RobertsonMultiplier

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 carrySelect_adder Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Generics

- M **NATURAL:=** 4

  M parallelismo dei ripplecarry adder.
- P **NATURAL:=** 2

  P parallelismo delle celle dell carry select Come metto M e P, marco e co fanno la stima dei tempi e mettono solo (M∗P) da cui ricavano poi M e P io direi di fare una versione con M e P espliciti e una versione come l'hanno fatta loro, ma su quella.

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 carrySelect_addSub Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164
- math_real
- numeric_std

Generics

- M **NATURAL:=** 4
- P **NATURAL:=** 2

  P parallelismo delle celle dell carry select Come metto M e P, marco e co fanno la stima dei tempi e mettono solo width da cui ricavano poi M e P io direi di fare una versione con M e P espliciti e una versione come l'hanno fatta loro, ma su quella.

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
- subtract **in STD_LOGIC**
- S **out STD_LOGIC_VECTOR(((M∗P)- 1)downto 0)**

  output somma
- overflow **out STD_LOGIC**
- c_out **out STD_LOGIC**

  output carry in uscita

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

- carrySelect_addSub.vhd

3.3 carrySelect_cell Entity Reference

Libraries

- IEEE

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**

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

- carrySelect_cell.vhd

3.4 counter_UpN_Re_Sr Entity Reference

Libraries

- IEEE

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

3.5 flipflop_d_risingEdge_asyncReset Entity Reference

flipflop_d_risingEdge_asyncReset implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono.

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Generics

- init_value **STD_LOGIC:=' 0 '**

  definisce il livello iniziale del flipflop
- reset_level **STD_LOGIC:=' 0 '**

  definisce il livello reset
- enable_level **STD_LOGIC:=' 1 '**

  definisce il livello enable

Ports

- clock **in STD_LOGIC**

  flipflop_d_risingEdge_asyncReset input : segnale di clock per sincronizzare
- enable **in STD_LOGIC**

  flipflop_d_risingEdge_asyncReset input : segnale enable
- reset **in STD_LOGIC**

  flipflop_d_risingEdge_asyncReset input : segnale reset
- d **in STD_LOGIC**

  flipflop_d_risingEdge_asyncReset input : input data
- q **out STD_LOGIC**

  flipflop_d_risingEdge_asyncReset output : output data

3.5.1 Detailed Description

flipflop_d_risingEdge_asyncReset implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono.

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: <14/11/2018> <13/11/2018> <log> create

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

- flipflop_d_risingEdge_asyncReset.vhd

3.6 flipflopmux Entity Reference

Libraries

- IEEE

  architecture behavioural end

Use Clauses

- STD_LOGIC_1164

Ports

- clock **in STD_LOGIC**

  clock
- en **in STD_LOGIC**

  enable
- reset_n **in STD_LOGIC**

  reset
- scan_en **in STD_LOGIC**

  segnale di selezione del multiplexer per modalità (0 = normale, 1 = controllo)
- d **in STD_LOGIC**

  ingresso del flipflop in modalità normale
- scan_in **in STD_LOGIC**

  ingresso del flipflop in modalità controllo
- q **out STD_LOGIC**

  uscita del flipflop

3.6.1 Detailed Description

flipflopmux è un flip flop D con multiplexer: scan_en è il segnale di controllo del multiplexer, se scan_en = 0 l'ingresso è d, se scan_en = 1 l'ingresso è scan_in.

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

- flipflopmux.vhd

3.7 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

3.7.1 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.

3.7.2 Member Data Documentation

3.7.2.1 IEEE

```
IEEE  [Library]
```

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

3.7.2.2 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

3.8 mux2_1 Entity Reference

definisco il componente e la sua interfaccia

Libraries

- IEEE

  architecture dataflow of full_adder end

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

3.8.1 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.

3.8.2 Member Data Documentation

3.8.2.1 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

3.9 overflow_checker Entity Reference

Libraries

- IEEE

  architecture dataflow of mux2_1 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

3.9.1 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

3.9.2 Member Data Documentation

3.9.2.1 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:

- overflow_checker.vhd

3.10 register_d_Re_Ar Entity Reference

Registro di diensione "width" che prende in ingresso un dato D e lo memorizza.

Libraries

- IEEE

  architecture behavioural of overflow_checker end

Use Clauses

- STD_LOGIC_1164

Generics

- width **NATURAL:= 8**

  definisce il parallelismo del registro
- reset_level **STD_LOGIC:=' 0 '**

  definisce il livello reset
- enable_level **STD_LOGIC:=' 1 '**

  definisce il livello enable

Ports

- clock **in STD_LOGIC**

  register_d_Re_Ar input : segnale di clock per sincronizzare
- enable **in STD_LOGIC**

  register_d_Re_Ar input : segnale enable
- reset **in STD_LOGIC**

  register_d_Re_Ar input : segnale reset
- d **in STD_LOGIC_VECTOR(width - 1 downto 0)**

  register_d_Re_Ar input : inpput data
- q **out STD_LOGIC_VECTOR(width - 1 downto 0)**

  register_d_Re_Ar input : output data

3.10.1 Detailed Description

Registro di diensione "width" che prende in ingresso un dato D e lo memorizza.

3.10.2 Member Data Documentation

3.10.2.1 STD_LOGIC_1164

`STD_LOGIC_1164 [Package]`

last changes: <16/11/2018> <16/11/2018> <log> create

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

- register_d_Re_Ar.vhd

3.11 rippleCarry_adder Entity Reference

Libraries

- IEEE

  architecture behavioral of register_d_Re_Ar 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

3.11.1 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.

