

mic1_fpga

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	ALU Entity Reference	5
3.1.1	Member Data Documentation	5
3.1.1.1	IEEE	6
3.1.1.2	STD_LOGIC_1164	6
3.2	anodes_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	baud_gen Entity Reference	7
3.3.1	Member Data Documentation	7
3.3.1.1	STD_LOGIC_1164	8
3.4	boot_rom Entity Reference	8
3.4.1	Detailed Description	8
3.4.2	Member Data Documentation	8
3.4.2.1	ieee	9
3.4.2.2	std_logic_1164	9
3.5	cathodes_manager Entity Reference	9

3.5.1	Detailed Description	10
3.5.2	Member Data Documentation	10
3.5.2.1	STD_LOGIC_1164	10
3.6	clk_wiz_v3_6 Entity Reference	10
3.7	clock_div Entity Reference	11
3.7.1	Member Data Documentation	11
3.7.1.1	ieee	11
3.8	clock_divisor Entity Reference	12
3.8.1	Detailed Description	12
3.8.2	Member Data Documentation	13
3.8.2.1	IEEE	13
3.8.2.2	STD_LOGIC_1164	13
3.9	control_store Entity Reference	13
3.10	counter_UpMod2n_Re_Sr Entity Reference	14
3.10.1	Detailed Description	14
3.11	cpool_rom Entity Reference	14
3.11.1	Detailed Description	15
3.11.2	Member Data Documentation	15
3.11.2.1	ieee	15
3.11.2.2	std_logic_1164	15
3.12	decoder Entity Reference	15
3.12.1	Member Data Documentation	16
3.12.1.1	IEEE	16
3.12.1.2	STD_LOGIC_1164	16
3.13	decodifica_indirizzi Entity Reference	16
3.13.1	Member Data Documentation	17
3.13.1.1	STD_LOGIC_1164	17
3.14	dff Entity Reference	17
3.14.1	Member Data Documentation	17
3.14.1.1	IEEE	18

3.14.1.2	STD_LOGIC_1164	18
3.15	display_7_segments Entity Reference	18
3.15.1	Detailed Description	19
3.16	divisor_freq Entity Reference	19
3.16.1	Detailed Description	19
3.16.2	Member Data Documentation	19
3.16.2.1	IEEE	19
3.16.2.2	STD_LOGIC_1164	20
3.17	high_bit Entity Reference	20
3.17.1	Member Data Documentation	20
3.17.1.1	IEEE	20
3.17.1.2	STD_LOGIC_1164	21
3.18	if_uart Entity Reference	21
3.18.1	Member Data Documentation	21
3.18.1.1	IEEE	22
3.18.1.2	STD_LOGIC_1164	22
3.19	io_controller Entity Reference	22
3.20	io_switch_led_display Entity Reference	23
3.21	mbr_register Entity Reference	24
3.21.1	Member Data Documentation	24
3.21.1.1	IEEE	24
3.21.1.2	STD_LOGIC_1164	24
3.22	mdr_register Entity Reference	25
3.22.1	Member Data Documentation	25
3.22.1.1	STD_LOGIC_1164	25
3.23	mic1 Entity Reference	25
3.23.1	Member Data Documentation	26
3.23.1.1	IEEE	26
3.23.1.2	STD_LOGIC_1164	26
3.24	mpc Entity Reference	26

3.24.1	Member Data Documentation	27
3.24.1.1	IEEE	27
3.24.1.2	STD_LOGIC_1164	27
3.25	neg_edge_reg Entity Reference	27
3.25.1	Member Data Documentation	28
3.25.1.1	IEEE	28
3.25.1.2	STD_LOGIC_1164	28
3.26	negedge_dff Entity Reference	28
3.26.1	Member Data Documentation	29
3.26.1.1	IEEE	29
3.26.1.2	STD_LOGIC_1164	29
3.27	oring Entity Reference	29
3.27.1	Member Data Documentation	30
3.27.1.1	IEEE	30
3.27.1.2	STD_LOGIC_1164	30
3.28	parte_di_controllo Entity Reference	30
3.28.1	Member Data Documentation	32
3.28.1.1	eob_mdr	32
3.28.1.2	f0	32
3.28.1.3	h	32
3.28.1.4	IEEE	32
3.28.1.5	sra1	33
3.28.1.6	wwrite	33
3.29	parte_operativa Entity Reference	33
3.29.1	Member Data Documentation	34
3.29.1.1	IEEE	34
3.29.1.2	STD_LOGIC_1164	34
3.30	posedge_reg_enable Entity Reference	35
3.30.1	Member Data Documentation	35
3.30.1.1	IEEE	35

3.30.1.2	STD_LOGIC_1164	35
3.31	ram Entity Reference	36
3.31.1	Member Data Documentation	36
3.31.1.1	IEEE	36
3.31.1.2	STD_LOGIC_1164	36
3.32	shifter Entity Reference	37
3.32.1	Member Data Documentation	37
3.32.1.1	IEEE	37
3.32.1.2	STD_LOGIC_1164	37
3.33	sistema_mic1 Entity Reference	37
3.33.1	Member Data Documentation	38
3.33.1.1	IEEE	38
3.33.1.2	STD_LOGIC_1164	38
3.34	tb_if_uart Entity Reference	39
3.35	tb_ram Entity Reference	39
3.35.1	Member Data Documentation	39
3.35.1.1	IEEE	39
3.35.1.2	STD_LOGIC_1164	40
3.36	TB_SISTEMA_COMPLETO Entity Reference	40
3.37	tb_uart_vhd Entity Reference	40
3.37.1	Member Data Documentation	40
3.37.1.1	ieee	41
3.37.1.2	std_logic_1164	41
3.38	uart_9600bps Entity Reference	41
3.38.1	Member Data Documentation	42
3.38.1.1	IEEE	42
3.38.1.2	STD_LOGIC_1164	42
3.39	uart_rx Entity Reference	42
3.39.1	Member Data Documentation	43
3.39.1.1	IEEE	43
3.39.1.2	STD_LOGIC_1164	43
3.40	uart_serial Entity Reference	43
3.40.1	Member Data Documentation	44
3.40.1.1	baud_sel	44
3.40.1.2	ieee	44
3.40.1.3	rx_data_serial	45
3.40.1.4	std_logic_1164	45
3.40.1.5	tx_data_serial	45
3.40.1.6	UNISIM	45
3.41	uart_tx Entity Reference	45
3.41.1	Member Data Documentation	46
3.41.1.1	IEEE	46
3.41.1.2	STD_LOGIC_1164	46

