Gamma Corrector IP

IP Version: 1.7.1

User Guide

FPGA-IPUG-02122-1.6

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
CRT	Cathode Ray Tube
DLP	Digital Light Processing
EBR	Embedded Block RAM
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
HDL	Hardware Description Language
I/O	Input/Output
IP	Intellectual Property
IPXACT	IP eXtensible Markup Language
LCOS	Liquid Crystal on Silicon
LUT	Look-Up Table
RAM	Random Access Memory
RTL	Register Transfer Level

1. Introduction

Gamma correction is a method used to adjust images or video frames to counteract the non-linear behavior of display systems, such as cathode ray tube (CRT) displays. CRT displays have a characteristic where the intensity produced is not directly proportional to the input voltage. Instead, the intensity is related to the signal amplitude raised to a power, known as *gamma*. Typically, gamma is greater than 1, which means that these displays have lower gain at low intensities and higher gain at higher intensities. Lattice Gamma Corrector IP core compensates for this by multiplying the input signal with the inverse of the display transfer function, resulting in a linear intensity response relative to the original input signal.

The non-CRT displays such as Plasma, LCOS (Liquid Crystal on Silicon), and DLP (Digital Light Processing) have different transfer characteristics. Several gamma correction methods and values are used in television and display systems. Sometimes, the display itself can have linear characteristics, but a gamma transformation (usually called *degamma*) may be required because of an earlier gamma correction made to the incoming signal.

The Lattice Gamma Corrector IP core is a widely parameterizable and multi-color plane gamma correction system. It can support almost any custom gamma correction requirement.

1.1. Quick Facts

Table 1.1 presents a summary of the Gamma Corrector IP Core.

Table 1.1. Quick Facts

IP Requirements	Supported Devices	CrossLink™-NX, Certus™-NX, CertusPro™-NX, MachXO5™-NX, Lattice Avant™, Certus-N2
	IP Changes[1]	For a list of changes to the IP, refer to the Gamma Corrector IP Release Notes (FPGA-RN-02052).
Resource Utilization	Resources	See the Resource Utilization section
Design Tool Support	Lattice Implementation	IP Core v1.7.1 – 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.

Note:

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 the Gamma Corrector IP Core include:

- Gamma correction characteristics can be specified as an equation using a gamma value or by the actual mapping values of the look-up table.
- Configurable number of color planes- 1 to 3.
- Configurable number of bits per color plane – 4 to 12.
- Gamma correction look-up table can be run-time programmable.
- Gamma correction enable/disable control.
- Option for sequential or parallel architecture for area or throughput trade-off.
- Optimized gamma look-up table memory when same gamma correction is used for multiple color planes.
- Registered input option for input set-up time improvement.

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:

- _*i* are input signals
- _*o* are output signals

1.3.3. Host

The logic unit inside the FPGA interacts with the Gamma Corrector IP Core.

1.3.4. Attribute

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

2. Functional Description

2.1. Overview

Gamma corrector is essentially a look-up-table (referred as g*amma LUT* in this document) that maps each input value to a corresponding output value having the same width. Some applications require simultaneous gamma correction of all the color components, for example, R, G, and B (Red, Green and Blue) require simultaneous correction. There may also be a need to reduce the memory utilization by performing the conversions sequentially for each color plane. Gamma correction for multiple color planes can also be used for gamma correction of multiple monochrome channels either parallel or sequentially. It is also useful to have the ability to dynamically load the gamma LUT values into the gamma corrector.

Figure 2.1 shows the interface diagram for the Gamma Corrector IP. The diagram shows all the available ports. Note that not all the I/O ports are available for a chosen configuration.

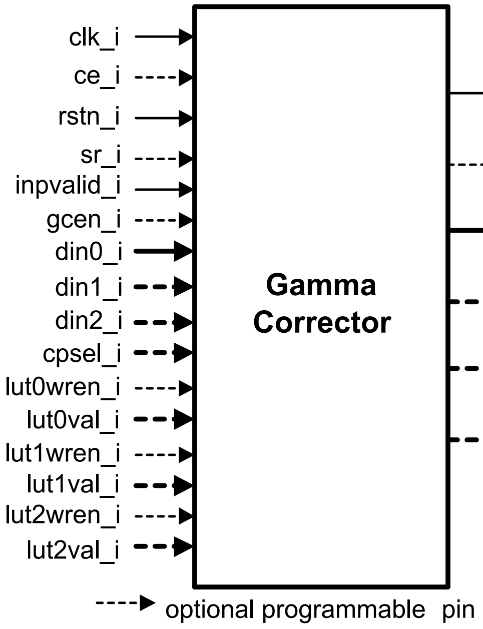


Figure 2.1. Top-level Interface Diagram for the Gamma Corrector IP Core

2.1.1. Gamma Correction Equation

Gamma correction is a non-linear exponent mapping applied to the normalized value of the input to result in a normalized corrected output. Gamma correction is defined by:

$$O_N = I_N^{\left(\frac{1}{\gamma}\right)} \qquad (1)$$

where O_N is the normalized output and I_N is the normalized input, both normalized to the range [0,1] and γ is the gamma for the display that is being corrected.

The actual pixel mapping depends on the bit width of the pixel component. For a pixel width of *b* bits, we have:

$$O = \left(\frac{I_N}{2^b - 1}\right)^{1/\gamma} \left(2^b - 1\right) \qquad (2)$$

Here O is the actual output pixel value represented as a *b*-bit binary number.

The original response of a CRT with gamma=2.2, gamma correction and the response of the CRT display for the gamma corrected input are shown in Figure 2.2.

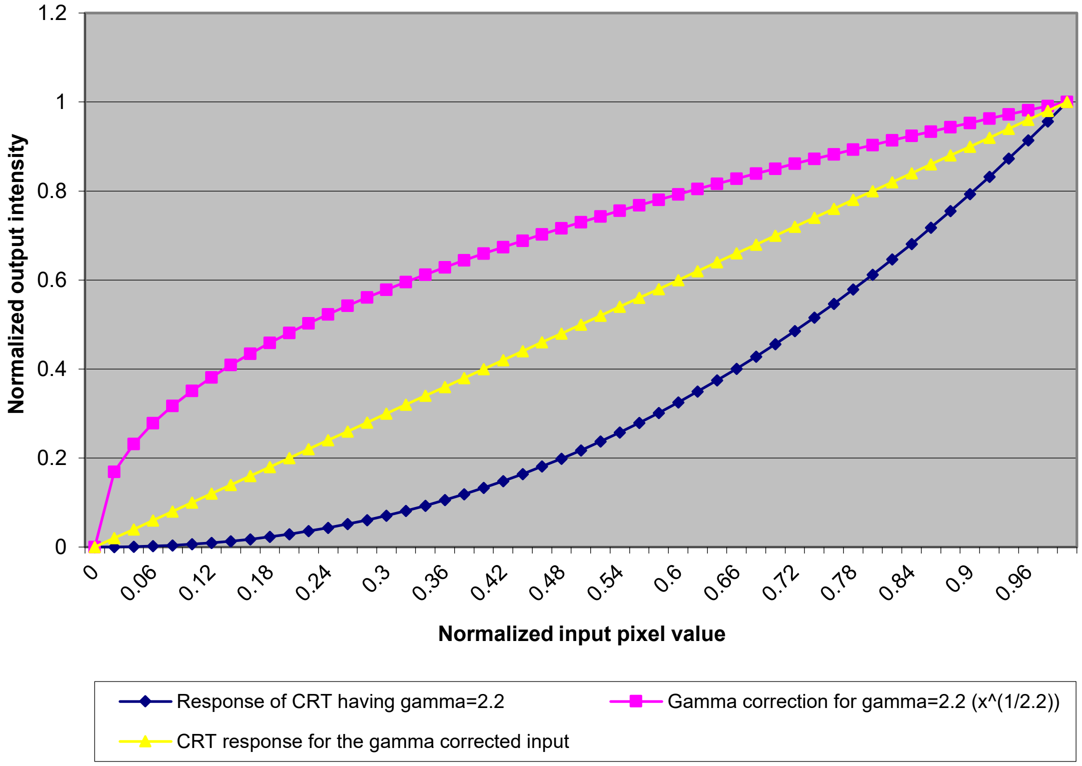


Figure 2.2. Original, Correction and After Correction Response Curves

2.1.2. Filling the Gamma LUT

Based on the above equation, the size of the gamma LUT is 2^b bits depth and *b* bits wide. The gamma LUT is, in most cases, pre-filled with the gamma correction value for each pixel value from 0 to 2^b-1. The gamma correction function can be specified by the gamma value used in Equation (1) or the actual gamma mapping values for each input pixel value. This IP also supports reloading of the gamma LUT through input ports. It takes as many cycles as the number of locations in the gamma LUT to program it completely. While the LUTs are being programmed, gamma correction cannot take place. The IP core can have one or more gamma LUTs when used for mapping different color plane data or channels. If there are multiple gamma LUTs in the IP, for example, one for each of the 3 color planes, a selection input (cpsel_i) determines which color plane or channel is being processed at any time. The Gamma Corrector IP optimizes the number of physical gamma LUTs required depending on the value of the gamma for each color plane and the number of simultaneous I/O (inputs/outputs) that you select.

