Echo uart

Generated by Doxygen 1.8.18

Chapter 1

Hierarchical Index

1.1 Design Unit Hierarchy

Here is a hierarchical list of all entities:

Chapter 2

Design Unit Index

2.1 Design Unit List

Here is a list of all design unit members with links to the Entities they belong to:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Behavioral Architecture Reference

Processes

- PROCESS_0(**Clk**)

Signals

- Clk **std_logic**
- Reset **std_logic**
- Rx **std_logic**

  signal for get
- Tx **std_logic**

  signal for transmitting
- RX_Data **std_logic_vector(DATA_WIDTH - 1 downto 0)**

  Inputs from uart_rx.
- RX_Ready **std_logic**

  Inputs from uart_rx.
- TX_Data **std_logic_vector(7 downto 0)**

  Inputs from uart_tx.
- TX_Ready **std_logic:=' 0 '**

  Inputs from uart_tx.
- TX_Start **std_logic**

  Outputs from uart_tx.

Instantiations

- uart_rx **UART_RX**

  declaration uart_rx
- uart_tx **uart_tx**

  declaration uart_tx

4.1.1 Member Function Documentation

4.1.1.1 PROCESS_0()

```
PROCESS_0 (
              Clk )
```

4.1.2 Member Data Documentation

4.1.2.1 Clk

```
Clk std_logic   [Signal]
```

4.1.2.2 Reset

```
Reset std_logic   [Signal]
```

4.1.2.3 Rx

```
Rx std_logic   [Signal]
```

signal for get

4.1.2.4 RX_Data

```
RX_Data std_logic_vector(DATA_WIDTH - 1 downto 0 )   [Signal]
```

Inputs from uart_rx.

4.1.2.5 RX_Ready

```
RX_Ready std_logic   [Signal]
```

Inputs from uart_rx.

4.1.2.6 Tx

`Tx` **std_logic** [Signal]

signal for transmitting

4.1.2.7 TX_Data

`TX_Data` **std_logic_vector(7 downto 0)** [Signal]

Inputs from uart_tx.

4.1.2.8 TX_Ready

`TX_Ready` **std_logic:=' 0 '** [Signal]

Inputs from uart_tx.

4.1.2.9 TX_Start

`TX_Start` **std_logic** [Signal]

Outputs from uart_tx.

4.1.2.10 uart_rx

`uart_rx` **UART_RX** [Instantiation]

declaration uart_rx

4.1.2.11 uart_tx

`uart_tx` **uart_tx** [Instantiation]

declaration uart_tx

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

- main.vhd

4.2 Behavioral Architecture Reference

Processes

- RX_PROCESS(**Clk** , **Reset**)
  waiting for data frame
- TO_OUTPUT(**Clk**)
  receiving data frame

Constants

- MAX_FREQ_COUNT **positive:=CLK_FREQUENCY /BAUD**
  length of one bit in clock cycles

Signals

- freq_count **naturalrange 0 toMAX_FREQ_COUNT - 1**
  used for counting clock cycles
- count **naturalrange 0 toDATA_WIDTH + 2**
  counting received bits
- last_Rx **std_logic**
  temporarily keeps last state of Rx input
- receiving **std_logic:=' 0 '**
  determinates if process is in receiving state
- data_buf **std_logic_vector(0 toDATA_WIDTH + 2)**
  buffer for incoming uart frame, 'to' is used for reverse trick on output assignment
- data_ready **std_logic:=' 0 '**
  determinates if data is ready to send to the output

4.2.1 Member Function Documentation

4.2.1.1 RX_PROCESS()

```
RX_PROCESS(
              Clk   ,
              Reset   )  [Process]
```

waiting for data frame

4.2.1.2 TO_OUTPUT()

```
TO_OUTPUT(
              Clk   )  [Process]
```

receiving data frame

4.2.2 Member Data Documentation

4.2.2.1 count

```
count naturalrange 0 toDATA_WIDTH + 2    [Signal]
```

counting received bits

4.2.2.2 data_buf

```
data_buf std_logic_vector( 0 toDATA_WIDTH + 2 )    [Signal]
```

buffer for incoming uart frame, 'to' is used for reverse trick on output assignment

4.2.2.3 data_ready

```
data_ready std_logic:=' 0 '    [Signal]
```

determinates if data is ready to send to the output

4.2.2.4 freq_count

```
freq_count naturalrange 0 toMAX_FREQ_COUNT - 1    [Signal]
```

used for counting clock cycles

4.2.2.5 last_Rx

```
last_Rx std_logic    [Signal]
```

temporarily keeps last state of Rx input

4.2.2.6 MAX_FREQ_COUNT

```
MAX_FREQ_COUNT positive:=CLK_FREQUENCY /BAUD    [Constant]
```

length of one bit in clock cycles

4.2.2.7 receiving

```
receiving std_logic:=' 0 '   [Signal]
```

determinates if process is in receiving state

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

- uart_rx.vhd

4.3 Behavioral Architecture Reference

Processes

- TX_PROCESS(**Clk** , **Reset**)

