

carry_look_ahead_adder

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	Class Documentation	3
2.1	carry_look_ahead_adder Entity Reference	3
2.1.1	Member Data Documentation	3
2.1.1.1	IEEE	4
2.1.1.2	STD_LOGIC_1164	4
2.2	carry_look_ahead_adder_timing Entity Reference	4
2.2.1	Detailed Description	4
2.2.2	Member Data Documentation	5
2.2.2.1	IEEE	5
2.2.2.2	STD_LOGIC_1164	5
2.3	d_edge Entity Reference	5
2.3.1	Member Data Documentation	6
2.3.1.1	ieee	6
2.3.1.2	std_logic_1164	6
2.4	full_adder Entity Reference	6
2.4.1	Detailed Description	7
2.4.2	Member Data Documentation	7
2.4.2.1	IEEE	7
2.4.2.2	STD_LOGIC_1164	7
2.5	propagation_generation_calculator Entity Reference	7
2.5.1	Member Data Documentation	8
2.5.1.1	STD_LOGIC_1164	8
	Index	9

Chapter 1

Class Index

1.1 Class List

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

entity carry_look_ahead_adder	3
entity carry_look_ahead_adder_timing	
Uncomment the following library declaration if instantiating any Xilinx primitives in this code . .	4
entity d_edge	5
entity full_adder	6
entity propagation_generation_calculator	7

Chapter 2

Class Documentation

2.1 carry_look_ahead_adder Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)

Generics

- [width](#) **NATURAL** := **8**

Ports

- [c_in](#) in **STD_LOGIC**
carry_look_ahead_adder input : carry ingresso
- [X](#) in **STD_LOGIC_VECTOR**(width- **1** downto **0**)
carry_look_ahead_adder inputs : primo addendo
- [Y](#) in **STD_LOGIC_VECTOR**(width- **1** downto **0**)
carry_look_ahead_adder inputs : secondo addendo
- [c_out](#) out **STD_LOGIC**
carry_look_ahead_adder output : carry uscita
- [S](#) out **STD_LOGIC_VECTOR**(width- **1** downto **0**)
carry_look_ahead_adder outputs : somma

2.1.1 Member Data Documentation

2.1.1.1 IEEE

IEEE [Library]

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

2.1.1.2 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:

- carry_look_ahead_adder.vhd

2.2 carry_look_ahead_adder_timing Entity Reference

Uncomment the following library declaration if instantiating any Xilinx primitives in this code.

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Generics

- width natural:= 16

Ports

- clock in STD_LOGIC
- X in STD_LOGIC_VECTOR(width- 1 downto 0)
- Y in STD_LOGIC_VECTOR(width- 1 downto 0)
- c_in in STD_LOGIC
- S out STD_LOGIC_VECTOR(width- 1 downto 0)
- c_out out STD_LOGIC

2.2.1 Detailed Description

Uncomment the following library declaration if instantiating any Xilinx primitives in this code.

Uncomment the following library declaration if using arithmetic functions with Signed or Unsigned values

2.2.2 Member Data Documentation

2.2.2.1 IEEE

[IEEE](#) [Library]

Company: Engineer:

Create Date: 14:00:52 02/16/2019 Design Name: Module Name: [carry_look_ahead_adder_timing](#) - Behavioral
Project Name: Target Devices: Tool versions: Description:

2.2.2.2 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:

- [carry_look_ahead_adder_timing.vhd](#)

2.3 d_edge Entity Reference

Libraries

- [ieee](#)

Use Clauses

- [std_logic_1164](#)
- [all](#)

Generics

- [width](#) natural := [8](#)

Ports

- [clock](#) in STD_LOGIC
- [D](#) in STD_LOGIC_VECTOR(width- [1](#) downto [0](#))
- [Q](#) out STD_LOGIC_VECTOR(width- [1](#) downto [0](#))

2.3.1 Member Data Documentation

2.3.1.1 ieee

`ieee` [Library]

D Flip-Flop (ESD book Chapter 2.3.1) by Weijun Zhang, 04/2001

2.3.1.2 std_logic_1164

`std_logic_1164` [Package]

Flip-flop is the basic component in sequential logic design we assign input signal to the output at the clock rising edge

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

- `d_edge_behav.vhd`

2.4 full_adder Entity Reference

Libraries

- `IEEE`

Use Clauses

- `STD_LOGIC_1164`

Ports

- **x in STD_LOGIC**
full_adder input : addendo
- **y in STD_LOGIC**
full_adder input : addendo
- **c_in in STD_LOGIC**
full_adder input : carry in ingresso
- **s out STD_LOGIC**
full_adder output : somma
- **c_out out STD_LOGIC**
full_adder output : carry

2.4.1 Detailed Description

Descrizione Somma i 3 bit in ingresso (2 addendi e 1 carry in ingresso).
In uscita abbiamo il risultato della somma sul bit S e il riporto sul bit C.

2.4.2 Member Data Documentation

2.4.2.1 IEEE

[IEEE](#) [Library]

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

2.4.2.2 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:

- [full_adder.vhd](#)

2.5 propagation_generation_calculator Entity Reference

Libraries

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

Use Clauses

- [STD_LOGIC_1164](#)

Generics

- [width](#) **NATURAL** := **2**

Ports

- **X** in STD_LOGIC_VECTOR(width- 1 downto 0)
propagation_generation_calculator inputs : primo addendo
- **Y** in STD_LOGIC_VECTOR(width- 1 downto 0)
propagation_generation_calculator inputs : secondo addendo
- **G** out STD_LOGIC_VECTOR(width- 1 downto 0)
propagation_generation_calculator inputs : riporto generato
- **P** out STD_LOGIC_VECTOR(width- 1 downto 0)
propagation_generation_calculator inputs : riporto propagato

2.5.1 Member Data Documentation

2.5.1.1 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:

- propagation_generation_calculator.vhd

Index

- carry_look_ahead_adder, [3](#)
 - IEEE, [3](#)
 - STD_LOGIC_1164, [4](#)
- carry_look_ahead_adder_timing, [4](#)
 - IEEE, [5](#)
 - STD_LOGIC_1164, [5](#)
- d_edge, [5](#)
 - ieee, [6](#)
 - std_logic_1164, [6](#)
- full_adder, [6](#)
 - IEEE, [7](#)
 - STD_LOGIC_1164, [7](#)
- IEEE
 - carry_look_ahead_adder, [3](#)
 - carry_look_ahead_adder_timing, [5](#)
 - full_adder, [7](#)
- ieee
 - d_edge, [6](#)
- propagation_generation_calculator, [7](#)
 - STD_LOGIC_1164, [8](#)
- STD_LOGIC_1164
 - carry_look_ahead_adder, [4](#)
 - carry_look_ahead_adder_timing, [5](#)
 - full_adder, [7](#)
 - propagation_generation_calculator, [8](#)
- std_logic_1164
 - d_edge, [6](#)