2.1.3. Multi-Color Plane/Channel Mapping

This IP supports gamma correction of multiple color planes or channels and they are done either sequentially or in parallel. In sequential correction, only one color component data is mapped during a clock cycle. For example, if there are three color planes, the Gamma Corrector requires three clocks to process a color pixel. The input, cpsel_i, is used to identify which color plane LUT needs to be used for the current input. The same cpsel_i input can be used to time multiplex multiple channels (multiple displays) having different gamma correction requirements. In parallel correction, gamma correction of more than one color plane or channel is done simultaneously.

2.1.4. Dynamically Loadable Gamma LUTs

The gamma LUTs can be dynamically loaded or re-programmed after the IP is generated. During LUT programming, each of the possible input values is applied at the input port dinx_i and the corresponding correction value at the input lutxval_i while keeping the corresponding LUT write enable (lutxwren_i) signal high (x in the port names stands for 0,1 or 2). If the core is configured for the sequential mode, each of the LUTs corresponding to the color planes or channels is programmed sequentially. The value at the input port cpsel_i is used to identify the LUT that is programmed.

2.1.5. Handshake I/O ports

The input gcen_i is used to enable gamma correction. If this signal is low, the gamma correction is bypassed and the input value appears at the output after the usual latency for that configuration. The vector port cpsel_i identifies the color plane or channel number that is being processed or programmed. The output cpout_o gives out the color plane or channel number information for the current output(s) appearing at doutx_o (x stands for 0, 1, or 2) ports.

2.2. Signal Description

Table 2.1 lists the input and output signals for Gamma Corrector IP Core.

Table 2.1. Gamma Corrector IP Core Signal Description

Port	Bits	I/O	Description
All configurations			
clk_i	1	I	System clock (reference clock for input and output data).
rstn_i	1	I	System wide asynchronous active-low reset signal.
inpvalid_i	1	I	Input data valid. Indicates valid data is present on din0_i (also on din1_i and din2_i when present).
din0_i	4–12	I	Input Data. When the *sequential* architecture is selected, this port is used to give input data for all the color planes in sequence. When the *parallel* architecture is selected, this port is used to give input data for the first color plane.
dout0_o	4–12	O	Output Data. When the *sequential* architecture is selected, this port is used to give the output data for all the color planes in sequence. When the *parallel* architecture is selected, this port is used to give output data for the first color plane.
outvalid_o	1	O	Output data valid. Indicates valid data is present on dout0_o (also on dout1 _o and dout2_o when present). This valid data may correspond to the gamma corrected output or the bypassed input data, depending on the state of the gcvalid_o signal.
When *Add bypass function* option is selected			
gcen_i	1	I	Gamma correction enable. This signal is valid only when inpvalid_i is asserted high. If gcen _i is high, gamma correction is performed for that input, otherwise no gamma correction is performed.
gcvalid_o	1	O	Gamma corrected output. This signal is valid only when outvalid _o is high. If gcvalid _o is high, the output is a gamma corrected value, otherwise the output is a bypassed value, same as input. The output gcvalid_o is a shifted version of the input signal gcen_i, the shift being equal to the output latency.
When *parallel* architecture is selected and Number of *color planes* is more than one			
din1_i	4–12	I	Input data for the second color plane.
dout1_o	4–12	O	Output data for the second color plane.
When *parallel* architecture is selected and Number of *color planes* is more than two			
din2_i	4–12	I	Input data for the third color plane.
dout2_o	4–12	O	Output data for the third color plane.
When first color plane gamma function is selected as *programmable*			
lut0val_i	4–12	I	Gamma value is programmed through lut0val_i port when: • lut0wren_i is asserted high • Make all color plane same is unchecked • *Programmable Gamma LUT* is Checked on either Second Color plane or Third Color plane when *Number of color planes* is 2 or 3 and *Architecture* is

Port	Bits	I/O	Description
			Sequential • *Architecture* is Sequential and any one of the planes *Programmable Gamma LUT* is Checked • *Architecture* is Parallel and first color plane *Programmable Gamma LUT* is Checked The value from lut0val _i port is written to the gamma LUT at the address provided at the din0_i input port.
lut0wren_i	1	I	Write enable for writing the gamma value for the first color plane when: • *Make all color plane same* is unchecked • *Programmable Gamma LUT* is Checked on either Second Color plane or Third Color plane when Number of color planes is 2 or 3 and Architecture is Sequential. • *Architecture* is Sequential and any one of the planes Programmable Gamma LUT is Checked. • *Architecture* is Parallel and first color plane Programmable Gamma LUT is Checked.
When second color plane gamma function is selected as *programmable, parallel* architecture is selected and Number of *color planes* is more than 1			
lut1val_i	4–12	I	Gamma value is programmed through lut1val port when lut1wren_i is asserted high. The value from lut1val _i port is written to the gamma LUT at the address provided at the din1_i input port.
lut1wren_i	1	I	Write enable for writing the gamma value for the second color plane.
When third color plane gamma function is selected as *programmable, parallel* architecture is selected and Number of *color planes* is more than 2			
lut2val_i	4–12	I	Gamma value is programmed through lut2val_i port when lut2wren_i is asserted high. The value from lut2val _i port is written to the gamma LUT at the address provided at the din2_i input port.
lut2wren_i	1	I	Write enable for writing the gamma value for the second color plane.
When *sequential* architecture is selected			
cpsel_i	1–2	I	Input color plane select. The inputs at din0_i, lut0val_i,lut0wren_i,din1_i, lut1val_i,lut1wren_i, din2_i, lut2val_i,lut2wren_i apply to the color plane provided at cpsel_i.
cpout_o	1–2	O	Output color plane select. The output at dout0_o,dout1_o,dout2_o corresponds to the color plane provided at cpout_o.
Optional I/O			
ce_i	1	I	Clock Enable. While this is low, the core will ignore all other synchronous inputs and maintain its current state. This optional signal should be selected only when required as it leads to increased resource utilization.
sr_i	1	I	Synchronous Reset. This signal must be asserted for at least one clock period duration in order to re-initialize the core. After synchronous reset, all the internal registers are cleared and the outvalid_o goes low. This optional signal should be selected only when required as it leads to increased resource utilization.

2.3. Attributes Summary

The configurable attributes of the Gamma Corrector IP Core 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
Configuration			
Number of color planes	1, 2, 3	3	—
Make all color plane same	Checked, Unchecked	Checked	Editable when Number of color planes not equal to 1.
Architecture	Sequential, Parallel	Sequential	Editable when Number of color planes not equal to 1
First Color Plane			
Data width	4–12	8	—
Gamma function input method	Gamma Value, Through LUT file	Gamma Value	Selectable values depends on *Programmable Gamma LUT*
Gamma value	0.1–10.0	2.2	Enabled only when *Gamma function input method* is *Gamma value.*
Programmable Gamma LUT	Checked, Unchecked	Checked	—
Gamma function through LUT file	NA	NA	Enabled only when *Gamma function input method* is *Through LUT file.* The *.mem file used for all color planes should be the same when *Make all color planes same* = Checked and *Gamma function input method* = ThroughLUT file.
Second Color Plane			
Data width	4–12	8	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 1
Gamma function input method	Gamma Value, Through LUT file	Gamma Value	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 1. Selectable values depends on *Programmable Gamma LUT*
Gamma value	0.1–10.0	2.2	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 1
Programmable Gamma LUT	Checked, Unchecked	Checked	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 1
Gamma function through LUT file	NA	NA	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 1. Enabled only when *Gamma function input method* is *Through LUT file.* The *.mem file used for all color planes should be the same when *Make all color planes same* = Checked and *Gamma function input method* = ThroughLUT file.

