UG0938
User Guide
Gamma Correction

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

50200938. 2.0 7/21

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 2.0

The following is the summary of changes made in this revision.

- Added Table 2, page 4 and Table 6, page 13
- Updated Table 3, page 4 and Table 5, page 13
- Added section IP Configurator, page 5
- Updated section RTL, page 6
- Replaced Figure 4, page 7 and Figure 5, page 7

1.2 Revision 1.0

The first publication of this document.

2 Introduction

A camera sensor converts the light intensity of a pixel into equivalent digital value. The digital value is directly proportional to the light intensity of a pixel and hence has a linear relationship. However, a human eye perceives the images as a logarithmic function of intensity instead of a linear function. If the images from a camera sensor are directly displayed on a display device, then the scene will appear unnatural because of the differences between the camera sensor and human eye perception. To compensate this, the image from the camera sensor goes through the gamma correction.

Gamma correction uses an exponent function to compensate for the logarithmic perception by using the below equation:

$$D_{out} = (D_{in})^{\gamma}$$

Equation 1

where,

D_{out} represents output image data (RGB). Represented in the range of 0 to 1.

D_{in} represents input image data (RGB). Represented in the range of 0 to 1.

γ represents the gamma correction factor. A fixed gamma correction factor of 0.4545 corresponding to a gamma of 2.2 is implemented in the IP.

The Gamma correction IP is implemented using a LUT for exponent function.

The width of the input data determines the number of entries in the LUT. For example, 8-bit input data would require 2^8 (256) entries in the LUT.

This section describes the inputs and outputs and configuration parameters of the Gamma Correction IP.

2.1 Key Features

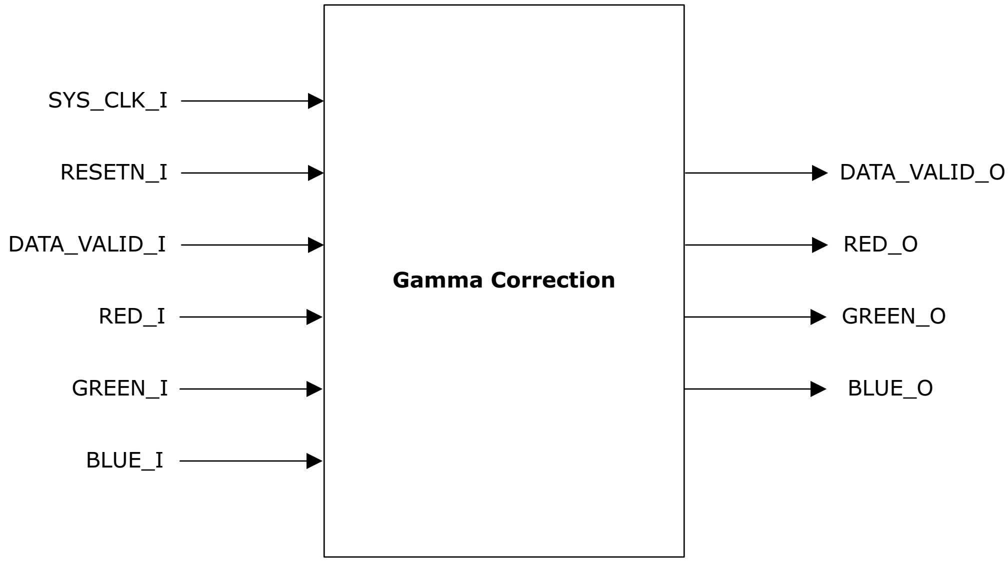
- LUT based gamma correction
- Fixed gamma factor of 2.2
- Supports data width of 8 and 10
- Supports Native and AXI4 Stream Video Interface for video data transfer

2.2 Supported Families

- PolarFire® SoC
- PolarFire®
- RTG4™
- IGLOO®2
- SmartFusion®2

2.3 Inputs and Outputs

Figure 1 • **Inputs and Outputs**



The following table lists the input and output ports of the Gamma Correction IP.

Table 1 • **Input and Output Ports for Native Video Interface**

Port Name	Direction	Width	Description
SYS_CLK_I	Input	1 bit	System clock.
RESETN_I	Input	1 bit	Active low asynchronous reset signal.
DATA_VALID_I	Input	1 bit	Input data valid signal. This signal should be asserted when the data is valid.
RED_I	Input	8 bits	Input Red pixel data.
GREEN_I	Input	8 bits	Input Green pixel data.
BLUE_I	Input	8 bits	Input Blue pixel data.
DATA_VALID_O	Output	1 bit	Output data valid signal. This signal is asserted when the output data is valid.
RED_O	Output	8 bits	Output Red pixel data.
GREEN_O	Output	8 bits	Output Green pixel data.
BLUE_O	Output	8 bits	Output Blue pixel data.

