Avant OSC Module User Guide

User Guide

FPGA-IPUG-02184-1.0

October 2022

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
FPGA	Field Programmable Gate Array
LSE	Lattice Synthesis Engine

1. Introduction

The Lattice Semiconductor OSC Module is designed to generate a clock signal that can be used for FPGA clock tree and any Avant IP that needs a non-PLL based clock source. It has dynamically selectable clock frequency between 400 or 320 MHz clock with 1-256 programmable divider options.

1.1. Features

The key features of this module are:

- Selectable 400 or 320 MHz clock with 1-256 programmable divider for global clock tree

1.2. Conventions

1.2.1. Nomenclature

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

1.2.2. Signal Names

Signal names that end with:

- _*n* are active low
- _*i* are input signals
- _*o* are output signals

1.2.3. Attribute Names

Attribute names in this document are formatted in title case and italicized (*Attribute Name*).

2. Functional Description

The Avant OSC Module generates a clock with maximum frequency of 400MHz. Figure 2.1 shows the top level block diagram of the OSC wherein the clk_out_o is selectable between 400MHz and 320 MHz using the input port sel_400n_i and has programmable feature.

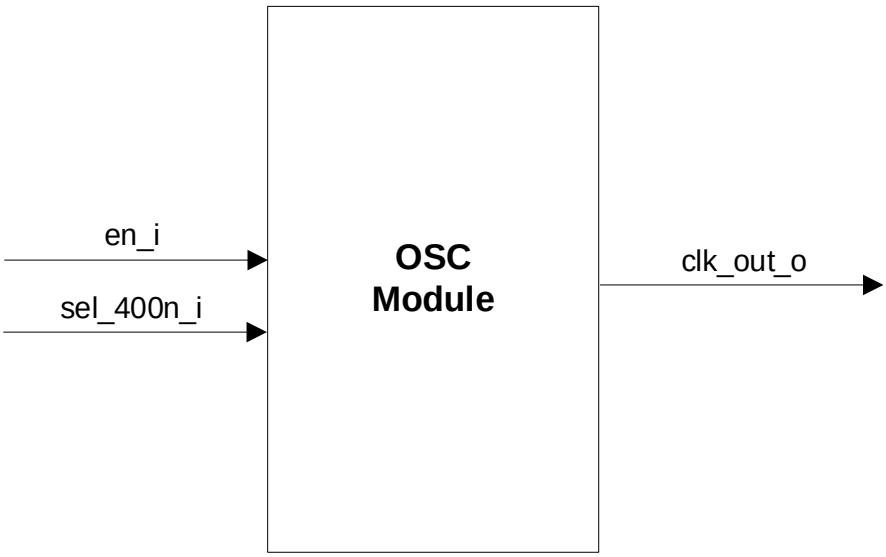

Figure 2.1. OSC Block Diagram

2.1. Signal Descriptions

Table 2.1. OSC Module Signal Description

Port Name	I/O	Width	Description
Clock ports			
sel_400n_i	In	1	Control signal to dynamically switch between 400MHz and 320MHz. 1'b1 – 320MHz 1'b0 – 400MHz
clk_out_o	Out	1	Programmable clock output.
Enable Ports			
en_i	In	1	Control signal to enable user clock

2.2. Attribute Summary

The configurable attributes of the OSC Module are shown in Table 2.2 and are described in Table 2.3. The attributes can be configured through the IP Catalog's Module/IP wizard of the Lattice Radiant software.

Table 2.2. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
General			
Oscillator Clock (MHz)	400, 320	400	
CLK Divider	Calculated	1	*Oscillator Clock/Result Frequency*
Output Clock			
Frequency (MHz) : sel_400n_i = 0	1.5625 - 400.0	400	Active if *Oscillator Clock = 400*
Frequency (MHz) : sel_400n_i = 1	1.25 - 320	320	Active if *Oscillator Clock = 320*

Table 2.3. Attributes Descriptions

Attribute	Description
General	
Oscillator Clock (MHz)	Displays the target maximum clk_out_o frequency.
CLK Divider	Displays the calculated divider of the user clock.
Result	
Frequency (MHz) : sel_400n_i = 0	Displays the selectable desired frequency when control signal sel_400n_i is low.
Frequency (MHz) : sel_400n_i = 1	Displays the selectable desired frequency when control signal sel_400n_i is high.