Third Color Plane			
Data width	4–12	8	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 2
Gamma function input method	Gamma Value, Through LUT file	Gamma Value	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 2. Selectable values depends on *Programmable Gamma LUT*
Gamma value	0.1–10.0	2.2	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 2
Programmable Gamma LUT	Checked, Unchecked	Checked	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 2
Gamma function through LUT file	NA	NA	Editable when *Make all color plane same* is disabled and when *Number of color planes* is greater than 2. Enabled only when *Gamma function input method* is *Through LUT file.* The *.mem file used for all color planes should be the same when *Make all color planes same* = Checked and *Gamma function input method* = ThroughLUT file.
Implementation			
Add Bypass Function	Checked, Unchecked	Checked	—
Registered Input	Checked, Unchecked	Checked	—
Memory Type	EBR, Distributed	EBR	—
Optional Input and Output Ports			
ce	Checked, Unchecked	Unchecked	—
sr	Checked, Unchecked	Unchecked	—
Output Latency	3–6	6	Not editable. The value depends on *Number of color planes, Architecture, Add bypass function, and Registered input.*

Table 2.3. Attributes Descriptions

Attribute	Description
Configuration	
Number of color planes	Number of color planes for gamma correction
Make all color plane same	Selecting this option will make all the other color plane parameters the same as those of the first color plane.
Architecture	Selects between parallel and sequential implementation architectures
First Color plane	
Data width	The bit width for the color plane
Gamma function input method	The method for specifying the gamma function. The function can be specified by the value of the gamma in the gamma correction equation or by the actual gamma mapping values for the all the pixel values in the input range.
Gamma value	This gamma value is used to create the gamma LUT using Equation (1). This parameter is available when Gamma function input method is selected as *Gamma value*
Programmable Gamma LUT	This parameter is used to indicate if the gamma LUT is also programmable through the input port.
Gamma function through LUT file	This browse button is enabled when Gamma function input method is set to *Through LUT file.* The gamma LUT values will be read from the text file specified. The *.mem file used for all color planes should be the same when *Make all color planes same* = Checked and *Gamma function input method* = ThroughLUT file.
Second Color plane	
Data width	The bit width for the color plane
Gamma function input method	The method for specifying the gamma function. The function can be specified by the value of the gamma in the gamma correction equation or by the actual gamma mapping values for the all the pixel values in the input range.
Gamma value	This gamma value is used to create the gamma LUT using Equation (1). This parameter is available when Gamma function input method is selected as *Gamma value*
Programmable Gamma LUT	This parameter is used to indicate if the gamma LUT is also programmable through the input port.
Gamma function through LUT file	This browse button is enabled when Gamma function input method is set to *Through LUT file.* The gamma LUT values will be read from the text file specified. The *.mem file used for all color planes should be the same when *Make all color planes same* = Checked and *Gamma function input method* = ThroughLUT file.
Third Color plane	
Data width	The bit width for the color plane
Gamma function input method	The method for specifying the gamma function. The function can be specified by the value of the gamma in the gamma correction equation or by the actual gamma mapping values for the all the pixel values in the input range.
Gamma value	This gamma value is used to create the gamma LUT using Equation (1). This parameter is available when Gamma function input method is selected as *Gamma value*
Programmable Gamma LUT	This parameter is used to indicate if the gamma LUT is also programmable through the input port.
Gamma function through LUT file	This browse button is enabled when Gamma function input method is set to *Through LUT file.* The gamma LUT values are read from the text file specified. The *.mem file used for all color planes should be the same when *Make all color planes same* = Checked and *Gamma function input method* = ThroughLUT file.
Implementation	
Add Bypass Function	Selecting this option will add the dynamic gamma correction bypass functionality. Input port gcen_i and output port gcvalid_i are added to the Gamma Corrector IP.
Registered Input	The inputs are registered if this option is selected. The core inputs' set-up times will improve by registering the inputs. This option is useful when the input data is provided on the device pins.
Memory Type	This parameter influences the type of memory used to implement the gamma LUT. If EBR option is selected then the device's EBR (Embedded Block RAM) resources are used for the gamma LUT if the data width is greater than 4. If Distributed option is selected, distributed memory (realized using FPGA's LUTs) is used for the gamma LUT if data width is less than 9.

Attribute	Description
Optional Input and output ports	
ce	Optional clock enable input port ce_i is added to the IP core if this option is checked.
sr	Optional clock enable input port sr_i is added to the IP core if this option is checked.
Output Latency	This static display shows the output latency for the selected core configuration.

2.4. Interfacing with the Gamma Corrector

2.4.1. Parallel and Sequential Architectures

The Gamma Corrector IP offers the choice of two different architectures: parallel and sequential. In the parallel architecture, all the color plane data are applied at the same time. The output data for all the color planes are also available at the same time after a latency of a few clock cycles. In the sequential architecture, the input data for the color planes is applied in sequence, one after the other, using the same input port din0_i. The output data for the color planes is given out sequentially using the same output port dout0_o after a latency of a few clock cycles.

When sequential architecture is selected, the input port din0_i and output port dout0_o are shared between all the color planes. If the data widths are not the same for all color planes, then the highest data width must correspond to the first color plane. When the data width is less than the input port (din0_i and lut0val_i) size, the data must be left aligned and the unused LSBs must be driven with zeros. Similarly, the output data is left aligned with unused LSBs driven to zero, when the data width is less than the output port (dout0_o) size. The color plane or channel number provided on cpsel_i can be in order, allowing the processing of multi-rate channels.

2.4.2. Valid Output

The data output of the Gamma Corrector IP is valid after the output latency for the selected configuration and is indicated by outvalid_i going high. Output latency for Gamma Corrector IP, defined as the number of clock cycles between the sampling of the input data and the availability of the gamma corrected data at the output port, is from 3 to 6 clock cycles depending on the parameters selected.

2.5. Timing Diagrams

2.5.1. Parallel Architecture Timing

Figure 2.3 shows the timing diagram for the parallel architecture. The input data for all the color planes are applied simultaneously on the input ports din0_i, din1_i, and din2_i.

The signal inpvalid_i is asserted to indicate a valid input data present on the input ports. After a latency of a few cycles, the output data for all the color planes appear on the output ports dout0_o, dout1_o, and dout2_o. The signal outvalid_o is asserted to indicate a valid output data present on the output ports. If signal gcen_i is asserted then the data from the input ports is gamma corrected and given at the output. A gamma corrected output at the output ports is indicated by a high gcvalid_o signal. If the signal gcen_i is not asserted, the data from the input ports is passed on directly to the output ports. This is indicated by a low gcvalid_o signal at the output.

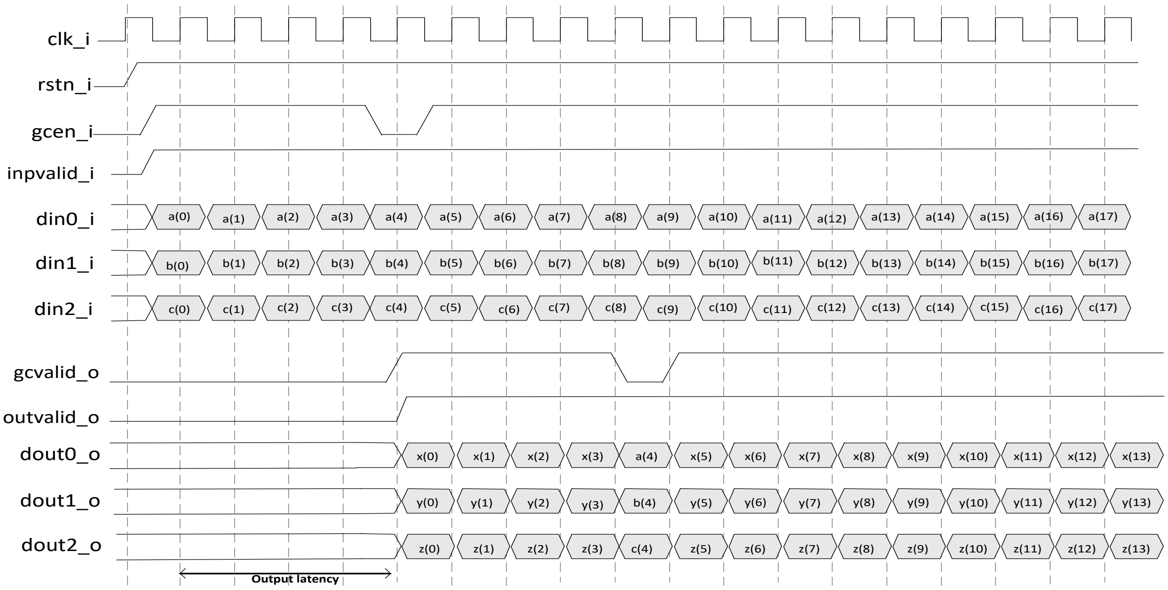

clk_i
rstn_i
gcen_i
inpvalid_i
din0_i
a(0) a(1) a(2) a(3) a(4) a(5) a(6) a(7) a(8) a(9) a(10) a(11) a(12) a(13) a(14) a(15) a(16) a(17)
din1_i
b(0) b(1) b(2) b(3) b(4) b(5) b(6) b(7) b(8) b(9) b(10) b(11) b(12) b(13) b(14) b(15) b(16) b(17)
din2_i
c(0) c(1) c(2) c(3) c(4) c(5) c(6) c(7) c(8) c(9) c(10) c(11) c(12) c(13) c(14) c(15) c(16) c(17)
gcvalid_o
outvalid_o
dout0_o
x(0) x(1) x(2) x(3) a(4) x(5) x(6) x(7) x(8) x(9) x(10) x(11) x(12) x(13)
dout1_o
y(0) y(1) y(2) y(3) b(4) y(5) y(6) y(7) y(8) y(9) y(10) y(11) y(12) y(13)
dout2_o
z(0) z(1) z(2) z(3) c(4) z(5) z(6) z(7) z(8) z(9) z(10) z(11) z(12) z(13)
Output latency

Figure 2.3 Parallel Architecture

2.5.2. Sequential Architecture Timing

Figure 2.4 shows the input and output signal timing for the sequential architecture. The input data for the three color planes are applied in sequence at the input port din0_i. The signal inpvalid_i is asserted to indicate a valid data on din0_i. After a latency of a few cycles the output data for the first color plane appears on the output port dout0. In the following two cycles, the second and third color plane data appear on dout0_o. The signal outvalid_o is asserted to indicate a valid data on dout0_o.

