CORDIC IP

IP Version: v1.6.1

User Guide

FPGA-IPUG-02136-1.7

December 2025

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, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. 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. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. 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.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CORDIC	Coordinate Rotation Digital Computer
FPGA	Field Programmable Gate Array
IP	Intellectual Property
LSB	Least Significant Bit
MSB	Most Significant Bit
RTL	Register Transfer Level

1. Introduction

This user guide describes the Lattice Coordinate Rotation Digital Computer (CORDIC) IP core. The CORDIC IP Core is configurable and supports several functions, including rotation, translation, sin and cos, and arctan. Two architecture configurations are supported for the arithmetic unit: parallel, in which the output data is calculated in a single clock cycle, and word-serial, in which the output data is calculated over multiple clock cycles. The input and output data widths and computation iterative numbers are configurable over a wide range of values. The IP core uses full precision arithmetic internally while supporting variable output precision and several choices of rounding algorithms.

1.1. Quick Facts

Table 1.1 presents a summary of the CORDIC IP Core.

Table 1.1. CORDIC Quick Facts

IP Requirements	Supported Devices	CrossLink™-NX, Certus™-NX, Certus-NX-RT, CertusPro™-NX, CertusPro-NX-RT, MachXO5™-NX, Lattice Avant™, and Certus-N2
	IP Changes[1]	For a list of changes to the IP, refer to the CORDIC IP Release Notes (FPGA-RN-02091).
Resource Utilization	Resources	See Appendix A. Resource Utilization
Design Tool Support	Lattice Implementation	IP Core v1.6.1 – Lattice Radiant™ Software 2025.2
	Synthesis	Lattice Synthesis Engine
		Synopsys® Synplify Pro® for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant software user guide.

Notes:

1. In some instances, the IP may be updated without changes to the user guide. This user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

1.2. Features

The key features of CORDIC IP Core include:

- Functional configurations
  - Vector rotation (Polar to rectangular)
  - Vector translation (Rectangular to Polar)
  - Sin and Cos
  - Arctan
- Input data widths from 8 to 32 bits
- Iterative number from 4 to 32
- Optional pre-rotation module
- Optional amplitude compensation scaling module to compensate for CORDIC algorithm's output amplitude scale factor
- Selectable rounding: Truncation, Rounding Up, Rounding away from zero, Convergent Rounding
- Parallel architectural configuration for high throughput
- Word serial architectural configuration for small area
- Signed 2's complement data
- Optional control signals: ce_i and sr_i
- Full precision arithmetic

1.3. Conventions

1.3.1. Nomenclature

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

1.3.2. Signal Names

Signal Names that end with:

- _*n* are active low
- _*i* are input signals
- _*o* are output signals
- _io are bi-directional input/output signals

1.4. Attributes

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

2. Functional Description

2.1. General Description of the CORDIC Algorithm

The CORDIC algorithm is an iterative method that uses simple arithmetic operations such as addition, subtraction, bit shift and table look up to perform hyperbolic and trigonometric functions. The CORDIC algorithm was initially designed to perform a vector rotation, where the vector (x, y) is rotated through the angle θ yielding a new vector (x', y'). Using a matrix form, a planar rotation for a vector of (x, y) is defined as:

$$x' = x\cos\theta - y\sin\theta \qquad (1)$$
$$y' = y\cos\theta + x\sin\theta$$

Note: θ is the angle to be traversed. With the CORDIC algorithm, the traversal is accomplished in iterative steps in which each step completes a small part of the rotation.

A single step is defined by the following equation:

$$x_{i+1} = \cos\theta_i(x_i - y_i\tan\theta_i) \qquad (2)$$
$$y_{i+1} = \cos\theta_i(y_i - x_i\tan\theta_i)$$

The number of multipliers required is reduced by selecting the angle steps such that the tangent of a step is a power of 2. The angle for each step is given by:

$$\theta_i = \arctan\left(\frac{1}{2^i}\right) \qquad (3)$$

Multiplying or dividing by a power of 2 can be implemented using a simple shift operation. All iteration-angles summed must equal the rotation angle θ.

$$\sum_{i=0}^{\infty} d_i\,\theta_i \; where \; d_i = \{-1; +1\} \qquad (4)$$

This results in the following equation for $\tan\theta_i$:

$$\tan\theta_i = d_i 2^{-i} \qquad (5)$$

Combining equations 2 and 5 results in:

$$x_{i+1} = \cos\theta_i(x_i - y_i\, d_i\, 2^{-i}) \qquad (6)$$
$$x_{i+1} = \cos\theta_i(y_i + x_i\, d_i\, 2^{-i})$$

The iterative rotation can now be expressed as:

$$x_{i+1} = \mathrm{K}_i(x_i - y_i\, d_i\, 2^{-i}) \qquad (7)$$
$$y_{i+1} = \mathrm{K}_i(y_i + x_i\, d_i\, 2^{-i})$$

where

$$K_{i+1} = \cos(\tan^{-1} 2^{-i}) = \frac{1}{\sqrt{1 + 2^{-2i}}}$$
$$\mathrm{d}_i = \pm 1$$

The CORDIC rotator is normally operated in one of two modes. The first, called rotation, rotates the input vector by a specified angle. The second mode, called vectoring, rotates the input vector to the x-axis while recording the angle required to make that rotation.

For rotation mode, the CORDIC equations are:

$$x_{i+1} = x_i - y_i\, d_i\, 2^{-i} \qquad (8)$$
$$y_{i+1} = y_i - x_i\, d_i\, 2^{-i}$$
$$z_{i+1} = z_i - d_i \tan^{-i}(2^{-i})$$

where d_i = –1 if z_i <0, +1 otherwise. Here z_i is the residual angle in the angle accumulator with the initial value z_0 as the angle to be rotated.

In vectoring mode, the CORDIC vectoring function works by seeking to minimize the y component of the residual vector at each rotation. The sign of the residual component is used to determine which direction to rotate next. If the angle accumulator is initialized with zero, it will contain the traversed angle at the end of the iterations. For vectoring mode, the CORDIC equations are:

$$x_{i+1} = x_i - y_i\, d_i\, 2^{-i} \qquad (9)$$
$$y_{i+1} = y_i - x_i\, d_i\, 2^{-i}$$
$$z_{i+1} = z_i - d_i \tan^{-i}(2^{-i})$$

where $d_i = -1\ if\ y_i < 0, +1$ otherwise

In sin/cos mode, the unit vector is rotated by the input phase angle θ, generating the output vector $(\cos(\theta), \sin(\theta))$. The rotation mode CORDIC operation can simultaneously compute the sine and cosine of the input angle θ. Setting the x component to 1 and y component to zero reduces the rotation mode. This results the equations 10 from equations 1:

$$x' = \cos\theta \qquad (10)$$
$$y' = \sin\theta$$

In arctangent mode, $\theta = \arctan(y_0/x_0)$ is directly computed using the vectoring mode if the angle accumulator is initialized with zero.

$$z_n = z_0 + \arctan\left(\frac{y_0}{x_0}\right) \qquad (11)$$

2.2. Block Diagram

The CORDIC IP core block diagram is shown in Figure 2.1.

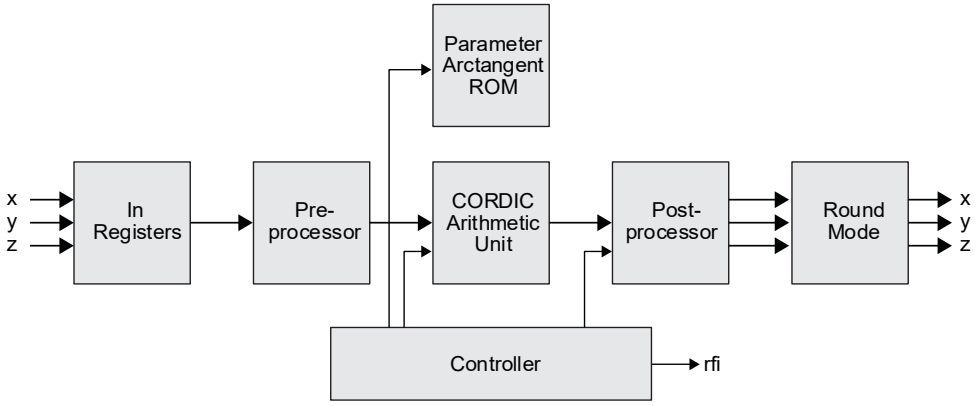

Figure 2.1. CORDIC IP Core Block Diagram

2.2.1. Data Path

2.2.1.1. Pre-processor

The CORDIC rotation and vectoring algorithms are limited to rotation angles between –π/2 and π/2. This limitation is due to the use of 2° for the tangent in the first iteration. For composite rotation angles larger than π/2, an additional rotation is required.