3. IP Generation, Simulation, and Validation

This section provides information on how to generate and synthesize this module using the Lattice Radiant software. For more on Lattice Radiant software, refer to the Lattice Radiant software user guide and relevant tutorials.

3.1. Generating the IP

The Lattice Radiant software allows you to customize and generate modules and IPs and integrate them into the device's architecture. The procedure for generating the OSC Module in Lattice Radiant software is described below.

To generate the OSC Module:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click on **OSC** under **Module, Architecture_Modules** category. The **Module/IP Block Wizard** opens as shown in Figure 3.1. Enter values in the **Component name** and the **Create in** fields and click **Next**.

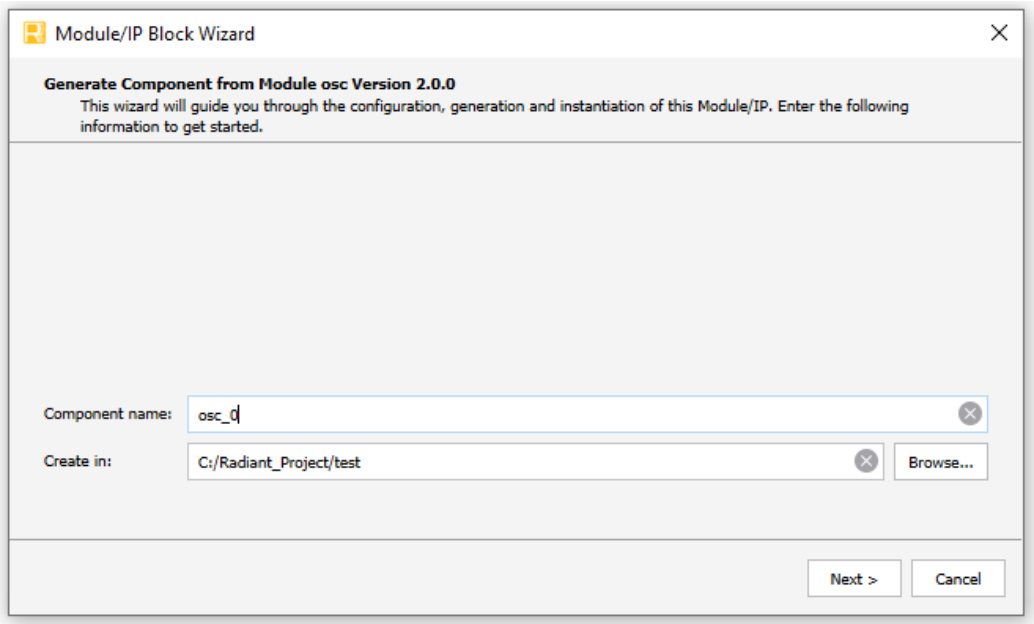

Figure 3.1. Module/IP Block Wizard

3. In the module's dialog box of the **Module/IP Block Wizard** window, customize the selected OSC Module using drop-down menus and check boxes. As a sample configuration, see Figure 3.2. For configuration options, see the Attribute Summary section.