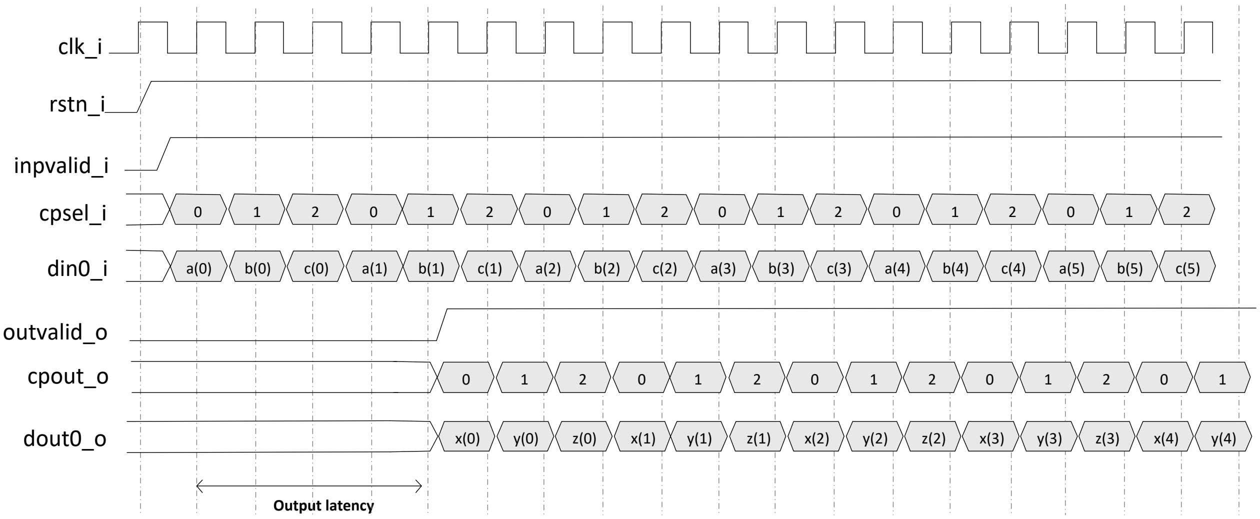

clk_i
rstn_i
inpvalid_i
cpsel_i
0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2
din0_i
a(0) b(0) c(0) a(1) b(1) c(1) a(2) b(2) c(2) a(3) b(3) c(3) a(4) b(4) c(4) a(5) b(5) c(5)
outvalid_o
cpout_o
0 1 2 0 1 2 0 1 2 0 1 2 0 1
dout0_o
x(0) y(0) z(0) x(1) y(1) z(1) x(2) y(2) z(2) x(3) y(3) z(3) x(4) y(4)
Output latency

Figure 2.4 Sequential Architecture

2.5.3. Dynamically Loadable Gamma LUT

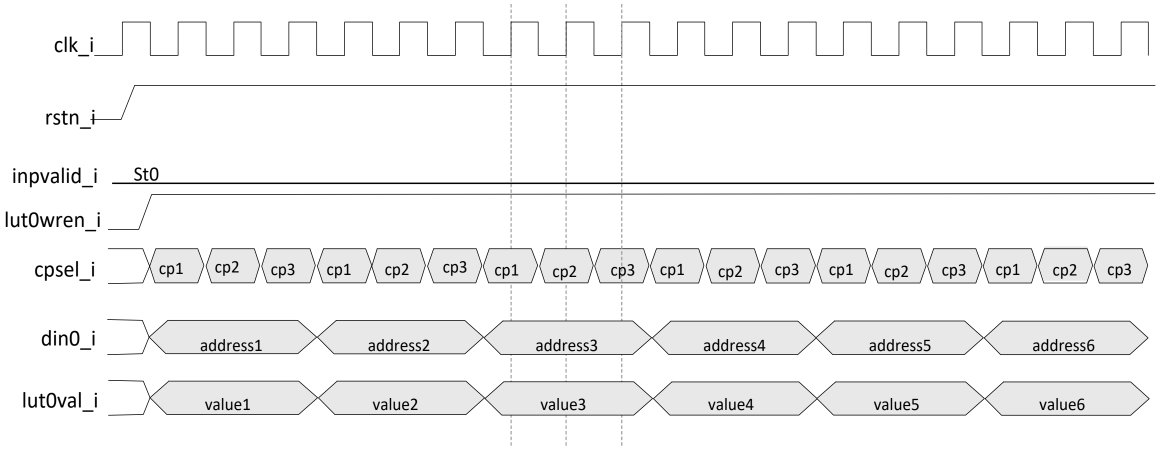


Figure 2.5. Example of Dynamic Gamma Value Programming, Three Color Planes, Sequential Mode

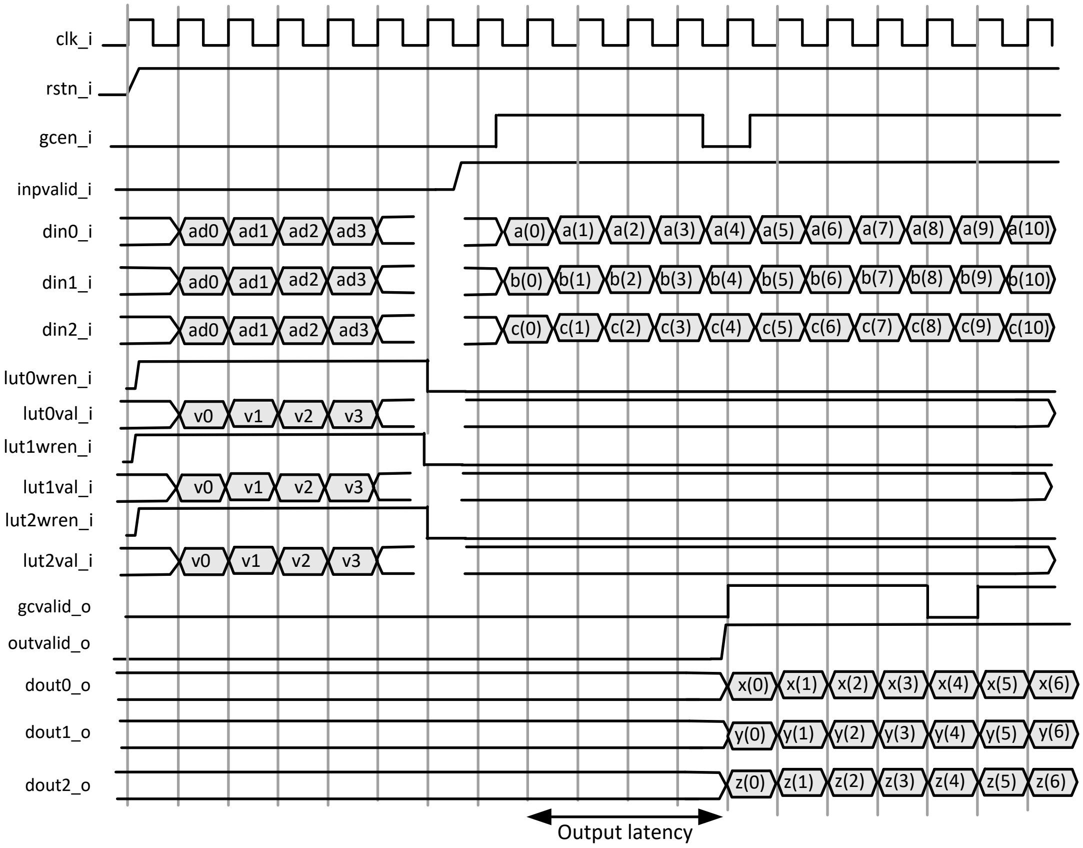


Figure 2.6. Example of Dynamic Gamma Value Programming, Three Color Planes, Parallel Mode

