UG0939
User Guide
White Balance IP

a Microchip company

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.

50200939. 1.0 11/20

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 1.0

The first publication of this document.

2 Introduction

White balance is the process of adjusting the color temperature so that the white objects which appear white in person are rendered as white. A scene can be lit by a light source of any temperature. A human eye adjusts to various color temperatures and detects white as white, where as a camera captures the light as it seen by the sensor.

As the color temperature rises, the color distribution becomes cooler. This may not seem intuitive but results from the fact that shorter wavelengths contain the light of higher energy. Such as 5000 K produces roughly neutral light, whereas 3000 K and 9000 K produce light spectrums, which shift to contain more orange and blue wavelengths, respectively. The temperature of commonly available light sources is described in the table below.

The white balance IP allows to set the temperature manually from user input to achieve white balance. The IP supports a temperature range of 1000 to 26500 Kelvin.

Table 1 • **Light Color Temperatures**

Color Temperature (Kelvin)	Light Source
1000-2000	Candlelight
2500-3500	Tungsten Bulb
3000-4000	Sunrise/Sunset
4000-5000	Fluorescent Lamps
5000-5500	Electronic Flash
5000-6500	Daylight with Clear Sky
6500-8000	Moderately Overcast Sky
9000-10000	Shade or Heavily Overcast Sky

This section describes the inputs and outputs and configuration parameters of the Image Scaler IP.

2.1 Key Features

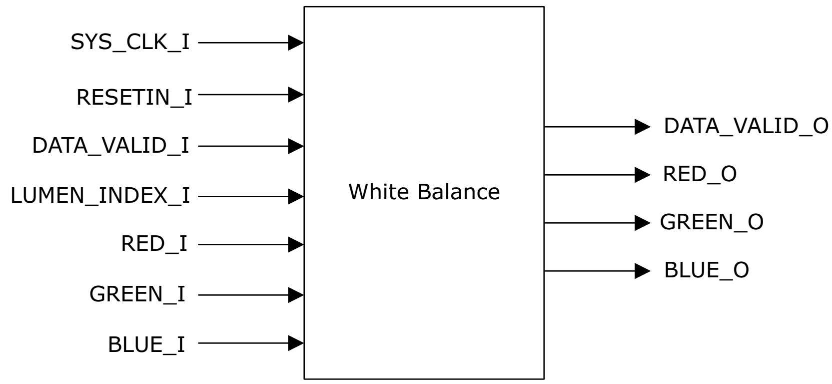
- Supports temperature range of 1000-26500 Kelvin.
- LUT based white balance
- Supports 8,10,12,14, and 16 data width

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 White Balance IP.

Table 2 • **Input and Output Ports**

Port Name	Direction	Width	Description
SYS_CLK_I	Input	1 bit	System clock
RESETN_I	Input	1 bit	Active low asynchronous reset signal
LUMEN_INDEX_I	Input	8 bits	Input luminance index varies between 0 to 255. A value of 0 corresponds to temperature of 1000K and each increment corresponds to a temperature rise of 100 Kelvin.
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

2.4 Configuration Parameters

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

Table 3 • **Configuration Parameters**

Parameter Name	Description
G_DATA_WIDTH	Represents bit-width of input and output data. Current version supports 8,10,12, and 16 bit input and output data.

3 Testbench

A testbench is provided to check the functionality of the White balance 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

White balance IP clear RTL is license locked and the obfuscated RTL available for free.

3.1.1 Obfuscated

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.

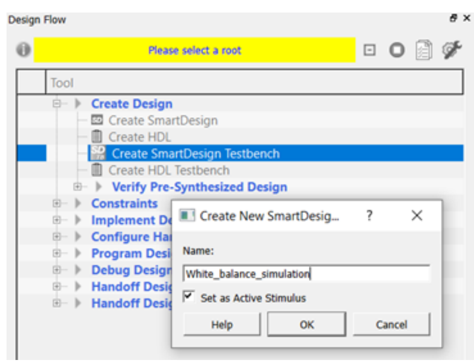
3.1.2 RTL

Complete RTL source code is provided for the core.

