Clock Reset Generator VIP - Lattice Propel Builder

User Guide

FPGA-IPUG-02147-1.0

October 2020

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
API	Application Programming Interface
DUT	Device Under Test
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
LUT	Lookup-Table
OSC	Oscillator
VIP	Verification Intellectual Property

1. Introduction

The Lattice Semiconductor Clock Reset Generator VIP (Verification IP) is used for verification purpose only. It generates clock and reset signals for the verification environment.

The Clock Reset Generator VIP design is implemented in System Verilog. It can be configured and generated using the Lattice Propel™ Builder software. It is for simulation only and is not targeted to any device.

1.1. Features

The clock and reset generator has the following features:

- Reset insertion at designated place
- Reset generation with configurable reset polarity
- Reset generation with configurable reset period
- Clock enable and disable at designated place
- Clock generation with configurable period
- Clock generation with configuration duty cycle
- Pass through output of DUT (Device Under Test) reset input
- Pass through output of DUT clock input

1.2. Conventions

1.2.1. Nomenclature

The nomenclature used in this document is based on System Verilog.

2. Functional Descriptions

2.1. Overview

The Clock Reset Generator VIP is used to generate clock and reset signals for the verification environment as shown in Figure 2.1. If the DUT's internal clock and reset are to be used, they should be connected to dut_clk_i and dut_rst_i. The pass through outputs dut_clk_o and dut_rst_o are then used to drive other components.

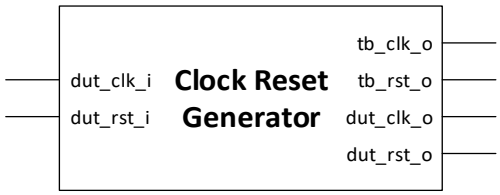

Figure 2.1. Clock Reset Generator Block Diagram

Note: Clocks and resets of all components should be driven by the outputs of this VIP. If the DUT has internally generated clock/reset (for example, the DUT instantiates OSC and uses its output clock to drive the design), connect the clock/reset to the corresponding input of this VIP. Thus, the verification project has less dependency on the DUT and you can benefit most from the provided APIs.

2.2. Signal Description

Table 2.1 lists the ports of the Clock Reset Generator VIP.

Table 2.1. Clock Reset Generator Ports

Name	Type	Width	Description
dut_clk_i	In	1	Clock input from the DUT
dut_rst_i	In	1	Reset input from the DUT
tb_clk_o	Out	1	Generated testbench Clock output
tb_rst_o	Out	1	Generated testbench Reset output
dut_clk_o	Out	1	Pass through output of dut_clk_i, enabled by DUT_CLK_IN_EN
dut_rst_o	Out	1	Pass through output of dut_rst_i, enabled by DUT_RST_IN_EN

2.3. Attribute Summary

The configurable attributes of the Clock Reset Generator VIP are shown in Table 2.2 and are described in Table 2.3. The attributes can be configured through the Propel Builder software.

Table 2.2. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
DUT_CLK_IN_EN	0, 1	0	—
DUT_RST_IN_EN	0, 1	0	—
TB_CLK_PERIOD	Floating value	10	—
TB_CLK_DUTY	Floating value	50	—
TB_RST_POL	0, 1	0	—
TB_RST_PERIOD	Floating value	50	—
INIT_RST	0, 1	1	—

Table 2.3. Attributes Description

Attribute	Description
DUT_CLK_IN_EN	Pass through output of dut_clk_i enable 0 – disable, 1 – enable
DUT_RST_IN_EN	Pass through output of dut_rst_i enable 0 – disable, 1 – enable
TB_CLK_PERIOD	Clock period (ns) of testbench Clock
TB_CLK_DUTY	Duty cycle of testbench Clock
TB_RST_POL	Polarity of valid tb_rst_o 0 – active low, 1 – active high
TB_RST_PERIOD	Valid period(ns) of tb_rst_o
INIT_RST	Inserting a reset period at the very beginning of simulation 0 – do not insert, 1 – insert.

2.4. API Summary

Table 2.4. API Description

API	Description
set_clk_period_ns(real period_ns)	Set clock period(ns) for the testbench clock
set_clk_freq_mhz(real freq_mhz)	Set frequency(MHz) for the testbench clock
set_clk_duty(real duty)	Set duty cycle for the testbench clock
start_clk()	Enable the testbench clock
stop_clk()	Disable the testbench clock
set_rst_period(real period_ns)	Set the testbench reset period (in ns)
set_rst_cycles(int num_cycles)	Set the testbench reset period (in testbench clock cycles)
inset_rst()	Insert a testbench reset period
wait_tb_clk_posedge(int num)	Wait specified positive testbench clock edges
wait_tb_clk_negedge(int num)	Wait specified negative testbench clock edges
wait_dut_clk_posedge(int num)	Wait specified positive DUT clock edges
wait_dut_clk_negedge(int num)	Wait specified negative DUT clock edges
gate_clk_ns(int gate_time)	Halt testbench clock for specified time(ns)
gate_clk_cycles(int num_cycles)	Halt testbench clock for specified cycles

The following is an example of API usage.

```
 initial begin
     clk_rst_u0.set_clk_freq_mhz(CLK_MHZ);
     clk_rst_u0.start_clk();
     clk_rst_u0.wait_tb_clk_posedges(10);
     clk_rst_u0.insert_rst();
  end
```

3. Clock Reset Generator VIP Generation

This section provides information on how to generate the Clock Reset Generator VIP module using Propel Builder.

