

BoothMultiplier

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	anodes_manager Entity Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Data Documentation	5
3.1.2.1	IEEE	6
3.1.2.2	STD_LOGIC_1164	6
3.2	booth_multiplier Entity Reference	6
3.3	booth_multiplier_control_unit Entity Reference	7
3.4	booth_multiplier_onBoard Entity Reference	7
3.5	BoothMultiplier_tb_onBoard Entity Reference	8
3.5.1	Detailed Description	8
3.6	carrySelect_adder Entity Reference	9
3.7	carrySelect_addSub Entity Reference	9
3.8	carrySelect_cell Entity Reference	10
3.9	cathodes_manager Entity Reference	11
3.9.1	Detailed Description	11
3.9.2	Member Data Documentation	11
3.9.2.1	IEEE	11

3.9.2.2	STD_LOGIC_1164	11
3.10	clock_divisor Entity Reference	12
3.10.1	Detailed Description	12
3.10.2	Member Data Documentation	12
3.10.2.1	STD_LOGIC_1164	13
3.11	counter_modN Entity Reference	13
3.12	counter_UpMod2n_Re_Sr Entity Reference	13
3.12.1	Detailed Description	14
3.12.2	Member Data Documentation	14
3.12.2.1	IEEE	14
3.12.2.2	STD_LOGIC_1164	14
3.13	display_7_segmenti Entity Reference	15
3.13.1	Detailed Description	15
3.14	flipflop_d_risingEdge_asyncReset Entity Reference	15
3.14.1	Detailed Description	16
3.14.2	Member Data Documentation	16
3.14.2.1	IEEE	16
3.14.2.2	STD_LOGIC_1164	16
3.15	flipflopmux Entity Reference	17
3.15.1	Detailed Description	17
3.16	full_adder Entity Reference	17
3.16.1	Detailed Description	18
3.17	half_adder Entity Reference	18
3.18	mux2_1 Entity Reference	19
3.18.1	Detailed Description	19
3.18.2	Member Data Documentation	19
3.18.2.1	IEEE	19
3.18.2.2	STD_LOGIC_1164	20
3.19	overflow_checker Entity Reference	20
3.19.1	Detailed Description	20

3.19.2 Member Data Documentation	21
3.19.2.1 STD_LOGIC_1164	21
3.20 register_d_Re_Ar Entity Reference	21
3.20.1 Detailed Description	22
3.20.2 Member Data Documentation	22
3.20.2.1 STD_LOGIC_1164	22
3.21 ripple_carry_adder Entity Reference	22
3.21.1 Member Data Documentation	22
3.21.1.1 std_logic_1164	23
3.22 rippleCarry_adder Entity Reference	23
3.22.1 Detailed Description	23
3.22.2 Member Data Documentation	23
3.22.2.1 c_in	24
3.22.2.2 c_out	24
3.22.2.3 IEEE	24
3.22.2.4 S	24
3.22.2.5 STD_LOGIC_1164	24
3.22.2.6 width	24
3.22.2.7 Y	24
3.23 scan_chain Entity Reference	25
3.23.1 Detailed Description	25

4 File Documentation	27
4.1 booth_multiplier_control_unit.vhd File Reference	27
4.1.1 Detailed Description	27
4.2 booth_On_Board.vhd File Reference	27
4.2.1 Detailed Description	28
4.3 carrySelect_adder.vhd File Reference	28
4.3.1 Detailed Description	28
4.4 carrySelect_addSub.vhd File Reference	29
4.4.1 Detailed Description	29
4.5 carrySelect_cell.vhd File Reference	29
4.5.1 Detailed Description	29
4.6 counter_modN.vhd File Reference	30
4.6.1 Detailed Description	30
4.7 display_7_segmenti.vhd File Reference	30
4.7.1 Detailed Description	30
4.8 flipflopmux.vhd File Reference	31
4.8.1 Detailed Description	31
4.9 full_adder.vhd File Reference	31
4.9.1 Detailed Description	31
4.10 half_adder.vhd File Reference	32
4.10.1 Detailed Description	32
4.11 scan_chain.vhd File Reference	32
4.11.1 Detailed Description	32
Index	33

