carry_select

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 carry_select_adder Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Generics

- M **NATURAL:= 8**
  parallelismo dei ripplecarry adder nei blocchi
- P **NATURAL:= 8**
  numero di blocchi del carry select

Ports

- A **in STD_LOGIC_VECTOR(((M ∗P)- 1)downto 0)**
  input addendo
- B **in STD_LOGIC_VECTOR(((M ∗P)- 1)downto 0)**
  input addendo
- c_in **in STD_LOGIC**
  input carry in ingresso
- S **out STD_LOGIC_VECTOR(((M ∗P)- 1)downto 0)**
  output somma
- c_out **out STD_LOGIC**
  output carry in uscita

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

- carrySelect_adder.vhd

3.2 carry_select_block Entity Reference

Use Clauses

- STD_LOGIC_1164

Generics

- width **NATURAL:= 4**

Ports

- A **in STD_LOGIC_VECTOR((width- 1)downto 0)**
- B **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**

3.2.1 Detailed Description

blocco carry select è formato da due RCA e due multiplexer 2-1. I due addizionatori si occuperanno di sommare le due stringhe in ingresso A e B (di lunghezza generica width) con c_in pari, rispettivamente, a 0 e 1. In base al valore c_in effettivo in ingresso al blocco, i due multiplexer sceglieranno quali dei due valori S e c_out, calcolati dai due RCA, riportare in uscita.

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

- carrySelect_cell.vhd

3.3 carry_select_timing Entity Reference

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

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Generics

- M **NATURAL:= 8**

  parallelismo dei ripplecarry adder nei blocchi
- P **NATURAL:= 8**

Ports

- clock **in STD_LOGIC**
- A **in STD_LOGIC_VECTOR((M** ∗**P)- 1 downto 0)**
- B **in STD_LOGIC_VECTOR((M** ∗**P)- 1 downto 0)**
- c_in **in STD_LOGIC**
- S **out STD_LOGIC_VECTOR((M** ∗**P)- 1 downto 0)**
- c_out **out STD_LOGIC**

3.3.1 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

3.3.2 Member Data Documentation

3.3.2.1 IEEE

`IEEE [Library]`

Company: Engineer:

Create Date: 12:07:17 02/16/2019 Design Name: Module Name: carry_select_timing - Behavioral Project Name: Target Devices: Tool versions: Description:

3.3.2.2 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_select_timing.vhd

3.4 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 **)**

3.4.1 Member Data Documentation

3.4.1.1 ieee

`ieee [Library]`

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

3.4.1.2 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:

- d_edge_behav.vhd

3.5 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

3.5.1 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.

3.5.2 Member Data Documentation

3.5.2.1 IEEE

`IEEE [Library]`

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

3.5.2.2 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

3.6 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

3.6.1 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.

3.6.2 Member Data Documentation

3.6.2.1 IEEE

`IEEE [Library]`

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

3.6.2.2 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

3.7 rippleCarry_adder Entity Reference

Libraries

- IEEE

  architecture dataflow of mux2_1 end

Use Clauses

- STD_LOGIC_1164

Generics

- width **NATURAL:= 8**

Ports

- 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**

  rippleCarry_adder output: carry

3.7.1 Detailed Description

Descrizione Somma le 2 stringe di bit in ingresso (2 addendi) e 1 bit (carry in ingresso). Caratterizzato da una serie di full_adder in cascata che propagano il riporto.
In uscita abbiamo il risultato della somma sul bit S e il riporto sul bit C.

3.7.2 Member Data Documentation

3.7.2.1 c_in

`c_in` **`in STD_LOGIC`** `[Port]`

rippleCarry_adder input: addendo

3.7.2.2 c_out

`c_out` **`out STD_LOGIC`** `[Port]`

rippleCarry_adder output: carry

rippleCarry_adder output: somma

3.7.2.3 S

```
S out STD_LOGIC_VECTOR(width - 1 downto 0 )   [Port]
```

rippleCarry_adder input : carry in ingresso

3.7.2.4 STD_LOGIC_1164

```
STD_LOGIC_1164  [Package]
```

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

3.7.2.5 width

```
width  NATURAL:= 8    [Generic]
```

usato per definire il parallelismo del rippleCarry_adder

3.7.2.6 Y

```
Y in STD_LOGIC_VECTOR(width - 1 downto 0 )   [Port]
```

rippleCarry_adder input: addendo

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

- rippleCarry_adder.vhd

Chapter 4

File Documentation

4.1 carrySelect_adder.vhd File Reference

Sommatore Carry Select.

Entities

- carry_select_adder entity

4.1.1 Detailed Description

Sommatore Carry Select.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.2 carrySelect_cell.vhd File Reference

Singolo blocco di un sommatore carry Select.

Entities

- carry_select_block entity

4.2.1 Detailed Description

Singolo blocco di un sommatore carry Select.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

Index