4	File Documentation	47
4.1	counter_UpMod2n_Re_Sr.vhd File Reference	47
4.1.1	Detailed Description	47
4.2	display_7_segmenti.vhd File Reference	47
4.2.1	Detailed Description	48
4.3	io_controller.vhd File Reference	48
4.3.1	Detailed Description	48
4.4	io_switch_led.vhd File Reference	49
4.4.1	Detailed Description	49
Index		51

Chapter 1

Class Index

1.1 Class List

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

entity ALU	5
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	6
entity baud_gen	7
entity boot_rom Use work.memory.all;	8
entity cathodes_manager	9
entity clk_wiz_v3_6	10
entity clock_div	11
entity clock_divisor Filtra i fronti del clock ad una frequenza "clock_frequency_in" per averli ad una frequenza più bassa "clock_frequency_out"	12
entity control_store	13
entity counter_UpMod2n_Re_Sr	14
entity cpool_rom Use work.memory.all;	14
entity decoder	15
entity decodifica_indirizzi	16
entity dff	17
entity display_7_segments	18
entity divisore_freq	19
entity high_bit	20
entity if_uart	21
entity io_controller	22
entity io_switch_led_display	23
entity mbr_register	24
entity mdr_register	25
entity mic1	25
entity mpc	26
entity neg_edge_reg	27
entity negedge_dff	28
entity oring	29
entity parte_di_controllo	30

entity parte_operativa	33
entity posedge_reg_enable	35
entity ram	36
entity shifter	37
entity sistema_mic1	37
entity tb_if_uart	39
entity tb_ram	39
entity TB_SISTEMA_COMPLETO	40
entity tb_uart_vhd	40
entity uart_9600bps	41
entity uart_rx	42
entity uart_serial	43
entity uart_tx	45

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	47
display_7_segmenti.vhd	
Componente che permette di pilotare le digit di un display a 7 segmenti	47
io_controller.vhd	
Componente che sceglie tra UART e Switch/Led/Display	48
io_switch_led.vhd	
Componente che gestisce il collegamento tra switch (input) e led (output)	49

Chapter 3

Class Documentation

3.1 ALU Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- [A](#) in `std_logic_vector(31 downto 0)`
- [B](#) in `std_logic_vector(31 downto 0)`
- [INVA](#) in `std_logic`
- [ENA](#) in `std_logic`
- [ENB](#) in `std_logic`
- [F0](#) in `std_logic`
- [F1](#) in `std_logic`
- [INC](#) in `std_logic`
- [Output](#) inout `std_logic_vector(31 downto 0)`
- [N](#) out `std_logic:= ' 0 '`
- [Z](#) out `std_logic:= ' 0 '`

3.1.1 Member Data Documentation

3.1.1.1 IEEE

[IEEE](#) [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 18:57:07 01/06/2008 Design Name: Module Name: alu - Behavioral Project Name: Target Devices: Tool versions: Description:

3.1.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [alu.vhd](#)

3.2 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](#)
calcola flag di zero

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

3.2.1 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.

3.2.2 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:

- [anodes_manager.vhd](#)

3.3 baud_gen Entity Reference

Libraries

- [IEEE](#)
architecture dataflow of [anodes_manager](#) end
- [ieee](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [STD_LOGIC_ARITH](#)
- [STD_LOGIC_UNSIGNED](#)
- [std_logic_1164](#)
- [numeric_std](#)

Generics

- **N integer:= 4**
number of bits
- **M integer:= 163**
mod 4

Ports

- [ck](#) in [std_logic](#)
- [max_tick](#) out [std_logic](#)
- [q](#) out [std_logic_vector](#)(**N - 1** downto **0**)

3.3.1 Member Data Documentation

3.3.1.1 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- `baud_rate_gen.vhd`

3.4 boot_rom Entity Reference

`use work.memory.all;`

Libraries

- [ieee](#)
- [work](#)

Use Clauses

- [std_logic_1164](#)
- [std_logic_arith](#)
- [std_logic_unsigned](#)
- [numeric_std](#)

Ports

- [ADDRESS](#) in `std_logic_vector(8 downto 0)`
- [CE](#) in `std_logic`
- [DATA](#) out `std_logic_vector(7 downto 0)`

3.4.1 Detailed Description

`use work.memory.all;`

3.4.2 Member Data Documentation

3.4.2.1 ieee

ieee [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 17:14:53 08/4/2007 Design Name: Module Name: - Behavioral Project Name: Target Devices: Tool versions: Description:

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 17:14:53 08/4/2007 Design Name: Module Name: - Behavioral Project Name: Target Devices: Tool versions: Description: Implementa la control store di MIC1 fatta da due banchi di 256 locazioni di byte

3.4.2.2 std_logic_1164

std_logic_1164 [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- rom8.vhd
- rom8_prod.vhd

3.5 cathodes_manager Entity Reference

Libraries

- IEEE
output logic

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

3.5.1 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.