Chapter 1

Class Index

1.1 Class List

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

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	5
entity booth_multiplier	6
entity booth_multiplier_control_unit	7
entity booth_multiplier_onBoard	7
entity BoothMultiplier_tb_onBoard	8
entity carrySelect_adder	9
entity carrySelect_addSub	9
entity carrySelect_cell	10
entity cathodes_manager	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 counter_modN	13
entity counter_UpMod2n_Re_Sr	13
entity display_7_segmenti	15
entity flipflop_d_risingEdge_asyncReset	
Flipflop_d_risingEdge_asyncReset implementa un flipflop di tipo d che commuta sul fronte di salita, con segnale di enable e reset asincrono	15
entity flipflopmux	17
entity full_adder	17
entity half_adder	18
entity mux2_1	
Definisco il componente e la sua interfaccia	19
entity overflow_checker	20
entity register_d_Re_Ar	
Registro di diensione "width" che prende in ingresso un dato D e lo memorizza	21
entity ripple_carry_adder	22
entity rippleCarry_adder	23
entity scan_chain	25

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

booth_multiplier_control_unit.vhd	
Unità di controllo del moltiplicatore di Booth	27
booth_On_Board.vhd	
Componente di alto livello per implementare il moltiplicatore di Booth su board	27
carrySelect_adder.vhd	
Sommatore Carry Select	28
carrySelect_addSub.vhd	
Sommatore Carry Select in grado di effettuare anche l'operazione di sottrazione	29
carrySelect_cell.vhd	
Singolo blocco di un sommatore carry Select	29
counter_modN.vhd	
Contatore modulo 2 alla N	30
display_7_segmenti.vhd	
Display a 7 segmenti	30
flipflopmux.vhd	
Flip flop D con multiplexer	31
full_adder.vhd	
Sommatore Full-Adder 3 ingressi 2 uscite	31
half_adder.vhd	
Sommatore half_adder 2 ingressi 2 uscite	32
scan_chain.vhd	
Registro di n flip flop D multiplexati	32

Chapter 3

Class Documentation

3.1 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](#)

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

3.1.1 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.

3.1.2 Member Data Documentation

3.1.2.1 IEEE

[IEEE](#) [Library]

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

3.1.2.2 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:

- [anodes_manager.vhd](#)

3.2 booth_multiplier Entity Reference

Libraries

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

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:

- [booth_multiplier.vhd](#)

3.3 booth_multiplier_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
- [counter_hit](#) in STD_LOGIC
segnala la fine della moltiplicazione
- [x_lsbs](#) in STD_LOGIC_VECTOR(**1** downto **0**)
ultimi due bit di x corrente
- [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
- [reset_a](#) out STD_LOGIC
- [reset_count](#) out STD_LOGIC
reset il conteggio

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

- [booth_multiplier_control_unit.vhd](#)

3.4 booth_multiplier_onBoard Entity Reference

Libraries

- [IEEE](#)

Use Clauses

- [STD_LOGIC_1164](#)

Generics

- **N NATURAL:= 4**

Ports

- [clock](#) in STD_LOGIC
- [start](#) in STD_LOGIC
- [start_led](#) out STD_LOGIC
- [stop](#) out STD_LOGIC
- [enable_a](#) in STD_LOGIC
- [enable_b](#) in STD_LOGIC
- [subtract](#) in STD_LOGIC
- [input](#) in STD_LOGIC_VECTOR(**3** downto **0**)
input addendo
- [overflow](#) out STD_LOGIC
- [c_out](#) out STD_LOGIC
- [anodes](#) out STD_LOGIC_VECTOR(**7** downto **0**)
- [leds](#) out STD_LOGIC_VECTOR(**2 * N - 1** downto **0**)
- [cathodes](#) out STD_LOGIC_VECTOR(**7** downto **0**)
output carry in uscita

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

- [booth_On_Board.vhd](#)

3.5 BoothMultiplier_tb_onBoard Entity Reference

Libraries

- [ieee](#)

Use Clauses

- [std_logic_1164](#)

3.5.1 Detailed Description

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

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

- BoothMultiplier_tb_onBoard.vhd

3.6 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](#)

3.7 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](#)

3.8 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](#)

3.9 cathodes_manager Entity Reference

Libraries

- [IEEE](#)

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

3.9.1 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.

3.9.2 Member Data Documentation

3.9.2.1 IEEE

[IEEE](#) [Library]

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

3.9.2.2 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:

- cathodes_manager.vhd

3.10 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](#)
architecture behavioral of [cathodes_manager](#) end

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

3.10.1 Detailed Description

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

3.10.2 Member Data Documentation

3.10.2.1 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:

- [clock_divisor.vhd](#)

3.11 counter_modN Entity Reference

Libraries

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

Use Clauses

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

Generics

- [count_max](#) integer:= **8**

Ports

- [clock](#) in STD_LOGIC
- [count_up](#) in STD_LOGIC
- [reset_n](#) in STD_LOGIC
- [value](#) out STD_LOGIC_VECTOR(((integer(ceil(log2(real(count_max)))))- **1** downto **0**)
- [hit](#) out STD_LOGIC

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

- [counter_modN.vhd](#)

3.12 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.