Table 2 • **Input and Output Ports for AXI4 Stream Video Interface**

Port Name	Type	Width	Description
RESETN_I	Input	1 bit	Active low asynchronous reset signal to design
SYS_CLK_I	Input	1 bit	System clock
TDATA_I	Input	3*G_PIXELS*G_DATA_WIDTH bit	Input Video Data
TVALID_I	Input	1 bit	Input Video Valid
TREADY_O	Output	1 bit	Output slave ready signal
TUSER_I	Input	4 bits	bit 0 = End of frame bit 1 = Unused bit 2 = Unused bit 3 = Unused
TDATA_O	Output	3*G_PIXELS*G_DATA_WIDTH bit	Output Video Data
TVALID_O	Output	1 bit	Output Video Valid
TUSER_O	Output	4 bits	bit 0 = End of frame bit 1 = Unused bit 2 = Unused bit 3 = Unused
TSTRB_O	Output	G_DATA_WIDTH/8	Output Video Data strobe
TKEEP_O	Output	G_DATA_WIDTH/8	Output Video Data Keep
TLAST_O	Output	1 bit	Output End of frame

2.4 Configuration Parameters

The following table lists the configuration parameters used in the hardware implementation of the Gamma correction. These parameters are generic and can be varied based on the application requirement.

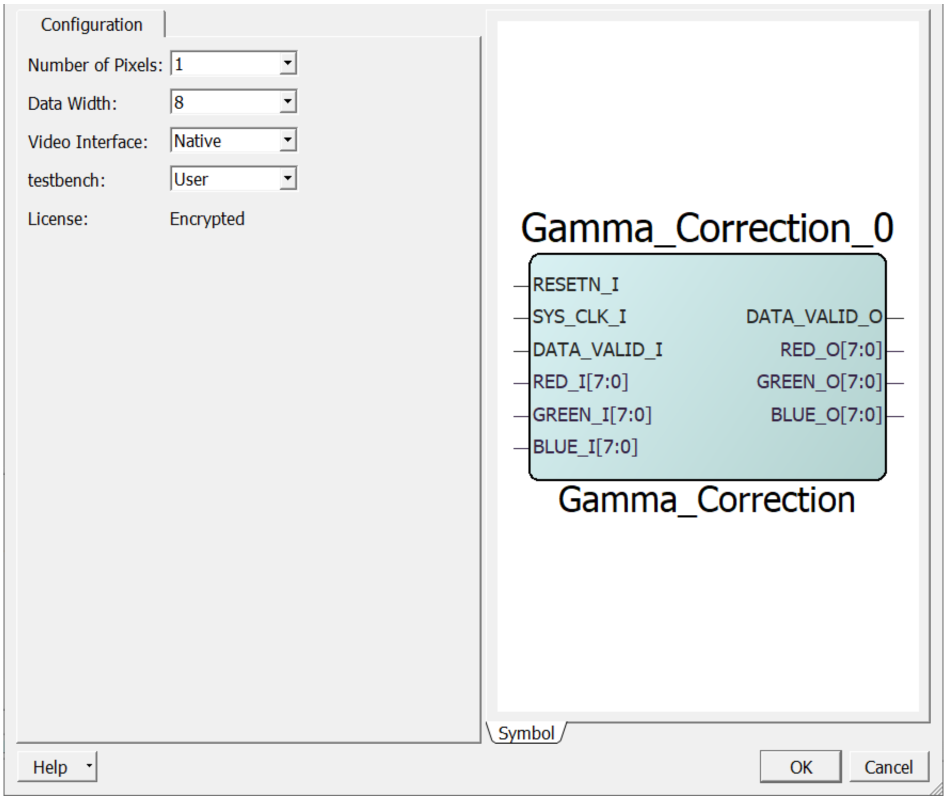
Table 3 • **Configuration Parameters**

Parameter Name	Description
Data Width	Represents bitwidth of input and output data. Current version supports 8-bit and 10-bit input and output data
Video Interface	Native Video Interface and AXI4 Stream Video Interface
Number of Pixel	Number of pixels per clock 1 and Number of pixels per clock 4

2.4.1 IP Configurator

The IP configurator is shown in the following figure.