2.2.1.2. CORDIC Arithmetic Unit

The CORDIC arithmetic unit performs the actual CORDIC algorithm. Two architecture configurations are available for the arithmetic unit: parallel (with single-cycle data throughput) and word-serial (with multiple-cycle throughput). The parallel configuration has a pipeline-structured core and can perform a CORDIC transformation each clock cycle, producing a new output every cycle. In contrast with the parallel structure, word-serial architecture produces a new output every N cycles. Here N is the user input in the Radiant GUI interface for the Iteration Number parameter.

2.2.1.3. Arctan ROM

The arc tangent ROM stores the tan^(–1) (2^(–i)) values. Its data width is variable, address width is log2 (number of iterations–1), address depth is 2 ^ log2 (number of iterations–1).

2.2.1.4. Controller

The controller module control generates all signals necessary for carrying out the iterations, including ROM addressing, ready for input (rfi) and output valid (outvalid). I/O port definition details are explained in Table 2.5.

2.2.1.5. Post-processor

The CORDIC algorithm introduces a scale factor that causes a magnitude gain that must be compensated for at the end. See Equation 7 in the General Description of the CORDIC Algorithm section. The post-processor module contains logic to correct the scale factor. In addition, it corrects the phase rotation introduced by the pre-processor module (if present).

2.2.1.6. Rounding

The rounding module provides four types of rounding, depending on the Rounding Method parameter:

- None (Truncation) – Discards all bits to the right of the output least significant bit and leaves the output uncorrected.
- Rounding up – Rounds up if the fractional part is exactly one-half.
- Rounding away from zero – Rounds away from zero if the fractional part is exactly one-half.
- Convergent rounding – Rounds to the nearest even value if the fractional part is exactly one-half.

2.2.2. CORDIC Functions

2.2.2.1. Vector Rotation

Polar to Rectangular Translation: In vector rotation mode, the input vector (x,y) is rotated by a specified angle, θ, giving a new output vector, (x', y'). Because of the CORDIC algorithm scale factor, a magnitude gain is introduced as shown in Figure 2.2. This magnitude gain is compensated for by the CORDIC IP post-processor module.

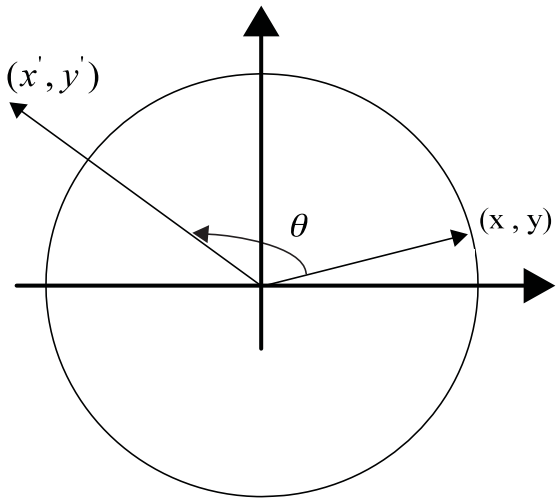

Figure 2.2. Vector Rotation

The inputs, xin_i, yin_i and phasein_i, are limited to the ranges provided in Table 2.1. Inputs outside the ranges will produce unpredictable results.

Table 2.1. Vector Rotation Input/Output

Signal	Description
xin_i	Input X Coordinate Range: –1 <= xin_i <= 1
yin_i	Input Y Coordinate Range: –1 <= yin_i <= 1
phasein_i	Input Rotation Angle Range: – π <= phasein_i <= π
xout_o	Output X Coordinate Range: $-\sqrt{2}$ <= xout_o <= $\sqrt{2}$
yout_o	Output Y Coordinate Range: $-\sqrt{2}$ <= yout_o <= $\sqrt{2}$

2.2.2.2. Vector Translation

Rectangular to Polar Translation: In vector translation mode, the input vector (x,y) is rotated through whatever angle is necessary to align the result vector with the x-axis, as shown in Figure 2.3. Output is the angle rotated and the magnitude on the x-axis after rotation.

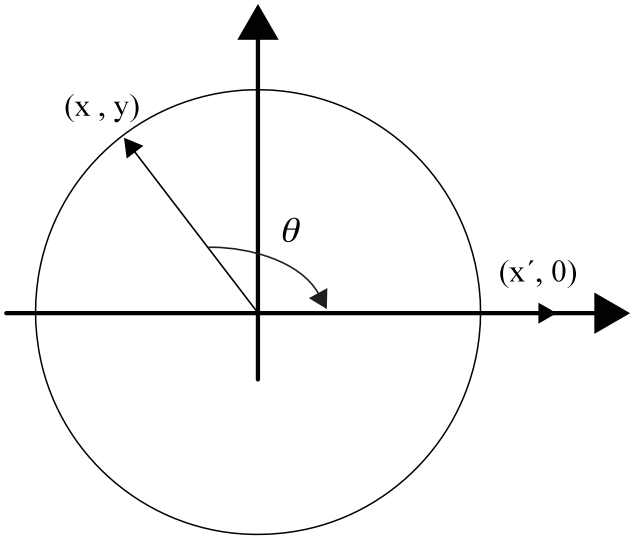

Figure 2.3. Vector Translation

The inputs, xin_i and yin_i are limited to the ranges provided in Table 2.2. Inputs outside the ranges will produce unpredictable results.

Table 2.2. Vector Translation Input/Output

Signal	Description
xin_i	Input X Coordinate Range: –1 <= xin_i <= 1
yin_i	Input Y Coordinate Range: –1 <= yin_i <= 1
xout_o	Output Magnitude Range: $-\sqrt{2}$ <= xout_o <= $\sqrt{2}$
phaseout_o	Output Phase Range: – π <= phaseout_o <= π

2.2.2.3. Sin and Cos

In sin/cos mode, the unit vector is rotated by the input phase angle θ providing the output vector (cos (θ), sin(θ)). The input angle, phasein_i, is limited to the range provided in Table 2.3. Inputs outside this range will produce unpredictable results.

Table 2.3. Sin and Cos Input/Output

Signal	Description
phasein_i	Input Phase Range: – π <= phasein_i <= π
xout_o	Output cos(θ) Range: –1 <= xout_o <= 1
yout_o	Output sin(θ) Range: –1 <= yout_o <= 1

2.2.2.4. Arctan

In arctan mode, the input vector, (x, y) is rotated until the y component is zero, yielding the output angle, (y/x). The inputs xin_i and yin_i are limited to the ranges given in Table 2.4. Inputs outside the ranges will produce unpredictable results.

Table 2.4. Arctan Input/Output

Signal	Description
xin_i	Input X Coordinate Range: –1 <= xin_i <= 1
yin_i	Input Y Coordinate Range: –1 <= yin_i <= 1
phaseout_o	Output Phase Range: – π <= phaseout_o <= π

2.3. Signal Description

The top-level interface diagram for the CORDIC IP Core is shown in Figure 2.4.

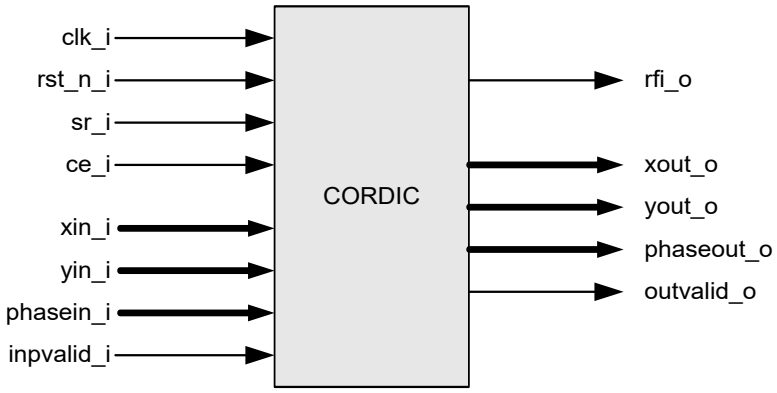

Figure 2.4. Top-Level Interface for CORDIC IP Core

Table 2.5 lists the input and output signals for CORDIC IP Core.

