AXIS_DATA_WIDTH_CONVERTER

AIR FORCE RESEARCH LABORATORY

November 19, 2024

Jay Convertino

Contents

1 Usage

1.1 Introduction

This data width converter is for even integer divides of slave to master or master to slave. Example this core can go from 4 bytes to 2 bytes or 2 bytes to 4 bytes. It can not go from 5 bytes to 2 bytes or 2 bytes to 5 bytes. 4/2 is 2, a round number. 5/2 is a fractional number that will not work with this core.

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:simulation:clock_stimulator
  - AFRL:utility:sim_helper

1.3 In a Project

Simply use this core between a sink and source AXIS devices. This will convert from one BUS size to another. Check the code to see if others will work correctly.

2 Architecture

The only module is the axis_data_width_converter module. It is listed below.

- **axis_data_width_converter** Impliment an algorithm to convert BUS data interfaces in even mutlples (see core for documentation).

This core only uses a combinatoral method to divide the accumulator. Since all weights are powers of two this is done with a part select based on bit position.

The always block has the following steps.

1. If there is valid data, sum the new data into the accumulator and remove the top element in the buffer from the accumulator.

2. Insert the new element into the buffer.

3. Shift the buffer to so that old elements at the top of the buffer are shifted out.

Please see 5 for more information.

3 Building

The AXIS data width converter core is 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

  Type: verilogSource
  - src/axis_data_width_converter.v

- tb

  - ’tb/tb_axis.v’: ’file_type’: ’verilogSource’

3.3 Targets

3.3.1 fusesoc_info Targets

- default

  Info: Default for IP intergration.

  - src
  - dep

- sim

  Info: Test 1:1 conversion.

  - src
  - dep
  - tb
  - dep_tb
  - IN_FILE_NAME
  - OUT_FILE_NAME
  - RAND_READY
  - MASTER_WIDTH
  - SLAVE_WIDTH

- sim_reduce

  Info: Test data reduction.

  - src
  - dep
  - tb
  - dep_tb
  - IN_FILE_NAME
  - OUT_FILE_NAME
  - RAND_READY
  - MASTER_WIDTH=2
  - SLAVE_WIDTH=4

- sim_rand_data_reduce

  Info: Test data reduction with random data

  - src
  - dep
  - tb

  - dep_tb
  - IN_FILE_NAME=random.bin
  - OUT_FILE_NAME=out_random.bin
  - RAND_READY
  - MASTER_WIDTH=2
  - SLAVE_WIDTH=4

- sim_rand_ready_rand_data_reduce

  Info: Test data reduction with random ready and random data.
  - src
  - dep
  - tb
  - dep_tb
  - IN_FILE_NAME=random.bin
  - OUT_FILE_NAME=out_random.bin
  - RAND_READY=1
  - MASTER_WIDTH=2
  - SLAVE_WIDTH=4

- sim_8bit_count_data_reduce

  Info: Test data reduction with counter data.
  - src
  - dep
  - tb
  - dep_tb
  - IN_FILE_NAME=8bit_count.bin
  - OUT_FILE_NAME=out_8bit_count.bin
  - RAND_READY
  - MASTER_WIDTH=2
  - SLAVE_WIDTH=4

- sim_rand_ready_8bit_count_data_reduce

  Info: Test data reduction with counter data, and random ready.
  - src
  - dep

  - tb
  - dep_tb
  - IN_FILE_NAME=8bit_count.bin
  - OUT_FILE_NAME=out_8bit_count.bin
  - RAND_READY=1
  - MASTER_WIDTH=2
  - SLAVE_WIDTH=4

- sim_increase

  Info: Test data increase.

  - src
  - dep
  - tb
  - dep_tb
  - IN_FILE_NAME
  - OUT_FILE_NAME
  - RAND_READY
  - MASTER_WIDTH=4
  - SLAVE_WIDTH=2

- sim_rand_data_increase

  Info: Test data increase with random data.

  - src
  - dep
  - tb
  - dep_tb
  - IN_FILE_NAME=random.bin
  - OUT_FILE_NAME=out_random.bin
  - RAND_READY
  - MASTER_WIDTH=4
  - SLAVE_WIDTH=2

- sim_rand_ready_rand_data_increase

  Info: Test data increase with random data, and random ready.

  - src
  - dep

  - tb
  - dep_tb
  - IN_FILE_NAME=random.bin
  - OUT_FILE_NAME=out_random.bin
  - RAND_READY=1
  - MASTER_WIDTH=4
  - SLAVE_WIDTH=2
- sim_8bit_count_data_increase

  Info: Test data increase with count data.
  - src
  - dep
  - tb
  - dep_tb
  - IN_FILE_NAME=8bit_count.bin
  - OUT_FILE_NAME=out_8bit_count.bin
  - RAND_READY
  - MASTER_WIDTH=4
  - SLAVE_WIDTH=2
- sim_rand_ready_8bit_count_data_increase

  Info: Test data increase with count data, and random ready.
  - src
  - dep
  - tb
  - dep_tb
  - IN_FILE_NAME=8bit_count.bin
  - OUT_FILE_NAME=out_8bit_count.bin
  - RAND_READY=1
  - MASTER_WIDTH=4
  - SLAVE_WIDTH=2

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

There is a single async module for this core.

- **axis_data_width_converter** AXIS data width converter, converts from one BUS data size to another.

The next sections document the module in great detail.

axis_data_width_converter.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/21

INFORMATION

Brief

AXIS DATA WIDTH CONVERTER

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_data_width_converter

```
module axis_data_width_converter #(
parameter
SLAVE_WIDTH
 =
1,
parameter
MASTER_WIDTH
 =
1,
parameter
REVERSE
```

```
  =
 0
) ( input aclk, input arstn, output [(MASTER_WIDTH*8)-1:0] m_axis_tdata, out
```

Change size of streaming bus in even integers of. 1/2 2/1 2/4 4/2 etc.

Parameters

SLAVE_WIDTH parameter	Width of the slave input bus in bytes
MASTER_WIDTH parameter	Width of the master output bus in bytes
REVERSE parameter	Change byte order

Ports

aclk	Clock for AXIS
arstn	Negative reset for AXIS
m_axis_tdata	Output data
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.
m_axis_tlast	Indicates last word in stream.
s_axis_tdata	Input data
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.
s_axis_tlast	Is this the last word in the stream (active high).