



***Lattice*CORE™**

Gamma Corrector IP Core User's Guide

Chapter 1. Introduction	4
Quick Facts	4
Features	5
Chapter 2. Functional Description	6
Block Diagram	6
Gamma Correction Equation	6
Filling the Gamma LUT	7
Multi-Color Plane/Channel Mapping	7
Dynamically Loadable Gamma LUTs	7
Handshake I/O Ports	8
Signal Descriptions	8
Interfacing with the Gamma Corrector	9
Parallel and Sequential Architectures	9
Valid Output	9
Timing Specifications	9
Parallel Architecture Timing	9
Sequential Architecture Timing	10
Dynamically Loadable Gamma LUT Timing	11
Chapter 3. Parameter Settings	13
Configuration Tab	13
Implementation Tab	14
Configuring the Gamma Corrector IP	15
Chapter 4. IP Core Generation	16
Licensing the IP Core	16
Getting Started	16
Configuring the Gamma Corrector IP Core in IPexpress	16
IPexpress-Created Files and Top-Level Directory Structure	18
Instantiating the Core	20
Running Functional Simulation	20
Synthesizing and Implementing the Core in a Top-Level Design	20
Hardware Evaluation	21
Enabling Hardware Evaluation in Diamond	21
Enabling Hardware Evaluation in ispLEVER	21
Updating/Regenerating the IP Core	21
Regenerating an IP Core in Diamond	21
Regenerating an IP Core in ispLEVER	22
Chapter 5. Support Resources	23
Lattice Technical Support	23
Online Forums	23
Telephone Support Hotline	23
E-mail Support	23
Local Support	23
Internet	23
References	23
Revision History	23
Appendix A. Resource Utilization	24
LatticeECP3 Devices	25
Ordering Part Number	25
LatticeECP2 and LatticeECP2S Devices	25

Ordering Part Number.....	25
LatticeECP2M and LatticeECP2MS Devices	25
Ordering Part Number.....	25
LatticeXP2 Devices	26
Ordering Part Number.....	26

Gamma correction is a type of pre-distortion correction made to images or video frames to offset the non-linear behavior of display systems, such as cathode ray tube (CRT) displays. A characteristic of CRT displays is that the intensity they generate is not a linear function of the input voltage. Instead, the intensity is proportional to a power of the signal amplitude, also referred to as gamma. Gamma is usually greater than 1 and therefore the displays have a lower gain at low intensities and progressively larger gain at higher intensities. The Gamma Corrector IP core multiplies the input signal with the inverse of the display transfer function which results in a linear intensity response with respect to the original input signal.

Several gamma correction methods and values are used in television and display systems. Plasma, LCOS (Liquid Crystal on Silicon) and DLP (Digital Light Processing) displays have transfer characteristics that are different from that of CRT displays. 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 Gamma Corrector IP core is a widely parameterizable, multi-color plane gamma correction system. It can support almost any custom gamma correction requirement.

Quick Facts

Table 1-1 gives quick facts about the Gamma Corrector IP core.

Table 1-1. Gamma Corrector IP Core Quick Facts

		Gamma Corrector IP Core			
		Sequential Architecture 3 Color Planes Same Color Planes	Parallel Architecture 3 Color Planes Same Color Planes	Sequential Architecture 3 Color Planes Different Color Planes	
Core Requirements	FPGA Families Supported	LatticeECP2™ and LatticeECP2S, LatticeECP2M™ and LatticeECP2MS, LatticeECP3™, LatticeXP2™			
	Minimum Device Required	LFE2-6E LFE2-6SE LFE2M20E LFE2M20SE LFE3-17EA LFXP2-5E	LFE2-6E LFE2-6SE LFE2M20E LFE2M20SE LFE3-17EA LFXP2-5E	LFE2-6E LFE2-6SE LFE2M20E LFE2M20SE LFE3-17EA LFXP2-5E	
Resource Utilization	LatticeECP2 & LatticeECP2S	LUTs	29	38	26
		EBRs	3	9	3
		Registers	113	265	104
		Slices	57	133	52
	LatticeECP2M & LatticeECP2MS	LUTs	28	37	25
		EBRs	3	9	3
		Registers	113	265	104
		Slices	57	133	52
	LatticeECP3	LUTs	32	38	29
		EBRs	3	9	3
		Registers	113	265	104
		Slices	57	133	52
	LatticeXP2	LUTs	27	36	24
		EBRs	3	9	3
		Registers	113	265	104
		Slices	57	133	52

© 2011 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 1-1. Gamma Corrector IP Core Quick Facts

		Gamma Corrector IP Core		
		Sequential Architecture 3 Color Planes Same Color Planes	Parallel Architecture 3 Color Planes Same Color Planes	Sequential Architecture 3 Color Planes Different Color Planes
Design Tool Support	Lattice Implementation	Lattice Diamond™ 1.1, ispLEVER® 8.1SP1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2010.03L-SP1 Mentor Graphics Precision RTL 2010a Update2.254		
	Simulation	Aldec Active-HDL 8.2 Edition II Mentor Graphics ModelSim SE 6.6b		

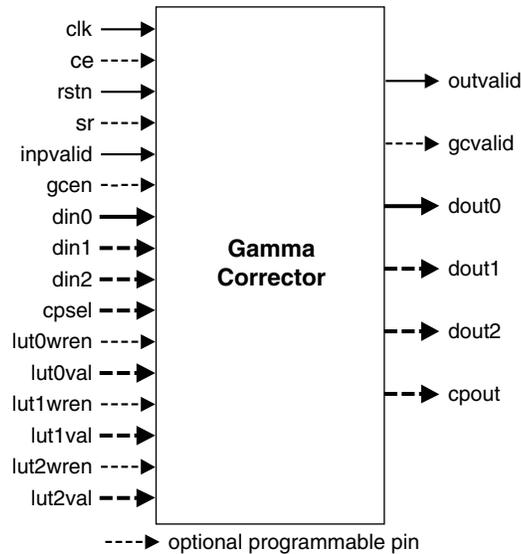
Features

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

Block Diagram

Figure 2-1 shows the interface diagram for the Gamma Corrector IP core. The diagram shows all of the available ports for the IP core. 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



The Gamma Corrector is essentially a look-up-table (referred to as gamma 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). 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 in parallel or sequentially. It is also useful to have the ability to dynamically load the gamma LUT values in to the gamma corrector.

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:

$$O_N = I_N^{1/\gamma} \quad (1)$$

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.