3.5.2 Member Data Documentation

3.5.2.1 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](#)

3.6 clk_wiz_v3_6 Entity Reference

Libraries

- [ieee](#)
architecture behavioral of [cathodes_manager](#) end
- [unisim](#)

Use Clauses

- [std_logic_1164](#)
_____primary_____100.000_____0.010
- [std_logic_unsigned](#)
- [std_logic_arith](#)
- [numeric_std](#)
- [vcomponents](#)

Ports

- [CLK_IN1](#) in [std_logic](#)
Clock in ports.
- [CLK_OUT1](#) out [std_logic](#)
Clock out ports.
- [RESET](#) in [std_logic](#)
Status and control signals.
- [LOCKED](#) out [std_logic](#)

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

- [clk_wiz_v3_6.vhd](#)

3.7 clock_div Entity Reference

Libraries

- [ieee](#)
- [unisim](#)

Use Clauses

- [std_logic_1164](#)

$$\text{primary} \quad 100.000 \quad 0.010$$
- [std_logic_unsigned](#)
- [std_logic_arith](#)
- [numeric_std](#)
- [vcomponents](#)

Ports

- [CLK_IN1](#) in [std_logic](#)
Clock in ports.
- [CLK_OUT1](#) out [std_logic](#)
Clock out ports.

3.7.1 Member Data Documentation

3.7.1.1 ieee

[ieee](#) [Library]

file: clock_div.vhd

(c) Copyright 2008 - 2011 Xilinx, Inc. All rights reserved.

This file contains confidential and proprietary information of Xilinx, Inc. and is protected under U.S. and international copyright and other intellectual property laws.

DISCLAIMER This disclaimer is not a license and does not grant any rights to the materials distributed herewith. Except as otherwise provided in a valid license issued to you by Xilinx, and to the maximum extent permitted by applicable law: (1) THESE MATERIALS ARE MADE AVAILABLE "AS IS" AND WITH ALL FAULTS, AND XILINX HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under or in connection with these materials, including for any direct, or any indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same.

CRITICAL APPLICATIONS Xilinx products are not designed or intended to be fail- safe, or for use in any application requiring fail-safe performance, such as life-support or safety devices or systems, Class III medical devices, nuclear facilities, applications related to the deployment of airbags, or any other applications that could lead to death, personal injury, or severe property or environmental damage (individually and collectively, "Critical Applications"). Customer assumes the sole risk and liability of any use of Xilinx products in Critical Applications, subject only to applicable laws and regulations governing limitations on product liability.

THIS COPYRIGHT NOTICE AND DISCLAIMER MUST BE RETAINED AS PART OF THIS FILE AT ALL TIMES.

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

- `clock_div.vhd`

3.8 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](#)

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

3.8.1 Detailed Description

Filtra i fronti del clock ad una frequenza "clock_frequency_in" per averli ad una frequenza più bassa "clock_frequency_out".

3.8.2 Member Data Documentation

3.8.2.1 IEEE

[IEEE](#) [Library]

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

3.8.2.2 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:

- clock_divisor.vhd

3.9 control_store Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- [addr](#) in [std_logic_vector](#)([8](#) downto [0](#))
- [mir](#) out [microinstruction:=microistruzione_tutti_zeri](#)

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

- control_store.vhd

3.10 counter_UpMod2n_Re_Sr Entity Reference

Libraries

- [IEEE](#)

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.

3.10.1 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](#)

3.11 cpool_rom Entity Reference

use work.memory.all;

Libraries

- [ieee](#)
architecture behavioral of [clock_divisor](#) end
- [work](#)

Use Clauses

- [std_logic_1164](#)
- [std_logic_arith](#)
- [std_logic_unsigned](#)

Ports

- [ADDRESS](#) in [std_logic_vector](#)([4](#) downto [0](#))
- [DATA](#) inout [std_logic_vector](#)([31](#) downto [0](#))
- [CE](#) in [std_logic](#)

3.11.1 Detailed Description

use work.memory.all;

3.11.2 Member Data Documentation

3.11.2.1 ieee

[ieee](#) [Library]

architecture behavioral of [clock_divisor](#) end

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 17:14:53 08/4/2007 Design Name: Module Name: - Behavioral Project Name: Target Devices: Tool versions: Description:

3.11.2.2 std_logic_1164

[std_logic_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- constant_pool_prod.vhd
- constant_pool_rom.vhd

3.12 decoder Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- [output](#) [out](#) [std_logic_vector](#)([15](#) [downto](#) [0](#))
- [sel](#) [in](#) [std_logic_vector](#)([3](#) [downto](#) [0](#))

3.12.1 Member Data Documentation

3.12.1.1 IEEE

[IEEE](#) [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 20:43:12 01/06/2008 Design Name: Module Name: 4to16decoder - Behavioral Project Name: Target Devices: Tool versions: Description:

3.12.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- 4to16decoder.vhd

3.13 decodifica_indirizzi Entity Reference

Libraries

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

Use Clauses

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

Ports

- `address` in `std_logic_vector(31 downto 0)`
- `ce_uart` out `std_logic:= ' 0 '`
- `ce_rom` out `std_logic:= ' 0 '`
- `ce_ram` out `std_logic:= ' 0 '`

3.13.1 Member Data Documentation

3.13.1.1 STD_LOGIC_1164

`STD_LOGIC_1164` [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- `decodifica_indirizzi.vhd`

3.14 dff Entity Reference

Libraries

- `IEEE`

Use Clauses

- `STD_LOGIC_1164`
- `STD_LOGIC_ARITH`
- `STD_LOGIC_UNSIGNED`

Ports

- `d` in `std_logic`
dato
- `ck` in `std_logic`
clock
- `q` out `std_logic:= ' 0 '`
uscita

3.14.1 Member Data Documentation

3.14.1.1 IEEE

IEEE [Library]

Company: Engineer:

Create Date: 20:30:50 01/06/2008 Design Name: Module Name: dff - Behavioral Project Name: Target Devices:
Tool versions: Description:

3.14.1.2 STD_LOGIC_1164

STD_LOGIC_1164 [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- dff.vhd

3.15 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.

