

AXIS_UART



November 14, 2024

Jay Convertino

Contents

1 Usage	2
1.1 Introduction	2
1.2 Dependencies	2
1.2.1 fusesoc_info Depenecies	2
1.3 In a Project	2
2 Architecture	2
3 Building	3
3.1 fusesoc	3
3.2 Source Files	3
3.2.1 fusesoc_info File List	3
3.3 Targets	4
3.3.1 fusesoc_info Targets	4
3.4 Directory Guide	6
4 Simulation	7
4.1 iverilog	7
4.2 cocotb	7
5 Module Documentation	8
5.1 axi_uart_rx	9
5.2 axi_uart_tx	11
5.3 axi_uart	13
5.4 uart_baud_gen	17

1 Usage

1.1 Introduction

Simple UART core for TTL rs232 software mode data communications. No hardware handshake. This contains its own internal baud rate generator that creates an enable to allow data output or sampling. Baud clock and aclk can be the same clock.

RTS/CTS is not implemented, it simply asserts it as if its always ready, and ignores CTS.

1.2 Dependencies

The following are the dependencies of the cores.

- fusesoc 2.X
- iverilog (simulation)
- cocotb (simulation)

1.2.1 fusesoc_info Depenecies

- dep
 - AFRL:utility:helper:1.0.0
- dep_tb
 - AFRL:simulation:axis_stimulator
 - AFRL:utility:sim_helper

1.3 In a Project

This core connects a UART to the AXIS bus. Meaning this is a streaming device only. Connect the RX/TX to the UART in question and connect the AXIS to its intended endpoints.

2 Architecture

This core is made up of other cores that are documented in detail in there source. The cores this is made up of are the,

- **axis_uart_tx** Interface with UART TX and present the data over AXIS interface (see core for documentation).

- **axis_uart_rx** Interface with UART RX and present the data over AXIS interface (see core for documentation).
- **uart_baud_gen** Generates BAUD clock for RX and TX based on modulo divide method (see core for documentation).

3 Building

The AXIS UART is written in Verilog 2001. It should synthesize in any modern FPGA software. The core comes as a fusesoc packaged core and can be included in any other core. Be sure to make sure you have met the dependencies listed in the previous section.

3.1 fusesoc

Fusesoc is a system for building FPGA software without relying on the internal project management of the tool. Avoiding vendor lock in to Vivado or Quartus. These cores, when included in a project, can be easily integrated and targets created based upon the end developer needs. The core by itself is not a part of a system and should be integrated into a fusesoc based system. Simulations are setup to use fusesoc and are a part of its targets.

3.2 Source Files

3.2.1 fusesoc_info File List

- src
 - Type: verilogSource
 - src/axis_uart.v
 - src/axis_uart_rx.v
 - src/axis_uart_tx.v
 - src/uart_baud_gen.v
- tb
 - Type: verilogSource
 - tb/tb_uart.v
 - tb/tb_uart_baud_gen.v
 - tb/tb_uart_rx.v
 - tb/tb_uart_tx.v

3.3 Targets

3.3.1 fusesoc_info Targets

- default
 - Info: Default for IP intergration.
 - src
 - dep
- sim
 - Info: Base simulation using icarus as default.
 - src
 - dep
 - tb
 - dep_tb
 - IN_FILE_NAME
 - OUT_FILE_NAME
 - RAND_READY
- sim_rand_data
 - Info: Use random data as sim input.
 - src
 - dep
 - tb
 - dep_tb
 - IN_FILE_NAME=random.bin
 - OUT_FILE_NAME=out_random.bin
 - RAND_READY
 - FIFO_DEPTH
- sim_rand_ready_rand_data
 - Info: Use random data with a random ready as sim input.
 - src
 - dep
 - tb
 - dep_tb
 - IN_FILE_NAME=random.bin
 - OUT_FILE_NAME=out_random.bin

- RAND_READY=1
- FIFO_DEPTH
- sim_8bit_count_data
 - Info: Use counter data as sim input.
 - src
 - dep
 - tb
 - dep_tb
 - IN_FILE_NAME=8bit_count.bin
 - OUT_FILE_NAME=out_8bit_count.bin
 - RAND_READY
 - FIFO_DEPTH
- sim_rand_ready_8bit_count_data
 - Info: Use counter data with a random ready as sim input.
 - src
 - dep
 - tb
 - dep_tb
 - IN_FILE_NAME=8bit_count.bin
 - OUT_FILE_NAME=out_8bit_count.bin
 - RAND_READY=1
 - FIFO_DEPTH
- sim_baud
 - Info: Simulate only the baud rate generator.
 - src
 - dep
 - tb
 - dep_tb
 - IN_FILE_NAME
 - OUT_FILE_NAME
 - RAND_READY
- sim_rx
 - Info: Simulate only the rx block.

- src
- dep
- tb
- dep_tb
- IN_FILE_NAME
- OUT_FILE_NAME
- RAND_READY
- sim_tx
 - Info: Simulate only the tx block.
 - src
 - dep
 - tb
 - dep_tb
 - IN_FILE_NAME
 - OUT_FILE_NAME
 - RAND_READY

3.4 Directory Guide

Below highlights important folders from the root of the directory.