Figure 2 • **IP Configurator**



3 Testbench

A testbench is provided to check the functionality of the Gamma Correction IP. To ensure that the testbench works correctly, the configuration parameters listed in Table 4 must be configured at the beginning of the testbench file.

Table 4 • **Testbench Configuration Parameters**

Name	Description
CLKPERIOD	Clock period
g_DATAWIDTH	Width of each pixel
HEIGHT	Vertical resolution
WIDTH	Horizontal resolution
IMAGE_FILE_NAME	Input image file

3.1 License

Gamma-Correction clear RTL is license locked and the encrypted RTL available for free.

3.1.1 Encrypted

Complete RTL code is provided for the core, allowing the core to be instantiated with the SmartDesign tool. Simulation, synthesis, and layout can be performed within Libero® System-on-Chip (SoC). The RTL code for the core is obfuscated.

3.1.2 RTL

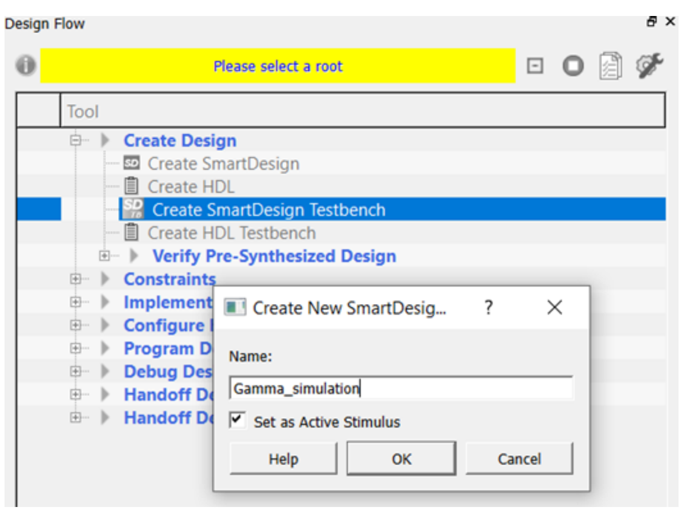
Complete RTL source code is provided for the core.

A testbench is provided to check the functionality of the Gamma-Correction IP. This testbench is working only in Native Video Interface with a data width of 8.

The following steps describe how to simulate the core using the testbench. The packaged testbench will gamma correct an input image.

1. In the **Design Flow** window, expand **Create Design**. Right-click **Create SmartDesign testbench** and click **Run**, as shown in the following figure.

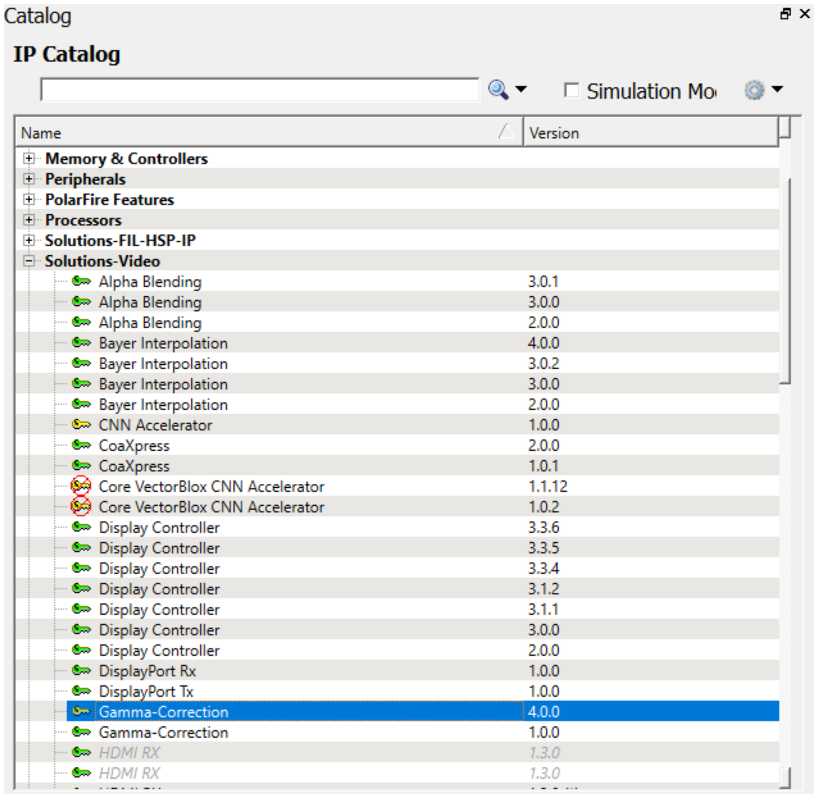
Figure 3 • **Design Flow**



SmartDesign testbench is created, and a canvas appears to the right of the Design Flow pane.