3.15.1 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](#)

3.16 divisore_freq Entity Reference

Libraries

- [IEEE](#)
- [UNISIM](#)

Use Clauses

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

Ports

- [clk_div](#) out std_logic
- [clk](#) in std_logic

3.16.1 Detailed Description

DCM_SP: Digital Clock Manager Circuit Spartan-3E/3A Xilinx HDL Language Template, version 9.2i

3.16.2 Member Data Documentation

3.16.2.1 IEEE

[IEEE](#) [Library]

Company: Engineer:

Create Date: 17:43:28 03/20/2009 Design Name: Module Name: main - Behavioral Project Name: Target Devices:
Tool versions: Description:

3.16.2.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [divisore.vhd](#)

3.17 high_bit Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- [high_bit_out](#) out std_logic:= '0'
- [ff_n](#) in std_logic
- [ff_z](#) in std_logic
- [jamn](#) in std_logic
- [jamz](#) in std_logic

3.17.1 Member Data Documentation

3.17.1.1 IEEE

[IEEE](#) [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 15:49:42 01/08/2008 Design Name: Module Name: [high_bit](#) - Behavioral Project Name: Target Devices: Tool versions: Description:

3.17.1.2 STD_LOGIC_1164

STD_LOGIC_1164 [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- high_bit.vhd

3.18 if_uart Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164
- STD_LOGIC_ARITH
- STD_LOGIC_UNSIGNED

Ports

- [illegible]

3.18.1 Member Data Documentation

3.18.1.1 IEEE

IEEE [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 16:17:31 01/12/2008 Design Name: Module Name: [if_uart](#) - Behavioral Project Name: Target Devices: Tool versions: Description:

3.18.1.2 STD_LOGIC_1164

STD_LOGIC_1164 [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [if_uart.vhd](#)

3.19 io_controller Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- [CLOCK](#) in [std_logic](#) := '0'
 clock
- [IO_SWITCH](#) in [std_logic](#) := '0'
 selettore componente_io '1' switch e led, '0' uart
- [CE](#) in [STD_LOGIC](#) := '0'
- [RD](#) in [std_logic](#) := '0'
 segnale di lettura
- [WR](#) in [std_logic](#) := '0'
 segnale di scrittura
- [RXD](#) in [std_logic](#) := '1'
 rxd seriale
- [START_READ](#) in [STD_LOGIC](#) := '0'
 avvia la lettura dagli switch
- [SWITCH](#) in [STD_LOGIC_VECTOR](#)([7](#) downto [0](#))

- [illegible]

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

- io_controller.vhd

3.20 io_switch_led_display Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164
- STD_LOGIC_ARITH
- STD_LOGIC_UNSIGNED

Ports

- CLOCK in std_logic:= '0'
clock
- CE in std_logic:= '0'
chip enable del componente
- RD in std_logic:= '0'
segnale di lettura
- WR in std_logic:= '0'
segnale di scrittura
- START_READ in STD_LOGIC:= '0'
avvia la lettura dagli switch, come se premessimo enter quando usiamo l'uart
- ENABLE_DISPLAY in STD_LOGIC:= '1'
- SWITCH in std_logic_vector(7 downto 0)
- LEDS out std_logic_vector(7 downto 0):=(others=>'0')
- 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.
- IO_MDR inout std_logic_vector(31 downto 0):= "ZZ"

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

- io switch led.vhd

3.21 mbr_register Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- [ck](#) in [std_logic](#)
- [eobb_mbr](#) in [std_logic](#)
- [eobb_mbru](#) in [std_logic](#)
- [ffetch_ritardato](#) in [std_logic](#)
- [data_mbr](#) in [std_logic_vector](#)([7](#) downto [0](#))
- [uscita_mbr](#) out [std_logic_vector](#)([31](#) downto [0](#))

3.21.1 Member Data Documentation

3.21.1.1 IEEE

[IEEE](#) [Library]

Company: Engineer:

Create Date: 14:14:15 03/30/2009 Design Name: Module Name: mbr - Behavioral Project Name: Target Devices:
Tool versions: Description:

3.21.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [mbr.vhd](#)

3.22 mdr_register Entity Reference

Libraries

- [IEEE](#)
per default

Use Clauses

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

Ports

- [ck](#) in [std_logic](#)
- [wcbtr_mdr](#) in [std_logic](#)
- [rread_ritardato](#) in [std_logic](#)
- [inguscita_mdr](#) in [std_logic_vector](#)([31](#) downto [0](#))
- [c_bus](#) in [std_logic_vector](#)([31](#) downto [0](#))
- [uscita_mdr](#) out [std_logic_vector](#)([31](#) downto [0](#)):=x" 00000000 "

3.22.1 Member Data Documentation

3.22.1.1 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [mdr_register.vhd](#)

3.23 mic1 Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- **cki** in std_logic
clock
- **uscita_mar** out std_logic_vector(31 downto 0):=x" 00000000 "
uscita verso la memoria reg. MAR
- **io_mdr** inout std_logic_vector(31 downto 0):=x" 00000000 "
ingr./uscita verso la memoria reg. MDR
- **uscita_pc** out std_logic_vector(31 downto 0):=x" 00000000 "
uscita verso la memoria reg. PC
- **ingresso_mbr** in std_logic_vector(7 downto 0):=x" 00 "
ingresso dalla memoria reg. MBR
- **wr** out std_logic
write
- **rd** out std_logic
read
- **fetch** out std_logic
fetch

3.23.1 Member Data Documentation

3.23.1.1 IEEE

IEEE [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 19:48:07 01/10/2008 Design Name: Module Name: mic1 - Behavioral Project Name: Target Devices: Tool versions: Description:

3.23.1.2 STD_LOGIC_1164

STD_LOGIC_1164 [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- mic1.vhd

3.24 mpc Entity Reference

Libraries

- IEEE

Use Clauses

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

Ports

- **din** in **std_logic_vector(8 downto 0)**
ingresso registro a 9 bit
- **dout** out **std_logic_vector(8 downto 0)** := "000000000 "
uscita registro a 9 bit
- **ck** in **std_logic**
clock