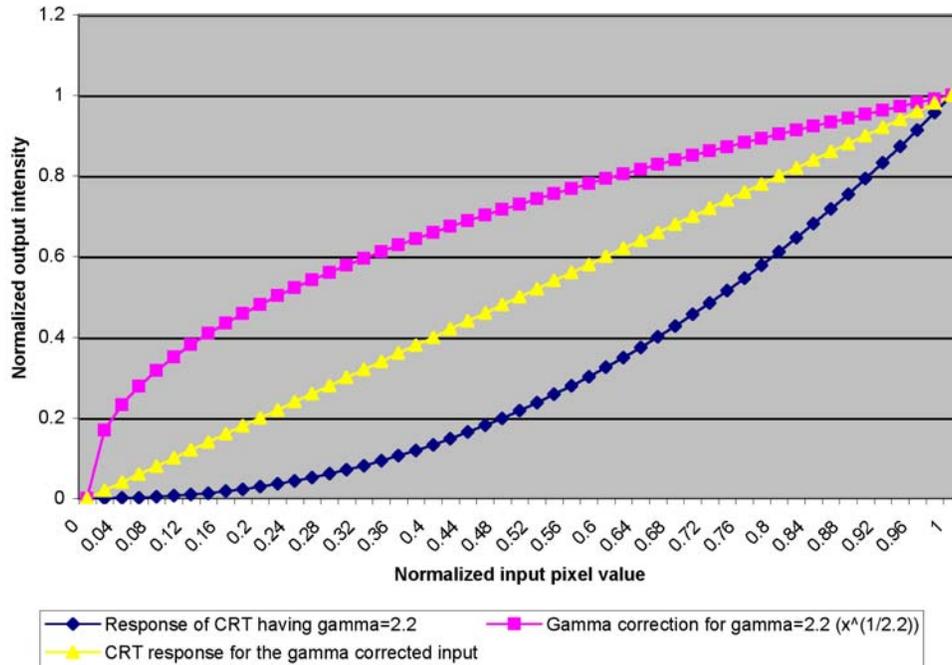
The actual pixel mapping depends on the bit width of the pixel component. For a pixel width of b bits, we have:

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

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



Filling the Gamma LUT

Based on the above equation, the size of the gamma LUT is 2^b bits deep and b bits wide. In most cases, the gamma LUT is 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 (2) or the actual gamma mapping values for each input pixel value. This IP supports reloading of the gamma LUT through input ports. It takes as many system clock 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 three color planes, a selection input (cpsel) 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/Os selected by the user.

Multi-Color Plane/Channel Mapping

Gamma correction of multiple color planes or channels can be done either sequentially or in parallel. In sequential correction, only one color component datum 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 is used to identify which color plane LUT needs to be used for the current input. The same cpsel 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.

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` and the corresponding correction value at the input `lutxval` while keeping the corresponding LUT write enable (`lutxwren`) 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 is used to identify the LUT that is programmed.

Handshake I/O Ports

The input gcen 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 identifies the color plane or channel number that is being processed or programmed. The output cpout provides the color plane or channel number information for the current output(s) appearing at doutx (x stands for 0, 1 or 2) port(s).

Signal Descriptions

The I/O port definitions for the Gamma Corrector IP core are given in [Table 2-1](#).

Table 2-1. I/O Interface Descriptions

Port	Bits	I/O	Description
All Configurations			
clk	1	I	System clock (reference clock for input and output data).
rstn	1	I	System wide asynchronous active-low reset signal.
inpvalid	1	I	Input data valid. Indicates valid data is present on din0 (also on din1 and din2 when present).
din0	4 to 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	4 to 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	1	O	Output data valid. Indicates valid data is present on dout0 (also on dout1 and dout2 when present). This valid data may correspond to the gamma corrected output or the bypassed input data, depending on the state of the gcvalid signal.
When 'Add Bypass Function' Option is Selected			
gcen	1	I	Gamma correction enable. This signal is valid only when inpvalid is asserted high. If gcen is high, gamma correction is performed for that input. Otherwise, no gamma correction is performed.
gcvalid	1	O	Gamma corrected output. This signal is valid only when outvalid is high. If gcvalid is high, the output is a gamma corrected value, otherwise the output is a bypassed value, same as input. The output gcvalid is a shifted version of the input signal gcen, the shift being equal to the output latency.
When Parallel Architecture is Selected and Number of Color Planes is More Than 1			
din1	4 to 12	I	Input data for the second color plane.
dout1	4 to 12	O	Output data for the second color plane.
When Parallel Architecture is Selected and Number of Color Planes is More Than 2			
din2	4 to 12	I	Input data for the third color plane.
dout2	4 to 12	O	Output data for the third color plane.
When First Color Plane Gamma Function is Selected as Programmable			
lut0val	4 to 12	I	Gamma value is programmed through the lut0val port when lut0wren is asserted high. The value from the lut0val port is written to the gamma LUT at the address provided at the din0 input port.
lut0wren	1	I	Write enable for writing the gamma value for the first color plane.
When Second Color Plane Gamma Function is Selected as Programmable, Parallel Architecture is Selected and Number of Color Planes is More Than 1			
lut1val	4 to 12	I	Gamma value is programmed through the lut1val port when lut1wren is asserted high. The value from the lut1val port is written to the gamma LUT at the address provided at the din1 input port.

Port	Bits	I/O	Description
lut1wren	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	4 to 12	I	Gamma value is programmed through the lut2val port when lut2wren is asserted high. The value from the lut2val port is written to the gamma LUT at the address provided at the din2 input port.
lut2wren	1	I	Write enable for writing the gamma value for the second color plane.
When sequential architecture is selected			
cpsel	1 to 2	I	Input color plane select. The inputs at din0, lut0val and lut0wren apply to the color plane provided at cpsel.
cpout	1 to 2	O	Output color plane select. The output at dout0 corresponds to the color plane provided at cpout.
Optional I/Os			
ce	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	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 goes low. This optional signal should be selected only when required as it leads to increased resource utilization.

Interfacing with the Gamma Corrector

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. The output data for the color planes is given out sequentially using the same output port dout0 after a latency of a few clock cycles.

When sequential architecture is selected, the input port din0 and output port dout0 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 and lut0val) 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) size. The color plane or channel number provided on cpsel can be in order, allowing the processing of multi-rate channels.

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 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 4 to 6 clock cycles depending on the parameters selected.

Timing Specifications

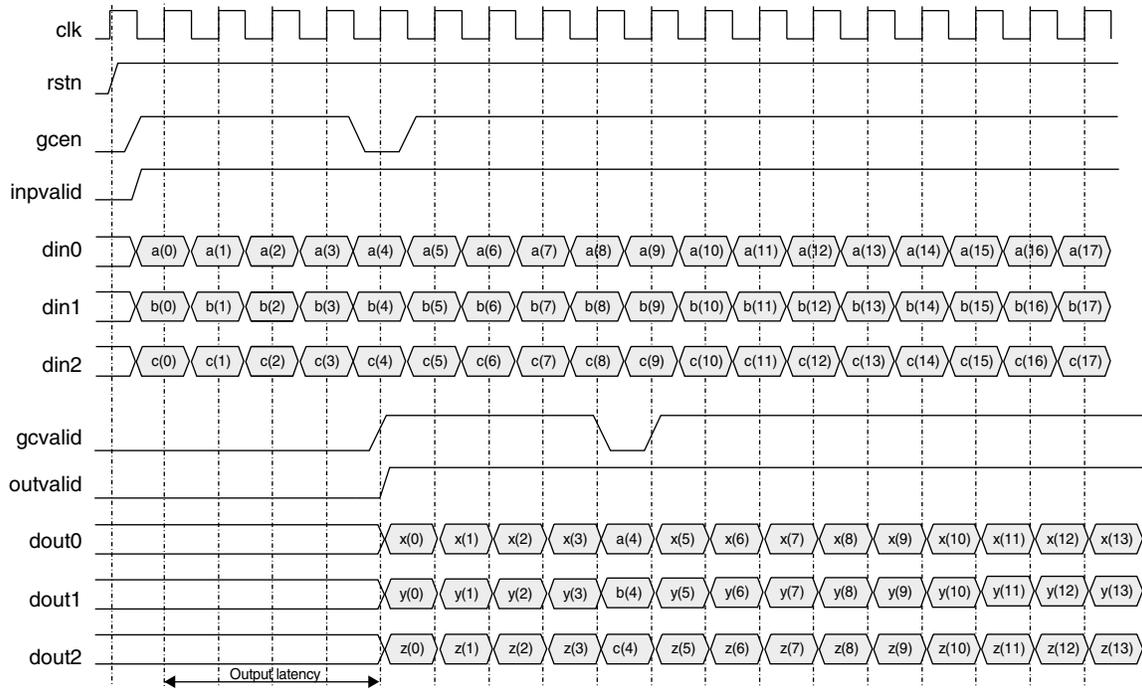
Parallel Architecture Timing

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, din1 and din2.