2. In the **Libero SoC Catalog** (**View > Windows > Catalog**), expand **Solutions-Video**, and drag the **Gamma-Correction** IP core onto the SmartDesign testbench canvas.

Figure 4 • **Libero SoC Catalog**



3. Select the default component name and click **OK**.
4. In the **Gamma-Correction Configurator** GUI window, update the G_DATA_WIDTH and click **OK**.
5. On Design Hierarchy tab, right-click GAMMA_CORRECTION_C0 and click **Set As Root**.
6. Select all the ports on the GAMMA_CORRECTION_C0 instance, right-click, and select **Promote to Top Level**, as shown in the following figure.

Figure 5 • **GAMMA_CORRECTION_C0 Instance**

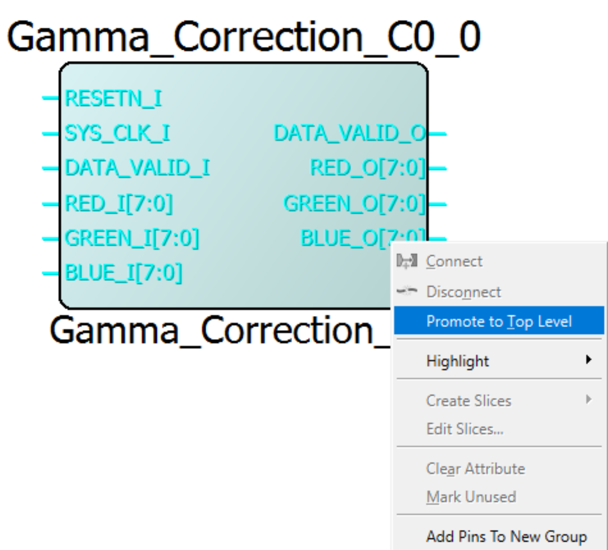
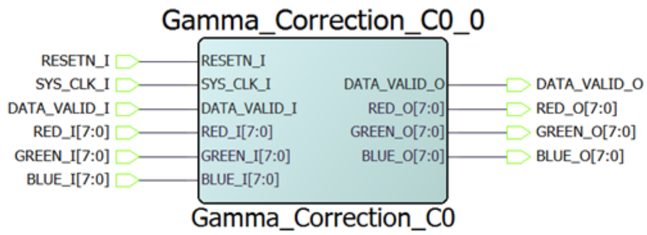
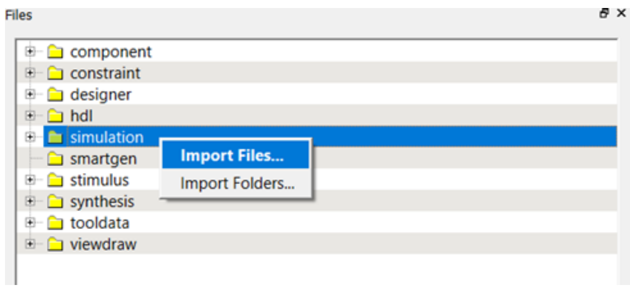


Figure 6 • **SmartDesign Toolbar**



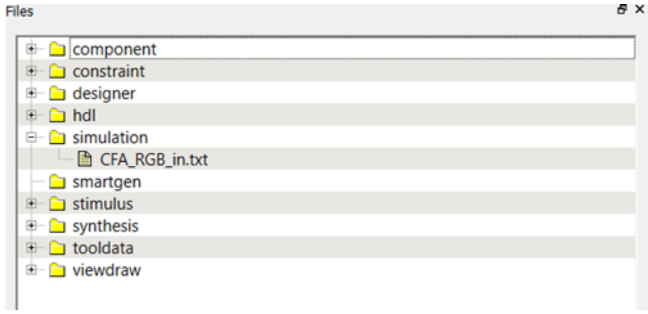
7. Click **Generate Component** from the SmartDesign toolbar.
8. Go to the Files tab and select **simulation** > **Import Files...,** as shown in the following figure.

