

# tb\_cocotb\_up.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/04/01

---

## INFORMATION

---

### Brief

---

Test bench wrapper for cocotb

### License MIT

---

Copyright 2025 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.BUS\_WIDTH

## tb\_cocotb

---

```
module tb_cocotb #(
  parameter
    ADDRESS_WIDTH
    =
    32,
  parameter
    BUS_WIDTH
    =
    4,
  parameter
    CLOCK_SPEED
    =
    100000000,
  parameter
```

```

    BAUD_RATE
    =
    115200,
    parameter
    PARITY_TYPE
    =
    0,
    parameter
    STOP_BITS
    =
    1,
    parameter
    DATA_BITS
    =
    8,
    parameter
    RX_BAUD_DELAY
    =
    0,
    parameter
    TX_BAUD_DELAY
    =
    0
) ( input clk, input rstn, input up_rreq, output up_rack, input [ADDRESS_WIDTH-1:0] up_addr, output [DATA_WIDTH-1:0] up_data )

```

uP UART testbench

## Parameters

<b>ADDRESS_WIDTH</b> parameter	Width of the axi address bus
<b>BUS_WIDTH</b> parameter	Number of bytes for the data bus.
<b>CLOCK_SPEED</b> parameter	This is the aclk frequency in Hz
<b>BAUD_RATE</b> parameter	Serial Baud, this can be any value including non-standard.
<b>PARITY_TYPE</b> parameter	Set the parity type, 0 = none, 1 = even, 2 = odd, 3 = mark, 4 = 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>RX_BAUD_DELAY</b> parameter	Delay in rx baud enable. This will delay when we sample a bit (default is midpoint when rx delay is 0).
<b>TX_BAUD_DELAY</b> parameter	Delay in tx baud enable. This will delay the time the bit output starts.

## Ports

<b>clk</b>	Clock for all devices in the core
<b>rstn</b>	Negative reset
<b>up_rreq</b>	uP bus read request
<b>up_rack</b>	uP bus read ack
<b>up_raddr</b>	uP bus read address
<b>up_rdata</b>	uP bus read data
<b>up_wreq</b>	uP bus write request
<b>up_wack</b>	uP bus write ack

<b>up_waddr</b>	uP bus write address
<b>up_wdata</b>	uP bus write data
<b>irq</b>	Interrupt when data is received
<b>tx</b>	transmit for UART (output to RX)
<b>rx</b>	receive for UART (input from TX)

## INSTANTIATED MODULES

---

### dut

---

```
up_uart_lite #(
    ADDRESS_WIDTH(ADDRESS_WIDTH),
    BUS_WIDTH(BUS_WIDTH),
    CLOCK_SPEED(CLOCK_SPEED),
    BAUD_RATE(BAUD_RATE),
    PARITY_TYPE(PARITY_TYPE),
    STOP_BITS(STOP_BITS),
    DATA_BITS(DATA_BITS),
    RX_BAUD_DELAY(RX_BAUD_DELAY),
    TX_BAUD_DELAY(TX_BAUD_DELAY)
) dut ( .clk(clk), .rstn(rstn), .up_rreq(up_rreq), .up_rack(up_rack), .up_ra
```

Device under test, up\_uart\_lite