3.24.1 Member Data Documentation

3.24.1.1 IEEE

[IEEE](#) [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 15:34:26 01/08/2008 Design Name: Module Name: mpc - Behavioral Project Name: Target Devices: Tool versions: Description:

3.24.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- mpc.vhd

3.25 neg_edge_reg Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- [input](#) in [std_logic_vector](#)([8](#) downto [0](#))
- [output](#) out [std_logic_vector](#)([8](#) downto [0](#)) := "00000000"
- [ck](#) in [std_logic](#)

3.25.1 Member Data Documentation

3.25.1.1 IEEE

[IEEE](#) [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 18:38:30 01/10/2008 Design Name: Module Name: [neg_edge_reg](#) - Behavioral Project Name: Target Devices: Tool versions: Description:

3.25.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [neg_edge_reg.vhd](#)

3.26 negedge_dff Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- `d` in `std_logic`
- `ck` in `std_logic`
- `q` out `std_logic:= '0'`

3.26.1 Member Data Documentation

3.26.1.1 IEEE

`IEEE` [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 20:30:50 01/06/2008 Design Name: Module Name: dff - Behavioral Project Name: Target Devices: Tool versions: Description:

3.26.1.2 STD_LOGIC_1164

`STD_LOGIC_1164` [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- `negedge_dff.vhd`

3.27 oring Entity Reference

Libraries

- `IEEE`

Use Clauses

- `STD_LOGIC_1164`
- `STD_LOGIC_ARITH`
- `STD_LOGIC_UNSIGNED`

Ports

- `uscita_oring` out `std_logic_vector(7 downto 0):=x" 01 "`
- `jmpc` in `std_logic`
- `addr` in `std_logic_vector(7 downto 0)`
- `reg_mbr` in `std_logic_vector(7 downto 0)`

3.27.1 Member Data Documentation

3.27.1.1 IEEE

[IEEE](#) [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 15:40:49 01/08/2008 Design Name: Module Name: oring - Behavioral Project Name: Target Devices: Tool versions: Description:

3.27.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [oring.vhd](#)

3.28 parte_di_controllo Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- **ff_n** in std_logic:= '0'
 - ingressi ingresso FF flag N*
- **ff_z** in std_logic:= '0'
 - ingresso FF flag Z*
- **reg_mbr** in std_logic_vector(7 downto 0)
 - ingresso registro MBR*
- **ck** in std_logic
- **wwrite** out std_logic:= '0'
 - linea write per la ram*
- **read** out std_logic:= '0'
 - linea read per la ram*
- **ffetch** out std_logic:= '0'
- **sra1** out std_logic:= '0'
 - linea sra1 shifter*
- **sll8** out std_logic:= '0'
- **f0** out std_logic:= '0'
 - linea f0 alu*
- **f1** out std_logic:= '0'
 - linea f1 alu*
- **ena** out std_logic:= '0'
 - linea ena alu*
- **enb** out std_logic:= '0'
 - linea enb alu*
- **inva** out std_logic:= '0'
 - linea inva alu*
- **inc** out std_logic:= '0'
- **h** out std_logic:= '0'
 - segnale h*
- **opc** out std_logic:= '0'
 - segnale opc*
- **tos** out std_logic:= '0'
 - segnale tos*
- **cpp** out std_logic:= '0'
 - segnale cpp*
- **lv** out std_logic:= '0'
 - segnale lv*
- **sp** out std_logic:= '0'
 - segnale sp*
- **pc** out std_logic:= '0'
 - segnale pc*
- **mdr** out std_logic:= '0'
 - segnale mdr*
- **mar** out std_logic:= '0'
- **eob_mdr** out std_logic:= '0'
 - segnale mdr*
- **eob_pc** out std_logic:= '0'
 - segnale pc*
- **eob_mbr** out std_logic:= '0'
 - segnale mbr*

- `eob_mbru out std_logic:= '0'`
segnale mbru
- `eob_sp out std_logic:= '0'`
segnale sp
- `eob_lv out std_logic:= '0'`
segnale lv
- `eob_cpp out std_logic:= '0'`
segnale cpp
- `eob_tos out std_logic:= '0'`
segnale tos
- `eob_opc out std_logic:= '0'`
segnale opc

3.28.1 Member Data Documentation

3.28.1.1 eob_mdr

`eob_mdr out std_logic:= '0'` [Port]

segnale mdr

segnale mar parte dei segnali di abilitazione per il B bus

3.28.1.2 f0

`f0 out std_logic:= '0'` [Port]

linea f0 alu

linea sll8 shifter uscite pilotaggio alu

3.28.1.3 h

`h out std_logic:= '0'` [Port]

segnale h

linea inc alu parte dei segnali di abilitazione per il C bus

3.28.1.4 IEEE

`IEEE` [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 10:08:15 01/07/2008 Design Name: Module Name: [parte_di_controllo](#) - Behavioral Project Name: Target Devices: Tool versions: Description:

Dependencies:

Revision: Revision 0.01 - File Created Additional Comments:

3.28.1.5 sra1

```
sra1 out std_logic:= ' 0 ' [Port]
```

linea sra1 shifter

linea read per la rom uscite pilotaggio shifter

3.28.1.6 wwrite

```
wwrite out std_logic:= ' 0 ' [Port]
```

linea write per la ram

ingresso clock uscite pilotaggio memoria

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

- pc.vhd

3.29 parte_operativa Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- **wwrite in std_logic:= ' 0 '**
ingressi dal MIR per pilotaggio memoria linea write per la ram
- **rread in std_logic:= ' 0 '**
linea read per la ram
- **ffetch in std_logic:= ' 0 '**
linea read per la rom
- **ck in std_logic**
clock
- **eobb_mdr in std_logic**
- **eobb_pc in std_logic**
- **eobb_mbr in std_logic**
- **eobb_mbru in std_logic**
- **eobb_sp in std_logic**
- **eobb_lv in std_logic**

3.30 posedge_reg_enable Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Generics