The signal inpsvalid 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, dout1 and dout2. The signal outvalid is asserted to indicate a valid output data present on the output ports. If signal gcen 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 gvalid signal. If the signal gcen is not asserted, the data from the input ports is passed on directly to the output ports. This is indicated by a low gvalid signal at the output.

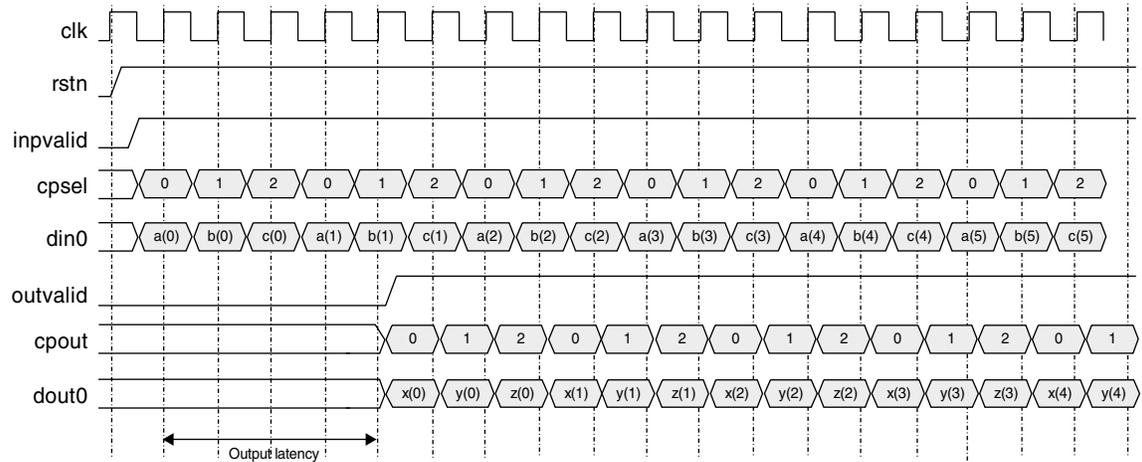
Figure 2-3. Parallel Architecture



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. The signal inval is asserted to indicate a valid data on din0. 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. The signal outvalid is asserted to indicate a valid data on dout0.

Figure 2-4. Sequential Architecture



Dynamically Loadable Gamma LUT Timing

Figure 2-5. Example of Dynamic Gamma Value Programming, Three Color Planes, Sequential Mode

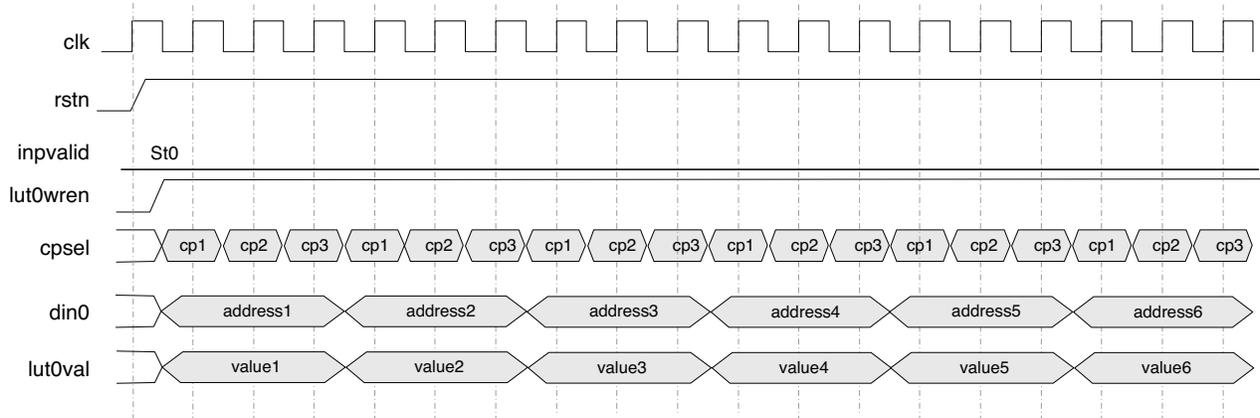


Figure 2-6. Example of Dynamic Gamma Value Programming, Three Color Planes, Parallel Mode