3.12.1 Detailed Description

Contatore modulo 2 alla N. Il conteggio viene effettuato sul fronte di salita del clock e il reset è sincrono.

3.12.2 Member Data Documentation

3.12.2.1 IEEE

[IEEE](#) [Library]

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

3.12.2.2 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:

- counter_UpMod2n_Re_Sr.vhd

3.13 display_7_segmenti Entity Reference

Libraries

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

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.

3.13.1 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](#)

3.14 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

3.14.1 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.

3.14.2 Member Data Documentation

3.14.2.1 IEEE

[IEEE](#) [Library]

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

3.14.2.2 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

3.15 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

3.15.1 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](#)

3.16 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

3.16.1 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](#)

3.17 half_adder Entity Reference

Libraries

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

Use Clauses

- [std_logic_1164](#)

Ports

- **x in std_logic**
- **y in std_logic**
- **s out std_logic**
- **c out std_logic**

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

- [half_adder.vhd](#)

3.18 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

3.18.1 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.

3.18.2 Member Data Documentation

3.18.2.1 IEEE

[IEEE](#) [Library]

3.18.2.2 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

3.19 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

3.19.1 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

3.19.2 Member Data Documentation

3.19.2.1 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](#)

3.20 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

3.20.1 Detailed Description

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

3.20.2 Member Data Documentation

3.20.2.1 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](#)

3.21 ripple_carry_adder Entity Reference

Libraries

- [ieee](#)
architecture behavioral of [register_d_Re_Ar](#) end

Use Clauses

- [std_logic_1164](#)

Generics

- [width](#) **natural**

Ports

- [X](#) in **std_logic_vector(width- 1 downto 0)**
- [Y](#) in **std_logic_vector(width- 1 downto 0)**
- [cin](#) in **std_logic**
- [cout](#) out **std_logic**
- [sum](#) out **std_logic_vector(width- 1 downto 0)**

3.21.1 Member Data Documentation

3.21.1.1 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:

- ripple_carry_adder.vhd

3.22 rippleCarry_adder Entity Reference

Libraries

- [IEEE](#)

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

3.22.1 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.

3.22.2 Member Data Documentation

3.22.2.1 c_in

`c_in` in STD_LOGIC [Port]

`rippleCarry_adder` input: addendo

3.22.2.2 c_out

`c_out` out STD_LOGIC [Port]

`rippleCarry_adder` output: carry

`rippleCarry_adder` output: somma

3.22.2.3 IEEE

`IEEE` [Library]

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

3.22.2.4 S

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

`rippleCarry_adder` input : carry in ingresso

3.22.2.5 STD_LOGIC_1164

`STD_LOGIC_1164` [Package]

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

3.22.2.6 width

`width` NATURAL := 8 [Generic]

usato per definire il parallelismo del `rippleCarry_adder`

3.22.2.7 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

3.23 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 temporizzazione
- **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

3.23.1 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 booth_multiplier_control_unit.vhd File Reference

Unità di controllo del moltiplicatore di Booth.

Entities

- [booth_multiplier_control_unit](#) entity

4.1.1 Detailed Description

Unità di controllo del moltiplicatore di Booth.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.2 booth_On_Board.vhd File Reference

Componente di alto livello per implementare il moltiplicatore di Booth su board.

Entities

- [booth_multiplier_onBoard](#) entity

4.2.1 Detailed Description

Componente di alto livello per implementare il moltiplicatore di Booth su board.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.3 carrySelect_adder.vhd File Reference

Sommatore Carry Select.

Entities

- [carrySelect_adder](#) entity

4.3.1 Detailed Description

Sommatore Carry Select.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.4 carrySelect_addSub.vhd File Reference

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

Entities

- [carrySelect_addSub](#) entity

4.4.1 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

4.5 carrySelect_cell.vhd File Reference

Singolo blocco di un sommatore carry Select.

Entities

- [carrySelect_cell](#) entity

4.5.1 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

4.6 counter_modN.vhd File Reference

Contatore modulo 2 alla N.

Entities

- [counter_modN](#) entity

4.6.1 Detailed Description

Contatore modulo 2 alla N.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.7 display_7_segmenti.vhd File Reference

Display a 7 segmenti.