- [valore_iniziale](#) `std_logic_vector(31 downto 0):=x" 00000000 "`

Ports

- [ck](#) in `std_logic`
- [en](#) in `std_logic`
- [input_reg](#) in `std_logic_vector(31 downto 0)`
- [output_reg](#) out `std_logic_vector(31 downto 0):=valore_iniziale`

3.30.1 Member Data Documentation

3.30.1.1 IEEE

[IEEE](#) [Library]

Company: Engineer:

Create Date: 15:40:14 03/30/2009 Design Name: Module Name: [posedge_reg_enable](#) - Behavioral Project
Name: Target Devices: Tool versions: Description:

3.30.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [posedge_reg_enable.vhd](#)

3.31 ram Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Generics

- [ADDR_WIDTH](#) integer:= **7**
- [DATA_WIDTH](#) integer:= **32**

Ports

- [cs](#) in std_logic
- [ck](#) in std_logic
- [we](#) in std_logic
- [rd](#) in std_logic
- [addr](#) in std_logic_vector(ADDR_WIDTH- **1** downto **0**)
- [data](#) inout std_logic_vector(DATA_WIDTH- **1** downto **0**)

3.31.1 Member Data Documentation

3.31.1.1 IEEE

[IEEE](#) [Library]

Company: Engineer:

Create Date: 15:53:29 03/21/2009 Design Name: Module Name: ram_core - Behavioral Project Name: Target
Devices: Tool versions: Description:

3.31.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- ram_core.vhd

3.32 shifter Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

Ports

- [sll8](#) in `std_logic`
- [sra1](#) in `std_logic`
- [input](#) in `std_logic_vector(31 downto 0)`
- [output](#) out `std_logic_vector(31 downto 0)`

3.32.1 Member Data Documentation

3.32.1.1 IEEE

[IEEE](#) [Library]

`others=>(others=>'0'));`

3.32.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- `shifter.vhd`

3.33 sistema_mic1 Entity Reference

Libraries

- [IEEE](#)
- [UNISIM](#)

Use Clauses

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

Ports

- [RXD](#) in `std_logic`
- [TXD](#) out `std_logic`
- [ck](#) in `std_logic`
- [led](#) out `std_logic_vector(7 downto 0)`
- [sw](#) in `std_logic_vector(7 downto 0)`
- [IO_SWITCH](#) in `STD_LOGIC`
- [read_led](#) out `STD_LOGIC`
- [start_read](#) in `STD_LOGIC`
- [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.

3.33.1 Member Data Documentation

3.33.1.1 IEEE

[IEEE](#) [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 16:17:31 01/12/2008 Design Name: Module Name: [sistema_mic1](#) - Behavioral Project Name: Target Devices: Tool versions: Description:

3.33.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [sistema_mic1.vhd](#)

3.34 tb_if_uart Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [STD_LOGIC_ARITH](#)
- [STD_LOGIC_UNSIGNED](#)
- [STD_LOGIC_TEXTIO](#)
- [TEXTIO](#)

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

- TB_IF_UART.vhd

3.35 tb_ram Entity Reference

Libraries

- [IEEE](#)

Use Clauses

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

3.35.1 Member Data Documentation

3.35.1.1 IEEE

[IEEE](#) [Library]

Company: Engineer:

3.35.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

// / / / / Vendor: Xilinx \ \ \ Version : 9.2i \ \ Application : ISE // Filename : tb_ram.vhw / / \ Timestamp : Fri
Apr 03 20:43:39 2009 \ \ \ / \ \

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

- [tb_ram.vhd](#)

3.36 TB_SISTEMA_COMPLETO Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [STD_LOGIC_ARITH](#)
- [STD_LOGIC_UNSIGNED](#)
- [STD_LOGIC_TEXTIO](#)
- [TEXTIO](#)

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

- [tb_mic1conuart.vhd](#)

3.37 tb_uart_vhd Entity Reference

Libraries

- [ieee](#)

Use Clauses

- [std_logic_1164](#)
- [std_logic_unsigned](#)
- [numeric_std](#)

3.37.1 Member Data Documentation

3.37.1.1 ieee

ieee [Library]

Company: Engineer:

3.37.1.2 std_logic_1164

std_logic_1164 [Package]

Notes: This testbench has been automatically generated using types std_logic and std_logic_vector for the ports of the unit under test. Xilinx recommends that these types always be used for the top-level I/O of a design in order to guarantee that the testbench will bind correctly to the post-implementation simulation model.

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

- tb_uart.vhd

3.38 uart_9600bps Entity Reference

Libraries

- IEEE
- UNISIM

Use Clauses

- STD_LOGIC_1164
- STD_LOGIC_ARITH
- STD_LOGIC_UNSIGNED
- VComponents

Ports

- reset in std_logic
- clk in std_logic
- rx_data_serial in std_logic
- tx_data_in in std_logic_vector(7 downto 0)
- tx_data_en in std_logic
- rx_data_out out std_logic_vector(7 downto 0)
- rx_data_en out std_logic
- rx_ovf_err out std_logic
- rx_parity_err out std_logic
- tx_data_serial out std_logic
- tx_ch_rdy out std_logic

3.38.1 Member Data Documentation

3.38.1.1 IEEE

[IEEE](#) [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 16:17:31 01/12/2008 Design Name: Module Name: uart_con_div - Behavioral Project Name: Target Devices: Tool versions: Description:

3.38.1.2 STD_LOGIC_1164

[STD_LOGIC_1164](#) [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- [uart_con_div.vhd](#)

3.39 uart_rx Entity Reference

Libraries

- [IEEE](#)
- [ieee](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [STD_LOGIC_ARITH](#)
- [STD_LOGIC_UNSIGNED](#)
- [std_logic_1164](#)
- [numeric_std](#)