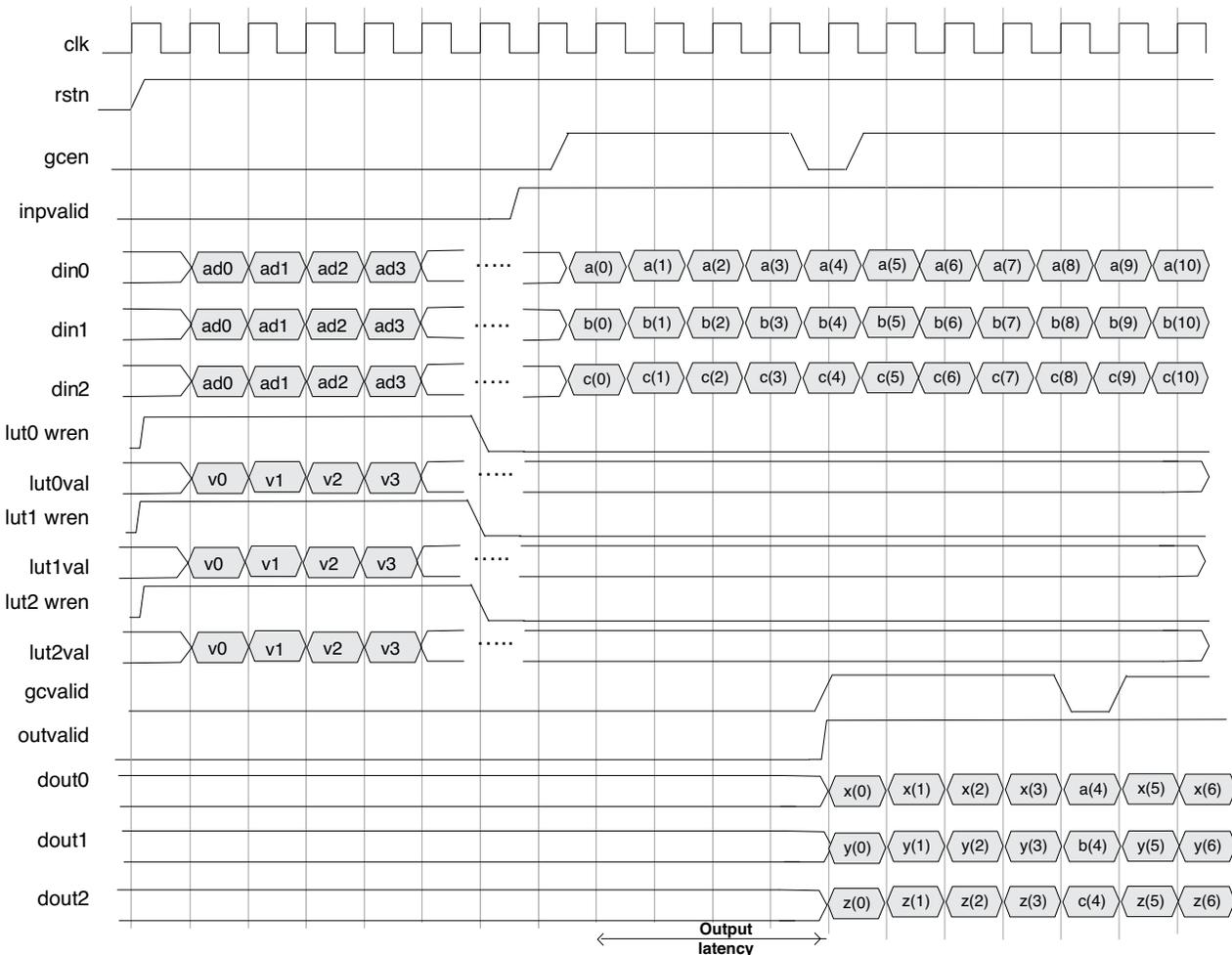


Figure 2-7. Functionality of Programmable Gamma through Look-Up Table (LUT)

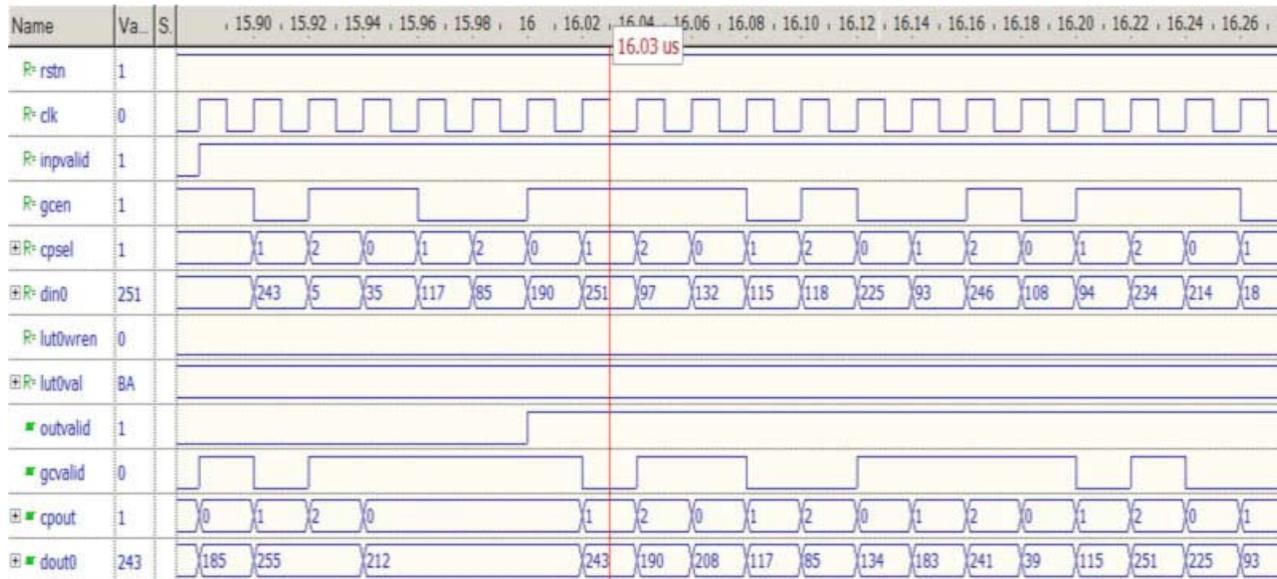
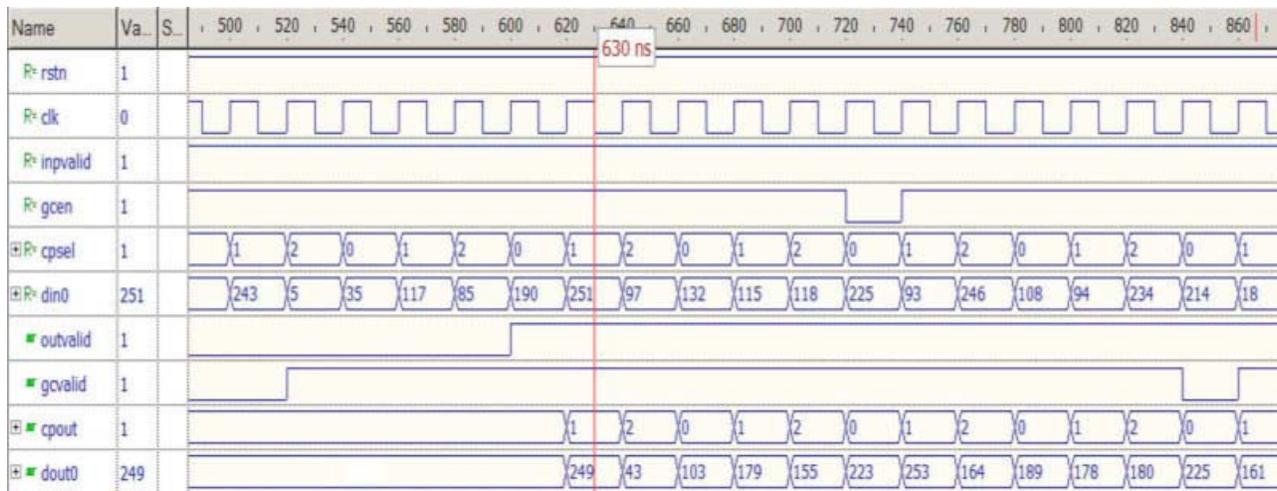


Figure 2-8. Functionality of Constant Gamma Value (= 2.2)



Parameter Settings

The IPexpress™ tool is used to create IP and architectural modules in the Diamond or ispLEVER software. Refer to “IP Core Generation” on page 16 for a description of how to generate the IP. The Gamma Corrector IP core can be customized to suit a specific application by adjusting parameters prior to core generation. Since the values of some parameters affect the size of the resultant core, the maximum value for these parameters may be limited by the size of the target device. Table 3-1 describes the user-configurable parameters available in the Gamma Corrector IP GUI. The GUI tabs for the default configuration are shown in Figures and 3-2.

