

# tb\_cocotb\_up.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/04/29

---

## 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
    WORD_WIDTH
    =
    4,
  parameter
```

```
CLOCK_SPEED
=
100000000,
parameter
SELECT_WIDTH
=
16,
parameter
DEFAULT_RATE_DIV
=
0,
parameter
DEFAULT_CPOL
=
0,
parameter
DEFAULT_CPHA
=
0
) ( input clk, input rstn, input up_rreq, output up_rack, input [ADDRESS_WID
```

SPI Master core with axis input/output data. Read/Write is size of BUS\_WIDTH bytes. Write activates core for read.

## Parameters

<b>ADDRESS_WIDTH</b> parameter	Width of the uP address port, max 32 bit.
<b>BUS_WIDTH</b> parameter	Width of the uP bus data port(can not be less than 2 bytes, max tested is 4).
<b>WORD_WIDTH</b> parameter	Width of each SPI Master word. This will also set the bits used in the TX/RX data registers. Must be less than or equal to BUS_WIDTH 1 to 4.
<b>CLOCK_SPEED</b> parameter	This is the ack frequency in Hz, this is the the frequency used for the bus and is divided by the rate.
<b>SELECT_WIDTH</b> parameter	Bit width of the slave select, defaults to 16 to match altera spi ip.
<b>DEFAULT_RATE_DIV</b> parameter	Default divider value of the main clock to use for the spi data output clock rate. 0 is 2 ( $2^{(X+1)}$ X is the DEFAULT_RATE_DIV)
<b>DEFAULT_CPOL</b> parameter	Default clock polarity for the core (0 or 1).
<b>DEFAULT_CPHA</b> parameter	Default clock phase for the core (0 or 1).

## 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>sclk</b>	spi clock, should only drive output pins to devices.

<b>mosi</b>	transmit for master output
<b>miso</b>	receive for master input
<b>ss_n</b>	slave select output

## INSTANTIATED MODULES

**dut**

```

up_spi_master #(
    ADDRESS_WIDTH(ADDRESS_WIDTH),
    BUS_WIDTH(BUS_WIDTH),
    WORD_WIDTH(WORD_WIDTH),
    CLOCK_SPEED(CLOCK_SPEED),
    SELECT_WIDTH(SELECT_WIDTH),
    DEFAULT_RATE_DIV(DEFAULT_RATE_DIV),
    DEFAULT_CPOL(DEFAULT_CPOL),
    DEFAULT_CPHA(DEFAULT_CPHA)
) dut ( .clk(clk), .rstn(rstn), .up_rreq(up_rreq), .up_rack(up_rack), .up_rack

```

Device under test, up\_spi\_master