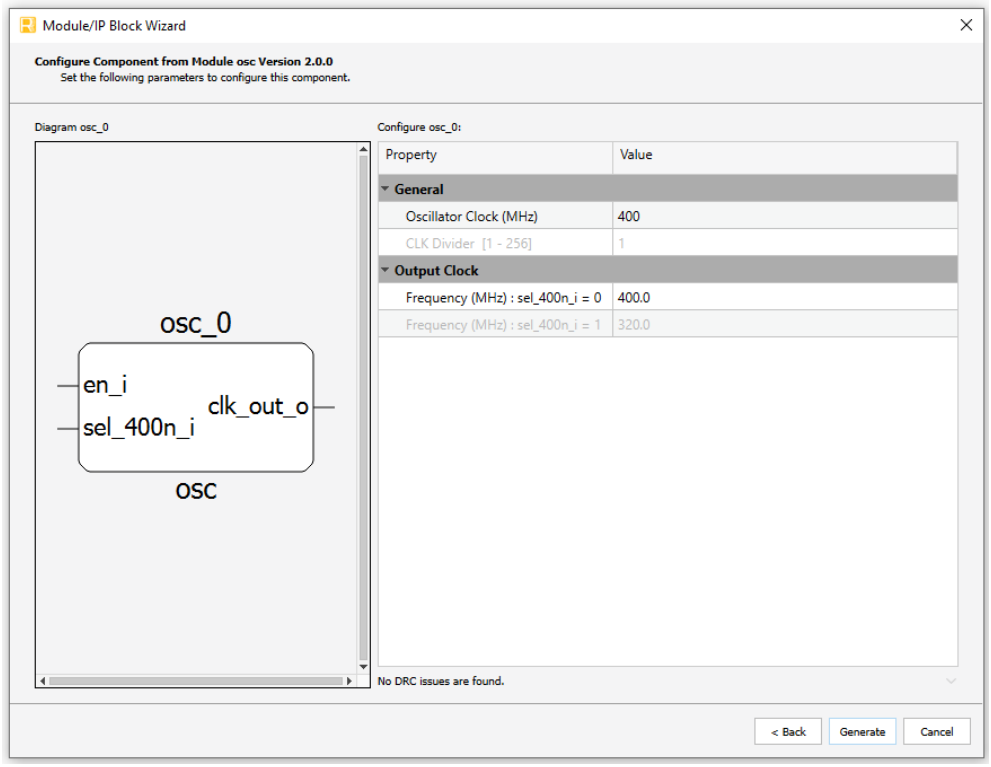

Figure 3.2. Configure User Interface of OSC Module

4. Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results as shown in Figure 3.3.

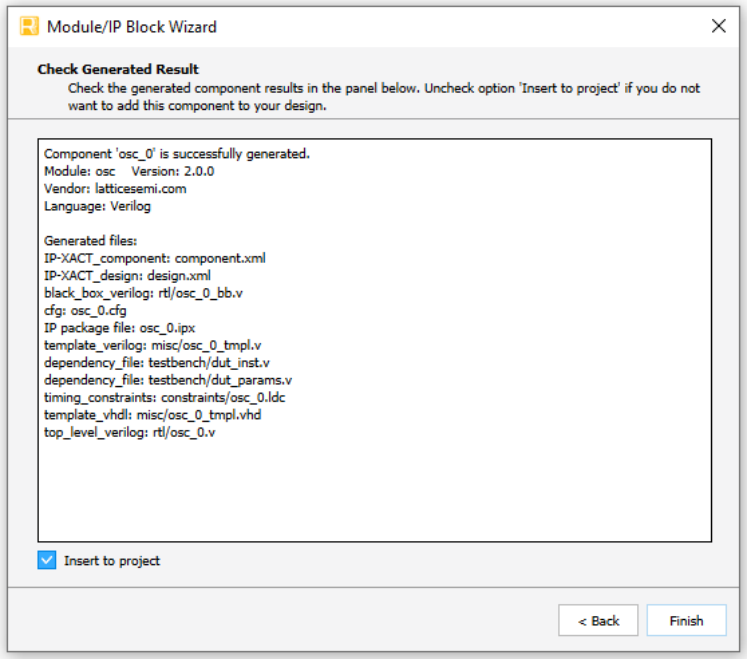

Figure 3.3. Check Generating Result

5. Click the **Finish** button. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 3.1.

The generated OSC Module package includes the black box (<Component Name>_bb.v) and component templates (<Component Name>_tmpl.v/vhd) that can be used to instantiate the module in a top-level design. An example RTL top-level reference source file (<Component Name>.v) that can be used as an instantiation template for the module is also provided. You may also use this top-level reference as the starting template for the top-level for their complete design. The generated files are listed in Table 3.1.