Entities

- [display_7_segmenti](#) entity

4.7.1 Detailed Description

Display a 7 segmenti.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.8 flipflopmux.vhd File Reference

flip flop D con multiplexer

Entities

- [flipflopmux](#) entity

4.8.1 Detailed Description

flip flop D con multiplexer

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.9 full_adder.vhd File Reference

Sommatore Full-Adder 3 ingressi 2 uscite.

Entities

- [full_adder](#) entity

4.9.1 Detailed Description

Sommatore Full-Adder 3 ingressi 2 uscite.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.10 half_adder.vhd File Reference

Sommatore [half_adder](#) 2 ingressi 2 uscite.

Entities

- [half_adder](#) entity

4.10.1 Detailed Description

Sommatore [half_adder](#) 2 ingressi 2 uscite.

Author

Gabriele Previtera, Mirko Pennone, Simone Penna

Date

04/03/2019

Version

0.2

Dependencies:

Nothings

4.11 scan_chain.vhd File Reference

Registro di n flip flop D multiplexati.

Entities

- [scan_chain](#) entity

4.11.1 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

- anodes_manager, [5](#)
 - IEEE, [5](#)
 - STD_LOGIC_1164, [6](#)
- booth_On_Board.vhd, [27](#)
- booth_multiplier, [6](#)
- booth_multiplier_control_unit, [7](#)
- booth_multiplier_control_unit.vhd, [27](#)
- booth_multiplier_onBoard, [7](#)
- BoothMultiplier_tb_onBoard, [8](#)
- c_in
 - rippleCarry_adder, [23](#)
- c_out
 - rippleCarry_adder, [24](#)
- carrySelect_addSub, [9](#)
- carrySelect_addSub.vhd, [29](#)
- carrySelect_adder, [9](#)
- carrySelect_adder.vhd, [28](#)
- carrySelect_cell, [10](#)
- carrySelect_cell.vhd, [29](#)
- cathodes_manager, [11](#)
 - IEEE, [11](#)
 - STD_LOGIC_1164, [11](#)
- clock_divisor, [12](#)
 - STD_LOGIC_1164, [12](#)
- counter_UpMod2n_Re_Sr, [13](#)
 - IEEE, [14](#)
 - STD_LOGIC_1164, [14](#)
- counter_modN.vhd, [30](#)
- counter_modN, [13](#)
- display_7_segmenti, [15](#)
- display_7_segmenti.vhd, [30](#)
- flipflop_d_risingEdge_asyncReset, [15](#)
 - IEEE, [16](#)
 - STD_LOGIC_1164, [16](#)
- flipflopmux, [17](#)
- flipflopmux.vhd, [31](#)
- full_adder, [17](#)
- full_adder.vhd, [31](#)
- half_adder, [18](#)
- half_adder.vhd, [32](#)
- IEEE
 - anodes_manager, [5](#)
 - cathodes_manager, [11](#)
 - counter_UpMod2n_Re_Sr, [14](#)
 - flipflop_d_risingEdge_asyncReset, [16](#)
 - mux2_1, [19](#)
 - rippleCarry_adder, [24](#)
- mux2_1, [19](#)
 - IEEE, [19](#)
 - STD_LOGIC_1164, [19](#)
- overflow_checker, [20](#)
 - STD_LOGIC_1164, [21](#)
- register_d_Re_Ar, [21](#)
 - STD_LOGIC_1164, [22](#)
- ripple_carry_adder, [22](#)
 - std_logic_1164, [22](#)
- rippleCarry_adder, [23](#)
 - c_in, [23](#)
 - c_out, [24](#)
 - IEEE, [24](#)
 - S, [24](#)
 - STD_LOGIC_1164, [24](#)
 - width, [24](#)
 - Y, [24](#)
- S
 - rippleCarry_adder, [24](#)
- STD_LOGIC_1164
 - anodes_manager, [6](#)
 - cathodes_manager, [11](#)
 - clock_divisor, [12](#)
 - counter_UpMod2n_Re_Sr, [14](#)
 - flipflop_d_risingEdge_asyncReset, [16](#)
 - mux2_1, [19](#)
 - overflow_checker, [21](#)
 - register_d_Re_Ar, [22](#)
 - rippleCarry_adder, [24](#)
- scan_chain, [25](#)
- scan_chain.vhd, [32](#)
- std_logic_1164
 - ripple_carry_adder, [22](#)
- width
 - rippleCarry_adder, [24](#)
- Y
 - rippleCarry_adder, [24](#)