wishbone_standard_uart.v

AUTHORS

JAY CONVERTINO

DATES

2024/02/29

INFORMATION

Brief

AXI Lite 1553 is a core for interfacing with 1553 devices over the AXI lite bus.

License MIT

wishbone_standard_uart

```
module wishbone_standard_uart #(
parameter
ADDRESS_WIDTH
 =
32,
parameter
BUS_WIDTH
 =
4,
parameter
CLOCK_SPEED
 =
100000000,
parameter
```

```
BAUD_RATE
 =
115200,
parameter
PARITY_ENA
 =
0,
parameter
PARITY_TYPE
 =
0,
parameter
STOP_BITS
 =
1,
parameter
DATA_BITS
 =
8,
parameter
RX_DELAY
 =
0,
parameter
RX_BAUD_DELAY
 =
0,
parameter
TX_DELAY
 =
0,
parameter
TX_BAUD_DELAY
 =
0
) ( input clk, input rst, input s_wb_cyc, input s_wb_stb, input s_wb_we, inp
```

Wishbone Standard based uart device.

Parameters

ADDRESS_WIDTH — Width of the address bus in bits.
parameter

BUS_WIDTH — Width of the data bus in bytes.
parameter

CLOCK_SPEED — This is the aclk frequency in Hz
parameter

BAUD_RATE — Serial Baud, this can be any value including non-standard.
parameter

PARITY_ENA — Enable Parity for the data in and out.
parameter

PARITY_TYPE — Set the parity type, 0 = even, 1 = odd, 2 = mark, 3 = space.
parameter

STOP_BITS — Number of stop bits, 0 to crazy non-standard amounts.
parameter

DATA_BITS — Number of data bits, 1 to crazy non-standard amounts.
parameter

RX_DELAY — Delay in rx data input.
parameter

RX_BAUD_DELAY — Delay in rx baud enable. This will delay when we sample a bit (default is midpoint when rx delay is 0).
parameter

TX_DELAY parameter	Delay in tx data output. Delays the time to output of the data.
TX_BAUD_DELAY parameter	Delay in tx baud enable. This will delay the time the bit output starts.

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_ack	Bus transaction terminated
s_wb_data_o	Output data
s_wb_err	Active high when a bus error is present
irq	Interrupt when data is received
tx	transmit for UART (output to RX)
rx	receive for UART (input from TX)
rts	request to send is a loop with CTS
cts	clear to send is a loop with RTS

up_rreq

```
wire up_rreq
```

uP read bus request

up_rack

```
wire up_rack
```

uP read bus acknowledge

up_raddr

```
wire [ADDRESS_WIDTH-(
BUS_WIDTH
                                                                /
2
)-1:0] up_raddr
```

uP read bus address

up_rdata

```
wire [31:0] up_rdata
```

uP read bus request

up_wreq

```
wire up_wreq
```

uP write bus request

up_wack

```
wire up_wack
```

uP write bus acknowledge

up_waddr

```
wire [ADDRESS_WIDTH-(
BUS_WIDTH
                                                                    /
2
)-1:0] up_waddr
```

uP write bus address

up_wdata

```
wire [31:0] up_wdata
```

uP write bus data

INSTANTIANTED MODULES

inst_up_wishbone_standard

```
up_wishbone_standard #(
                                                                    .
ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                    .
BUS_WIDTH(BUS_WIDTH)
) inst_up_wishbone_standard ( .clk(clk), .rst(rst), .s_wb_cyc(s_wb_cyc), .s_
```

Module instance of up_wishbone_standard for the Wishbone Classic Standard bus to the uP bus.

inst_up_uart

```
up_uart #(
                                                                                .
ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                                .
BUS_WIDTH(BUS_WIDTH),
                                                                                .
CLOCK_SPEED(CLOCK_SPEED),
                                                                                .
BAUD_RATE(BAUD_RATE),
                                                                                .
PARITY_ENA(PARITY_ENA),
                                                                                .
PARITY_TYPE(PARITY_TYPE),
                                                                                .
STOP_BITS(STOP_BITS),
                                                                                .
DATA_BITS(DATA_BITS),
                                                                                .
RX_DELAY(RX_DELAY),
                                                                                .
RX_BAUD_DELAY(RX_BAUD_DELAY),
                                                                                .
TX_DELAY(TX_DELAY),
                                                                                .
TX_BAUD_DELAY(TX_BAUD_DELAY)
) inst_up_uart ( .clk(clk), .rstn(~rst), .up_rreq(up_rreq), .up_rack(up_rack
```

Module instance of up_uart creating a Logic wrapper for uart axis bus cores to interface with uP bus.