To generate the Clock Reset Generator VIP module:

1. In Propel Builder, create a new verification design, and select the Clock Reset Generator VIP package.
2. Enter the module name and click **Next**, as shown in Figure 3.1.

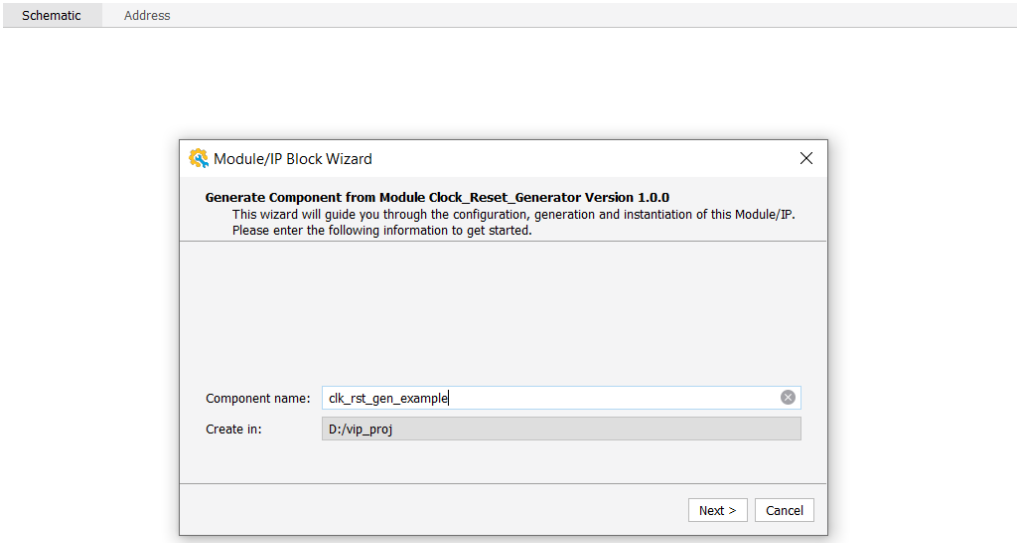

Figure 3.1. Entering the Module Name

3. Configure the parameters as needed, then click **Generate**.

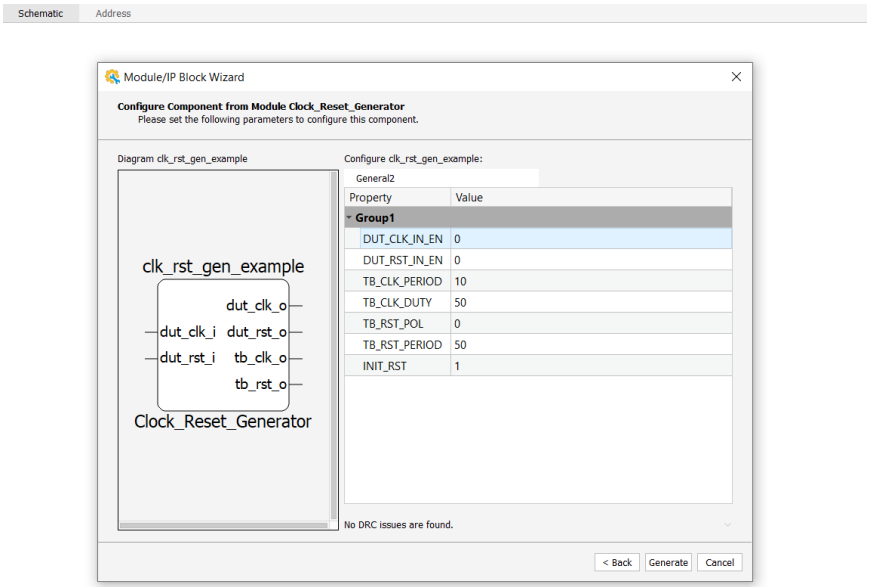

Figure 3.2. Configuring the Parameters

4. Verify the information and click **Finish**.

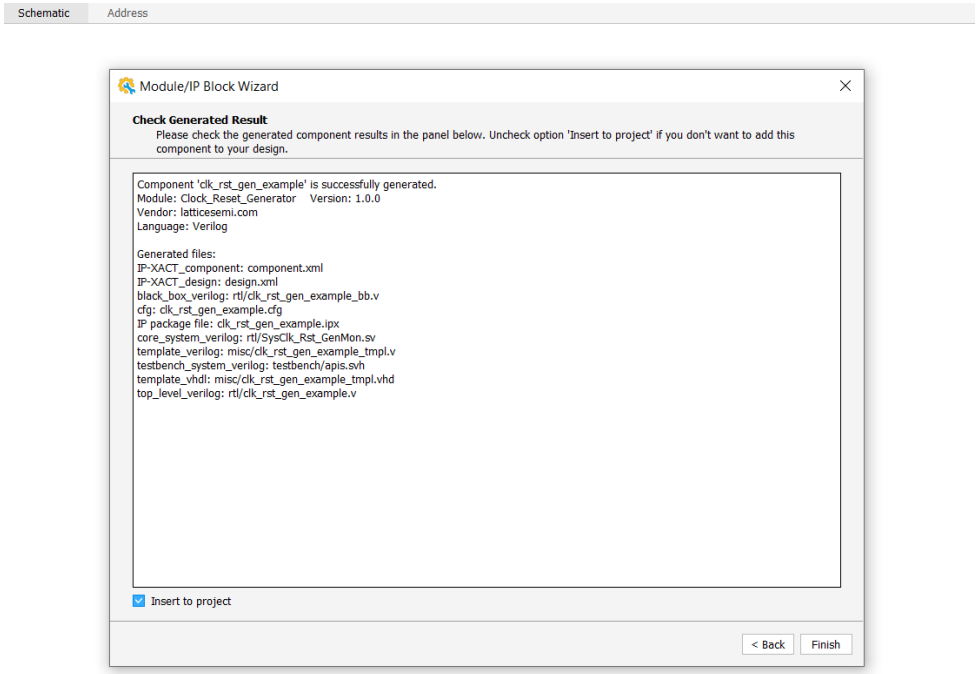

Figure 3.3. Verifying the Result

5. Confirm or modify the module instance name, then click **OK**.