Figure 7 • **Import Files**



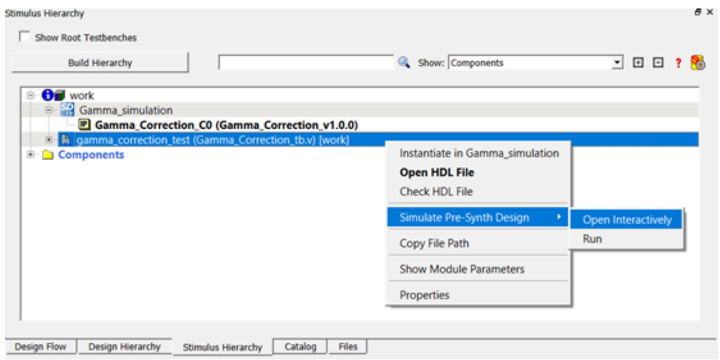
9. Import the Input Image file "`CFA_RGB_in.txt`" from the following path: ..\<Project_name>\component\Microsemi\SolutionCore\Gamma-Correction\4.0.0\Stimulus. To import a different file, browse the folder that contains the required file, and click **Open**. The imported file is listed under simulation as shown in the following figure.

Figure 8 • **Simulation**



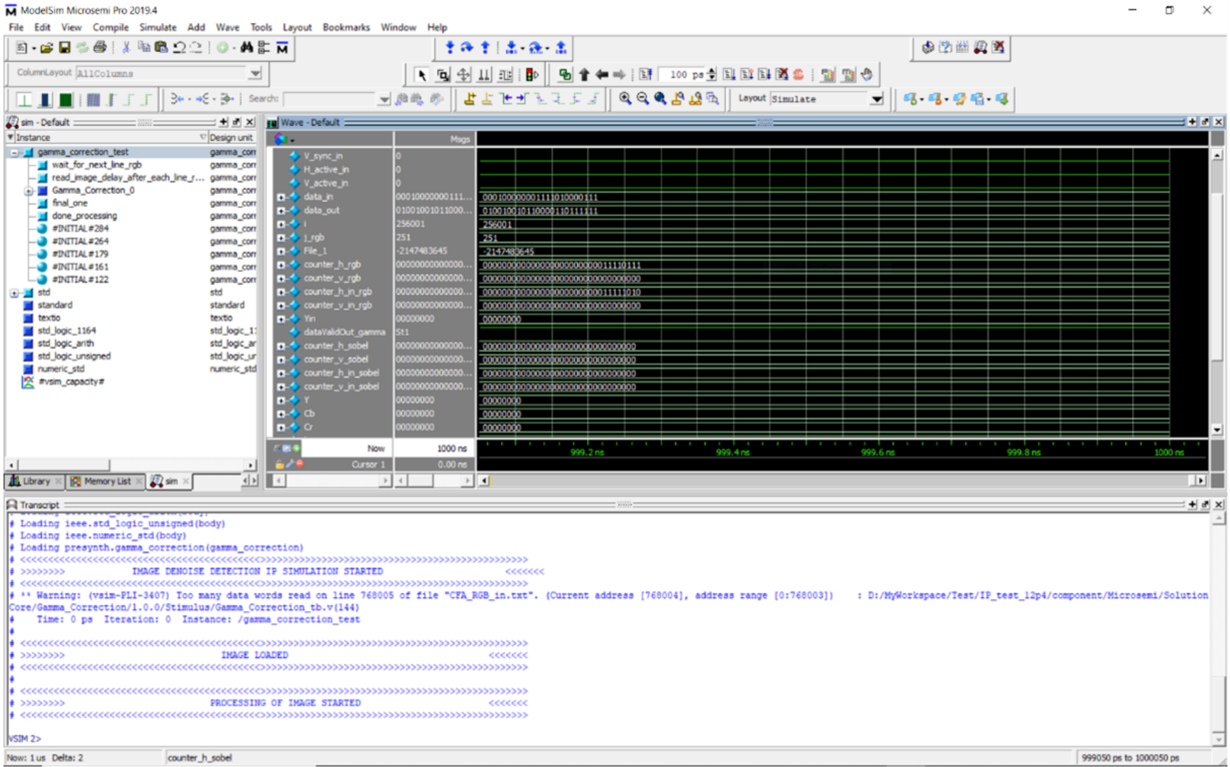
10. On Stimulus Hierarchy tab, right-click gamma_correction_test testbench file and click **Open Interactively** from Simulate Pre-Synth Design.

Figure 9 • **Stimulus Hierarchy**



The ModelSim tool appears with the test bench file loaded onto it, as shown in the following figure.