Table 2.5. CORDIC IP Core Signal Description

Port Name	I/O	Size	Description
Clock and Reset			
clk_i	I	1	System clock for data and control inputs and outputs.
rst_n_i	I	1	System wide asynchronous active-low reset signal.
General I/O			
xin_i	I	Width depends on *Input data width*	X component of input sample, Will be available for *Mode* other than Sin/Cos
yin_i	I	Width depends on *Input data width*	Y component of input sample, Will be available for *Mode* other than Sin/Cos
phasein_i	I	Width depends on *Input data width*	Phase component of input sample, Will be available for *Mode* Rotation or Sin/Cos
inpvalid_i	I	1	Input valid signal. The input data is read-in only when inpvalid_i is high.
xout_o	O	Width depends on *Output data width*	X component of output sample, Will be available for *Mode* other than Arctan
yout_o	O	Width depends on *Output data width*	Y component of output sample, Will be available for *Mode* Rotation or Sin/Cos

Port Name	I/O	Size	Description
phaseout_o	O	Width depends on *Output data width*	Phase component of output sample, Will be available for *Mode* Translation or Arctan
outvalid_o	O	1	Output data qualifier. Output data is valid only when this signal is high.
rfi_o	O	1	Ready for input. This output, when high, indicates that the IP core is ready to receive the next input data. A valid data may be applied at xin_i, yin_i and phasein_i only if rfi_o was high during the previous clock cycle.
Optional I/O			
ce_i	I	1	Clock Enable. Applicable when *Clock enable* is Checked.
sr_i	I	1	Synchronous Reset, Applicable when *Synchronous reset* is Checked.

2.4. Attributes Summary

Table 2.6. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
Basic Options			
Mode	Rotation, Translation, Sin/Cos, Arctan	Rotation	—
Architecture	Word-Serial, Parallel	Parallel	—
Number of iterations	4–32	16	—
Compensation	None, LUT-Based, DSP-Based	None	—
Pre-Rotation	Checked, Unchecked	Checked	—
Advanced Options			
Rounding Method	Truncation, Rounding up, Rounding away from zero, Convergent rounding	Truncation	—
Input data width	8–32	16	—
Output data width	8–32	16	—
Optional Ports			
Synchronous reset	Checked, Unchecked	Unchecked	—
Clock enable	Checked, Unchecked	Unchecked	—

Table 2.7. Attributes Descriptions

Attribute	Description
Basic Options	
Mode	Specifies whether the CORDIC is: rotate, translate, sin and cos, arctan.
Architecture	Specifies two architecture configurations for the CORDIC IP Core: Parallel, with single cycle data throughput, and Word-serial, with multiple cycles throughput.
Number of iterations	Specifies the number of internal add-sub iterations to perform
Compensation	Specifies three compensation configurations for CORDIC magnitude scaling. The outputs are compensated using a LUT-based multiplier or using the Block multiplier
Pre-Rotation	Specifies whether the core is instantiation of the pre-rotation module.
Advanced Options	
Rounding Method	Specifies the rounding method when there is a need to drop one or more LSBs from the true output.
Input data width	Input data width of CORDIC IP Core.
Output data width	Output data width of CORDIC IP Core.
Optional Ports	
Synchronous reset	Specifies if a synchronous reset port is needed in the IP. Synchronous reset signal resets all the registers in the CORDIC IP Core.
Clock enable	Specifies if a clock enable port is needed in the IP. Clock enable control can be used for power saving when the core is not used. Use of clock enable port increases the resource utilization and may affect the performance due to the increased routing congestion.

2.5. Configuring the CORDIC IP Core

2.5.1. Basic Options

The options for mode, architecture, number of iterations and compensation are independent and specified in the Basic Options tab of the interface.

2.5.1.1. Architecture specification

The CORDIC IP Core provides two architecture configurations for the arithmetic unit: parallel (with single cycle data throughput) and word-serial (with multiple-cycle throughput). Because of the pipelined structure, the CORE can perform a CORDIC transformation each clock cycle, thus producing a new output every cycle. In contrast with parallel structures, word-serial architecture produces a new output every N cycles. Figure 2.5 shows a basic CORDIC arithmetic unit.

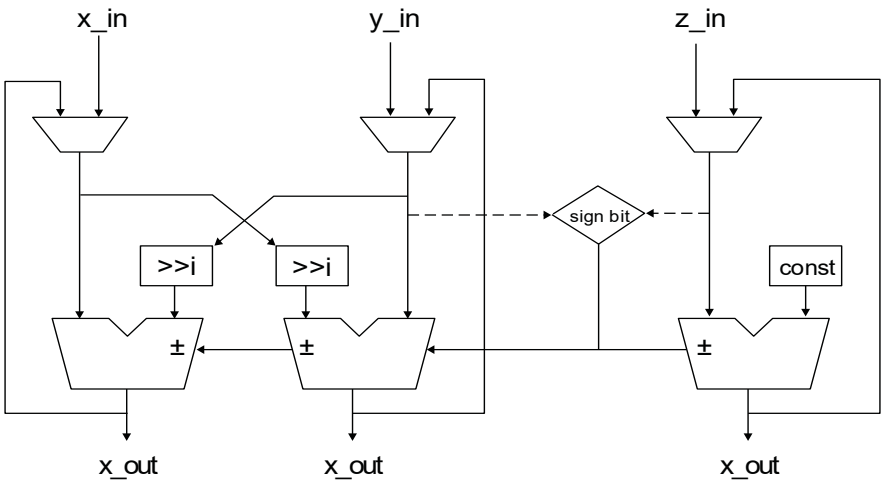

Figure 2.5. Basic CORDIC Arithmetic Unit

2.5.1.2. Iterations Specification

Parameter iteration specifies the number of internal add-sub iterations performed by the CORDIC processor in deriving the result. It determines the accuracy of the output: if the number is larger, the accuracy of the output is higher.

2.5.1.3. Pre-rotation Specification

When the pre-rotation module is selected, the CORDIC operational range extends to the full circle; otherwise the operational range is limited between $-\pi/2$ and $\pi/2$. Angle ranges outside the ranges will produce an unpredictable result if the pre-rotation module is not selected. The following describes an initial pre-rotation $\pm\pi/2$:

$$\begin{aligned} x' &= d \cdot \mathrm{y} \\ y' &= d \cdot \mathrm{x} \\ z' &= z + \mathrm{d} \cdot \frac{\pi}{2} \end{aligned} \qquad (12)$$

2.5.1.4. Compensation Specification

In the CORDIC algorithm, the magnitude outputs, xout_o and yout_o, are generated with a magnitude gain. The compensation module provides three configurations to compensate for the CORDIC magnitude scale factor.

- None – The outputs xout_o and yout_o will not be compensated. It is the user's obligation to compensate and scale for the magnitude outputs gain introduced by the CORDIC algorithm. Refer to General Description of the CORDIC Algorithm of this document for details, especially the K factor in Equation.
- LUT-based – The outputs xout_o and yout_o are compensated using a LUT-based multiplier.
- DSP-based – The outputs xout_o and yout_o are compensated using a DSP-based multiplier

2.5.2. Advanced Options

The controls in this tab are used to define the various data widths and rounding methods used in the data path. The widths of the input data and output data can be defined independently.

2.5.2.1. Round Method Specification

The CORDIC IP Core provides four rounding modes. Examples of round method are provided in Table 2.8.

- Truncation – The outputs, xout_o, yout_o and phaseout_o, are truncated. The LSBs are removed to match the specified output width.
- Rounding up – The outputs, xout_o, yout_o and phaseout_o, are rounded up (0.5 rounded up).
- Rounding away from zero – The outputs, xout_o, yout_o and phaseout_o, are rounded (0.5 rounded up, –0.5 rounded down).
- Convergent rounding – The outputs, xout_o, yout_o and phaseout_o, are rounded towards the nearest even number.

Table 2.8. Round Method

Value	Truncation	Rounding Up	Rounding Away from Zero	Convergent Rounding
1.50	1	2	2	2
–1.50	–2	–1	–2	–2
0.50	0	1	1	0
–0.50	–1	0	–1	0
0.25	0	0	0	0
–0.25	–1	0	0	0
0.65	0	1	1	0

2.5.2.2. Input/Output Width Specification

The input/output data widths can be configured in the range 8 to 32 bits.