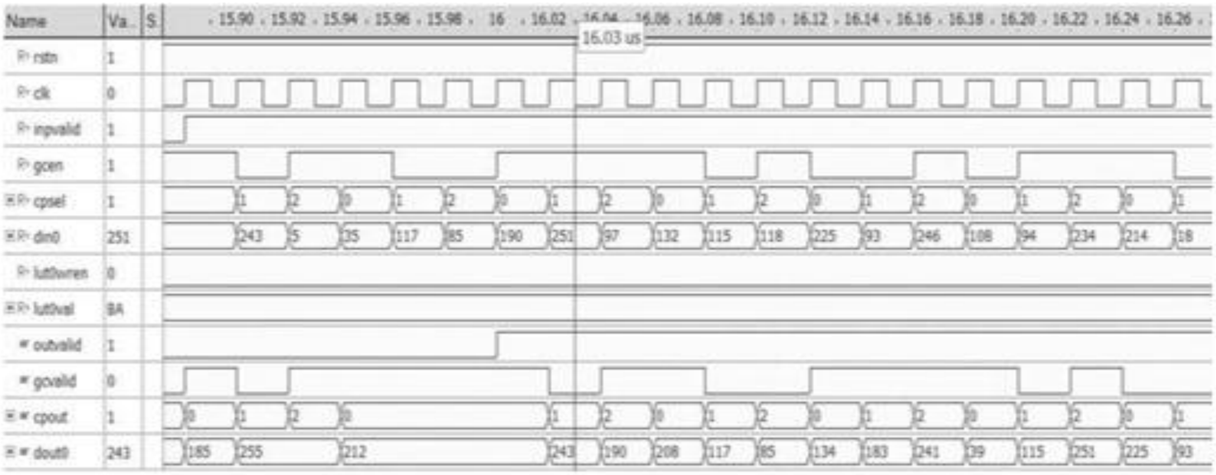


Figure 2.7. Functionality of Programmable Gamma through Look-Up Table (LUT)

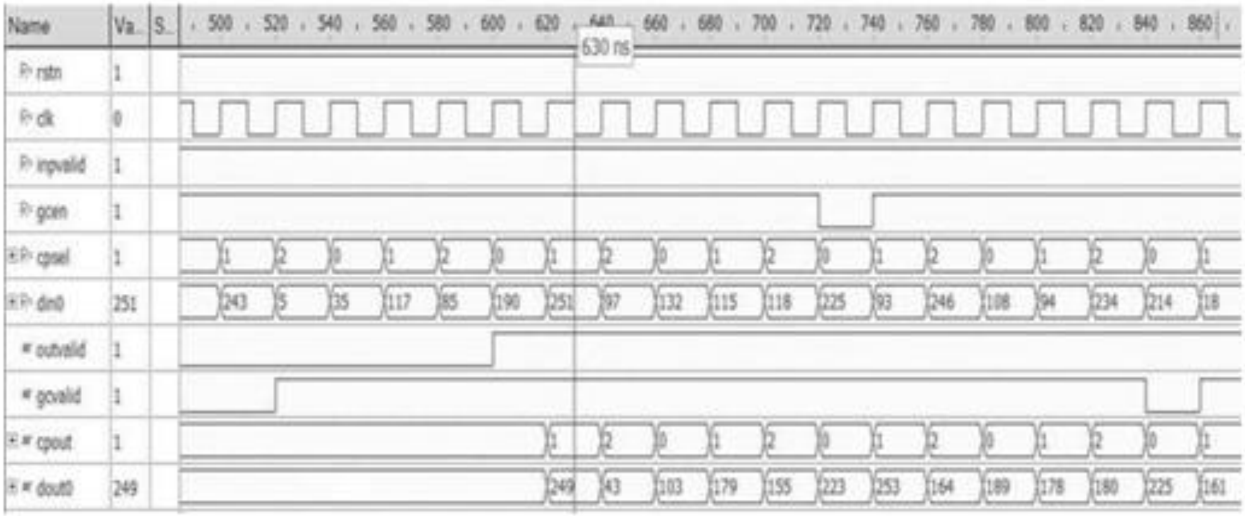


Figure 2.8. Functionality of Constant Gamma Value (= 2.2)

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

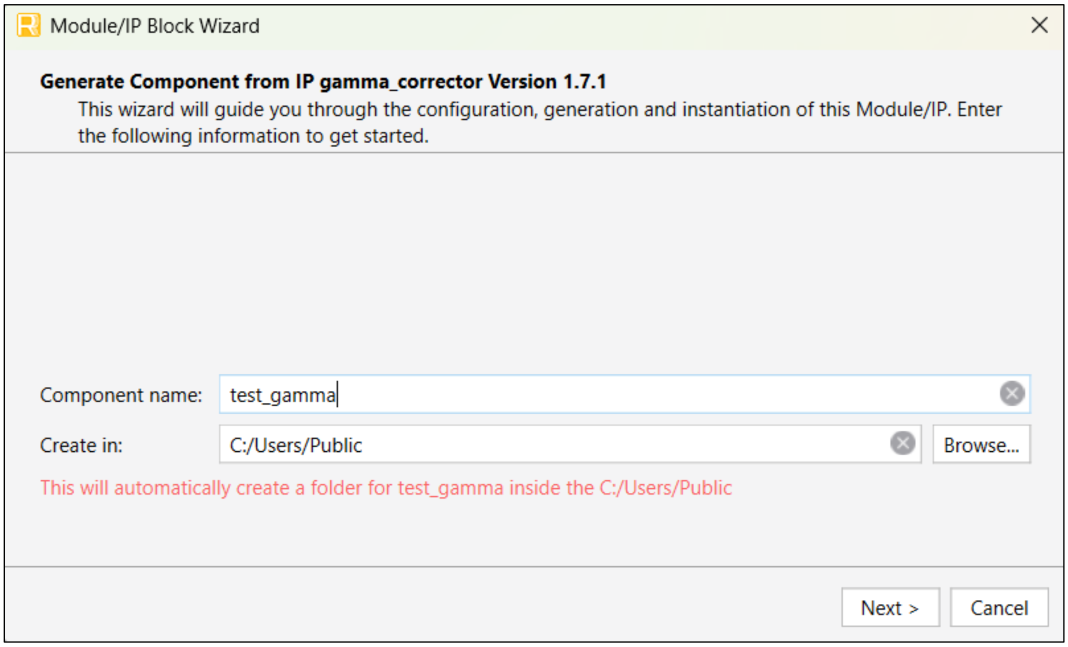
The Gamma Corrector IP is provided at no additional cost with the Lattice Radiant software.

3.2. 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 Gamma Corrector IP Core in Lattice Radiant software is described below.

To generate the Gamma Corrector IP Core:

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



Module/IP Block Wizard

Generate Component from IP gamma_corrector Version 1.7.1

This wizard will guide you through the configuration, generation and instantiation of this Module/IP. Enter the following information to get started.

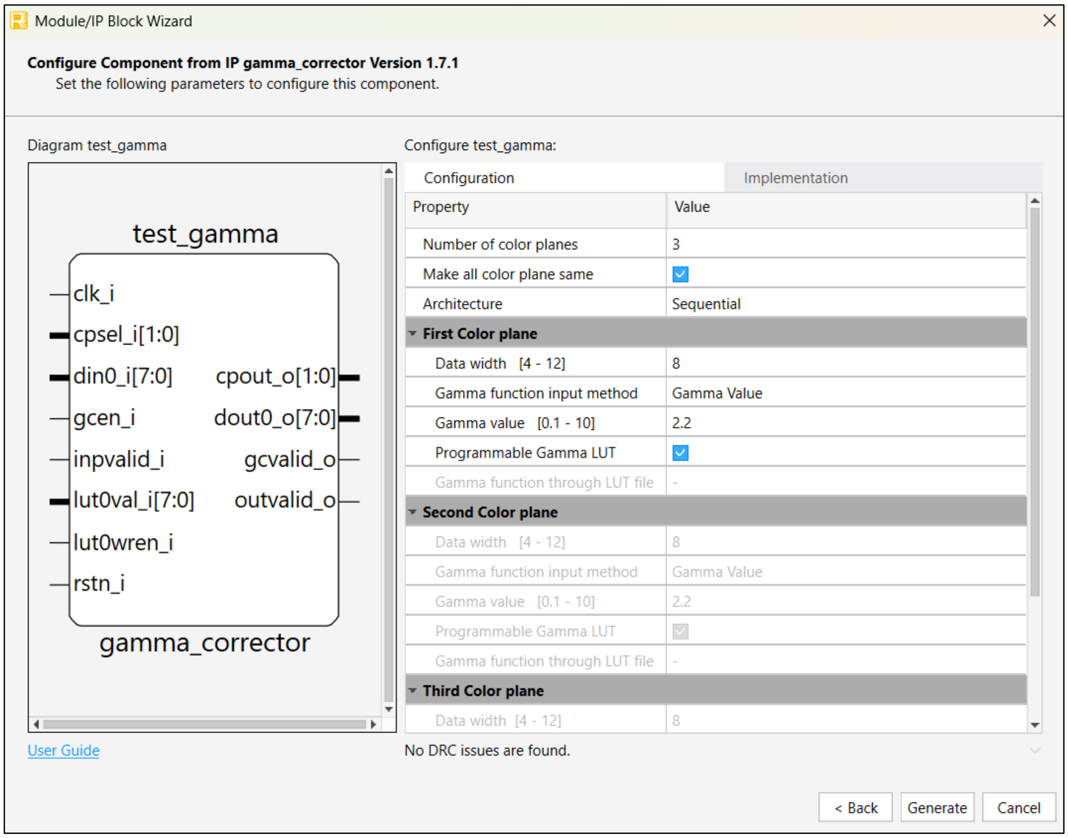
Component name: test_gamma

Create in: C:/Users/Public

This will automatically create a folder for test_gamma inside the C:/Users/Public

Figure 3.1. Module/IP Block Wizard

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



Configure Component from IP gamma_corrector Version 1.7.1
Set the following parameters to configure this component.