Figure 10 • **ModelSim tool**



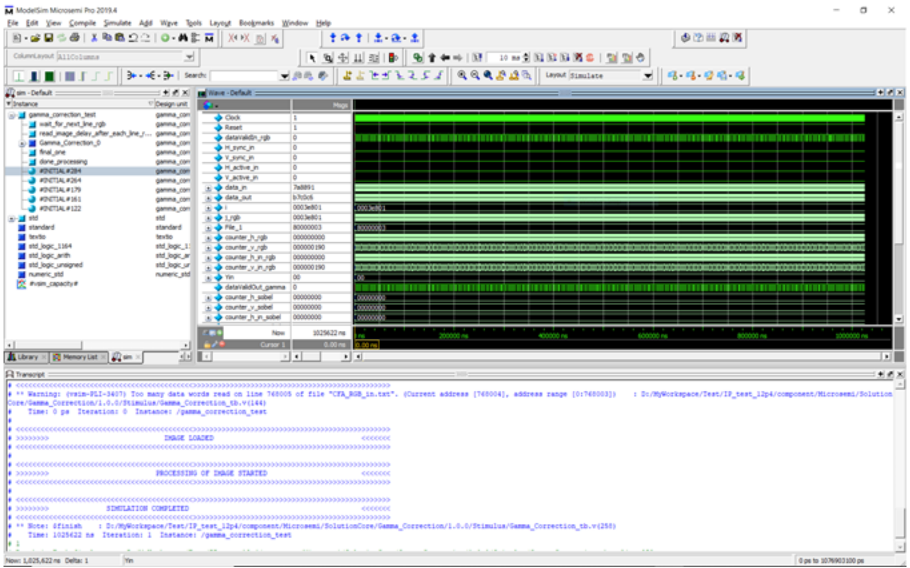
If the simulation is interrupted because of the runtime limit in the DO file, use the run `-all` command to complete the simulation. By default, the output image file is placed in the Files/simulation directory and uses the `CFA_RGB_out.txt`.

4 Simulation Results

4.1 Timing Diagram

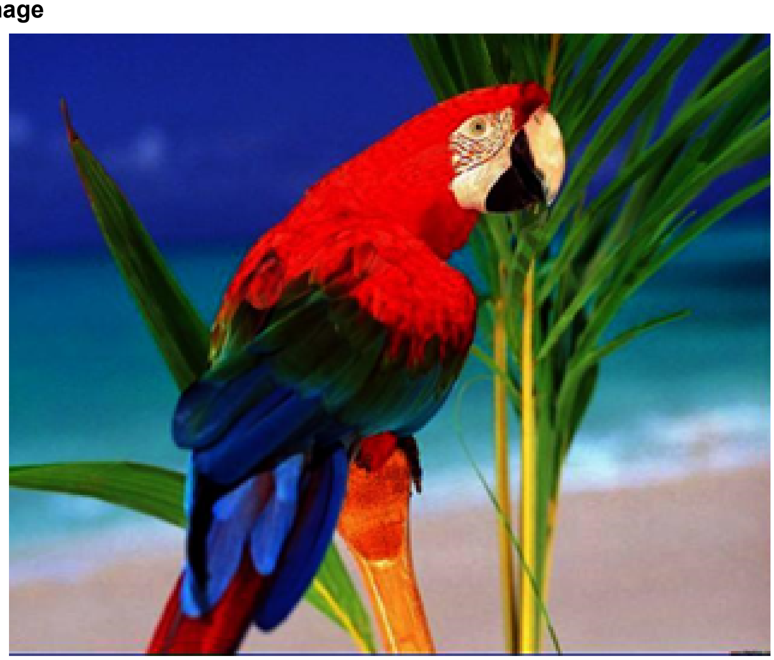
The following is the timing diagram for Gamma Correction IP showing video data and output image.

Figure 11 • Gamma Correction IP



4.2 Input Image

Figure 12 • Input Image



4.3 Output Image

Figure 13 • **Output Image**

5 Resource Utilization

Gamma correction is implemented on PolarFire FPGA (MPF500T -1FCG1152I package). The following table shows the resource utilization report after synthesis.

Table 5 • **Gamma Correction IP Resource Utilization for 1 pixel**

Gamma Correction IP	LUT	DFF	RAM1K20	MACC
Data Width = 8	354	26	0	0
Data Width = 10	1439	32	0	0

Table 6 • **Gamma Correction IP Resource Utilization for 4 pixels**

Gamma Correction IP	LUT	DFF	RAM1K20	MACC
Data Width = 8	1429	98	0	0
Data Width = 10	5760	122	0	0