

# tb\_coctb.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/03/26

---

## 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
    =
    16,
  parameter
    BUS_WIDTH
    =
    4
) ( input clk, input rstn, input [ADDRESS_WIDTH-1:0] s_apb_paddr, input [0:0] s_apb_wdata,
```

APB3 slave to uP interface DUT

Parameters

<b>ADDRESS_WIDTH</b> <small>parameter</small>	Width of the APB3 address port in bits.
<b>BUS_WIDTH</b> <small>parameter</small>	Width of the APB3 bus data port in bytes.

Ports

<b>clk</b>	Clock
<b>rstn</b>	negative reset
<b>s_apb_paddr</b>	APB3 address bus, up to 32 bits wide.
<b>s_apb_psel</b>	APB3 select per slave (1 for this core).
<b>s_apb_penable</b>	APB3 enable device for multiple transfers after first.
<b>s_apb_pready</b>	APB3 ready is a output from the slave to indicate its able to process the request.
<b>s_apb_pwrite</b>	APB3 Direction signal, active high is a write access. Active low is a read access.
<b>s_apb_pwdata</b>	APB3 write data port.
<b>s_apb_prdata</b>	APB3 read data port.
<b>s_apb_pslverror</b>	APB3 error indicates transfer failure, not implimented.
<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

INSTANTIATED MODULES

---

dut

```
up_apb3 #(
    ADDRESS_WIDTH(ADDRESS_WIDTH),
    BUS_WIDTH(BUS_WIDTH)
) dut ( .clk(clk), .rstn(rstn), .s_apb_paddr(s_apb_paddr), .s_apb_psel(s_apb_psel), .s_apb_penable(s_apb_penable), .s_apb_pready(s_apb_pready), .s_apb_pwrite(s_apb_pwrite), .s_apb_pwdata(s_apb_pwdata), .s_apb_prdata(s_apb_prdata), .s_apb_pslverror(s_apb_pslverror), .up_rreq(up_rreq), .up_rack(up_rack), .up_raddr(up_raddr), .up_rdata(up_rdata), .up_wreq(up_wreq), .up_wack(up_wack), .up_waddr(up_waddr), .up_wdata(up_wdata))
```

Device under test, up\_apb3