

# tb\_cocotb\_axi\_lite.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/04/30

---

## 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
aclk,
input
arstn,
input
s_axi_awvalid,
input
[ADDRESS_WIDTH-1:0]
s_axi_awaddr,
input
[ 2:0]
s_axi_awprot,
output
s_axi_awready,
input
s_axi_wvalid,
input
[(BUS_WIDTH*8)-1:0]
s_axi_wdata,
input
[ 3:0]
s_axi_wstrb,
output
s_axi_wready,
output
s_axi_bvalid,
output
[ 1:0]
s_axi_bresp,
input
s_axi_bready,
input
s_axi_arvalid,
input
[ADDRESS_WIDTH-1:0]
s_axi_araddr,
input
[ 2:0]
s_axi_arprot,
output
s_axi_arready,
output
s_axi_rvalid,
output
[(BUS_WIDTH*8)-1:0]

```

(

```
s_axi_rdata,  
output  
[ 1:0]  
s_axi_rresp,  
input  
s_axi_rready,  
output  
irq,  
output  
sclk,  
output  
mosi,  
input  
miso,  
output  
[SELECT_WIDTH-1:0]  
ss_n  
)
```

AXI Lite based SPI Master device.

Parameters

ADDRESS_WIDTH parameter	Width of the uP address port, max 32 bit.
BUS_WIDTH parameter	Width of the uP bus data port, only valid values are 2 or 4.
WORD_WIDTH 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.
CLOCK_SPEED parameter	This is the aclk frequency in Hz, this is the the frequency used for the bus and is divided by the rate.
SELECT_WIDTH parameter	Bit width of the slave select, defaults to 16 to match altera spi ip.
DEFAULT_RATE_DIV 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)
DEFAULT_CPOL parameter	Default clock polarity for the core (0 or 1).
DEFAULT_CPHA parameter	Default clock phase for the core (0 or 1).

Ports

aclk input	Clock for all devices in the core
arstn input	Negative reset
s_axi_awvalid input	Axi Lite aw valid
s_axi_awaddr input [ADDRESS_WIDTH- 1:0]	Axi Lite aw addr
s_axi_awprot input [2:0]	Axi Lite aw prot
s_axi_awready output [2:0]	Axi Lite aw ready
s_axi_wvalid input [2:0]	Axi Lite w valid
s_axi_wdata input [(BUS_WIDTH* 8)- 1:0]	Axi Lite w data

<b>s_axi_wstrb</b> input [3:0]	Axi Lite w strb
<b>s_axi_wready</b> output [3:0]	Axi Lite w ready
<b>s_axi_bvalid</b> output [3:0]	Axi Lite b valid
<b>s_axi_bresp</b> output [1:0]	Axi Lite b resp
<b>s_axi_bready</b> input [1:0]	Axi Lite b ready
<b>s_axi_arvalid</b> input [1:0]	Axi Lite ar valid
<b>s_axi_araddr</b> input [ADDRESS_WIDTH- 1:0]	Axi Lite ar addr
<b>s_axi_arprot</b> input [2:0]	Axi Lite ar prot
<b>s_axi_arready</b> output [2:0]	Axi Lite ar ready
<b>s_axi_rvalid</b> output [2:0]	Axi Lite r valid
<b>s_axi_rdata</b> output [(BUS_WIDTH* 8)- 1:0]	Axi Lite r data
<b>s_axi_rresp</b> output [1:0]	Axi Lite r resp
<b>s_axi_rready</b> input [1:0]	Axi Lite r ready
<b>irq</b> output [1:0]	Interrupt when data is received
<b>sclk</b> output [1:0]	spl clock, should only drive output pins to devices.
<b>mosi</b> output [1:0]	transmit for master output
<b>miso</b> input [1:0]	receive for master input
<b>ss_n</b> output [SELECT_WIDTH- 1:0]	slave select output

## INSTANTIATED MODULES

---

### dut

---

Device under test, axi\_lite\_spi\_master