Table 3.1. Generated File List

Attribute	Description
<Component Name>.ipx	This file contains the information on the files associated to the generated IP.
<Component Name>.cfg	This file contains the attribute values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration attributes of the IP in IP-XACT 2014 format.
rtl/<Component Name>.v	This file provides an example RTL top file that instantiates the module.
rtl/< Component Name>_bb.v	This file provides the synthesis black box.
misc/< Component Name>_tmpl.v misc /< Component Name>_tmpl.vhd	These files provide component templates for the module.
testbench/dut_inst.v	Instantiated version of the <Component Name>.v file for simulation use.
testbench/dut_params.v	Top Level parameters of the generated RTL file.
testbench/tb_top.v	Testbench template, you can modify this to match your specific needs.

3.2. Running the Functional Simulation

Running functional simulation can be performed after the IP is generated.

To run Verilog simulation:

1. Click the  button located on the Toolbar to initiate the Simulation Wizard shown in Figure 3.4.

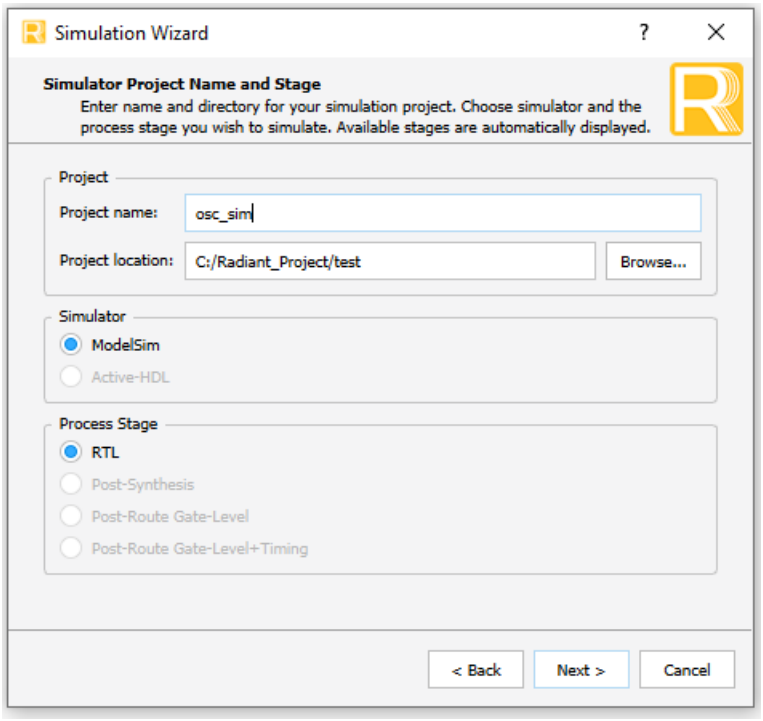

Figure 3.4. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 3.5.