2.5.2.3. Data Format Specification

The data signals are: xin_i, yin_i, xout_o and yout_o. The input data signals, xin_i and yin_i, must be in the range [–1,1]. Input data outside the range will produce unpredictable results.

- Input Data Signals
  Input data signals are represented in decimal format using bus format (as little endian). For N-bit input data signal, the (N-2) LSB represent the fractional component to the left of the decimal place and the MSB represents the sign bit.
  For example, when the *Input data width* is 8, +1 and –1 are represented as:
  01000000 => 01.000000 => +1.0
  11000000 => 11.000000 => –1.0
  When the *Input data width* is 12, +1 and –1 are represented as:
  010000000000 => 01.0000000000 => +1.0
  110000000000 => 11.0000000000 => –1.0
- Output Data Signals
  If *Compensation* is LUT- based or DSP-based, the output data signal format is the same as the input data signal format. The range of the output data signal is $[-\sqrt{2}, \sqrt{2}]$.
  For N-bit output data signal, the (N-2) LSB represent the fractional component to the left of the decimal place and the MSB represents the sign bit.

  For example, when the *Output data width* is 8, in the data format, +1 and –1 are represented:

  01000000 => 01.000000 => +1.0
  11000000 => 11.000000 => –1.0

  When the *Output data width* is 12, in the data format, +1 and –1 are represented:

  010000000000 => 01.0000000000 => +1.0
  110000000000 => 11.0000000000 => –1.0

  If *Compensation* is None, the output data signals format is different from the input data signals. Due to the magnitude gain introduced by the CORDIC algorithm, without the compensation, the range of the output data signal can be larger than 2 or less than –2, so it will need 2 bits to represent the decimal number.
  For the N-bit output data signal, the (N-3) LSB represent the fractional component to the left of the decimal place and the MSB represents the sign bit.

  For example, when the *Output data width* is 8, in the data format, +1 and –1 are represented:

  00100000 => 001.00000 => +1.0
  11000000 => 111.00000 => –1.0

  When the *Output data width* is 12, in the data format, +2 and –2 are represented:

  010000000000 => 010.000000000 => +2.0
  110000000000 => 110.000000000 => –2.0

  +2.25 and –2.25 are represented:

  010010000000 => 010.010000000 => +2.25
  101110000000 => 101.110000000 => –2.25

2.5.2.4. Phase Format Specification

- Phase Signals
  The phase signals are phasein_i and phaseout_o. The input phase signal, phasein_i, must be in the range [–π,π]. Input phase outside this range will produce unpredictable results.
  The phase signals, phasein_i and phaseout_o, are always the same representation.
  For N-bit phase signal, the (N-3) LSB represents the fractional component to the left of the decimal place and the MSB represents the sign bit.

2.6. Timing Descriptions

Timing diagrams for the CORDIC IP Core are shown in Figure 2.6, Figure 2.7, and Figure 2.8.

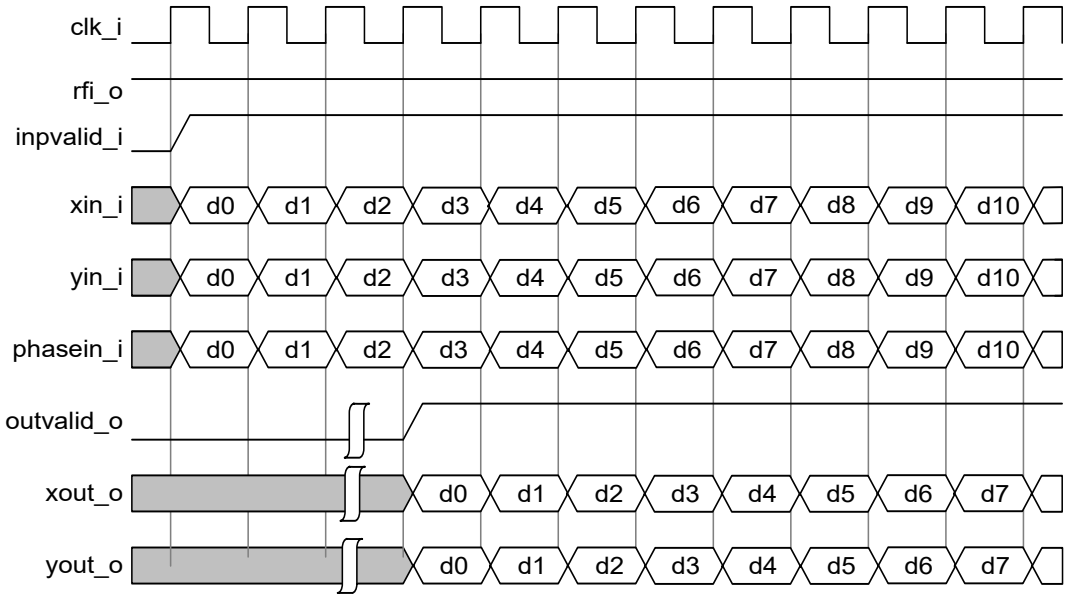

Figure 2.6. Timing Diagram for Parallel CORDIC (Rotation Mode) with Continuous Inputs

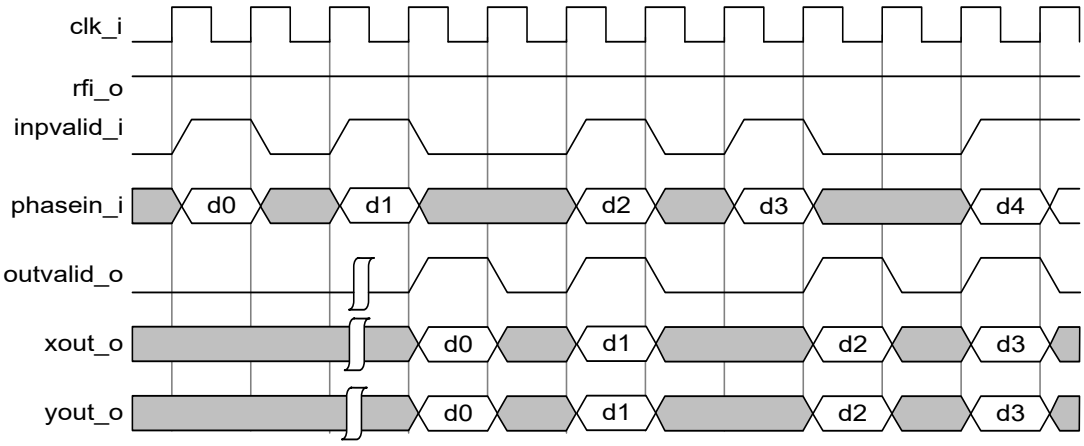

Figure 2.7. Timing Diagram for Parallel CORDIC (Sin/Cos Mode) with Gap Inputs

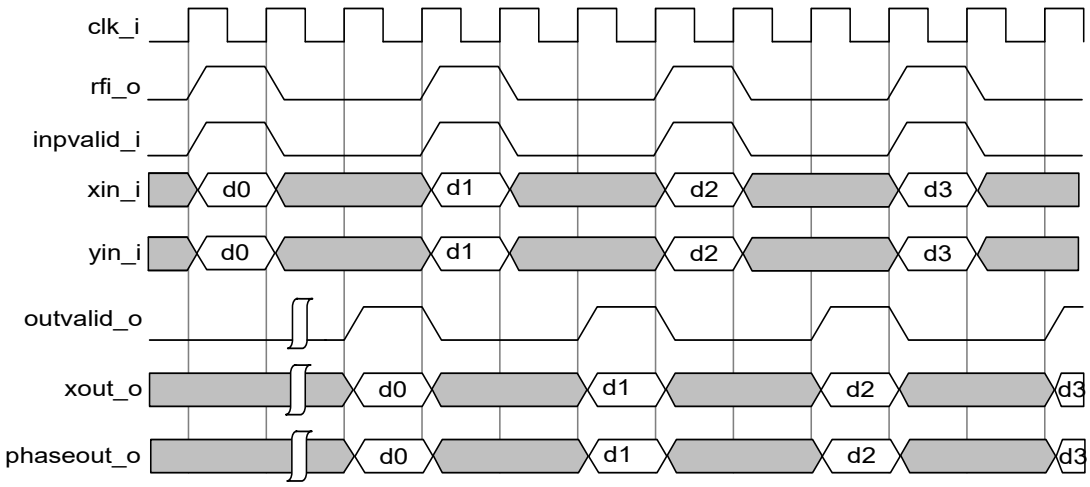

Figure 2.8. Timing Diagram for Serial CORDIC (Translation Mode)

3. IP Generation, Simulation, and Validation

This section provides information on how to generate the IP Core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, refer to the Lattice Radiant software user guide.

