flip Flop D con multiplexer.

Entities

- [flipflopmux](#) entity

4.6.1 Detailed Description

Flip Flop D con multiplexer.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.7 scan_chain.vhd File Reference

Registro di N flip flop D multiplexati.

Entities

- [scan_chain](#) entity

4.7.1 Detailed Description

Registro di N flip flop D multiplexati.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.8 uart.vhd File Reference

Componente che permette la comunicazione asincrona tra 2 device.

Entities

- [uart](#) entity

4.8.1 Detailed Description

Componente che permette la comunicazione asincrona tra 2 device.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.9 uart_onBoard.vhd File Reference

Componente di alto livello per implementare l'UART su board.

Entities

- [uart_onBoard](#) entity

4.9.1 Detailed Description

Componente di alto livello per implementare l'UART su board.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.10 `uart_rx.vhd` File Reference

Parte di ricezione dell'UART.

Entities

- `uart_rx` entity

4.10.1 Detailed Description

Parte di ricezione dell'UART.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.11 `uart_rx_cu.vhd` File Reference

Unità di controllo della parte di ricezione dell'UART.

Entities

- `uart_rx_cu` entity

4.11.1 Detailed Description

Unità di controllo della parte di ricezione dell'UART.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.12 uart_rx_po.vhd File Reference

Parte operativa della parte di ricezione dell'UART.

Entities

- [uart_rx_po](#) entity

4.12.1 Detailed Description

Parte operativa della parte di ricezione dell'UART.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.13 uart_tx.vhd File Reference

Parte di trasmissione dell'UART.

Entities

- [uart_tx](#) entity

4.13.1 Detailed Description

Parte di trasmissione dell'UART.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.14 `uart_tx_cu.vhd` File Reference

Unità di controllo della parte di trasmissione dell'UART.

Entities

- [uart_tx_cu](#) entity

4.14.1 Detailed Description

Unità di controllo della parte di trasmissione dell'UART.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.15 `uart_tx_po.vhd` File Reference

Parte operativa della parte di trasmissione dell'UART.

Entities

- [uart_tx_po](#) entity

4.15.1 Detailed Description

Parte operativa della parte di trasmissione dell'UART.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

Index

- anodes_manager, [5](#)
 - IEEE, [5](#)
 - STD_LOGIC_1164, [6](#)
- cathodes_manager, [6](#)
 - STD_LOGIC_1164, [7](#)
- clock_divisor, [7](#)
 - STD_LOGIC_1164, [8](#)
- counter_UpMod2n_Re_Sr, [8](#)
- counter_UpMod2n_Re_Sr.vhd, [25](#)
- counter_UpN_Re_Sr, [10](#)
- counter_UpN_Re_Sr.vhd, [26](#)
- counter_UpN_Re_preset_Sr, [9](#)
- counter_UpN_preset_Re_Sr.vhd, [25](#)
- display_7_segments, [10](#)
- display_7_segments.vhd, [27](#)
- flag_FF.vhd, [27](#)
- flag_FF, [11](#)
- flipflop_d_risingEdge_asyncReset, [12](#)
 - IEEE, [13](#)
 - STD_LOGIC_1164, [13](#)
- flipflopmux, [13](#)
- flipflopmux.vhd, [28](#)
- IEEE
 - anodes_manager, [5](#)
 - flipflop_d_risingEdge_asyncReset, [13](#)
 - mux2_1, [15](#)
- mux2_1, [14](#)
 - IEEE, [15](#)
 - STD_LOGIC_1164, [15](#)
- rd_uart
 - uart, [18](#)
- register_d_Re_Ar, [15](#)
 - STD_LOGIC_1164, [16](#)
- STD_LOGIC_1164
 - anodes_manager, [6](#)
 - cathodes_manager, [7](#)
 - clock_divisor, [8](#)
 - flipflop_d_risingEdge_asyncReset, [13](#)
 - mux2_1, [15](#)
 - register_d_Re_Ar, [16](#)
- scan_chain, [16](#)
- scan_chain.vhd, [28](#)
- uart, [17](#)
 - rd_uart, [18](#)
- uart.vhd, [29](#)
- uart_onBoard, [18](#)
- uart_onBoard.vhd, [29](#)
- uart_rx, [19](#)
- uart_rx.vhd, [30](#)
- uart_rx_cu, [20](#)
- uart_rx_cu.vhd, [30](#)
- uart_rx_po, [21](#)
- uart_rx_po.vhd, [31](#)
- uart_tx, [22](#)
- uart_tx.vhd, [31](#)
- uart_tx_cu, [22](#)
- uart_tx_cu.vhd, [32](#)
- uart_tx_po, [23](#)
- uart_tx_po.vhd, [32](#)