AXIS_STIMULATOR

AFRL

AIR FORCE RESEARCH LABORATORY

November 25, 2024

Jay Convertino

Contents

1 Usage

1.1 Introduction

This core contains two modules. A writer, and reader that should be placed on the output, and input of the device under test. This will stream data through till is has read all data. Then once all data has been written AND tlast is set to high the writer module will end the simulation.

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:clock_stimulator
  - AFRL:utility:sim_helper
- dep_vpi
  - AFRL:vpi:binary_file_io:1.0.0

2 Architecture

The project contains two modules master_axis_stimulus and slave_axis_stimulus. The master_axis_stimulus is used to take input data from the slave axis interface (input) and write it to a file. Essentially it goes DUT_MASTER to MASTER_AXIS_STIMULUS. slave_axis_stimulus is used to read a file and push that data to the master axis interface (output). Essentially it goes SLAVE_AXIS_STIMULUS to DUT_SLAVE.

This core uses a custom library for reading and writing files called vpi_binary_file_io. This library provides multithreaded file reads using a ring buffer between processes. The core will also puncture data according to its bit type. X/Z values are tossed if they are contained in a byte.

- **tm_stim_axis** Contains two modules master_axis_stimulus, slave_axis_stimulus.

Please see 5 for more information per target.

3 Building

The all AXIS stimulator modules are written in Verilog 2001. They 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 meet 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
  - src/tm_stim_axis.v
- tb
  - ’tb/tb_axis.v’: ’file_type’: ’verilogSource’

3.3 Targets

3.3.1 fusesoc_info Targets

- default

  Info: Default file set.

- sim

  Info: Default icarus sim.

- sim_rand_data

Info: Use random data as a input.

- sim_rand_ready_rand_data

  Info: Use random data as a input, with random ready.

- sim_8bit_count_data

  Info: Use counter data as a input.

- sim_rand_ready_8bit_count_data

  Info: Use counter data as a input, with random 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 axis_stimulator.
3. **tb** Contains test bench files.

4 Simulation

There is no simulation at the moment. Maybe a future addition?

5 Module Documentation

There project has multiple modules. The targets are the top system wrappers.

- **tm_stim_axis**
- **tb_axis**

The next sections document the module in great detail.

tm_stim_axis.v

AUTHORS

JAY CONVERTINO

DATES

2022/10/24

INFORMATION

Brief

All modules for AXIS test bench top are here. There will be loop of tests the axis core must pass. In these tests is where the end user must alter the checks if the input does not equal the output. As these were designed with a FIFO in mind.

License MIT

slave_axis_stimulus

```
module slave_axis_stimulus #(
parameter
BUS_WIDTH
 =
1,
parameter
USER_WIDTH
 =
1,
parameter
DEST_WIDTH
 =
1,
parameter
BYTE_SWAP
 =
0,
parameter
FILE
 =
"test.bin"
) ( input m_axis_aclk, input m_axis_arstn, output reg m_axis_tvalid, input
```

Simulator core to read file data, and output it over master axis dut.

Parameters

BUS_WIDTH parameter	bus width in bytes for data bus
USER_WIDTH parameter	user width in bits
DEST_WIDTH parameter	dest width in bits
BYTE_SWAP parameter	swap bytes fed to the DUT
FILE parameter	input file name

Ports

m_axis_aclk	master axis clock
m_axis_arstn	master axis negative reset
m_axis_tvalid	master axis data valid active high
m_axis_tready	master axis, is the input device ready?
m_axis_tdata	master axis data input.
m_axis_tkeep	master axis byte indicator
m_axis_tlast	master axis data is last word
m_axis_tuser	master axis user defined
m_axis_tdest	master axis desitination
eof	end of input file has been reached.

master_axis_stimulus

```
module master_axis_stimulus #(
parameter
BUS_WIDTH
 =
1,
parameter
USER_WIDTH
 =
1,
parameter
DEST_WIDTH
 =
1,
parameter
RAND_READY
 =
0,
parameter
FILE
 =
"out.bin"
) ( input s_axis_aclk, input s_axis_arstn, input s_axis_tvalid, output reg
```

Simulator core to write file data, from input over slave axis dut. This module will keep a constant ready to the dut.

Parameters

BUS_WIDTH parameter	bus width in bytes for data bus
USER_WIDTH parameter	user width in bits
DEST_WIDTH parameter	dest width in bits
RAND_READY parameter	random ready if set anything other than 0
FILE parameter	output file name

Ports

s_axis_aclk	slave axis clock
s_axis_arstn	slave axis negative reset
s_axis_tvalid	slave data valid
s_axis_tready	slave ready
s_axis_tdata	slave data
s_axis_tkeep	slave keep
s_axis_tlast	slave last word of data
s_axis_tuser	slave user port
s_axis_tdest	slave destination port
eof	end of file will trigger $finish to end sim

tb_axis.v

AUTHORS

JAY CONVERTINO

DATES

2022/10/24

INFORMATION

Brief

Generic AXIS test bench top with verification.

License MIT

tb_axis

```
module tb_axis #(
parameter
OUT_FILE_NAME
 =
in.bin,
parameter
IN_FILE_NAME
 =
out.bin,
parameter
RAND_READY
```

```
  =
 0
)
```

Generic AXIS test bench top with verification.

Parameters

OUT_FILE_NAME parameter — Name of the output file to write.

IN_FILE_NAME parameter — Name of the input file to read from.

RAND_READY parameter — Randomize the Ready signal from the writer (master_axis_stim) core.

INSTANTIANTED MODULES

clk_stim

```
 clk_stimulus #(
                                                                          .
 CLOCKS(1),
                                                                          .
 CLOCK_BASE(1000000),
                                                                          .
 CLOCK_INC(1000),
                                                                          .
 RESETS(1),
                                                                          .
 RESET_BASE(2000),
                                                                          .
 RESET_INC(100)
) clk_stim ( .clkv(tb_stim_clk), .rstnv(tb_stim_rstn), .rstv() )
```

Generate a clock for the modules.

slave_axis_stim

```
 slave_axis_stimulus #(
                                                                          .
 BUS_WIDTH(BUS_WIDTH),
                                                                          .
 USER_WIDTH(USER_WIDTH),
                                                                          .
 DEST_WIDTH(DEST_WIDTH),
                                                                          .
 FILE(IN_FILE_NAME)
) slave_axis_stim ( .m_axis_aclk(tb_stim_clk), .m_axis_arstn(tb_stim_rstn),
```

Read a file and output to a SLAVE AXIS interface from the master.

master_axis_stim

```
 master_axis_stimulus #(
                                                                          .
 BUS_WIDTH(BUS_WIDTH),
                                                                          .
 USER_WIDTH(USER_WIDTH),
                                                                          .
 DEST_WIDTH(DEST_WIDTH),
                                                                          .
 RAND_READY(RAND_READY),
                                                                          .
 FILE(OUT_FILE_NAME)
) master_axis_stim ( .s_axis_aclk(tb_stim_clk), .s_axis_arstn(tb_stim_rstn),
```

Write a file from the input from a MASTER AXIS interface to the slave.