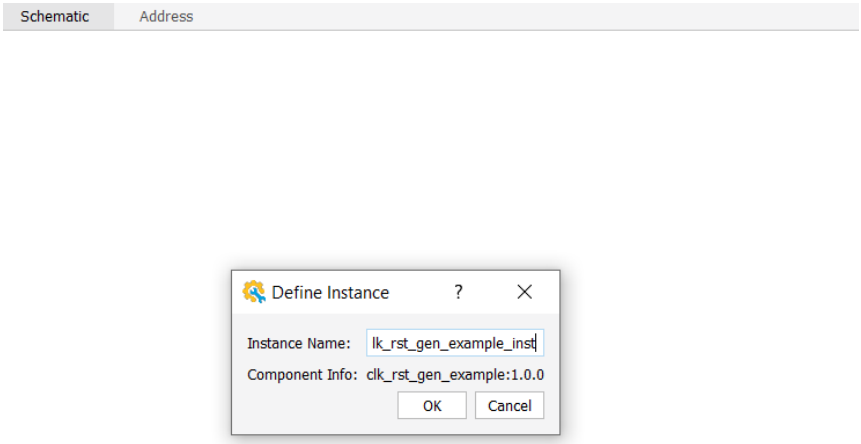

Figure 3.4. Specifying the Instance Name

The Clock Reset Generator VIP instance is successfully generated.

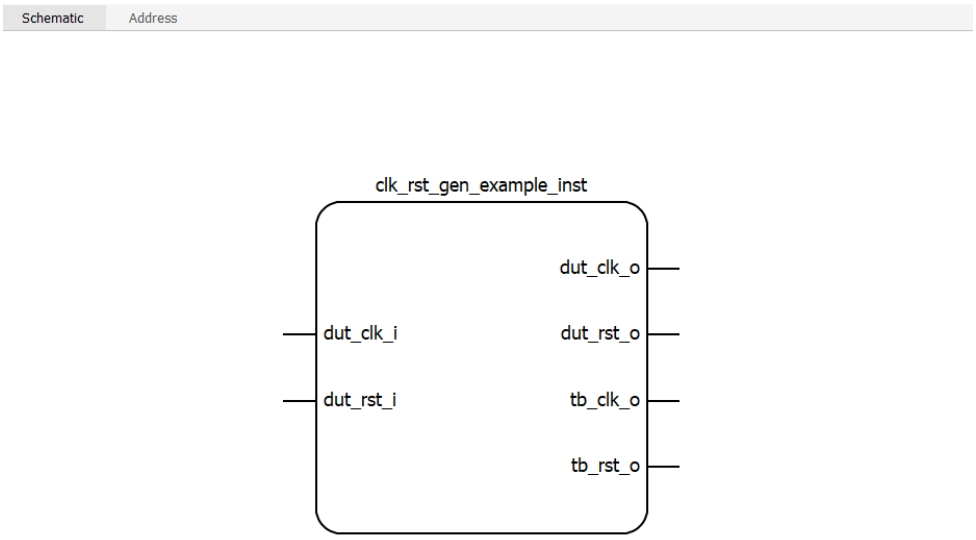

Figure 3.5. Generated Instance

References

- MachXO3D FPGA web page in latticesemi.com
- CrossLink-NX FPGA web page in latticesemi.com
- Lattice Propel 1.0 User Guide
- Lattice Diamond Software 3.11 User Guide
- Lattice Radiant™ Software 2.1 User Guide

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.0, October 2020

Section	Change Summary
All	Initial release.

www.latticesemi.com