Diagram test_gamma

Configure test_gamma:

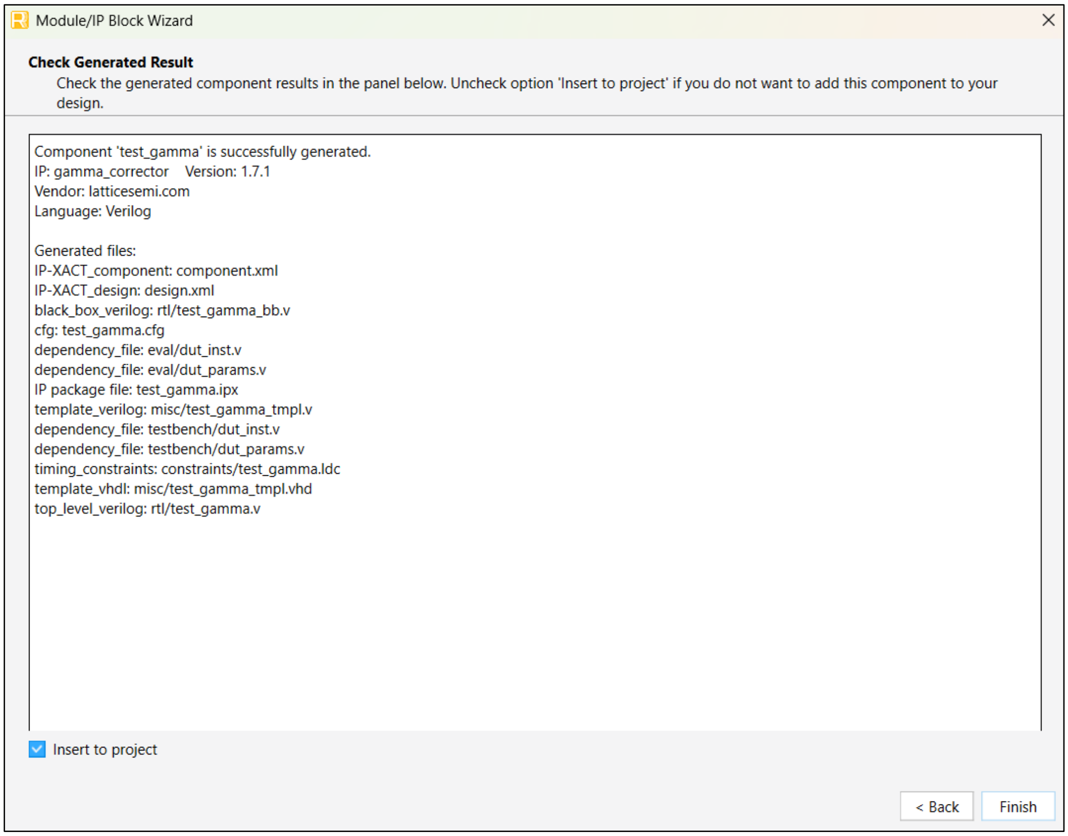
Configuration | Implementation

Property	Value
Number of color planes	3
Make all color plane same	☑
Architecture	Sequential
First Color plane	
Data width [4 - 12]	8
Gamma function input method	Gamma Value
Gamma value [0.1 - 10]	2.2
Programmable Gamma LUT	☑
Gamma function through LUT file	-
Second Color plane	
Data width [4 - 12]	8
Gamma function input method	Gamma Value
Gamma value [0.1 - 10]	2.2
Programmable Gamma LUT	☑
Gamma function through LUT file	-
Third Color plane	
Data width [4 - 12]	8

No DRC issues are found.

Figure 3.2. Configure User Interface of Gamma Corrector IP Core

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



Check Generated Result

Check the generated component results in the panel below. Uncheck option 'Insert to project' if you do not want to add this component to your design.

```
Component 'test_gamma' is successfully generated.
IP: gamma_corrector    Version: 1.7.1
Vendor: latticesemi.com
Language: Verilog

Generated files:
IP-XACT_component: component.xml
IP-XACT_design: design.xml
black_box_verilog: rtl/test_gamma_bb.v
cfg: test_gamma.cfg
dependency_file: eval/dut_inst.v
dependency_file: eval/dut_params.v
IP package file: test_gamma.ipx
template_verilog: misc/test_gamma_tmpl.v
dependency_file: testbench/dut_inst.v
dependency_file: testbench/dut_params.v
timing_constraints: constraints/test_gamma.ldc
template_vhdl: misc/test_gamma_tmpl.vhd
top_level_verilog: rtl/test_gamma.v
```

☑ Insert to project

Figure 3.3. Check Generated 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.

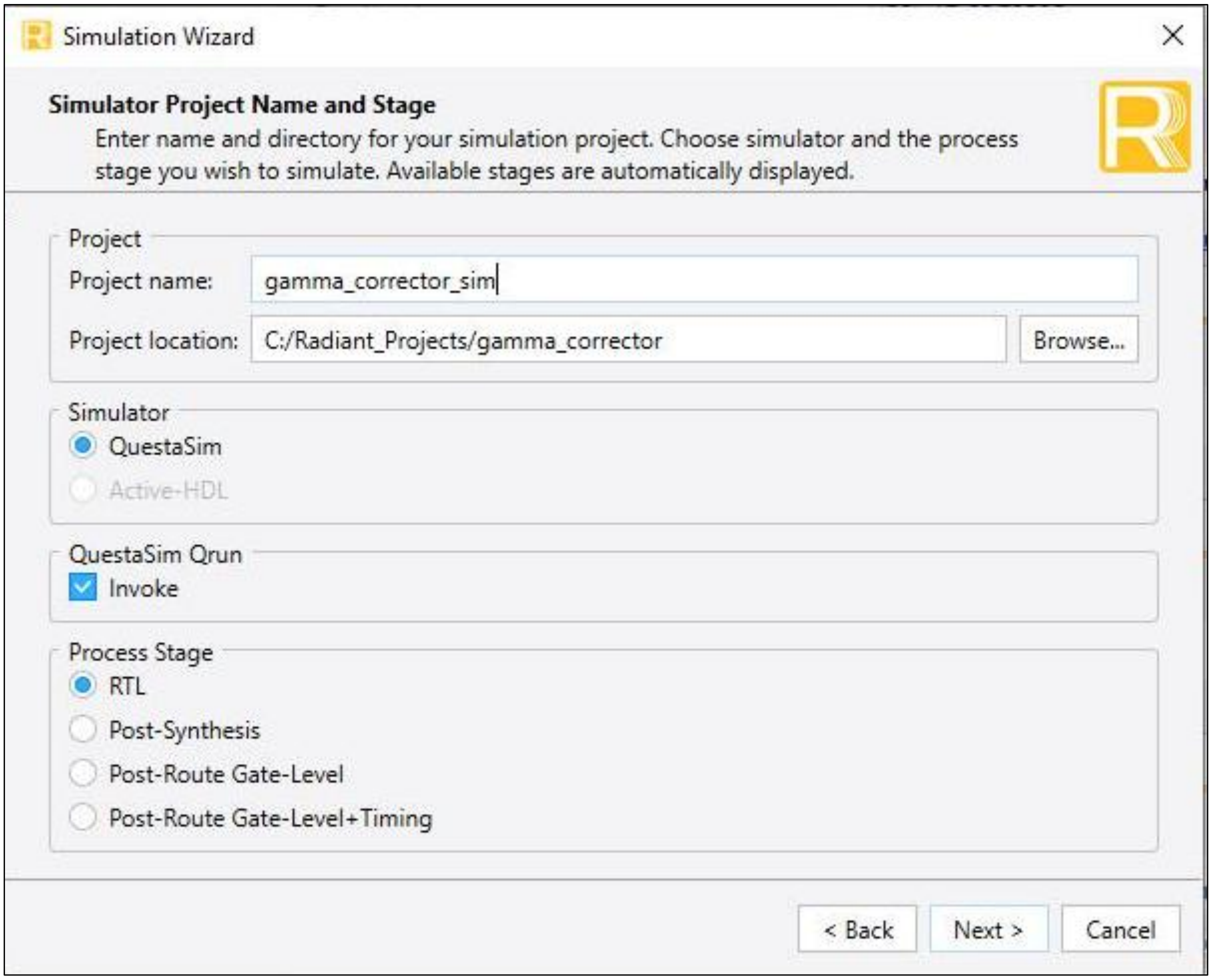
The generated Gamma Corrector IP Core package includes the closed-box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (<Component name>.v) that can be used as an instantiation template for the IP core is also provided. 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 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 the Constraining the IP section on how to use this file.