Constants

- MAX_FREQ_COUNT **positive:=CLK_FREQUENCY /BAUD**
  length of one bit in clock cycles

Signals

- freq_count **naturalrange 0 toMAX_FREQ_COUNT - 1**
  used for counting clock cycles
- count **naturalrange 0 to 11 := 11**
  counting received bits

4.3.1 Member Function Documentation

4.3.1.1 TX_PROCESS()

```
TX_PROCESS(
            Clk   ,
            Reset   )  [Process]
```

4.3.2 Member Data Documentation

4.3.2.1 count

```
count naturalrange 0 to 11 := 11    [Signal]
```

counting received bits

4.3.2.2 freq_count

```
freq_count naturalrange 0 toMAX_FREQ_COUNT - 1    [Signal]
```

used for counting clock cycles

4.3.2.3 MAX_FREQ_COUNT

```
MAX_FREQ_COUNT positive:=CLK_FREQUENCY /BAUD    [Constant]
```

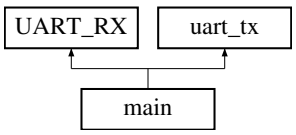
length of one bit in clock cycles

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

- uart_tx.vhd

4.4 main Entity Reference

Inheritance diagram for main:



Entities

- Behavioral architecture

Libraries

- IEEE

  use standard library

Use Clauses

- STD_LOGIC_1164

  use logic elements
- NUMERIC_STD

  use numeric elements

Generics

- CLK_FREQUENCY **positive:=** 12000000

  clock frequency
- DATA_WIDTH **positive:=** 8

  UART message length.
- BAUD **positive:=** 19200

  UART baud rate.

Ports

- Clk_input **in std_logic_vector(0 downto 0)**
- Reset_input **in std_logic_vector(0 downto 0)**
- Rx_input **in std_logic_vector(0 downto 0)**

  RX pin to get signals.
- Tx_output **out std_logic_vector(0 downto 0)**

  Tx pin for transmitting.

4.4.1 Member Data Documentation

4.4.1.1 BAUD

```
BAUD  positive:= 19200    [Generic]
```

UART baud rate.

4.4.1.2 CLK_FREQUENCY

```
CLK_FREQUENCY  positive:= 12000000    [Generic]
```

clock frequency

4.4.1.3 Clk_input

```
Clk_input in std_logic_vector( 0 downto 0 )   [Port]
```

4.4.1.4 DATA_WIDTH

```
DATA_WIDTH  positive:= 8    [Generic]
```

UART message length.

4.4.1.5 IEEE

```
IEEE  [Library]
```

use standard library

4.4.1.6 NUMERIC_STD

```
NUMERIC_STD  [use clause]
```

use numeric elements

4.4.1.7 Reset_input

```
Reset_input in std_logic_vector( 0 downto 0 )   [Port]
```

4.4.1.8 Rx_input

```
Rx_input in std_logic_vector( 0 downto 0 )   [Port]
```

RX pin to get signals.

4.4.1.9 STD_LOGIC_1164

```
STD_LOGIC_1164  [use clause]
```

use logic elements

4.4.1.10 Tx_output

```
Tx_output out std_logic_vector( 0 downto 0 )   [Port]
```

Tx pin for transmitting.

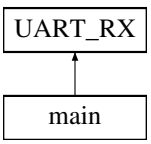
The documentation for this class was generated from the following file:

- main.vhd

4.5 UART_RX Entity Reference

Definition of UART RX.

Inheritance diagram for UART_RX:



Entities

- Behavioral architecture

Libraries

- IEEE

  use standard library

Use Clauses

- STD_LOGIC_1164

  use logic elements
- NUMERIC_STD

  use numeric elements

Generics

- CLK_FREQUENCY **positive:= 12000000**
- DATA_WIDTH **positive:= 8**

  UART message length.
- BAUD **positive:= 19200**

  UART baud rate.

Ports

- Clk **in std_logic**
- Reset **in std_logic**
- Rx **in std_logic**

  Rx pin for receiving.
- RX_Data_Out **out std_logic_vector(DATA_WIDTH - 1 downto 0)**

  received data
- RX_Ready **out std_logic:=' 0 '**

  determinates if data on output is ready

4.5.1 Detailed Description

Definition of UART RX.

4.5.2 Member Data Documentation

4.5.2.1 BAUD

```
BAUD  positive:= 19200    [Generic]
```

UART baud rate.