Generics

- [DBIT](#) integer:= **8**
#data bits
- [SB_TICK](#) integer:= **16**

ticks for stop bits

Ports

- `ck` in `std_logic`
- `rx` in `std_logic`
- `s_tick` in `std_logic`
- `rx_done_tick` out `std_logic`
- `dout` out `std_logic_vector(7 downto 0)`

3.39.1 Member Data Documentation

3.39.1.1 IEEE

`IEEE` [Library]

Company: Engineer: taken from Pong Chu , Wiley , "FPGA PROTOTYPING BY VHDL EXAMPLES"

Create Date: 10:10:15 01/12/2008 Design Name: Module Name: `uart_rx` - Behavioral Project Name: Target Devices: Tool versions: Description:

3.39.1.2 STD_LOGIC_1164

`STD_LOGIC_1164` [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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

- `uart_rx.vhd`

3.40 uart_serial Entity Reference

Libraries

- `UNISIM`
- `ieee`

Use Clauses

- `VComponents`
- `std_logic_1164`
- `std_logic_unsigned`
- `numeric_std`

Ports

- **reset** in **std_logic**
Global signal reset control signal.
- **clk** in **std_logic**
- **rx_data_serial** in **std_logic**
Received Serial data from RS232.
- **rx_data_out** out **std_logic_vector(7 downto 0)**
Received Data.
- **rx_data_en** out **std_logic**
Received data enable control signal.
- **rx_ovf_err** out **std_logic**
Received data over frame error detected.
- **rx_parity_err** out **std_logic**
- **tx_data_serial** out **std_logic**
Transmitted Serial data to RS232.
- **tx_data_in** in **std_logic_vector(7 downto 0)**
Transmitted data.
- **tx_data_en** in **std_logic**
Transmitted data latch enable.
- **tx_ch_rdy** out **std_logic**
- **baud_sel** in **std_logic_vector(3 downto 0)**
Baud value see Note.
- **parity_en** in **std_logic**
Enable parity control signal active HIGH.
- **parity_type** in **std_logic**
1:ODD parity / 0:EVEN parity

3.40.1 Member Data Documentation

3.40.1.1 baud_sel

baud_sel in **std_logic_vector(3 downto 0)** [Port]

Baud value see Note.

Transmission channel ready status signal Control command

3.40.1.2 ieee

ieee [Library]

Use of this source code through a simulator and/or a compiler tool is illegal if not authorised through Author License agreement.

3.40.1.3 rx_data_serial

`rx_data_serial` in `std_logic` [Port]

Received Serial data from RS232.

14.7456 Mhz Clock frequency Reception channel

3.40.1.4 std_logic_1164

`std_logic_1164` [Package]

Version : 1.0 Date : Modifier : Modif. :

3.40.1.5 tx_data_serial

`tx_data_serial` out `std_logic` [Port]

Transmitted Serial data to RS232.

Received data parity error Transmission channel

3.40.1.6 UNISIM

`UNISIM` [Library]

Company: UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO SECONDO Engineer: AIELLO MARCO MATR. 045/004437

Create Date: 16:17:31 01/12/2008

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

- `uart_serial.vhd`

3.41 uart_tx Entity Reference

Libraries

- [IEEE](#)
- [ieee](#)

Use Clauses

- [STD_LOGIC_1164](#)
- [STD_LOGIC_ARITH](#)
- [STD_LOGIC_UNSIGNED](#)
- [std_logic_1164](#)
- [numeric_std](#)

Generics

- `DBIT integer:= 8`
data bits
- `SB_TICK integer:= 16`

ticks for stop bits

Ports

- `ck in std_logic`
- `tx_start in std_logic`
- `s_tick in std_logic`
- `din in std_logic_vector(7 downto 0)`
- `tx_done_tick out std_logic`
- `tx out std_logic:= ' 1 '`

3.41.1 Member Data Documentation

3.41.1.1 IEEE

`IEEE` [Library]

Company: Engineer: taken from Pong Chu , Wiley , "FPGA PROTOTYPING BY VHDL EXAMPLES"

Create Date: 10:10:15 01/12/2008 Design Name: Module Name: `uart_tx` - Behavioral Project Name: Target
Devices: Tool versions: Description:

3.41.1.2 STD_LOGIC_1164

`STD_LOGIC_1164` [Package]

Revision: Revision 0.01 - File Created Additional Comments:

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 display_7_segmenti.vhd File Reference

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

Entities

- [display_7_segments](#) entity

4.2.1 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

4.3 io_controller.vhd File Reference

Componente che sceglie tra UART e Switch/Led/Display.

Entities

- [io_controller](#) entity

4.3.1 Detailed Description

Componente che sceglie tra UART e Switch/Led/Display.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.4 io_switch_led.vhd File Reference

Componente che gestisce il collegamento tra switch (input) e led (output)

Entities

- [io_switch_led_display](#) entity

4.4.1 Detailed Description

Componente che gestisce il collegamento tra switch (input) e led (output)