Note: The screenshots provided are for reference only. Details may vary depending on the version of the IP or software being used. If there have been no significant changes to the GUI, a screenshot may reflect an earlier version of the IP.

3.1. Licensing the IP

The CORDIC IP is provided at no additional cost with the Lattice Radiant software.

3.2. Generating the IP

The Lattice Radiant software allows the user to customize and generate modules and IPs and integrate them into the device's architecture. The procedure for generating the CORDIC IP Core in Lattice Radiant software is described below.

To generate the CORDIC IP Core:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click on **CORDIC** under **IP, DSP** 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**.

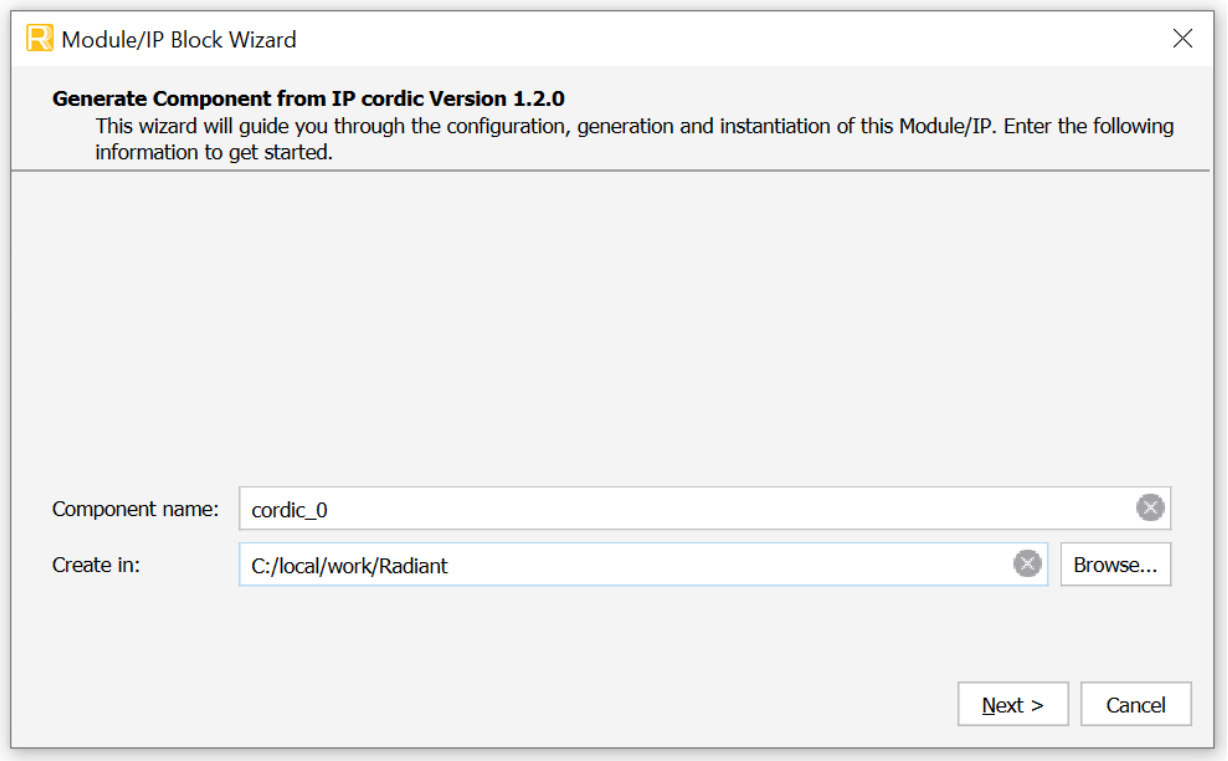

Figure 3.1. Module/IP Block Wizard

3. In the module's dialog box of the **Module/IP Block Wizard** window, customize the selected CORDIC IP Core using drop-down menus and check boxes. As a sample configuration, see Figure 3.2. For configuration options, see Table 2.6.