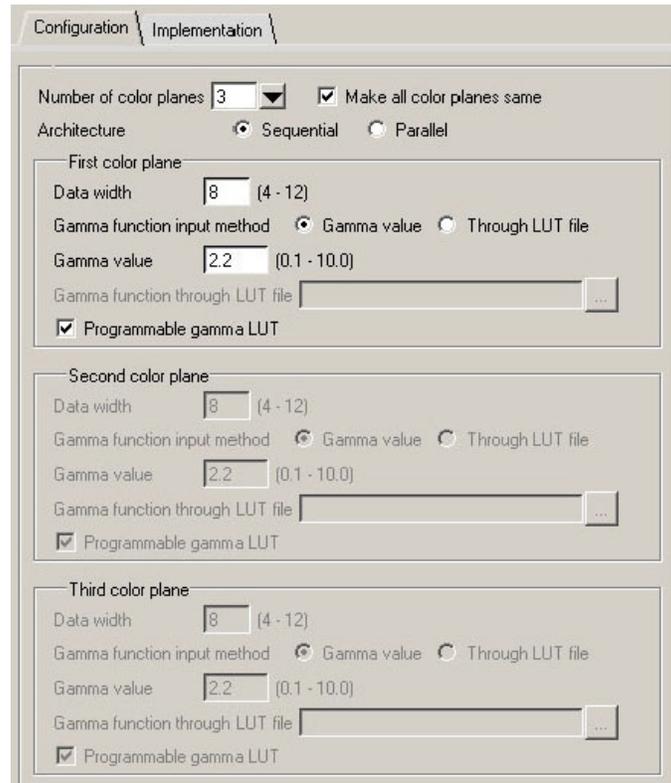
Table 3-1. User Configurable Parameters

Name	Range	Default
Common Color Plane Options		
Number of color planes	{1,2,3}	3
Make all color planes same	{Yes, No}	Yes
Architecture		
Architecture	{Sequential, Parallel}	Sequential
Individual Color Plane Options		
Data width	{4, 5, 6, 7, 8, 9, 10, 11, 12}	8
Gamma function input method	{Gamma value, Through LUT file}	Gamma value
Gamma value	0.1 to 10.0	2.2
Gamma function through file	N/A	N/A
Programmable gamma LUT	{Yes, No}	No
Implementation		
Add bypass function	{Yes, No}	Yes
Registered input	{Yes, No}	Yes
Memory type	{EBR, Distributed, Automatic}	Automatic
Output latency	{4,5,6}	6
Optional Input Ports		
ce	{Yes, No}	No
sr	{Yes, No}	No

Configuration Tab

The Configuration tab provides settings for common color plane options, architecture and individual color plane options. See Figure 3-1.

Figure 3-1. Configuration Tab of the Gamma Corrector IP Core GUI



Number of color planes: Sets the number of color planes for gamma correction.

Make all color planes 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.

Data width: Sets 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 the Gamma function input method is selected as “Gamma value”.

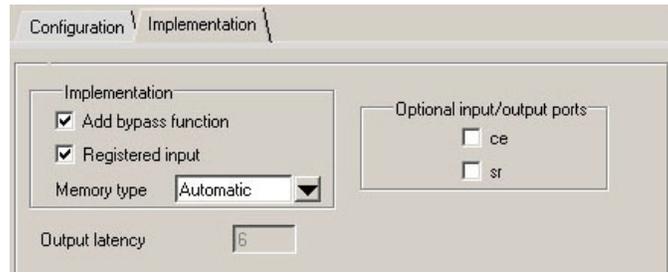
Gamma function through file: This browse button is enabled when the Gamma function input method is set to “Through LUT file”. The gamma LUT values will be read from the text file specified.

Programmable gamma LUT: This parameter is used to indicate if the gamma LUT is also programmable through the input port.

Implementation Tab

The Implementation tab provides settings for implementation, optional input/output ports and output latency options. See [Figure 3-2](#).

Figure 3-2. Implementation Tab of the Gamma Corrector IP Core GUI



Add bypass function: Selecting this option will add the dynamic gamma correction bypass functionality. Input port `gcn` and output port `gvalid` are added to the Gamma Corrector IP core.

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 the “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 the “Distributed” option is selected, distributed memory (realized using the FPGA’s LUTs) is used for the gamma LUT if data width is less than 9. If the “Automatic” option is selected, the memory resources are automatically selected in an optimal way, typically using distributed memories for data widths less than 7 and block memories for higher data widths. There are exceptions to the above-mentioned split of EBR/distributed memory usage for complex configurations that require optimization of the gamma LUT for multiple color planes.

Output latency: This static display shows the output latency for the selected core configuration.

Optional input ports:

- **ce:** Optional clock enable input port `ce` is added to the IP core if this option is checked.
- **sr:** Optional synchronous reset input port `sr` is added to the IP core if this option is checked.

Configuring the Gamma Corrector IP

When the gamma function input method is selected as Through LUT file, the gamma values are provided through a data file as decimal numbers. This data file contains the actual mapped gamma values. If the data width is 4 bits, then 16 values for the look up table are provided with the value for address 0 in the first line, the value for address 1 in the second line, and so on. Only the gamma-mapped values and not the addresses, are provided in the data file and the address is assumed to start from 0, running sequentially. The following is an example of the gamma LUT data file for a data width of 4 bits and gamma of 2.2.

```
0
4
6
7
8
9
10
11
11
12
12
13
14
14
15
15
```

This section provides information on how to generate the Gamma Corrector IP core using the Diamond or ispLEVER IPexpress tool, and how to include the core in a top-level design.

Licensing the IP Core

An IP core- and device-specific license is required to enable full, unrestricted use of the Gamma Corrector IP core in a complete, top-level design. Instructions on how to obtain licenses for Lattice IP cores are given at www.lattice-semi.com/products/intellectualproperty/aboutip/islevercoreonlinepurchas.cfm.

Users may download and generate the Gamma Corrector IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The Gamma Corrector IP core supports the Lattice IP hardware evaluation capability, which makes it possible to create versions of the IP core which operate in hardware for a limited time (approximately four hours) without requiring an IP license. See “[Hardware Evaluation](#)” on page 21 for further details. However, a license is required to enable timing simulation, to open the design in the Diamond or ispLEVER EPIC tool, and to generate bitstreams that do not include the hardware evaluation timeout limitation.

Getting Started

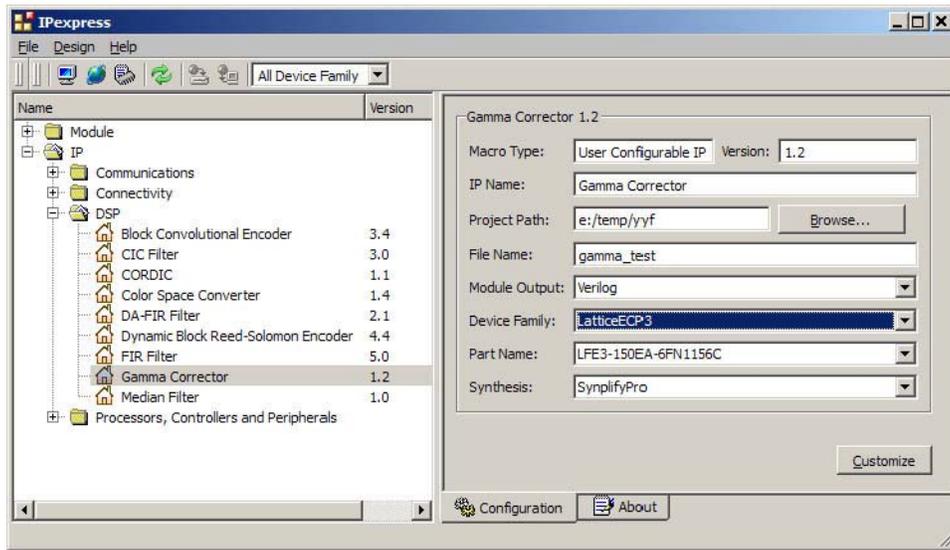
The Gamma Corrector IP core is available for download from the Lattice IP server using the IPexpress tool. The IP files are automatically installed using ispUPDATE technology in any user-specified directory. After the IP core has been installed, the IP core will be available in the IPexpress GUI dialog box shown in [Figure 4-1](#). To generate a specific IP core configuration, the user specifies:

- **Project Path** – Path to the directory where the generated IP files will be located.
- **File Name** – “username” designation given to the generated IP core and corresponding folders and files.
- **(Diamond) Module Output** – Verilog or VHDL.
- **(ispLEVER) Design Entry Type** – Verilog HDL or VHDL.
- **Device Family** – Device family to which IP is to be targeted (e.g. LatticeECP2M, LatticeECP2, etc.). Only families that support the particular IP core are listed.
- **Part Name** – Specific targeted part within the selected device family.

Configuring the Gamma Corrector IP Core in IPexpress

The Gamma Corrector configuration GUI is accessed via the ispLEVER IPexpress tool, and provides an interface for setting the desired parameters and invoking the IP core generator. The start-up IPexpress page allows the user to select the IP to be generated, project directory, user-designated module name, design entry type, and target device. The “File Name” will be used as username in the core generation. The Gamma Corrector IP core is found under **IP > DSP**, as shown below.

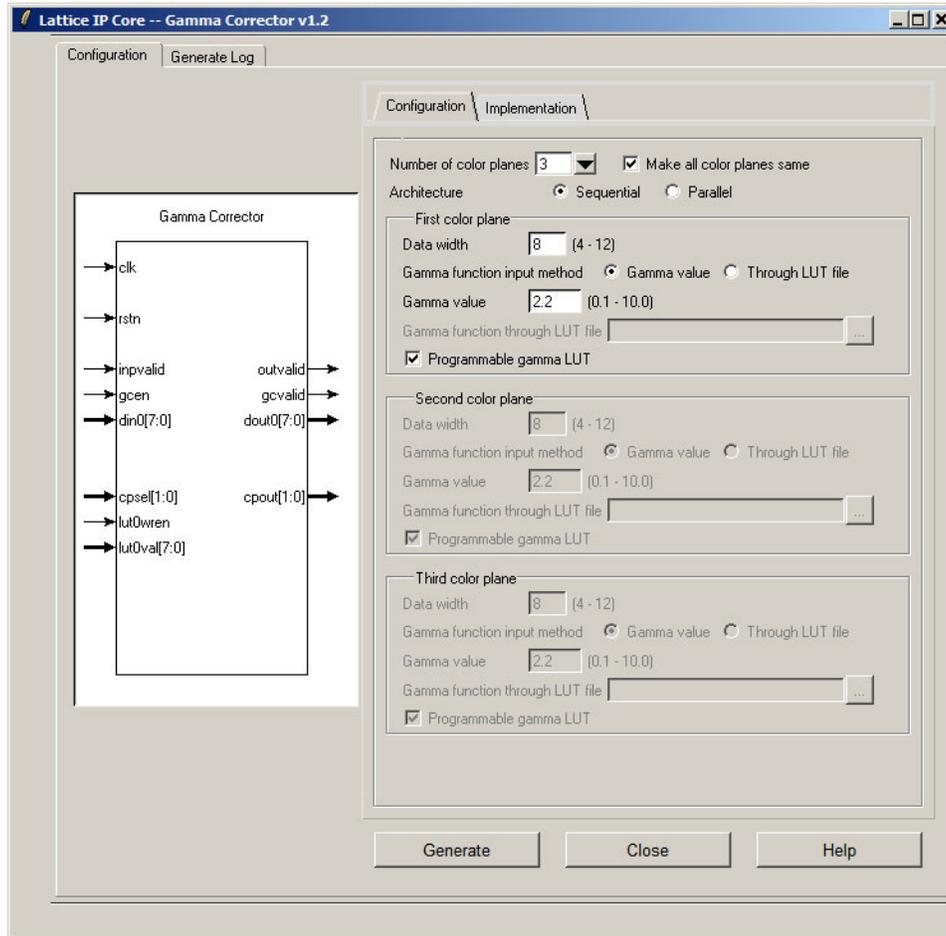
Figure 4-1. IPexpress Dialog Box (Diamond Version)



Note that if the IPexpress tool is called from within an existing project, Project Path, Module Output (Design Entry in ispLEVER), Device Family and Part Name default to the specified project parameters. Refer to the IPexpress tool online help for further information.

To create a custom configuration, the user clicks the **Customize** button in the IPexpress tool dialog box to display the Gamma Corrector IP core configuration GUI, as shown in [Figure 4-2](#). From this dialog box, the user can select the IP parameter options specific to their application. Refer to [“Parameter Settings” on page 13](#) for more information on the Gamma Corrector IP core parameter settings.

Figure 4-2. Configuration GUI (Diamond Version)



IPexpress-Created Files and Top-Level Directory Structure

When the user clicks the **Generate** button in the IP Configuration dialog box, the IP core and supporting files are generated in the specified “Project Path” directory. The directory structure of the generated files is shown in [Figure 4-3](#).

Figure 4-3. Gamma Corrector IP Core Directory Structure

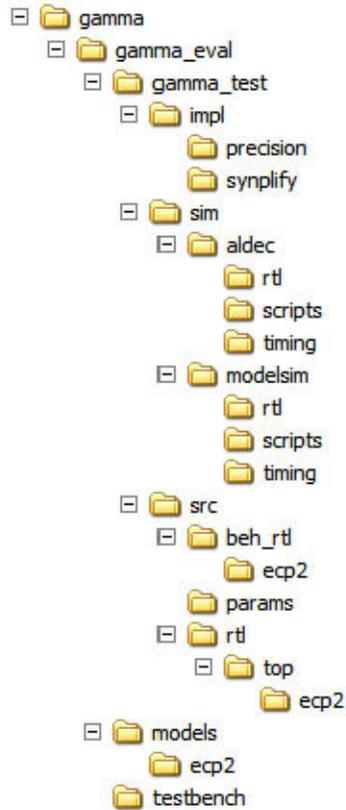


Table 4-1 provides a list of key files and directories created by the IPexpress tool and how they are used. The IPexpress tool creates several files that are used throughout the design cycle. The names of most of the created files are customized to the user’s module name specified in the IPexpress tool.

