axis_uart.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/24

INFORMATION

Brief

V2 upgrade to UART that will create a full UART compatible IP DTE device.

License MIT

axis_uart

```
module axis_uart #(
parameter
CLOCK_SPEED
 =
2000000,
parameter
BUS_WIDTH
 =
1,
parameter
RX_BAUD_DELAY
 =
0,
parameter
```

```
 TX_BAUD_DELAY
  =
 0
) ( input wire aclk, input wire arstn, input wire [ 2:0] reg_parity, input w
```

AXIS UART DTE, a UART with AXI Streaming interface.

Parameters

CLOCK_SPEED parameter	This is the aclk frequency in Hz
BUS_WIDTH parameter	AXIS data bus width in bytes.
RX_BAUD_DELAY parameter	DELAY RX internal baud rate by CLOCK_SPEED number of cycles.
TX_BAUD_DELAY parameter	DELAY TX internal baud rate by CLOCK_SPEED number of cycles.

Ports

aclk	Clock for AXIS
arstn	Negative reset for AXIS
reg_parity	Set the parity type, 0 = none, 1 = odd, 2 = even, 3 = mark , 4 = space
reg_stop_bits	Set the number of stop bits (0 to 3, 0=0, 1=1, 2=2, 3=??).
reg_data_bits	Set the number of data bits up to the BUS_WIDTH*8 (1 to 16, all values are biased by 1, 0+1=1).
reg_baud_rate	Frequency in Hz for the output/input data rate. This can be up to half of AXIS clock (any 32 bit unsigned value in Hz).
reg_istatus_bits	Collection of input status bits for dtr,cts,dts,dcd.
reg_ostatus_bits	Collection of output status bits for rx/tx frame, rx parity.
s_axis_tdata	Input data for UART TX.
s_axis_tvalid	When set active high the input data is valid
s_axis_tready	When active high the device is ready for input data.
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
tx	transmit for UART (output to RX)
rx	receive for UART (input from TX)
rts	request to send is a loop with CTS
dtr	data terminal ready
cts	clear to send is a loop with RTS
ri	ring indicator

INSTANTIATED MODULES

uart_baud_gen_tx

```
mod_clock_ena_gen #(
                                                                         .
 CLOCK_SPEED(CLOCK_SPEED),
                                                                         .
 DELAY(TX_BAUD_DELAY)
) uart_baud_gen_tx ( .clk(aclk), .rstn(arstn), .start0(1'b1), .clr(s_axis_t
```

Generates TX BAUD rate for UART modules using modulo divide method.

uart_baud_gen_rx

```
 mod_clock_ena_gen #(
                                                                         .
 CLOCK_SPEED(CLOCK_SPEED),
                                                                         .
 DELAY(RX_BAUD_DELAY)
) uart_baud_gen_rx ( .clk(aclk), .rstn(arstn), .start0(1'b0), .clr(r_rx_uar
```

Generates RX BAUD rate for UART modules using modulo divide method.

inst_piso

```
 piso #(
                                                                         .
 BUS_WIDTH(BUS_WIDTH),
                                                                         .
 DEFAULT_RESET_VAL(1),
                                                                         .
 DEFAULT_SHIFT_VAL(1)
) inst_piso ( .clk(aclk), .rstn(arstn), .ena(s_tx_uart_ena), .rev(1'b1), .l
```

take axis input parallel data at bus size, and output the word to the UART TX

inst_sipo

```
 sipo #(
                                                                         .
 BUS_WIDTH(BUS_WIDTH)
 ) inst_sipo ( .clk(aclk), .rstn(arstn), .ena(s_rx_uart_ena), .rev(1'b1), .
```

take UART RX data, and output the word to the parallel data bus.