3.3. Running Functional Simulation

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



Simulation Wizard

Simulator Project Name and Stage

Enter name and directory for your simulation project. Choose simulator and the process stage you wish to simulate. Available stages are automatically displayed.

Project

Project name: gamma_corrector_sim

Project location: C:/Radiant_Projects/gamma_corrector

Simulator

● QuestaSim

○ Active-HDL

QuestaSim Qrun

☑ Invoke

Process Stage

● RTL

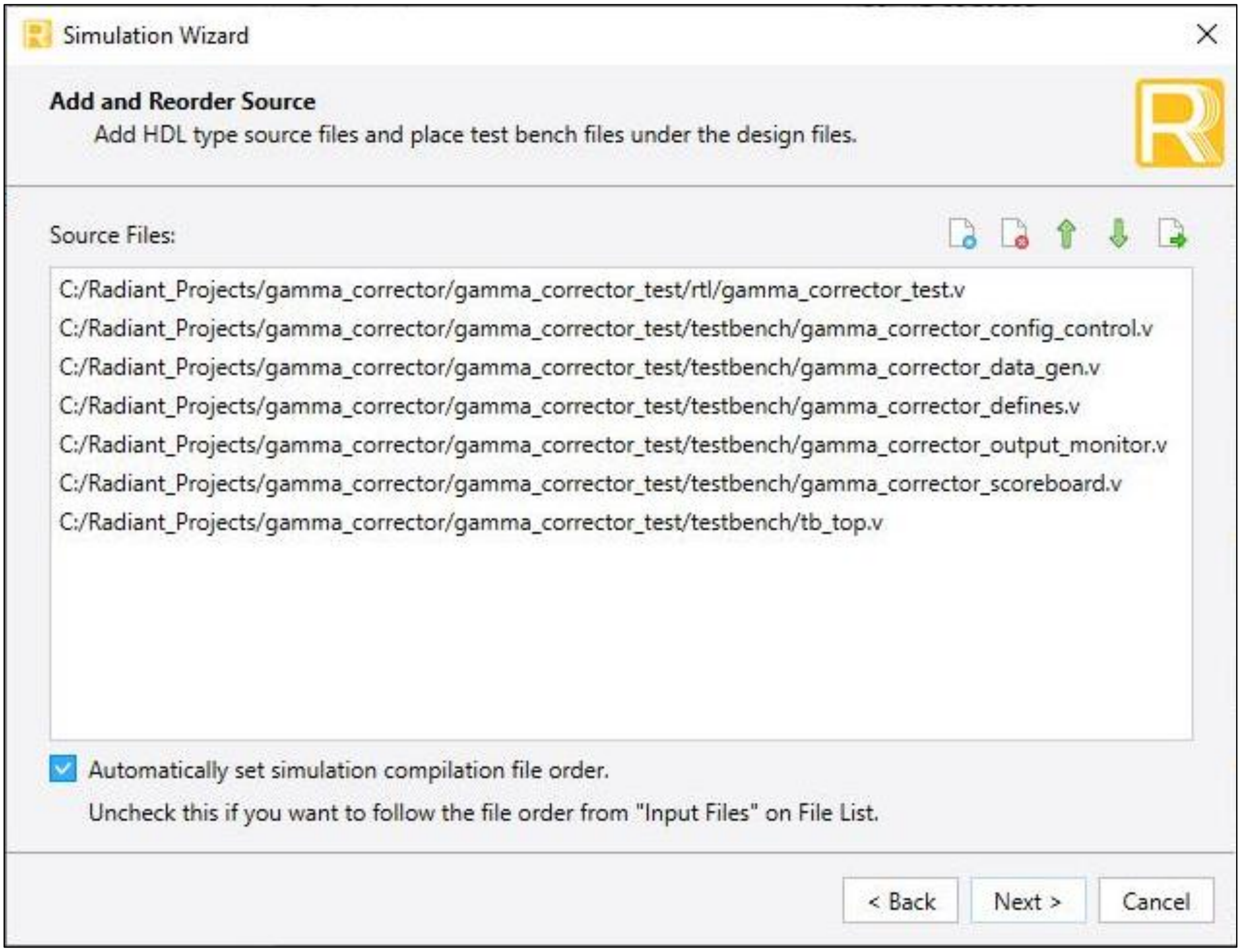
○ Post-Synthesis

○ Post-Route Gate-Level

○ Post-Route Gate-Level+Timing

Figure 3.4. Simulation Wizard

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



Simulation Wizard

Add and Reorder Source

Add HDL type source files and place test bench files under the design files.

Source Files:

```
C:/Radiant_Projects/gamma_corrector/gamma_corrector_test/rtl/gamma_corrector_test.v
C:/Radiant_Projects/gamma_corrector/gamma_corrector_test/testbench/gamma_corrector_config_control.v
C:/Radiant_Projects/gamma_corrector/gamma_corrector_test/testbench/gamma_corrector_data_gen.v
C:/Radiant_Projects/gamma_corrector/gamma_corrector_test/testbench/gamma_corrector_defines.v
C:/Radiant_Projects/gamma_corrector/gamma_corrector_test/testbench/gamma_corrector_output_monitor.v
C:/Radiant_Projects/gamma_corrector/gamma_corrector_test/testbench/gamma_corrector_scoreboard.v
C:/Radiant_Projects/gamma_corrector/gamma_corrector_test/testbench/tb_top.v
```

☑ Automatically set simulation compilation file order.

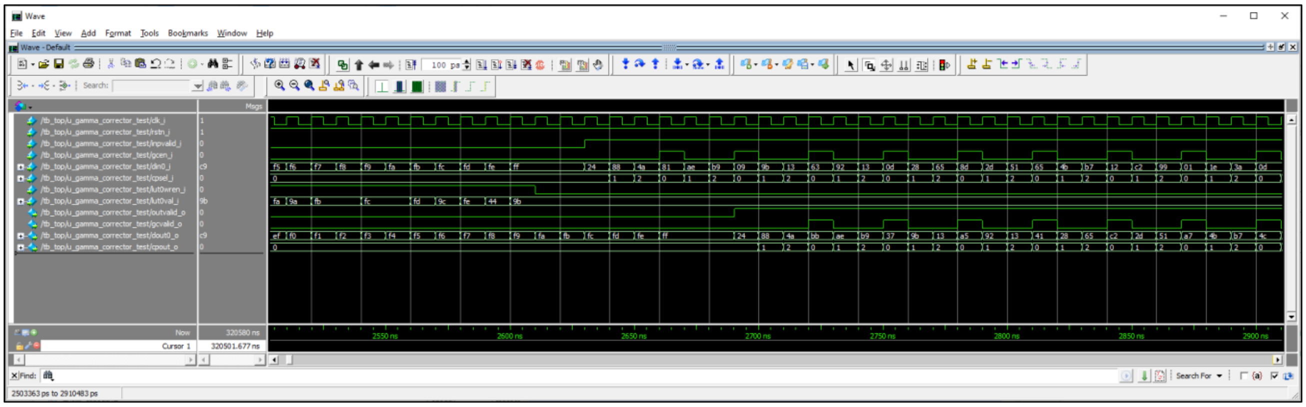
Uncheck this if you want to follow the file order from "Input Files" on File List.

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 Figure 3.6.



Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/tb_top/u_gamma_corrector_test/clk_i 1
/tb_top/u_gamma_corrector_test/rstn_i 1
/tb_top/u_gamma_corrector_test/inpvalid_i 0
/tb_top/u_gamma_corrector_test/gcen_i 0
/tb_top/u_gamma_corrector_test/din0_i c9
/tb_top/u_gamma_corrector_test/cpsel_i 0
/tb_top/u_gamma_corrector_test/lut0wren_i 0
/tb_top/u_gamma_corrector_test/lut0val_i 9b
/tb_top/u_gamma_corrector_test/outvalid_o 0
/tb_top/u_gamma_corrector_test/gcvalid_o 0
/tb_top/u_gamma_corrector_test/dout0_o c9
/tb_top/u_gamma_corrector_test/cpout_o 0
Now 320580 ns
Cursor 1 320501.677 ns
2550 ns
2600 ns
2650 ns
2700 ns
2750 ns
2800 ns
2850 ns
2900 ns
Find:
Search For
2503363 ps to 2910483 ps

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 the Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059) for details on how to constraint your design.

4. Design Considerations

4.1. Limitations

The ce_i and sr_i ports are currently unused by the IP. It is recommended to uncheck the *ce* and *sr* attributes to disable the ports.

Appendix A. Resource Utilization

Table A.1 shows configuration and resource utilization for LIFCL-40-9BG400I using Synplify Pro of Lattice Radiant software 2025.2.