4.5.2.2 Clk

```
Clk in std_logic   [Port]
```

4.5.2.3 CLK_FREQUENCY

```
CLK_FREQUENCY  positive:= 12000000    [Generic]
```

4.5.2.4 DATA_WIDTH

```
DATA_WIDTH  positive:= 8    [Generic]
```

UART message length.

4.5.2.5 IEEE

```
IEEE  [Library]
```

use standard library

4.5.2.6 NUMERIC_STD

```
NUMERIC_STD  [use clause]
```

use numeric elements

4.5.2.7 Reset

```
Reset in std_logic   [Port]
```

4.5.2.8 Rx

```
Rx in std_logic   [Port]
```

Rx pin for receiving.

4.5.2.9 RX_Data_Out

```
RX_Data_Out out std_logic_vector(DATA_WIDTH - 1 downto 0 )   [Port]
```

received data

4.5.2.10 RX_Ready

```
RX_Ready out std_logic:=' 0 '   [Port]
```

determinates if data on output is ready

4.5.2.11 STD_LOGIC_1164

```
STD_LOGIC_1164  [use clause]
```

use logic elements

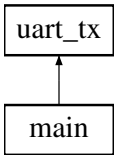
The documentation for this class was generated from the following file:

- uart_rx.vhd

4.6 uart_tx Entity Reference

Definition of UART TX.

Inheritance diagram for uart_tx:



Entities

- Behavioral architecture

Libraries

- IEEE

  use standart library

Use Clauses

- STD_LOGIC_1164

  use logic elements
- NUMERIC_STD

  use numeric elements

Generics

- CLK_FREQUENCY **positive:=** 12000000
- DATA_WIDTH **positive:=** 8

  UART message length.
- BAUD **positive:=** 19200

  UART baud rate.

Ports

- Clk **in std_logic**
- Reset **in std_logic**
- TX_Data_In **in std_logic_vector(DATA_WIDTH - 1 downto 0)**

  data to transmit
- TX_Ready **in std_logic**

  definition when new data come
- Tx **out std_logic:=' 1 '**

  Tx pin for transmitting.
- TX_Start **out std_logic:=' 1 '**

  definition when new data can come

4.6.1 Detailed Description

Definition of UART TX.

4.6.2 Member Data Documentation

4.6.2.1 BAUD

```
BAUD  positive:= 19200    [Generic]
```

UART baud rate.

4.6.2.2 Clk

```
Clk in std_logic   [Port]
```

4.6.2.3 CLK_FREQUENCY

```
CLK_FREQUENCY  positive:= 12000000    [Generic]
```

4.6.2.4 DATA_WIDTH

```
DATA_WIDTH  positive:= 8    [Generic]
```

UART message length.

4.6.2.5 IEEE

```
IEEE  [Library]
```

use standart library

4.6.2.6 NUMERIC_STD

```
NUMERIC_STD  [use clause]
```

use numeric elements

4.6.2.7 Reset

```
Reset in std_logic   [Port]
```

4.6.2.8 STD_LOGIC_1164

```
STD_LOGIC_1164  [use clause]
```

use logic elements

4.6.2.9 Tx

```
Tx out std_logic:=' 1 '   [Port]
```

Tx pin for transmitting.

4.6.2.10 TX_Data_In

```
TX_Data_In in std_logic_vector(DATA_WIDTH - 1 downto 0 )   [Port]
```

data to transmit

4.6.2.11 TX_Ready

```
TX_Ready in std_logic   [Port]
```

definition when new data come

4.6.2.12 TX_Start

```
TX_Start out std_logic:=' 1 '   [Port]
```

definition when new data can come

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

- uart_tx.vhd

Chapter 5

File Documentation

5.1 elbertv2_pin.ucf File Reference

Constraints

- VCCAUX " 3 . 3 "
- "Clk_input[0]" LOC=P129|IOSTANDARD=LVCMOS33|PERIOD=12MHz
- "Rx_input[0]" LOC=P125|IOSTANDARD=LVCMOS33|SLEW=SLOW|DRIVE= 12
- "Tx_output[0]" LOC=P127|IOSTANDARD=LVCMOS33|SLEW=SLOW|DRIVE= 12

5.1.1 Variable Documentation

5.1.1.1 ""Clk_input[0]""

```
[Constraints]
```

5.1.1.2 ""Rx_input[0]""

```
[Constraints]
```

5.1.1.3 ""Tx_output[0]""

```
[Constraints]
```

5.1.1.4 VCCAUX

```
[Constraints]
```

5.2 main.vhd File Reference

Entities

- main entity
- Behavioral architecture

5.3 uart_rx.vhd File Reference

Entities

- UART_RX entity

  Definition of UART RX.
- Behavioral architecture

5.4 uart_tx.vhd File Reference

Entities

- uart_tx entity

  Definition of UART TX.
- Behavioral architecture

Index