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

Index

- ALU, [5](#)
 - IEEE, [5](#)
 - STD_LOGIC_1164, [6](#)
- anodes_manager, [6](#)
 - STD_LOGIC_1164, [7](#)
- baud_gen, [7](#)
 - STD_LOGIC_1164, [7](#)
- baud_sel
 - uart_serial, [44](#)
- boot_rom, [8](#)
 - ieee, [8](#)
 - std_logic_1164, [9](#)
- cathodes_manager, [9](#)
 - STD_LOGIC_1164, [10](#)
- clk_wiz_v3_6, [10](#)
- clock_div, [11](#)
 - ieee, [11](#)
- clock_divisor, [12](#)
 - IEEE, [13](#)
 - STD_LOGIC_1164, [13](#)
- control_store, [13](#)
- counter_UpMod2n_Re_Sr, [14](#)
- counter_UpMod2n_Re_Sr.vhd, [47](#)
- cpool_rom, [14](#)
 - ieee, [15](#)
 - std_logic_1164, [15](#)
- decoder, [15](#)
 - IEEE, [16](#)
 - STD_LOGIC_1164, [16](#)
- decodifica_indirizzi, [16](#)
 - STD_LOGIC_1164, [17](#)
- dff, [17](#)
 - IEEE, [17](#)
 - STD_LOGIC_1164, [18](#)
- display_7_segmenti.vhd, [47](#)
- display_7_segments, [18](#)
- divisore_freq, [19](#)
 - IEEE, [19](#)
 - STD_LOGIC_1164, [19](#)
- eob_mdr
 - parte_di_controllo, [32](#)
- f0
 - parte_di_controllo, [32](#)
- h
 - parte_di_controllo, [32](#)
- high_bit, [20](#)
 - IEEE, [20](#)
 - STD_LOGIC_1164, [20](#)
- IEEE
 - ALU, [5](#)
 - clock_divisor, [13](#)
 - decoder, [16](#)
 - dff, [17](#)
 - divisore_freq, [19](#)
 - high_bit, [20](#)
 - if_uart, [21](#)
 - mbr_register, [24](#)
 - mic1, [26](#)
 - mpc, [27](#)
 - neg_edge_reg, [28](#)
 - negedge_dff, [29](#)
 - oring, [30](#)
 - parte_di_controllo, [32](#)
 - parte_operativa, [34](#)
 - posedge_reg_enable, [35](#)
 - ram, [36](#)
 - shifter, [37](#)
 - sistema_mic1, [38](#)
 - tb_ram, [39](#)
 - uart_9600bps, [42](#)
 - uart_rx, [43](#)
 - uart_tx, [46](#)
- ieee
 - boot_rom, [8](#)
 - clock_div, [11](#)
 - cpool_rom, [15](#)
 - tb_uart_vhd, [40](#)
 - uart_serial, [44](#)
- if_uart, [21](#)
 - IEEE, [21](#)
 - STD_LOGIC_1164, [22](#)
- io_controller, [22](#)
- io_controller.vhd, [48](#)
- io_switch_led.vhd, [49](#)
- io_switch_led_display, [23](#)
- mbr_register, [24](#)
 - IEEE, [24](#)
 - STD_LOGIC_1164, [24](#)
- mdr_register, [25](#)
 - STD_LOGIC_1164, [25](#)
- mic1, [25](#)
 - IEEE, [26](#)
 - STD_LOGIC_1164, [26](#)

- mpc, [26](#)
 - IEEE, [27](#)
 - STD_LOGIC_1164, [27](#)
- neg_edge_reg, [27](#)
 - IEEE, [28](#)
 - STD_LOGIC_1164, [28](#)
- negedge_dff, [28](#)
 - IEEE, [29](#)
 - STD_LOGIC_1164, [29](#)
- oring, [29](#)
 - IEEE, [30](#)
 - STD_LOGIC_1164, [30](#)
- parte_di_controllo, [30](#)
 - eob_mdr, [32](#)
 - f0, [32](#)
 - h, [32](#)
 - IEEE, [32](#)
 - sra1, [32](#)
 - wwrite, [33](#)
- parte_operativa, [33](#)
 - IEEE, [34](#)
 - STD_LOGIC_1164, [34](#)
- posedge_reg_enable, [35](#)
 - IEEE, [35](#)
 - STD_LOGIC_1164, [35](#)
- ram, [36](#)
 - IEEE, [36](#)
 - STD_LOGIC_1164, [36](#)
- rx_data_serial
 - uart_serial, [44](#)
- STD_LOGIC_1164
 - ALU, [6](#)
 - anodes_manager, [7](#)
 - baud_gen, [7](#)
 - cathodes_manager, [10](#)
 - clock_divisor, [13](#)
 - decoder, [16](#)
 - decodifica_indirizzi, [17](#)
 - dff, [18](#)
 - divisore_freq, [19](#)
 - high_bit, [20](#)
 - if_uart, [22](#)
 - mbr_register, [24](#)
 - mdr_register, [25](#)
 - mic1, [26](#)
 - mpc, [27](#)
 - neg_edge_reg, [28](#)
 - negedge_dff, [29](#)
 - oring, [30](#)
 - parte_operativa, [34](#)
 - posedge_reg_enable, [35](#)
 - ram, [36](#)
 - shifter, [37](#)
 - sistema_mic1, [38](#)
 - tb_ram, [39](#)
 - uart_9600bps, [42](#)
 - uart_rx, [43](#)
 - uart_tx, [46](#)
- shifter, [37](#)
 - IEEE, [37](#)
 - STD_LOGIC_1164, [37](#)
- sistema_mic1, [37](#)
 - IEEE, [38](#)
 - STD_LOGIC_1164, [38](#)
- sra1
 - parte_di_controllo, [32](#)
- std_logic_1164
 - boot_rom, [9](#)
 - cpool_rom, [15](#)
 - tb_uart_vhd, [41](#)
 - uart_serial, [45](#)
- TB_SISTEMA_COMPLETO, [40](#)
- tb_if_uart, [39](#)
- tb_ram, [39](#)
 - IEEE, [39](#)
 - STD_LOGIC_1164, [39](#)
- tb_uart_vhd, [40](#)
 - ieee, [40](#)
 - std_logic_1164, [41](#)
- tx_data_serial
 - uart_serial, [45](#)
- UNISIM
 - uart_serial, [45](#)
- uart_9600bps, [41](#)
 - IEEE, [42](#)
 - STD_LOGIC_1164, [42](#)
- uart_rx, [42](#)
 - IEEE, [43](#)
 - STD_LOGIC_1164, [43](#)
- uart_serial, [43](#)
 - baud_sel, [44](#)
 - ieee, [44](#)
 - rx_data_serial, [44](#)
 - std_logic_1164, [45](#)
 - tx_data_serial, [45](#)
 - UNISIM, [45](#)
- uart_tx, [45](#)
 - IEEE, [46](#)
 - STD_LOGIC_1164, [46](#)
- wwrite
 - parte_di_controllo, [33](#)