The following steps describe how to simulate the core using the testbench. The packaged testbench will correct the white balance of 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.

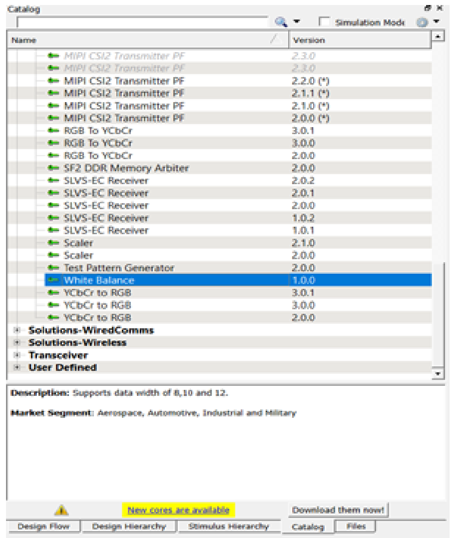
Figure 2 • **Design Flow**



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

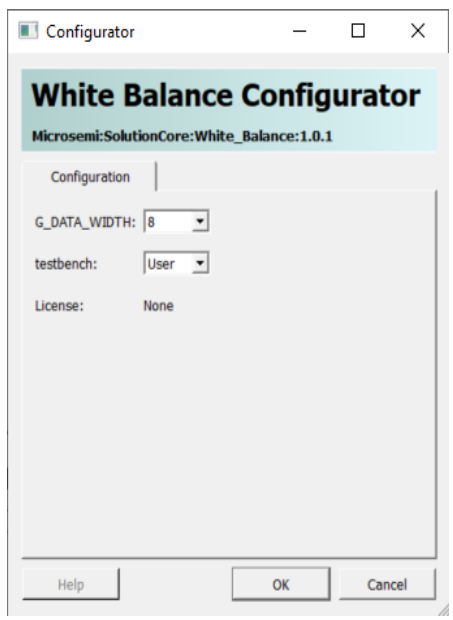
2. In the **Libero SoC Catalog (View > Windows > Catalog)**, expand **Solutions-Video**, and drag the **White balance** IP core onto the SmartDesign testbench canvas.

Figure 3 • **Libero SoC Catalog**



3. Select the default component name and click **OK**.
4. In the **White balance Configurator** GUI window, update the G_DATA_WIDTH and click **OK**.

Figure 4 • **Configurator**



5. On the Design Hierarchy tab, right-click White_Balance_C0 and click **Set As Root**.
6. Select all the ports on the White_Balance_C0 instance, right-click, and select **Promote to Top Level**, as shown in the following figure.

Figure 5 • **White_Balance_C0 Instance**

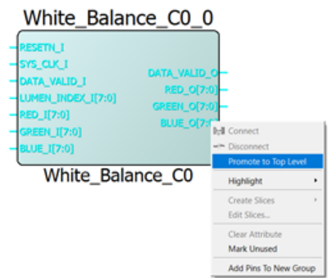
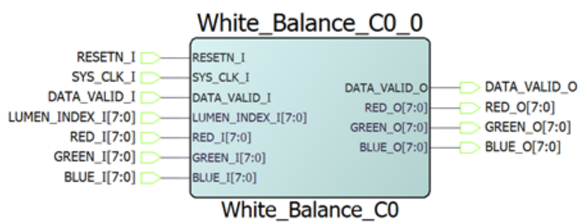
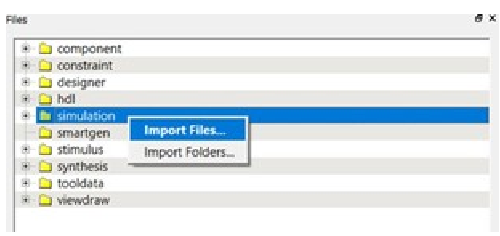


Figure 6 • **SmartDesign Toolbar**



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.

