tb_coctb.v

AUTHORS

JAY CONVERTINO

DATES

2025/01/23

INFORMATION

Brief

Test bench wrapper for cocotb

License MIT

tb_cocotb

```
module tb_cocotb #(
parameter
PARITY_ENA
 =
0,
parameter
PARITY_TYPE
 =
0,
parameter
STOP_BITS
 =
1,
parameter
```

```
DATA_BITS
 =
8,
parameter
DELAY
 =
0,
parameter
BUS_WIDTH
 =
1
) ( input aclk, input arstn, output parity_err, output frame_err, output [BU
```

Test bench for axis_uart_rx.

Parameters

PARITY_ENA parameter	Enable Parity for the data in and out.
PARITY_TYPE parameter	Set the parity type, 0 = even, 1 = odd, 2 = mark, 3 = space.
STOP_BITS parameter	Number of stop bits, 0 to crazy non-standard amounts.
DATA_BITS parameter	Number of data bits, 1 to crazy non-standard amounts.
DELAY parameter	Delay in rx data input.
BUS_WIDTH parameter	BUS_WIDTH for axis bus in bytes.

Ports

aclk	Clock for AXIS
arstn	Negative reset for AXIS
parity_err	Indicates error with parity check (active high)
frame_err	Indicates error with frame (active high)
m_axis_tdata	Output data from UART RX
m_axis_tvalid	When active high the output data is valid
m_axis_tready	When set active high the output device is ready for data.
uart_clk	Clock used for BAUD rate generation
uart_rstn	Negative reset for UART, for anything clocked on uart_clk
uart_ena	Enable UART data processing from RX.
uart_hold	Output to hold back clock in reset state till uart is in receive state.
rxd	receive for UART (input from TX)

INSTANTIATED MODULES

dut

```
axis_uart_rx #(
                                                                    .
PARITY_ENA(PARITY_ENA),
                                                                    .
```

```
 PARITY_TYPE(PARITY_TYPE),                                                    .
 STOP_BITS(STOP_BITS),                                                        .
 DATA_BITS(DATA_BITS),                                                        .
 DELAY(DELAY),                                                                .
 BUS_WIDTH(BUS_WIDTH)
) dut ( .aclk(aclk), .arstn(arstn), .parity_err(parity_err), .frame_err(fram
```

Device under test, axis_uart_rx