CLOCK_STIMULATOR

November 26, 2024

Jay Convertino

Contents

1 Usage

1.1 Introduction

This modules creates multiple clocks and multiple negative or positive resets. The module outputs a vector based on the number requested.

1.2 Dependencies

The following are the dependencies of the cores.

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

2 Architecture

Generates a clock and reset for simulation. The clock is for loop generated using delay controls in a always block. The reset is geneated in a for loop for each reset in a initial block with a delay based upon the increment amount.

Please see 5 for more information per target.

3 Building

The all clock 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_clk.v
- tb
  - tb/tb_clk.v

3.3 Targets

3.3.1 fusesoc_info Targets

- default

  Info: Default file set.

- sim

  Info: Default for sim intergration.

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 clock_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_clk**
- **tb_clk**

The next sections document the module in great detail.

tm_stim_clk.v

AUTHORS

JAY CONVERTINO

DATES

2022/10/24

INFORMATION

Brief

clock simulator creates multiple clocks and resets for testing.

License MIT

clk_stimulus

```
module clk_stimulus #(
parameter
CLOCKS
 =
2,
parameter
CLOCK_BASE
 =
1000000,
parameter
CLOCK_INC
```

```
 =
100,
parameter
CLOCK_ASYNC
 =
0,
parameter
RESETS
 =
2,
parameter
RESET_BASE
 =
200,
parameter
RESET_INC
 =
100
) ( output reg [CLOCKS-1:0] clkv, output reg [RESETS-1:0] rstnv, output reg
```

clock simulator creates multiple clocks and resets for testing.

Parameters

CLOCKS Number of clocks
parameter

CLOCK_BASE Clock time base mhz
parameter

CLOCK_INC Time diff for other clocks, only used for async mode
parameter

CLOCK_ASYNC Used time diff to generate async clocks (1), 0 is sync clock
parameter

RESETS Number of resets
parameter

RESET_BASE Time to stay in reset
parameter

RESET_INC Time diff for other resets
parameter

Ports

clkv clock output vector

rstnv negative reset output vector

rstv positive reset output vector

tb_clk.v

AUTHORS

JAY CONVERTINO

DATES

2022/10/24

INFORMATION

Brief

Generic clock test bench top.

License MIT

tb_clk

```
module tb_clk
```

Generic clock test bench top.

INSTANTIANTED MODULES

clk_stim

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

Generate a clock for the modules.