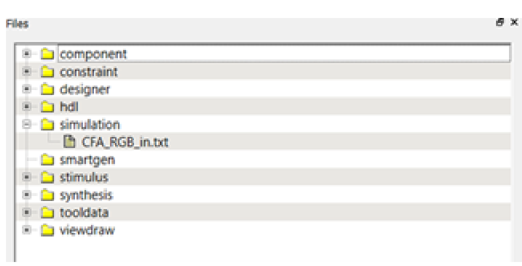
Figure 7 • **Import Files**



9. Import the Input Image file "`CFA_RGB_in.txt`" from the following path:

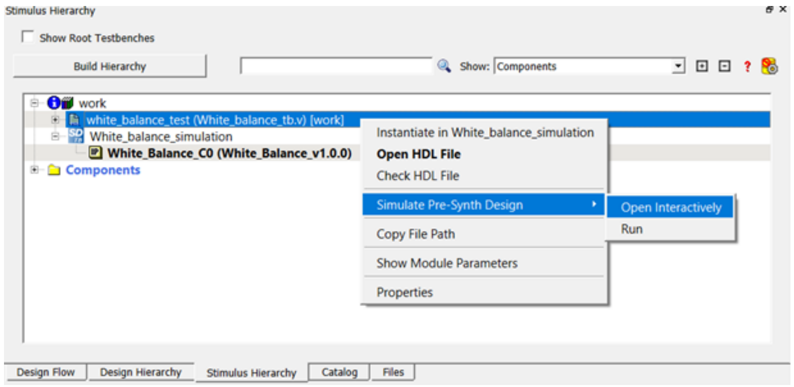
   ..\<Project_name>\component\Microsemi\SolutionCore\White_Balance\1.0.0\Stimulus. To import a different file, browse the folder that contains the required file, and click **Open**. The imported file is listed under simulation, as shown in the following figure.

Figure 8 • **Simulation**



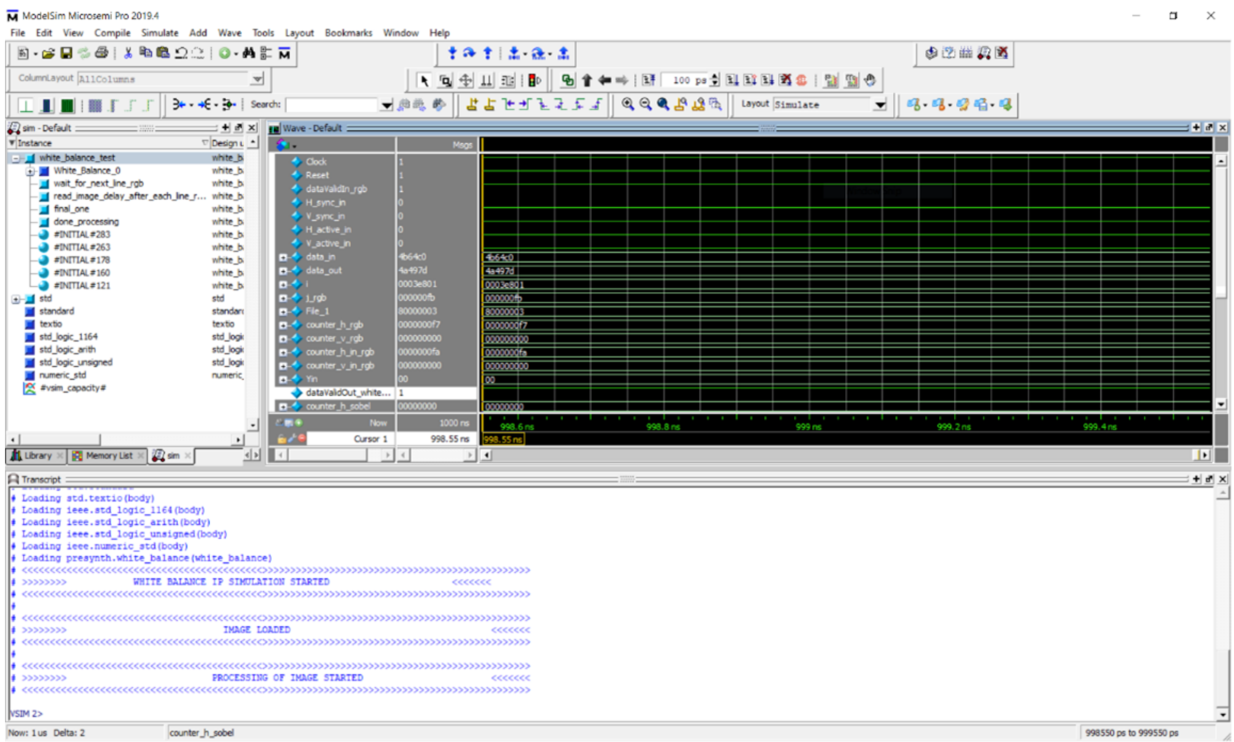
10. On the Stimulus Hierarchy tab, right-click white_balance_test testbench file and click **Open Interactively** from Simulate Pre-Synth Design.