Table A.1. Resource Utilization for LIFCL-40-9BG400I Device

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs	EBRs	Programmable I/O
Configuration 1: Sequential architecture, 3 color planes, same color planes	200	130	35	3	35
Configuration 2: Parallel architecture, 3 color planes, same color planes	200	232	49	3	81
Configuration 3: Sequential architecture, 3 color planes, different color planes	200	130	35	3	35

Note:

1. Fmax is generated when the FPGA design only contains Gamma Corrector 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.2 shows configuration and resource utilization for LIFCL-17-7BG256I using Synplify Pro of Lattice Radiant software 2025.2.

Table A.2. Resource Utilization for LIFCL-17-7BG256I Device

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs	EBRs	Programmable I/O
Configuration 1: Sequential architecture, 3 color planes, same color planes	200	130	35	3	35
Configuration 2: Parallel architecture, 3 color planes, same color planes	200	232	49	3	81
Configuration 3: Sequential architecture, 3 color planes, different color planes	200	130	35	3	35

Note:

1. Fmax is generated when the FPGA design only contains Gamma Corrector 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 configuration and resource utilization for LFD2NX-40-8BG256I using Synplify Pro of Lattice Radiant software 2025.2.

Table A.3. Resource Utilization for LFD2NX-40-8BG256I Device

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs	EBRs	Programmable I/O
Configuration 1: Sequential architecture, 3 color planes, same color planes	200	130	35	3	35
Configuration 2: Parallel architecture, 3 color planes, same color planes	200	232	49	3	81
Configuration 3: Sequential architecture, 3 color planes, different color planes	200	130	35	3	35

Note:

1. Fmax is generated when the FPGA design only contains Gamma Corrector 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 configuration and resource utilization for LFMXO5-25-9BBG400I using Synplify Pro of the 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. Resource Utilization for LFMXO5-25-9BBG400I Device

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs	EBRs
Configuration 1: Sequential architecture, 3 color planes, same color planes	200	130	35	1
Configuration 2: Parallel architecture, 3 color planes, same color planes	200	232	49	3
Configuration 3: Sequential architecture, 3 color planes, different color planes	200	130	35	1

Note:

1. Fmax is generated when the FPGA design only contains Gamma Corrector 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.5 shows configuration and resource utilization for LAV-AT-E70-3LFG1156I using Synplify Pro of the 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. Resource Utilization for LAV-AT-E70-3LFG1156I Device

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs	EBRs
Configuration 1: Sequential architecture, 3 color planes, same color planes	250	130	34	1
Configuration 2: Parallel architecture, 3 color planes, same color planes	207.25	232	48	3
Configuration 3: Sequential architecture, 3 color planes, different color planes	250	130	34	1

Note:

1. Fmax is generated when the FPGA design only contains Gamma Corrector 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 configuration and resource utilization for LN2-CT-20-1ASG410I using Synplify Pro of the 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. Resource Utilization for LN2-CT-20-1ASG410I Device

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs	EBRs
Configuration 1: Sequential architecture, 3 color planes, same color planes	250	130	34	3
Configuration 2: Parallel architecture, 3 color planes, same color planes	210.84	232	48	3
Configuration 3: Sequential architecture, 3 color planes, different color planes	250	130	34	3

Note:

1. Fmax is generated when the FPGA design only contains Gamma Corrector 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.7 shows configuration and resource utilization for LFD2NX-9-9MG121C using Synplify Pro of the 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.7. Resource Utilization for LFD2NX-9-9MG121C Device

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs	EBRs
Configuration 1: Sequential architecture, 3 color planes, same color planes	200	130	35	3
Configuration 2: Parallel architecture, 3 color planes, same color planes	200	232	129	3
Configuration 3: Sequential architecture, 3 color planes, different color planes	200	130	35	3

Note:

1. Fmax is generated when the FPGA design only contains Gamma Corrector IP core and the target frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

References

- Gamma Corrector IP Release Notes (FPGA-RN-02052)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- Certus-NX web page
- Certus-N2 web page
- CertusPro-NX web page
- CrossLink-NX web page
- MachXO5-NX web page
- Avant-E web page
- Avant-G web page
- Avant-X web page
- Gamma Corrector IP Core web page
- Lattice Radiant Software web page
- Lattice Insights 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 1.6, IP v1.7.1, December 2025

Section	Change Summary
All	• Added a note on IP version in Quick Facts and *Revision History* sections. • Performed minor formatting and editorial edits.
Acronyms in This Document	Updated list of acronyms.
Introduction	• Updated Table 1.1. Quick Facts as follows: • Added IP version. • Removed earlier IP versions.
IP Generation, Simulation, and Validation	• Added a note on IP version in GUI in the IP Generation, Simulation, and Validation section. • Updated the Licensing the IP section. • Updated the following figures: • Figure 3.1. Module/IP Block Wizard • Figure 3.2. Configure User Interface of Gamma Corrector IP Core • Figure 3.3. Check Generated Result • Removed the *IP Evaluation* section.
Ordering Part Number	Removed this section.
Design Considerations	Added this section.
Resource Utilization	Updated the following tables: • Table A.2. Resource Utilization for LIFCL-17-7BG256I Device • Table A.4. Resource Utilization for LFMXO5-25-9BBG400I Device • Table A.5. Resource Utilization for LAV-AT-E70-3LFG1156I Device • Table A.6. Resource Utilization for LN2-CT-20-1ASG410I Device
References	Updated references.

Revision 1.5, IP v1.7.0, July 2025

Section	Change Summary
Introduction	• Updated the Introduction section. • Updated Table 1.1. Quick Facts as follows: • Renamed *Supported FPGA Families* to *Supported Devices*. • Removed the *Targeted Devices* row. • Added IP version.
IP Generation, Simulation, and Validation	Updated the following figures: • Figure 3.1. Module/IP Block Wizard • Figure 3.2. Configure User Interface of Gamma Corrector IP Core • Figure 3.3. Check Generating Result • Figure 3.4. Simulation Wizard • Figure 3.5. Adding and Reordering Source
Ordering Part Number	Changed from *Multi-site Perpetual* to *Single Seat Perpetual*.
Resource Utilization	Updated resource utilization for LIFCL-40-9BG400I, LIFCL-17-7BG256I, and LFD2NX-40-8BG256I devices.
References	Updated references.

Revision 1.4, IP v1.6.0, December 2024

Section	Change Summary
All	Renamed document from *Gamma Corrector IP Core* to *Gamma Corrector IP*.
Introduction	Updated Table 1.1. Quick Facts as follows: • Added support for LFD2NX-9, LFD2NX-28, and LN2-CT-20 devices. • Added IP changes. • Updated resources. • Updated IP version.
IP Generation, Simulation, and Validation	Changed *black box* to *closed-box* in the Generating the IP section.
Ordering Part Number	• Added OPN for Certus-N2, Avant-G, and Avant-X devices. • Changed from *Single Machine Annual* to *Single Seat Annual* license.
Resource Utilization	Added resource utilization tables for LN2-CT-20-1ASG410I and LFD2NX-9-9MG121C devices.
References	Added links to the Certus-N2 web page and IP release notes.

Revision 1.3, December 2023

Section	Change Summary
All	• Renamed document from *Gamma Corrector IP Core – Lattice Radiant Software* to *Gamma Corrector IP Core*. • Performed minor formatting and typo edits.
Disclaimers	Updated this section.
Inclusive Language	Added inclusive language boilerplate.
Introduction	Updated Target Devices in Table 1.1 to include the following devices: LIFCL-33, LFCPNX-50, LFMXO5-55T, LFMXO5-100T, UT24C40, UT24CP100, LAV-AT-G70, and LAV-AT-X70, and updated LAV-AT-E500 to LAV-AT-E70.
IP Generation, Simulation, and Validation	• Added section 3.1. Licensing the IP. • Updated Figure 3.1, Figure 3.2, Figure 3.3, Figure 3.4, Figure 3.5, and Figure 3.6. • 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 1.2, November 2022

Section	Change Summary
Introduction	• Added MachXO5™-NX and Lattice Avant families in Table 1.1. Quick Facts. • Add LFMXO5-25 and LAV-AT-500E devices in Table 1.1. Quick Facts.
IP Generation, Simulation, and Validation	• Updated title of Section 3 from *Core Generation, Simulation, and Validation* to *IP Generation, Simulation, and Validation.* • Deleted the *Licensing the IP* section. • Updated Section 3.1 from *Generation and Synthesis* to *Generating the IP.*
Ordering Part Number	Updated OPNs for Gamma Corrector.
Resource Utilization	Added Table A.4. Resource Utilization and Table A.5. Resource Utilization.

Revision 1.1, June 2021

Section	Change Summary
Introduction	Updated Table 1.1 to add CertusPro-NX support.
Ordering Part Number	Added this section.
Appendix A. Resource Utilization	Updated section content.

Section	Change Summary
References	Added CertusPro-NX web page.

Revision 1.0, December 2020

Section	Change Summary
All	Initial release

LATTICE
SEMICONDUCTOR®

www.latticesemi.com