UART

Corso di ASE anno 18/19

Gruppo 14
PREVITERA GABRIELE
PENNONE MIRKO
PENNA SIMONE

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

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_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:=** 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

3.6 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.

3.6.1 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

3.7 io_buffer Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Generics

- width **NATURAL:= 8**

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
- din **in STD_LOGIC_VECTOR((width- 1)downto 0)**
- flag **out STD_LOGIC**

  segnala lo stato del buffer
- dout **out STD_LOGIC_VECTOR((width- 1)downto 0)**

3.7.1 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:

- io_buffer.vhd

3.8 uart Entity Reference

RICEZIONE quando uart_rx riceve il dato, lo mette nell'output input e alza rx_done_int. Tale segnale pilota il set_↩ flag del buffer, segnalando che il buffer è pieno e il dato può essere consumato. Per leggere si aspetta dunque che rx_empty sia basso, ossia che rx_full sia alto: tale segnale è pilotato dal flag del buffer in uscita, alto solo quando è stato settato con rx_done (ricezione completata). Dopodiché si setta rd_uart a 1, il quale pilota clr_flag del buffer in uscita segnalando che il dato non è più consumabile (flag basso), e si preleva il dato su d_out.

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**

  se alto segnala al buffer in uscita che il dato è stato consumato
- 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 alto il buffer in uscita è vuoto
- tx_full **out STD_LOGIC**

  se alto il buffer in ingresso è pieno
- dout **out STD_LOGIC_VECTOR(data_bits- 1 downto 0)**

  byte ricevuto

3.8.1 Detailed Description

RICEZIONE quando uart_rx riceve il dato, lo mette nell'output input e alza rx_done_int. Tale segnale pilota il set_↩ flag del buffer, segnalando che il buffer è pieno e il dato può essere consumato. Per leggere si aspetta dunque che rx_empty sia basso, ossia che rx_full sia alto: tale segnale è pilotato dal flag del buffer in uscita, alto solo quando è stato settato con rx_done (ricezione completata). Dopodiché si setta rd_uart a 1, il quale pilota clr_flag del buffer in uscita segnalando che il dato non è più consumabile (flag basso), e si preleva il dato su d_out.

top level entity che utilizza un trasmettitore e ricevitore con opportuni buffer di input e output è in grado di effettuare trasmissione e ricezione

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

- uart.vhd

3.9 uart_onBoard Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Ports

- clock **in STD_LOGIC**
- rx **in STD_LOGIC**
- rx_empty **out STD_LOGIC**

  segnala se il buffer in uscita è vuoto
- tx **out STD_LOGIC**
- tx_full **out STD_LOGIC**

  segnala se il buffer in ingresso è pieno
- anodes **out STD_LOGIC_VECTOR(7 downto 0)**
- cathodes **out STD_LOGIC_VECTOR(7 downto 0)**

3.9.1 Detailed Description

per testare l'uart su board, si è istanziato un'entità top level UART: i valori da trasmettere sono comunicati tramite gli switch della board all'uart. i valori vengono ricevuti dallo stesso uart su rx e riportati sul display a 7 segmenti

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

- uart_onBoard.vhd

3.10 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.
- stop_Ticks **NATURAL:= 16**

  Numero di conteggi per determinare la fine della trasmissione.

Ports

- clock **in STD_LOGIC**
- reset **in STD_LOGIC**
- rx **in STD_LOGIC**

  linea di ricezione
- tick **in STD_LOGIC**

  segnale dal baud rate gen
- rx_done **out STD_LOGIC**

  va alto quando è stato ricevuto un byte
- dout **out STD_LOGIC_VECTOR(data_bits - 1 downto 0)**

  byte ricevuto

3.10.1 Detailed Description

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

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

- uart_rx.vhd

3.11 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 per ogni trasmissione (tra un mark e l'altro)
- stop_ticks **NATURAL:= 16**

  numero di tick da lasciar passare alla fine della trasmissione di ogni byte

Ports

- clock **in STD_LOGIC**
- reset **in STD_LOGIC**
- tx_start **in STD_LOGIC**

  alto quando deve partire la trasmissione
- tick **in STD_LOGIC**

  baud rate
- din **in STD_LOGIC_VECTOR(data_bits - 1 downto 0)**

  byte da trasmettere
- tx_done **out STD_LOGIC**

  alto quando la trasmissione è completata
- tx **out STD_LOGIC**

  linea di trasmissione

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

- uart_tx.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_Re_Sr.vhd File Reference

Contatore modulo N.

Entities

- counter_UpN_Re_Sr entity

4.2.1 Detailed Description

Contatore modulo N.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.3 display_7_segments.vhd File Reference

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

Entities

- display_7_segments entity

4.3.1 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

4.4 io_buffer.vhd File Reference

buffer utilizzato nell'UART in fase di trasmissione e ricezione

Entities

- io_buffer entity

4.4.1 Detailed Description

buffer utilizzato nell'UART in fase di trasmissione e ricezione

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.5 uart_onBoard.vhd File Reference

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

Entities

- uart_onBoard entity

4.5.1 Detailed Description

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

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.6 uart_rx.vhd File Reference

Parte di ricezione dell'UART.

Entities

- uart_rx entity

4.6.1 Detailed Description

Parte di ricezione dell'UART.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.7 uart_tx.vhd File Reference

Parte di trasmissione dell'UART.

Entities

- uart_tx entity

4.7.1 Detailed Description

Parte di trasmissione dell'UART.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

Index