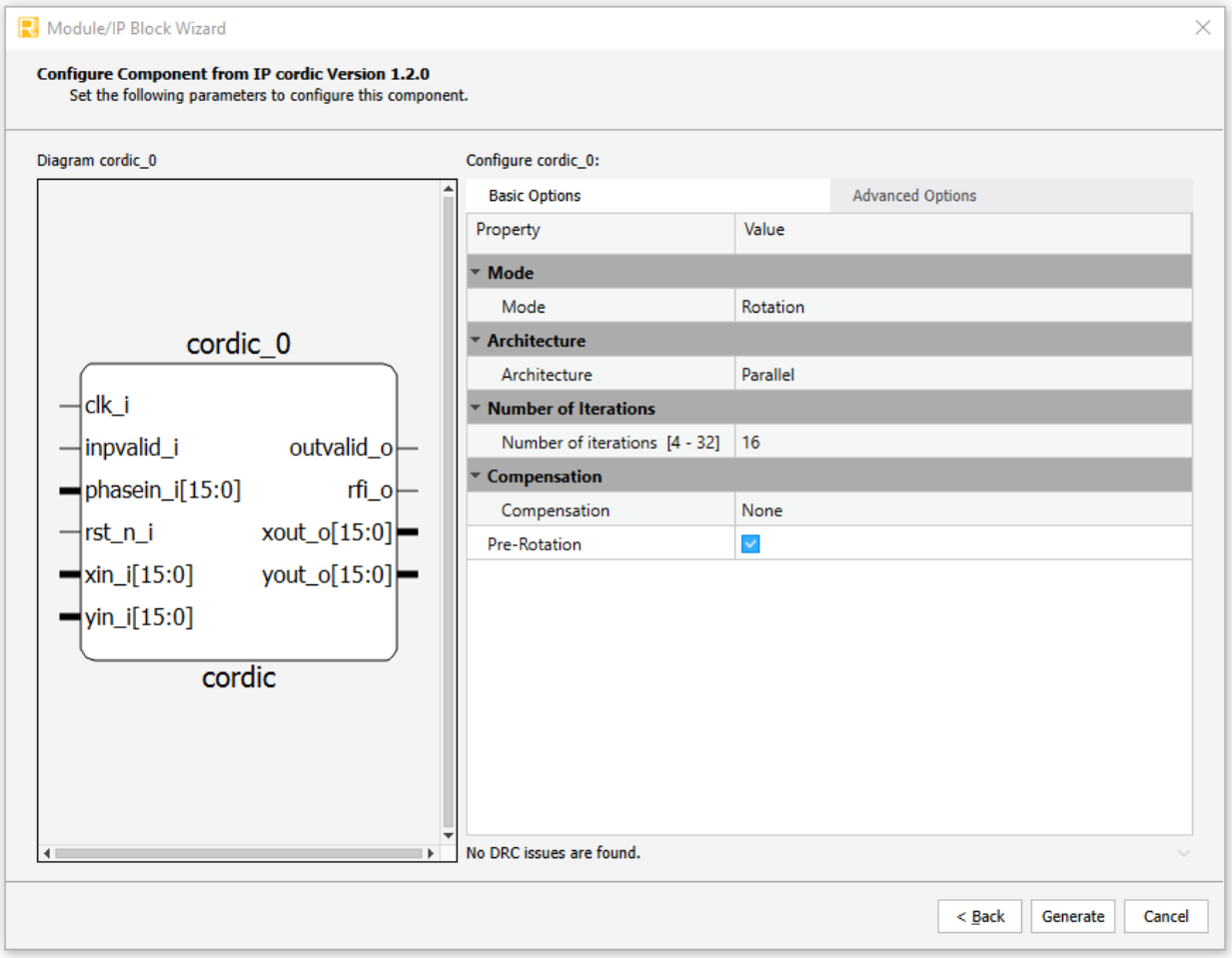

Figure 3.2. Configure User Interface of CORDIC IP Core

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

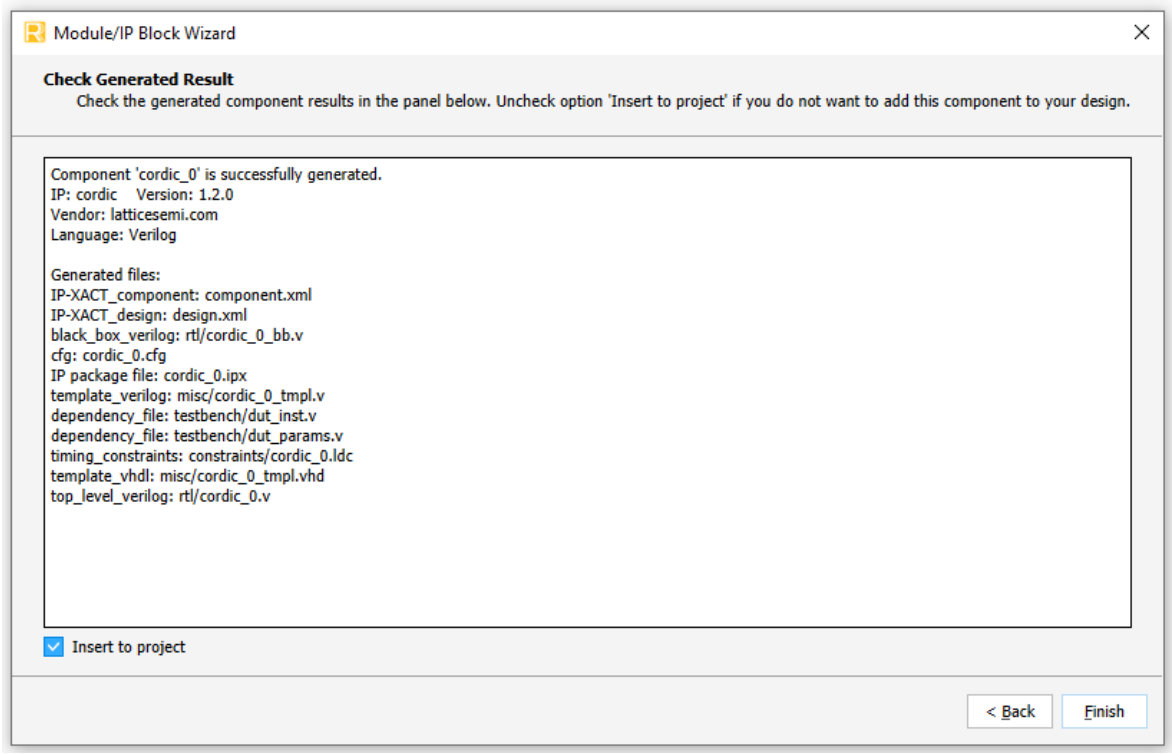

Figure 3.3. Check Generating Result

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 CORDIC IP Core package includes the closed-box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core 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 IP core is also provided. The user 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 parameter values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl/<Component name>.v	This file provides an example RTL top file that instantiates the IP core.
rtl/<Component name>_bb.v	This file provides the synthesis closed-box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	These files provide instance templates for the IP core.
eval/constraint.pdc	This file provides information on how to constrain this IP. Refer to section 3.4 on how to use this file.

3.3. Running Functional Simulation

To run functional simulation:

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

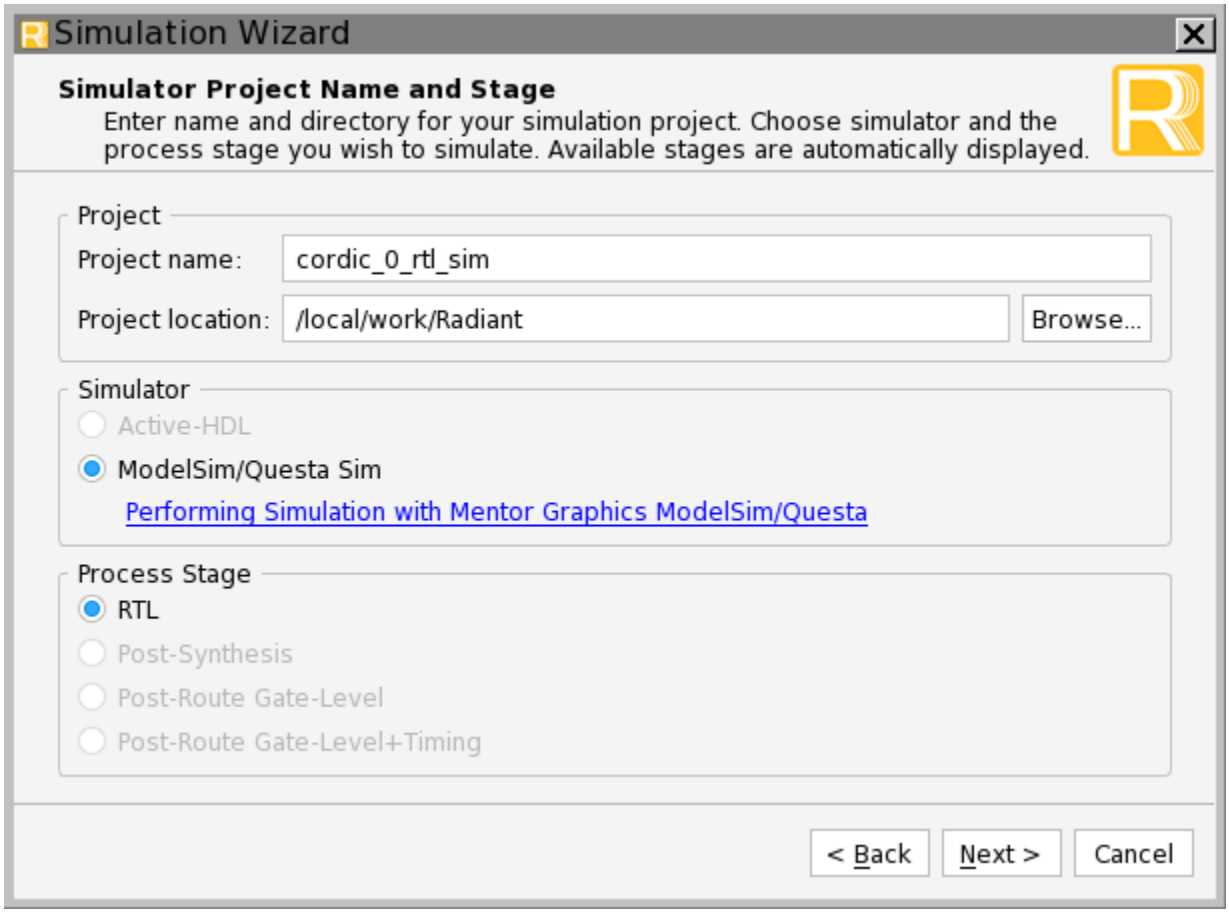

Figure 3.4. Simulation Wizard

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

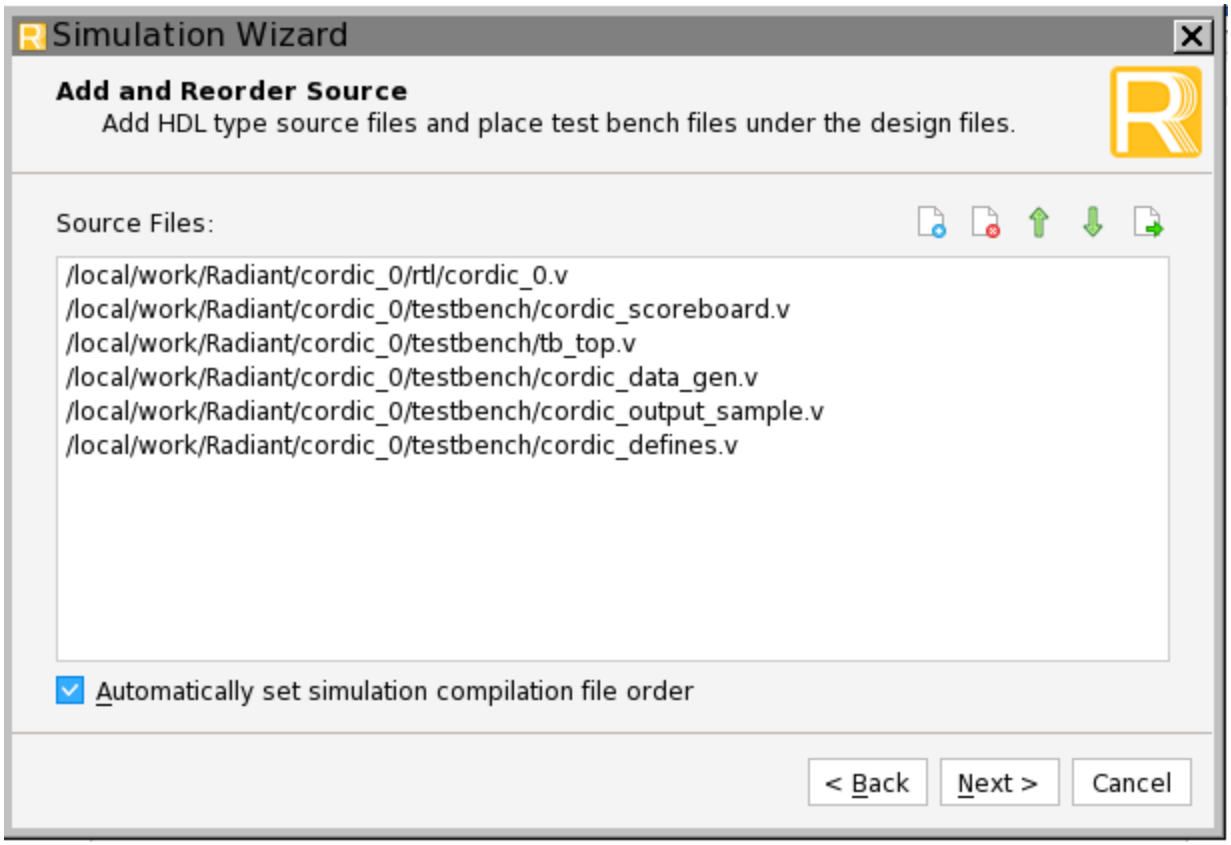

Figure 3.5. Adding and Reordering Source

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.

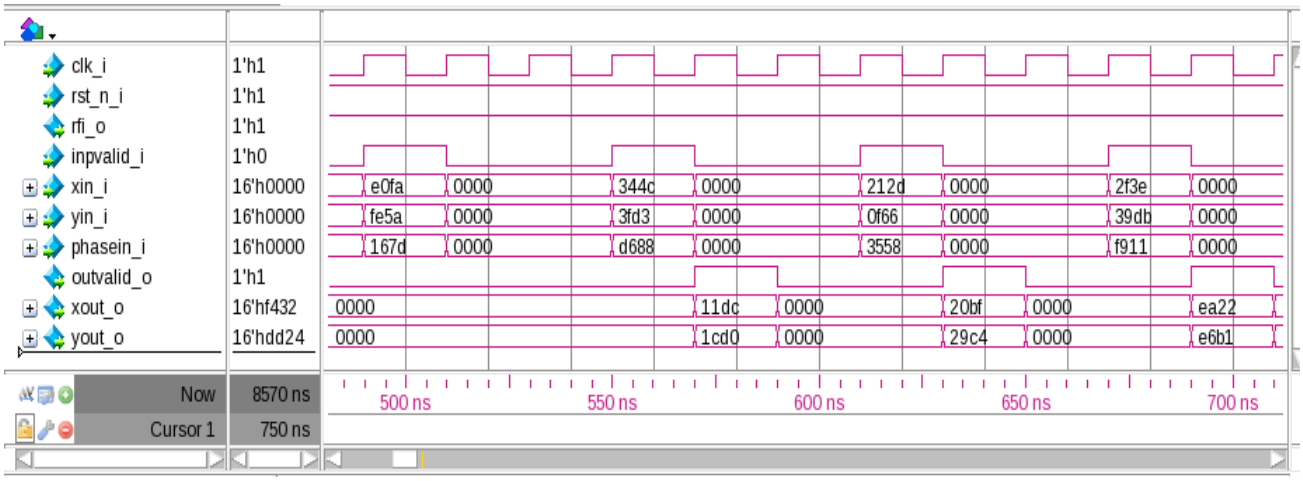

Figure 3.6. Simulation Waveform

3.4. Constraining the IP

You need to provide proper timing and physical design constraints to ensure that your design meets the desired performance goals on the FPGA. Add the content of the following IP constraint file to your design constraints:

<IP_Instance_Path>/<IP_Instance_Name>/eval/constraint.pdc.

The constraint file has been verified during IP evaluation with the IP instantiated directly in the top-level module. You can modify the constraints in this file with thorough understanding of the effect of each constraint.

Refer to Lattice Radiant Timing Constraints Methodology for details on how to constraint your design.

3.5. IP Evaluation

The IP Core supports Lattice's IP evaluation capability when used in the supported FPGA family and targeted device. This makes it possible to create versions of the IP core that operates in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. The IP evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default. To change this setting, go to Project > Active Strategy > LSE/Synplify Pro Settings.

Appendix A. Resource Utilization

Table A.2 shows the resource utilization of the CORDIC IP Core for the LIFCL-40-9BG400I device using Synplify Pro of Lattice Radiant software. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.2. LIFCL-40-9BG400I Device Resource Utilization

Configuration	CLK Fmax (MHz)*	Registers	LUTs	DSP
Default	200	1157	1334	0
Mode: Translation, Others = Default	200	1145	1320	0
Mode: Sin/Cos, Others = Default	200	1003	1146	0
Architecture= Word-Serial, *Others = Default*	200	302	585	0
Compensation: DSP-based, Others = Default	200	1242	1311	8

***Note:** Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 100 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.3 shows the resource utilization of the CORDIC IP Core for the LFD2NX-40-9BG256I device using Synplify Pro of Lattice Radiant software. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.3. LFD2NX-40-9BG256I Device Resource Utilization

Configuration	CLK Fmax (MHz)*	Registers	LUTs	DSP
Default	200	1157	1334	0
Mode: Translation, Others = Default	200	1145	1320	0
Mode: Sin/Cos, Others = Default	200	1003	1146	0
Mode: Arctan, Others = Default	200	302	585	0
Compensation: DSP-based, *Rounding Method*: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default	200	1242	1311	8

***Note:** Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 100 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.4 shows the resource utilization of the CORDIC IP Core for the LFMXO5-25-9BBG400I device using Synplify Pro of Lattice Radiant Software. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.4. LFMXO5-25-9BBG400I Device Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Default	200	1157	1334	0
Mode: Translation, Others = Default	200	1145	1320	0
Mode: Sin/Cos, Others = Default	200	1003	1146	0
Mode: Arctan, Others = Default	200	1072	1276	0
Compensation: DSP-based,	197	1437	2621	8

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Rounding Method: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default				

***Note:** Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 100MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.5 shows the resource utilization of the CORDIC IP Core for the LFMXO5-25-7BBG400I device using Synplify Pro of Lattice Radiant Software. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.5. LFMXO5-25-7BBG400I Device Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Default	145	1157	1334	0
Mode: Translation, Others = Default	156	1145	1320	0
Mode: Sin/Cos, Others = Default	153	1003	1146	0
Mode: Arctan, Others = Default	158	1072	1276	0
Compensation: DSP-based, *Rounding Method*: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default	160.694	1437	2621	8

***Note:** Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 100 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.6 shows the resource utilization of the CORDIC IP Core for the LAV-AT-E70-1LFG1156I device using Synplify Pro of Lattice Radiant Software. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.6. LAV-AT-E70-1LFG1156I Device Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Default	201.45	1165	1710	0
Mode: Translation, Others = Default	201.45	1153	1458	0
Mode: Sin/Cos, Others = Default	201.45	1003	1446	0
Mode: Arctan, Others = Default	201.45	1080	1414	0
Compensation: DSP-based, *Rounding Method*: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default	201.45	1610	3099	8

***Note:** Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.7 shows the resource utilization of the CORDIC IP Core for the LFD2NX-9-7MG121C device using Synplify Pro of Lattice Radiant Software version 2024.2. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.7. LFD2NX-9-7MG121C Device Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Default	200	1157	1398	0
Mode: Translation, Others = Default	198.059	1145	1370	0
Mode: Sin/Cos, Others = Default	199.601	1003	1147	0
Mode: Arctan, Others = Default	193.498	1072	1278	0
Compensation: DSP-based, *Rounding Method*: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default	154.967	1437	2685	8

*__Note:__ Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.8 shows the resource utilization of the CORDIC IP Core for the LFD2NX-17-7MG121C device using Synplify Pro of Lattice Radiant Software version 2024.2. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.8. LFD2NX-17-7MG121C Device Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Default	200	1157	1398	0
Mode: Translation, Others = Default	198.059	1145	1370	0
Mode: Sin/Cos, Others = Default	199.601	1003	1147	0
Mode: Arctan, Others = Default	193.498	1072	1278	0
Compensation: DSP-based, *Rounding Method*: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default	154.967	1437	2685	8

