

tb_cocotb.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
    =
    16,
  parameter
    BUS_WIDTH
    =
    4
) ( input clk, input rst, output rstn, input s_wb_cyc, input s_wb_stb, input
```

Wishbone Classic slave to uP up_wishbone_classic DUT

Parameters

ADDRESS_WIDTH parameter	Width of the Wishbone address port in bits.
BUS_WIDTH parameter	Width of the Wishbone bus data port in bytes.

Ports

clk	Clock
rst	Positive reset
s_wb_cyc	Bus Cycle in process
s_wb_stb	Valid data transfer cycle
s_wb_we	Active High write, low read
s_wb_addr	Bus address
s_wb_data_i	Input data
s_wb_sel	Device Select
s_wb_ack	Bus transaction terminated
s_wb_data_o	Output data
s_wb_err	Active high when a bus error is present
up_rreq	uP bus read request
up_rack	uP bus read ack
up_raddr	uP bus read address
up_rdata	uP bus read data
up_wreq	uP bus write request
up_wack	uP bus write ack
up_waddr	uP bus write address
up_wdata	uP bus write data

INSTANTIATED MODULES

dut

```
up_wishbone_standard #(
    ADDRESS_WIDTH(ADDRESS_WIDTH),
    BUS_WIDTH(BUS_WIDTH)
) dut ( .clk(clk), .rst(rst), .s_wb_cyc(s_wb_cyc), .s_wb_stb(s_wb_stb), .s_wb_we(s_wb_we), .s_wb_addr(s_wb_addr), .s_wb_data_i(s_wb_data_i), .s_wb_sel(s_wb_sel), .s_wb_ack(s_wb_ack), .s_wb_data_o(s_wb_data_o), .s_wb_err(s_wb_err), .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_wishbone_standard