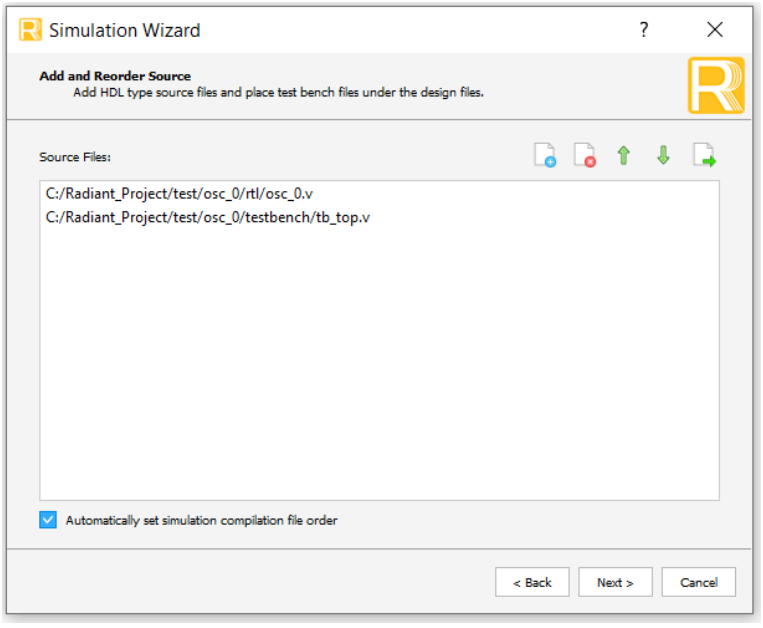

Figure 3.5. Adding and Reordering Source

3. Click **Next**. The **Summary** window is shown. Click **Finish** to run the simulation.

Note: It is necessary to follow the procedure above until it is fully automated in the Lattice Radiant software Suite.

The results of the simulation in our example are provided in Figure 3.6.

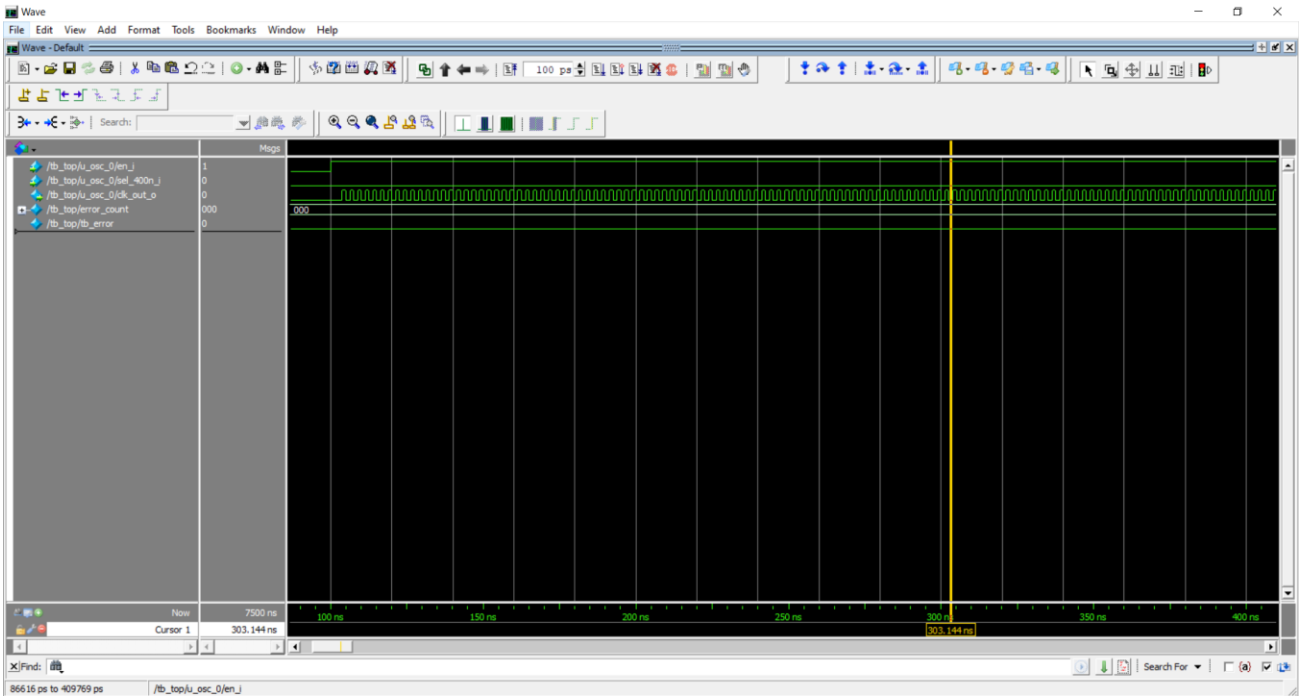

Figure 3.6. Simulation Waveform

Technical Support Assistance

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

Revision History

Revision 1.0, October 2022

Section	Change Summary
All	Initial release.

www.latticesemi.com