rca_add_sub

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 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

3.1.1 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.

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

- full_adder.vhd

3.2 half_adder Entity Reference

definisco il componente e la sua interfaccia

Libraries

- IEEE

  architecture dataflow of full_adder end

Use Clauses

- STD_LOGIC_1164

Ports

- X **in STD_LOGIC**

  half_adder input : addendo
- Y **in STD_LOGIC**

  half_adder input : addendo
- C **out STD_LOGIC**

  half_adder output : carry
- S **out STD_LOGIC**

  half_adder output : somma

3.2.1 Detailed Description

definisco il componente e la sua interfaccia

Descrizione Somma i due bit in ingresso.
In uscita abbiamo il risultato della somma sul bit S e il riporto sul bit C.

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

- half_adder.vhd

3.3 overflow_checker Entity Reference

Libraries

- IEEE

  architecture dataflow of half_adder end

Use Clauses

- STD_LOGIC_1164

Ports

- a **in STD_LOGIC**

  bit più significativo (segno) di A
- b **in STD_LOGIC**

  bit più significativo (segno) di B
- subtract **in STD_LOGIC**

  bit di operazione: 1 se sottrazione, 0 se addizione
- s **in STD_LOGIC**

  bit più significativo (segno) di S
- overflow **out STD_LOGIC**

  bit alto se ho una condizione di overflow

3.3.1 Detailed Description

Descrizione La macchina controlla se vi è overflow nel risultato confrontando le cifre più significative (segno) dei due operandi e del risultato con subtract. Ho overflow in caso di:

- somma di due positivi con risultato negativo
- somma di due negativi con risultato positivo
- differenza di positivo e negativo con risultato negativo
- differenza di negativo e positivo con risultato positivo

3.3.2 Member Data Documentation

3.3.2.1 STD_LOGIC_1164

STD_LOGIC_1164 [Package]

Dependencies:
full_adder

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

- overflow_checker.vhd

3.4 rippleCarry_adder Entity Reference

Libraries

- IEEE

  architecture behavioural of overflow_checker 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

3.4.1 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.

3.4.2 Member Data Documentation

3.4.2.1 c_in

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

rippleCarry_adder input: addendo

3.4.2.2 c_out

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

rippleCarry_adder output: carry

rippleCarry_adder output: somma

3.4.2.3 S

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

rippleCarry_adder input : carry in ingresso

3.4.2.4 STD_LOGIC_1164

```
STD_LOGIC_1164  [Package]
```

Dependencies:
full_adder

3.4.2.5 width

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

usato per definire il parallelismo del rippleCarry_adder

3.4.2.6 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

3.5 rippleCarry_addsub Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Generics

- width **NATURAL:= 8**

  usato per definire il parallelismo del sommatore

Ports

- A **in STD_LOGIC_VECTOR(width - 1 downto 0)**
- B **in STD_LOGIC_VECTOR(width - 1 downto 0)**
- subtract **in STD_LOGIC**
- S **out STD_LOGIC_VECTOR(width - 1 downto 0)**
- overflow **out STD_LOGIC**

  rippleCarry_addsub output: condizione di overflow

3.5.1 Detailed Description

Descrizione Effettua la somma o la sottrazione di due stringhe di bit (A e B) di lunghezza width in ingresso. Il bit SUBTRACT in ingresso sarà 0 in caso di una somma, 1 in caso di una differenza. Il secondo operando del RCA (Y) è determinato dalla XOR tra SUBTRACT e B: se subtract = 1, ne farà il complemento. Il subtract è portato anche come c_in del RCA, in modo tale che, se 1, B diventerà !B + 1 = -B, e dunque il RCA farà (A - B). Se subtract è 0, Y sarà B e c_in sarà 0 (eseguirà A + B). L'overflow è alto se ho una somma di numeri positivi e risultato negativo, somma di numeri negativi e risultato positivo, differenza positivo e negativo con risultato negativo o differenza negativo e positivo con risultato positivo. Usiamo una macchina overflow_checker che fa tale controllo usando i bit più significativi (segno) di A, B, S e il bit di subtract.

3.5.2 Member Data Documentation

3.5.2.1 B

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

rippleCarry_addsub input: addendo

3.5.2.2 overflow

```
overflow out STD_LOGIC   [Port]
```

rippleCarry_addsub output: condizione di overflow

rippleCarry_addsub output: risultato

3.5.2.3 S

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

rippleCarry_addsub input: subtract sarà 0 per somma, 1 per differenza

3.5.2.4 subtract

```
subtract in STD_LOGIC   [Port]
```

rippleCarry_addsub input: addendo

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

- rippleCarry_addsub.vhd

Chapter 4

File Documentation

4.1 full_adder.vhd File Reference

Implementazione di un full adder in data flow.

Entities

- full_adder entity

4.1.1 Detailed Description

Implementazione di un full adder in data flow.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.2 half_adder.vhd File Reference

Implementazione di un half adder dataflow.

Entities

- half_adder entity

  definisco il componente e la sua interfaccia

4.2.1 Detailed Description

Implementazione di un half adder dataflow.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.3 rippleCarry_addsub.vhd File Reference

sommatore che effettua sia addizione che sottrazione di due operandi (settando subtract)

Entities

- rippleCarry_addsub entity

4.3.1 Detailed Description

sommatore che effettua sia addizione che sottrazione di due operandi (settando subtract)

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

Index