Table 4-1. File List

File	Description
<username>.lpc	This file contains the IPexpress tool options used to recreate or modify the core in the IPexpress tool.
<username>.ipx	The IPX file holds references to all of the elements of an IP or Module after it is generated from the IPexpress tool (Diamond version only). The file is used to bring in the appropriate files during the design implementation and analysis. It is also used to re-load parameter settings into the IP/Module generation GUI when an IP/Module is being re-generated.
<username>.ngo	This file provides the synthesized IP core.
<username>_bb.v	This file provides the synthesis black box for the user’s synthesis.
<username>_inst.v	This file provides an instance template for the Gamma Corrector IP core.
<username>_beh.v	This file provides the front-end simulation library for the Gamma Corrector IP core.

Table 4-2 provides a list of additional key files that provide IP core generation status information and command line generation capability that are generated in the user's project directory.

Table 4-2. Additional Files

File	Description
<username>_generate.tcl	This file is created when the GUI Generate button is pushed. This file may be run from command line.
<username>_generate.log	This is the synthesis and map log file.
<username>_gen.log	This is the IPexpress IP generation log file.

Instantiating the Core

The generated Gamma Corrector IP core package includes black-box (<username>_bb.v) and instance (<username>_inst.v) templates that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file that can be used as an instantiation template for the IP core is provided in \<project_dir>\gamma_eval\<username>\src\rtl\top. Users may also use this top-level reference as the starting template for the top-level for their complete design.

Running Functional Simulation

Simulation support for the Gamma Corrector IP core is provided for the Aldec Active-HDL (Verilog and VHDL) simulator and Mentor Graphics ModelSim simulator. The functional simulation includes a configuration-specific behavioral model of the Gamma Corrector IP core. The test bench sources stimulus to the core, and monitors output from the core. The generated IP core package includes the configuration-specific behavior model (<username>_beh.v) for functional simulation in the "Project Path" root directory. The simulation scripts supporting ModelSim evaluation simulation is provided in \<project_dir>\gamma_eval\<username>\sim\modelsim\scripts. The simulation script supporting Aldec evaluation simulation is provided in \<project_dir>\gamma_eval\<username>\sim\aldec\scripts. Both ModelSim and Active-HDL simulation is supported via test bench files provided in \<project_dir>\gamma_eval\test-bench. Models required for simulation are provided in the corresponding \models folder.

Users may run the Active-HDL evaluation simulation by doing the following:

1. Open Active-HDL.
2. Under the **Tools** tab, select **Execute Macro**.
3. Browse to folder \<project_dir>\gamma_eval\<username>\sim\aldec\scripts and execute one of the "do" scripts shown.

Users may run the ModelSim evaluation simulation by doing the following:

1. Open ModelSim.
2. Under the **File** tab, select **Change Directory** and choose the folder <project_dir>\gamma_eval\<username>\sim\modelsim\scripts.
3. Under the **Tools** tab, select **Execute Macro** and execute the ModelSim "do" script shown.

Note: When the simulation is complete, a pop-up window will appear asking "Are you sure you want to finish?"

Choose **No** to analyze the results. Choosing **Yes** closes ModelSim.

Synthesizing and Implementing the Core in a Top-Level Design

Synthesis support for the Gamma Corrector IP core is provided for Mentor Graphics Precision or Synopsys Synplify. The Gamma Corrector IP core itself is synthesized and is provided in NGO format when the core is generated in IPexpress. Users may synthesize the core in their own top-level design by instantiating the core in their top level as described previously and then synthesizing the entire design with either Synplify or Precision RTL synthesis.

The top-level file `<username>_eval_top.v` provided in `\<project_dir>\gamma_eval\<username>\src\top` supports the ability to implement the Gamma Corrector IP core in isolation. Push-button implementation of this top-level design with either Synplify or Precision RTL Synthesis is supported via the project files `<username>_eval.lfd` (Diamond) or `.syn` (ispLEVER) located in the `\<project_dir>\gamma_eval\<username>\impl\synplify` and the `\<project_dir>\gamma_eval\<username>\impl\precision` directories, respectively.

To use this project file in Diamond:

1. Choose **File > Open > Project**.
2. Browse to `\<project_dir>\gamma_eval\<username>\impl\<synplify or precision>` in the Open Project dialog box.
3. Select and open `<username>.lfd`. At this point, all of the files needed to support top-level synthesis and implementation will be imported to the project.
4. Select the **Process** tab in the left-hand GUI window.
5. Implement the complete design via the standard Diamond GUI flow.

To use this project file in ispLEVER:

1. Choose **File > Open Project**.
2. Browse to `\<project_dir>\gamma_eval\<username>\impl\<synplify or precision>` in the Open Project dialog box.
3. Select and open `<username>.syn`. At this point, all of the files needed to support top-level synthesis and implementation will be imported to the project.
4. Select the device top-level entry in the left-hand GUI window.
5. Implement the complete design via the standard ispLEVER GUI flow.

Hardware Evaluation

The Gamma Corrector IP core supports the Lattice IP hardware evaluation capability, which 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.

Enabling Hardware Evaluation in Diamond

Choose **Project > Active Strategy > Translate Design Settings**. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default.

Enabling Hardware Evaluation in ispLEVER

In the **Processes for Current Source** pane, right-click the **Build Database** process and choose **Properties** from the drop-down menu. The hardware evaluation capability may be enabled/disabled in the Properties dialog box. It is enabled by default.

Updating/Regenerating the IP Core

By regenerating an IP core with the IPexpress tool, you can modify any of its settings including device type, design entry method, and any of the options specific to the IP core. Regenerating can be done to modify an existing IP core or to create a new but similar one.

Regenerating an IP Core in Diamond

To regenerate an IP core in Diamond:

1. In IPexpress, click the **Regenerate** button.

2. In the Regenerate view of IPexpress, choose the IPX source file of the module or IP you wish to regenerate.
3. IPexpress shows the current settings for the module or IP in the Source box. Make your new settings in the **Target** box.
4. If you want to generate a new set of files in a new location, set the new location in the **IPX Target File** box. The base of the file name will be the base of all the new file names. The IPX Target File must end with an .ipx extension.
5. Click **Regenerate**. The module's dialog box opens showing the current option settings.
6. In the dialog box, choose the desired options. To get information about the options, click **Help**. Also, check the **About** tab in IPexpress for links to technical notes and user's guides. IP may come with additional information. As the options change, the schematic diagram of the module changes to show the I/O and the device resources the module will need.
7. To import the module into your project, if it's not already there, select Import **IPX to Diamond Project** (not available in stand-alone mode).
8. Click **Generate**.
9. Check the **Generate Log** tab to check for warnings and error messages.
10. Click **Close**.

The IPexpress package file (.ipx) supported by Diamond holds references to all of the elements of the generated IP core required to support simulation, synthesis and implementation. The IP core may be included in a user's design by importing the .ipx file to the associated Diamond project. To change the option settings of a module or IP that is already in a design project, double-click the module's .ipx file in the File List view. This opens IPexpress and the module's dialog box showing the current option settings. Then go to step 6 above.