1. **docs** Contains all documentation related to this project.
 - **manual** Contains user manual and github page that are generated from the latex sources.
2. **src** Contains source files for the core
3. **tb** Contains test bench files for iverilog and cocotb
 - **cocotb** testbench files

4 Simulation

There are a few different simulations that can be run for this core.

4.1 iverilog

iverilog is used for simple test benches for quick verification, visually, of the core.

4.2 cocotb

Future simulations will use cocotb. This feature is not yet implemented.

5 Module Documentation

- **axis_uart_tx** Interfaces AXIS to the UART transmit line.
- **axis_uart_rx** Interfaces AXIS to the UART receive line.
- **uart_baud_gen** Generates a Baud rate clock for the UART based on the input clock using modulo division method. The modulo method works by loading the clock speed (in hertz) used for generation into a register. This register is then subtracted from by the desired baud clock (in hertz). Once the counter is depleted the remainder, if it exists, is added to the counter plus the original clock speed and the operation repeats. Each remainder is added back to the counter. This allows non-even divisions to eventually average out to the target rate.
- **axis_uart Wrapper** for all of the above modules to create a singular device to interface with.

axis_uart_rx.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/24

INFORMATION

Brief

UART RX to AXIS bus.

License MIT

Copyright 2021 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.

axis_uart_rx

```
module axis_uart_rx #(
    parameter
    PARITY_ENA
    =
    0,
    parameter
    PARITY_TYPE
    =
    0,
    parameter
    STOP_BITS
```

```

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

```

AXIS UART, simple UART with AXI Streaming interface.

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.

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 clock till in receive state.
rxdata	receive for UART (input from TX)

axis_uart_tx.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/24

INFORMATION

Brief

UART TX from AXIS bus.

License MIT

Copyright 2021 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.

axis_uart_tx

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

```

    =
    1,
    parameter
    DATA_BITS
    =
    8,
    parameter
    DELAY
    =
    0
) ( input aclk, input arstn, input [DATA_BITS-1:0] s_axis_tdata, input s_axi

```

AXIS UART TX, simple UART TX from AXI Streaming interface.

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 tx data output. Delays the time to output of the data.

Ports

aclk	Clock for AXIS
arstn	Negative reset for AXIS
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.
uart_clk	Clock used for BAUD rate generation
uart_rstn	Negative reset for UART, for anything clocked on uart_clk
uart_ena	When active high enable UART transmit state.
txd	transmit for UART (output to RX)

axis_uart.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/24

INFORMATION

Brief

Core for interfacing with simple UART communications. Output is always the size of DATA_BITS.

License MIT

Copyright 2021 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.

axis_uart

```
module axis_uart #(
    parameter
    BAUD_CLOCK_SPEED
    =
    2000000,
    parameter
    BAUD_RATE
    =
    2000000,
    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 aclk, input arstn, output parity_err, output frame_err, input [DA

```

AXIS UART, simple UART with AXI Streaming interface.

Parameters

BAUD_CLOCK_SPEED parameter	This is the aclk frequency in Hz
BAUD_RATE parameter	Serial Baud, this can be any value including non-standard.
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.
RX_DELAY parameter	Delay in rx data input.
RX_BAUD_DELAY parameter	Delay in rx baud enable. This will delay when we sample a bit (default is midpoint when rx delay is 0).
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

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)
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
cts	clear to send is a loop with RTS

INSTANTIATED MODULES

uart_baud_gen_tx

```
uart_baud_gen #(
    BAUD_CLOCK_SPEED(BAUD_CLOCK_SPEED),
    BAUD_RATE(BAUD_RATE),
    DELAY(TX_BAUD_DELAY)
) uart_baud_gen_tx ( .uart_clk(uart_clk), .uart_rstn(uart_rstn), .uart_hold(
```

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

uart_baud_gen_rx

```
uart_baud_gen #(
    BAUD_CLOCK_SPEED(BAUD_CLOCK_SPEED),
    BAUD_RATE(BAUD_RATE),
    DELAY(RX_BAUD_DELAY)
) uart_baud_gen_rx ( .uart_clk(uart_clk), .uart_rstn(uart_rstn), .uart_hold(
```

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

uart_tx

Produces transmit data for tx UART from AXIS.

Consumes receive data for rx UART to AXIS.

uart_baud_gen.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/24

INFORMATION

Brief

Generate UART BAUD rate by dividing input clock rate using modulo divide (subtract and carry remainder)

License MIT

Copyright 2021 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.

uart_baud_gen

```
module uart_baud_gen #(
    parameter
    BAUD_CLOCK_SPEED
    =
    2000000,
    parameter
    BAUD_RATE
    =
    115200,
    parameter
```

```

    DELAY
    =
    0
) ( input uart_clk, input uart_rstn, input uart_hold, output uart_ena )

```

Baud rate generator

Parameters

BAUD_CLOCK_SPEED parameter	This is the aclk frequency in Hz
BAUD_RATE parameter	Serial Baud, this can be any value including non-standard.
DELAY parameter	Delay in rx data input.

Ports

uart_clk	Clock used for BAUD rate generation
uart_rstn	Negative reset for UART, for anything clocked on uart_clk
uart_hold	Output to hold clock till in receive state.
uart_ena	Enable UART data processing from RX.