*__Note__: Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.9 shows the resource utilization of the CORDIC IP Core for the LFD2NX-28-7MG121C device using Synplify Pro of Lattice Radiant Software version 2024.2. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.9. LFD2NX-28-7MG121C Device Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Default	200	1157	1398	0
Mode: Translation, Others = Default	166.030	1145	1370	0
Mode: Sin/Cos, Others = Default	183.621	1003	1147	0
Mode: Arctan, Others = Default	174.490	1072	1278	0

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Compensation: DSP-based, *Rounding Method*: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default	150.966	1437	2685	8

*__Note__: Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.10 shows the resource utilization of the CORDIC IP Core for the LFD2NX-40-7MG121C device using Synplify Pro of Lattice Radiant Software version 2024.2. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.10. LFD2NX-40-7MG121C Device Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Default	200	1157	1398	0
Mode: Translation, Others = Default	166.030	1145	1370	0
Mode: Sin/Cos, Others = Default	183.621	1003	1147	0
Mode: Arctan, Others = Default	174.490	1072	1278	0
Compensation: DSP-based, *Rounding Method*: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default	150.966	1437	2685	8

*__Note__: Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.11 shows the resource utilization of the CORDIC IP Core for the LN2-CT-20-1CBG484C device using Synplify Pro of Lattice Radiant Software version 2024.2. Default configuration is used, and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.11. LN2-CT-20-1CBG484C Device Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP
Default	250	1165	1710	0
Mode: Translation, Others = Default	250	1154	1458	0
Mode: Sin/Cos, Others = Default	250	1004	1429	0
Mode: Arctan, Others = Default	250	1081	1414	0
Compensation: DSP-based, *Rounding Method*: Convergent, *Synchronous Reset*: true, *Clock Enable*: true, Others = Default	181.851	1568	3037	8

*__Note__: Fmax is generated when the FPGA design only contains CORDIC IP Core, and the target frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

References

- CORDIC IP Release Notes (FPGA-RN-02091)
- Avant-E web page
- Avant-G web page
- Avant-X web page
- Certus-N2 web page
- Certus-NX web page
- CertusPro-NX web page
- CrossLink-NX web page
- MachXO5-NX web page
- Lattice Radiant Software web page
- Lattice Solutions IP Cores web page
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- Lattice Insights web page for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Note: In some instances, the IP may be updated without changes to the user guide. The user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

Revision 1.7, IP v1.6.1, December 2025

Section	Change Summary

Quick Facts	• In Table 1.1. CORDIC Quick Facts:
	  • Updated from *IP Core v1.6.0 to IP Core v1.6.1.*
	  • Updated from *Lattice Radiant software 2025.1* to *Lattice Radiant software 2025.2*.
	• Added note.

IP Generation, Simulation, and Validation	• Added note, *The screenshots provided are for reference only. Details may vary depending on the version of the IP or software being used. If there have been no significant changes to the GUI, a screenshot may reflect an earlier version of the IP.* • Updated content of Licensing the IP, *The CORDIC IP is provided at no additional cost with the Lattice Radiant software.*
Ordering Part Number	Removed this section.
Revision History	Added note.

Revision 1.6, IP v1.6.0, July 2025

Section	Change Summary
Introduction	Updated Table 1.1. CORDIC Quick Facts: • Renamed Table 1.1 Quick Facts to Table 1.1 CORDIC Quick Facts. • Changed *FPGA Families Supported* to *Supported Devices* in *IP Requirements*. • Added the *Certus-NX-RT* and *CertusPro-NX-RT* devices to *Supported Devices*. • Added *IP Changes* to *IP Requirements*. • Removed *Targeted Devices* from *Resource Utilization*. • Updated the *Lattice Implementation* information in *Design Tool Support*.
Error! Reference source not found.	Updated instances of *Multi-site Perpetual* to *Single Seat Perpetual*.
References	Added CORDIC IP Release Notes (FPGA-RN-02091) document.

Revision 1.5, IP v1.5.0, December 2024

Section	Change Summary
All	• Removed *Core* from the document title. • Added the IP version information on the cover page.
Introduction	Updated Table 1.1. Quick Facts: • Added the *Certus-N2* device family to *Supported FPGA Family*. • Added the *LFD2NX-9*, *LFD2NX-28*, and *LN2-CT-20* devices to *Targeted Devices*. • Updated the Resources and Lattice Implementation information.
IP Generation, Simulation, and Validation	Updated instances of *black box* to *closed-box* in the Generating the IP section.
Ordering Part Number	• Updated instances of *Single Machine* to *Single Seat*. • Added the *CORDIC-AVG-US*, *CORDIC-AVX-UT*, and Certus-N2 OPNs. • Made editorial fixes.
Resource Utilization	Added resource utilizations for the Lattice Radiant software version 2024.2 and made editorial fixes.
References	• Added the *Certus-N2*, *Lattice Solutions IP Cores*, and *Lattice Insights* web pages. • Added the *Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)* document.

Section	Change Summary
	• Made editorial fixes.

Revision 1.4, December 2023

Section	Change Summary
All	• Renamed document *from CORDIC IP Core – Lattice Radiant Software* to *CORDIC IP Core*. • Performed minor formatting and typo edits.
Disclaimers	Updated this section.
Inclusive Language	Added inclusive language boilerplate.
IP Generation, Simulation, and Validation	• Added section 3.1 Licensing the IP. • Updated Table 3.1. Generated File List to include information on the constraint.pdc file. • Added section 3.4 Constraining the IP.
Ordering Part Number	Updated OPNs.
Resource Utilization	Updated LAV-AT-E500-3LFG1156I to LAV-AT-E70-3LFG1156I.
Technical Support Assistance	Added link to the Lattice Answer Database.

Revision 1.3, November 2022

Section	Change Summary
Introduction	Added Lattice Avant and LAV-AT-500E to Table 1.1. Quick Facts.
Functional Description	Signal Description, Attribute Summary, and Configuring the CORDIC IP Core sections are moved to under Section 2.
IP Generation, Simulation, and Validation	• Updated the heading of Section 3 from “IP Generation and Evaluation” to “IP Generation, Simulation, and Validation”. • Removed “Licensing the IP” section. • Updated the heading of Section 3.1 from “Generation and Synthesis” to “Generating the IP”. • Updated the heading of Section 3.3 from “Hardware Evaluation” to “IP Evaluation”. • Updated the Section 3.3 from *The CORDIC IP Core supports Lattice’s IP hardware evaluation capability when used with CrossLink-NX and Certus-NX devices. This makes it possible to create versions of the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default. To change this setting, go to Project > Active Strategy > LSE/Synplify Pro Settings.* to *The IP Core supports Lattice’s IP evaluation capability when used in the supported FPGA family and targeted device. This makes it possible to create versions of the IP core that operates in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. The IP evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default. To change this setting, go to Project > Active Strategy > LSE/Synplify Pro Settings.*
Resource Utilization	• Updated Table A.1. Resource Utilization and Table A.2. Resource Utilization. • Added Table A.3. Resource Utilization, Table A.4. Resource Utilization, and Table A.5. Resource Utilization.
Ordering Part Number	• Added CNX-US, CPNX-US, CTNX-US, and Avant-E part numbers.

Revision 1.2, May 2022

Section	Change Summary
Disclaimers	Updated.
Introduction	Added MachXO5-NX device support to Table 1.1. Quick Facts.
Core Generation, Simulation, and Validation	Updated Figure 4.1. Module/IP Block Wizard, Figure 4.2. Configure User Interface of CORDIC IP Core, and Figure 4.3. Check Generating Result to reflect the most recent IP CORDIC version 1.2.
Ordering Part Number	Added MachXO5-NX OPNs.
Resource Utilization	Updated description and data for Table A.1. Resource Utilization and Table A.2. Resource Utilization to reflect the support for MachXO5-NX family.

Revision 1.1, June 2021

Section	Change Summary
Acronyms in This Document	Added this section.
Introduction	Updated content, including Table 1.1 to add CertusPro-NX support.
Ordering Part Number	Added part number for CertusPro-NX.
References	Updated this section to add CertusPro-NX web page.

Revision 1.0, October 2020

Section	Change Summary
All	Initial release.

www.latticesemi.com