Regenerating an IP Core in ispLEVER

To regenerate an IP core in ispLEVER:

1. In the IPexpress tool, choose **Tools > Regenerate IP/Module**.
2. In the **Select a Parameter File** dialog box, choose the Lattice Parameter Configuration (.lpc) file of the IP core you wish to regenerate, and click **Open**.
3. The **Select Target Core Version, Design Entry, and Device** dialog box shows the current settings for the IP core in the Source Value box. Make your new settings in the **Target Value** box.
4. If you want to generate a new set of files in a new location, set the location in the **LPC Target File** box. The base of the .lpc file name will be the base of all the new file names. The LPC Target File must end with an .lpc extension.
5. Click **Next**. The IP core's dialog box opens showing the current option settings.
6. In the dialog box, choose the desired options. To get information about the options, click **Help**. Also, check the **About** tab in the IPexpress tool for links to technical notes and user's guides. The IP core may come with additional information. As the options change, the schematic diagram of the IP core changes to show the I/O and the device resources the IP core will need.
7. Click **Generate**.
8. Click the **Generate Log** tab to check for warnings and error messages.

Lattice Technical Support

There are a number of ways to receive technical support as listed below.

Online Forums

The first place to look is Lattice Forums (www.latticesemi.com/support/forums.cfm). Lattice Forums contain a wealth of knowledge and are actively monitored by Lattice Applications Engineers.

Telephone Support Hotline

Receive direct technical support for all Lattice products by calling Lattice Applications from 5:30 a.m. to 6 p.m. Pacific Time.

- For USA and Canada: 1-800-LATTICE (528-8423)
- For other locations: +1 503 268 8001

In Asia, call Lattice Applications from 8:30 a.m. to 5:30 p.m. Beijing Time (CST), +0800 UTC. Chinese and English language only.

- For Asia: +86 21 52989090

E-mail Support

- techsupport@latticesemi.com
- techsupport-asia@latticesemi.com

Local Support

Contact your nearest Lattice sales office.

Internet

www.latticesemi.com

References

- [LatticeECP3 Family Handbook](#)
- [LatticeECP2/M Family Handbook](#)
- [LatticeXP2 Family Handbook](#)

Revision History

Date	Document Version	IP Core Version	Change Summary
January 2007	01.0	1.0	Initial release.
February 2009	01.1	1.1	Updated to show device utilization using ispLEVER 7.1 SP1.
February 2011	01.2	1.2	Updated to support Lattice Diamond 1.1 and ispLEVER 8.1 SP1 design software. Updated to support LatticeECP3 and LatticeXP2 FPGA families.

Resource Utilization

This appendix gives resource utilization information for Lattice FPGAs using the Gamma Corrector IP core. The IP configurations shown in this chapter were generated using the IPexpress software tool. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the Diamond and ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and Diamond and ispLEVER help systems. For more information on the Diamond or ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Table A-1 lists the parameter settings that were used to derive the performance and utilization data shown in Tables A-3, A-4, A-4 and A-5. You can use the IPexpress software tool to help generate new configurations of this IP core.

Table A-1. Example Configurations for Gamma Corrector IP Core

	Configuration 1	Configuration 2	Configuration 3
Architecture	Sequential	Parallel	Sequential
Mark all color planes same	Yes	Yes	No
Number of color planes	3	3	3
First Color Plane			
Data width	8	12	8
Gamma function input method	Gamma value	Gamma value	Gamma value
Gamma value	2.2	2.2	2.2
Gamma function through LUT file	NA	NA	NA
Programmable Gamma LUT	Yes	Yes	No
Second Color Plane			
Data width	8	12	8
Gamma function input method	Gamma value	Gamma value	Gamma value
Gamma value	2.2	2.2	3.0
Gamma function through LUT file	NA	NA	NA
Programmable Gamma LUT	Yes	Yes	No
Third Color Plane			
Data width	8	12	8
Gamma function input method	Gamma value	Gamma value	Gamma value
Gamma value	2.2	2.2	3.2
Gamma function through LUT file	NA	NA	NA
Programmable Gamma LUT	Yes	Yes	No
Implementation			
Add bypass function	Yes	Yes	Yes
Registered input	Yes	Yes	Yes
Memory type	Automatic	Automatic	Automatic
Output latency	6	5	6
ce	No	No	No
sr	No	No	No

LatticeECP3 Devices

Table A-2. Performance and Resource Utilization¹

IPexpress User-Configurable Mode	Slices	LUTs	Registers	sysMEM EBRs	f _{MAX} (MHz)
Configuration 1: Sequential architecture, 3 color planes, same color planes	57	32	113	3	305
Configuration 2: Parallel architecture, 3 color planes, same color planes	133	38	265	9	305
Configuration 3: Sequential architecture, 3 color planes, different color planes	52	29	104	3	305

1. Performance and utilization characteristics are generated using LFE3-70EA-7F672C with Lattice Diamond 1.1 software. When using this IP core in a different density, speed, or grade within the LatticeECP3 family, performance and utilization may vary.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Gamma Corrector targeting LatticeECP3 devices is GAMMA-E3-U1.

LatticeECP2 and LatticeECP2S Devices

Table A-3. Performance and Resource Utilization¹

IPexpress User-Configurable Mode	Slices	LUTs	Registers	sysMEM™ EBRs	f _{MAX} (MHz)
Configuration 1: Sequential architecture, 3 color planes, same color planes	57	29	113	3	370
Configuration 2: Parallel architecture, 3 color planes, same color planes	133	38	265	9	370
Configuration 3: Sequential architecture, 3 color planes, different color planes	52	26	104	3	370

1. Performance and utilization characteristics are generated using LFE2-50E-7F672C and LFE2-50SE-7F672C with Lattice Diamond 1.1 software. When using this IP core in a different density, speed, or grade within the LatticeECP2 and LatticeECP2S families, performance and utilization may vary.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Gamma Corrector targeting LatticeECP2 and LatticeECP2S devices is GAMMA-P2-U1.

LatticeECP2M and LatticeECP2MS Devices

Table A-4. Performance and Resource Utilization¹

IPexpress User-Configurable Mode	Slices	LUTs	Registers	sysMEM EBRs	f _{MAX} (MHz)
Configuration 1: Sequential architecture, 3 color planes, same color planes	57	28	113	3	370
Configuration 2: Parallel architecture, 3 color planes, same color planes	133	37	265	9	370
Configuration 3: Sequential architecture, 3 color planes, different color planes	52	25	104	3	370

1. Performance and utilization characteristics are generated using LFE2M50E-7F672C and LFE2M50SE-7F672C with Lattice Diamond 1.1 software. When using this IP core in a different density, speed, or grade within the LatticeECP2M and LatticeECP2MS families, performance and utilization may vary.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Gamma Corrector targeting LatticeECP2M and LatticeECP2MS devices is GAMMA-PM-U1.

LatticeXP2 Devices

Table A-5. Performance and Resource Utilization¹

IPexpress User-Configurable Mode	Slices	LUTs	Registers	sysMEM EBRs	f_{MAX} (MHz)
Configuration 1: Sequential architecture, 3 color planes, same color planes	57	27	113	3	310
Configuration 2: Parallel architecture, 3 color planes, same color planes	133	36	265	9	310
Configuration 3: Sequential architecture, 3 color planes, different color planes	52	24	104	3	310

1. Performance and utilization characteristics are generated using LFXP2-17E-7F484C with Lattice Diamond 1.1 software. When using this IP core in a different density, speed, or grade within the LatticeXP2 family, performance and utilization may vary.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Gamma Corrector targeting LatticeXP2 devices is GAMMA-X2-U1.