tb_wishbone_cocotb.v

AUTHORS

JAY CONVERTINO

DATES

2024/12/10

INFORMATION

Brief

Test bench wrapper for cocotb

License MIT

tb_cocotb

```
module tb_cocotb #(
parameter
ADDRESS_WIDTH
 =
32,
parameter
BUS_WIDTH
 =
4,
parameter
DEPTH
 =
512,
parameter
```

```
RAM_TYPE
  =
"block",
parameter
HEX_FILE
  =
""
) ( input clk, input rst, input s_wb_cyc, input s_wb_stb, input s_wb_we, inp
```

Test bench for wishbone.

Parameters

ADDRESS_WIDTH	Width of the axi address bus in bits.	parameter
BUS_WIDTH	Bus width for data paths in bytes.	parameter
DEPTH	Depth of the RAM in terms of data width words.	parameter
RAM_TYPE	Used to set the ram_style atribute.	parameter
HEX_FILE	Hex file to write to RAM.	parameter

Ports

clk	Clock for all devices in the core
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_bte	Burst Type Extension
s_wb_cti	Cycle Type
s_wb_ack	Bus transaction terminated
s_wb_data_o	Output data
s_wb_err	Active high when a bus error is present

INSTANTIATED MODULES

dut

```
wishbone_classic_block_ram #(
                                                                    .
ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                    .
BUS_WIDTH(BUS_WIDTH),
                                                                    .
DEPTH(DEPTH),
                                                                    .
RAM_TYPE(RAM_TYPE),
```

```
                                                                        .
 HEX_FILE(HEX_FILE)
) dut ( .clk(clk), .rst(rst), .s_wb_cyc(s_wb_cyc), .s_wb_stb(s_wb_stb), .s_w
```

Device under test, wishbone_classic_block_ram