3.11.2 Member Data Documentation

3.11.2.1 c_in

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

rippleCarry_adder input: addendo

3.11.2.2 c_out

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

rippleCarry_adder output: carry

rippleCarry_adder output: somma

3.11.2.3 S

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

rippleCarry_adder input : carry in ingresso

3.11.2.4 STD_LOGIC_1164

```
STD_LOGIC_1164  [Package]
```

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

3.11.2.5 width

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

usato per definire il parallelismo del rippleCarry_adder

3.11.2.6 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

3.12 robertson_control_unit Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164
- numeric_std
- math_real

Generics

- N **NATURAL:= 8**

  parallelismo di X

Ports

- clock **in STD_LOGIC**
- start **in STD_LOGIC**
- reset_n **in STD_LOGIC**
- current_multiplicand **in STD_LOGIC**

  moltiplicando corrente
- counter_hit **in STD_LOGIC**

  segnala la fine della moltiplicazione
- stop **out STD_LOGIC**
- en_a **out STD_LOGIC**

  se scan_en =1 la scan chain funziona come shifter register
- en_q **out STD_LOGIC**
- en_m **out STD_LOGIC**
- shift **out STD_LOGIC**
- subtract **out STD_LOGIC**
- count_up **out STD_LOGIC**
- sel **out STD_LOGIC**

  sel pilota il secondo input dell'adder 0 in input è Y 1 sono tutti 0
- reset_a **out STD_LOGIC**
- reset_count **out STD_LOGIC**

  reset il conteggio

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

- robertson_control_unit.vhd
- robertson_control_unit_old.vhd

3.13 robertson_multiplier Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164
- numeric_std
- math_real

Generics

- N **INTEGER:=** 8

Ports

- X **in STD_LOGIC_VECTOR(N-** 1 **downto** 0 **)**
- Y **in STD_LOGIC_VECTOR(N-** 1 **downto** 0 **)**
- start **in STD_LOGIC**
- clock **in STD_LOGIC**
- reset_n **in STD_LOGIC**
- stop **out STD_LOGIC**
- Z **out STD_LOGIC_VECTOR((** 2 ∗**N)-** 1 **downto** 0 **)**

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

- robertson_multiplier.vhd

3.14 scan_chain Entity Reference

Libraries

- IEEE

Use Clauses

- STD_LOGIC_1164

Generics

- width **integer:=** 8
  dimensione del registro
- shift_direction **std_logic:='** 1 **'**
  shift a sinistra

Ports

- clock **in STD_LOGIC**

  segnale clock di tempificazione
- en **in STD_LOGIC**

  segnale di abilitazione 1-attivo
- reset_n **in STD_LOGIC**

  segnale di reset 0-attivo
- scan_en **in STD_LOGIC**

  segnale di selezione modalità (0 = normale, 1 = controllo)
- scan_in **in STD_LOGIC**

  primo valore scan-in
- d_reg **in STD_LOGIC_VECTOR(width - 1 downto 0)**

  valore in ingresso nel registro
- scan_out **out STD_LOGIC**

  ultimo valore scan-out
- q_reg **out STD_LOGIC_VECTOR(width - 1 downto 0)**

  valore in uscita del registro

3.14.1 Detailed Description

Scan chain è un registro di width flipflop D multiplexati. Quando scan_en = 0, il componente si comporta come un normale registro. Quando scan_en = 1, diventa uno shift register che shifta ad ogni colpo di clock. La direzione dello shift è regolata dal generic shift_direction (0 = right, 1 = left)

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

- scan_chain.vhd

Chapter 4

File Documentation

4.1 carrySelect_adder.vhd File Reference

Sommatore Carry Select.

Entities

- carrySelect_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_addSub.vhd File Reference

Sommatore Carry Select in grado di effettuare anche l'operazione di sottrazione.

Entities

- carrySelect_addSub entity

4.2.1 Detailed Description

Sommatore Carry Select in grado di effettuare anche l'operazione di sottrazione.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.3 carrySelect_cell.vhd File Reference

Singolo blocco di un sommatore carry Select.

Entities

- carrySelect_cell entity

4.3.1 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

4.4 counter_UpN_Re_Sr.vhd File Reference

Contatore modulo N.

Entities

- counter_UpN_Re_Sr entity

4.4.1 Detailed Description

Contatore modulo N.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.5 flipflopmux.vhd File Reference

flip flop D con multiplexer

Entities

- flipflopmux entity

4.5.1 Detailed Description

flip flop D con multiplexer

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.6 robertson_control_unit.vhd File Reference

Unità di controllo del moltiplicatore di Robertson.

Entities

- robertson_control_unit entity

4.6.1 Detailed Description

Unità di controllo del moltiplicatore di Robertson.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.7 robertson_multiplier.vhd File Reference

Moltiplicatore di Robertson che implementa l'algoritmo di Robertson.

Entities

- robertson_multiplier entity

4.7.1 Detailed Description

Moltiplicatore di Robertson che implementa l'algoritmo di Robertson.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

4.8 scan_chain.vhd File Reference

Registro di n flip flop D multiplexati.

Entities

- scan_chain entity

4.8.1 Detailed Description

Registro di n flip flop D multiplexati.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:
Nothings

Index