

# tb\_coctb.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2024/12/10

---

## INFORMATION

---

### Brief

---

Test bench wrapper for cocotb

### License MIT

---

Copyright 2024 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## tb\_cocotb

---

```
module tb_cocotb #(
  parameter
  PARITY_ENA
  =
  0,
  parameter
  PARITY_TYPE
  =
  1,
  parameter
  STOP_BITS
  =
  1,
  parameter
```

```

DATA_BITS
=
8,
parameter
DELAY
=
0,
parameter
BUS_WIDTH
=
1
) ( input aclk, input arstn, input [BUS_WIDTH*8-1:0] s_axis_tdata, input s_

```

Test bench for AXIS UART TX, simple UART TX from AXI Streaming interface.

### Parameters

<b>PARITY_ENA</b> parameter	Enable Parity for the data in and out.
<b>PARITY_TYPE</b> parameter	Set the parity type, 0 = even, 1 = odd, 2 = mark, 3 = space.
<b>STOP_BITS</b> parameter	Number of stop bits, 0 to crazy non-standard amounts.
<b>DATA_BITS</b> parameter	Number of data bits, 1 to crazy non-standard amounts.
<b>DELAY</b> parameter	Delay in tx data output. Delays the time to output of the data.
<b>BUS_WIDTH</b> parameter	BUS_WIDTH for axis bus in bytes.

### Ports

<b>aclk</b>	Clock for AXIS
<b>arstn</b>	Negative reset for AXIS
<b>s_axis_tdata</b>	Input data for UART TX.
<b>s_axis_tvalid</b>	When set active high the input data is valid
<b>s_axis_tready</b>	When active high the device is ready for input data.
<b>uart_clk</b>	Clock used for BAUD rate generation
<b>uart_rstn</b>	Negative reset for UART, for anything clocked on uart_clk
<b>uart_ena</b>	When active high enable UART transmit state.
<b>uart_hold</b>	Output to hold back clock in reset state till uart is in transmit state.
<b>txd</b>	transmit for UART (output to RX)

## INSTANTIATED MODULES

### dut

```

axis_uart_tx #(
    PARITY_ENA(PARITY_ENA),
    PARITY_TYPE(PARITY_TYPE),
    STOP_BITS(STOP_BITS),

```

```
DATA_BITS(DATA_BITS),  
DELAY(DELAY),  
BUS_WIDTH(BUS_WIDTH)  
) dut ( .aclk(aclk), .arstn(arstn), .s_axis_tdata(s_axis_tdata), .s_axis_tva
```

Device under test, axis\_uart\_tx