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**



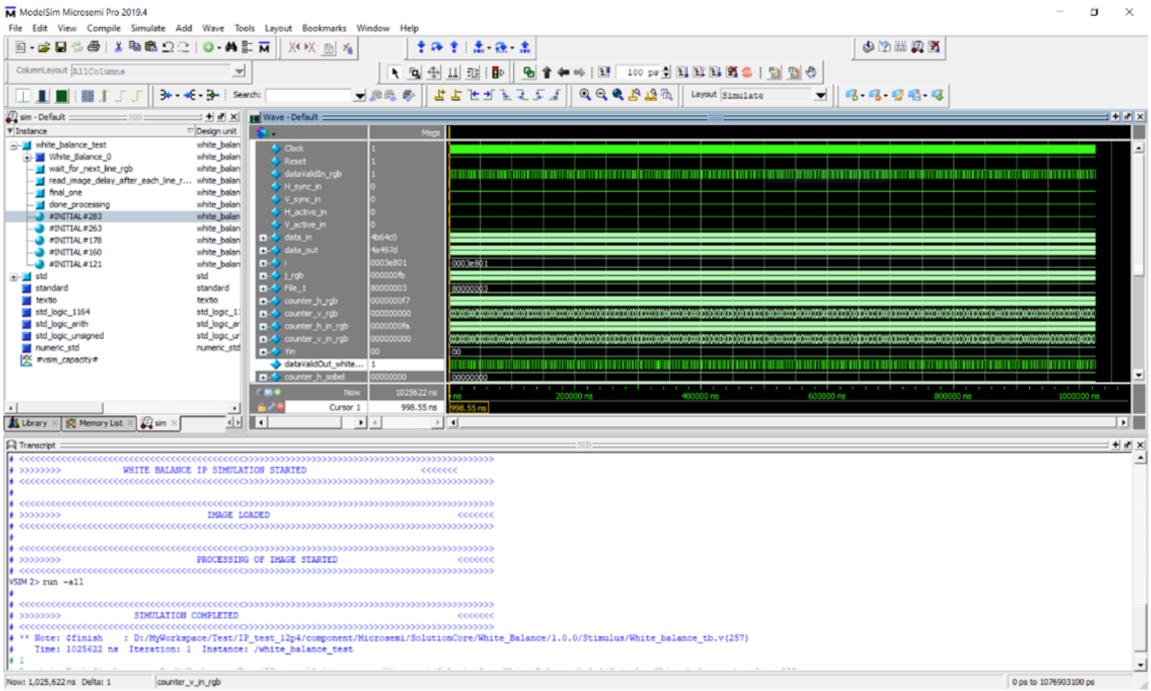
If the simulation is interrupted because of the runtime limit in the DO file, use the run `-all` command to complete the simulation. By default, the output image file is placed in the Files/simulation directory and uses the `CFA_RGB_out.txt`.

4 Simulation Results

4.1 Timing Diagram

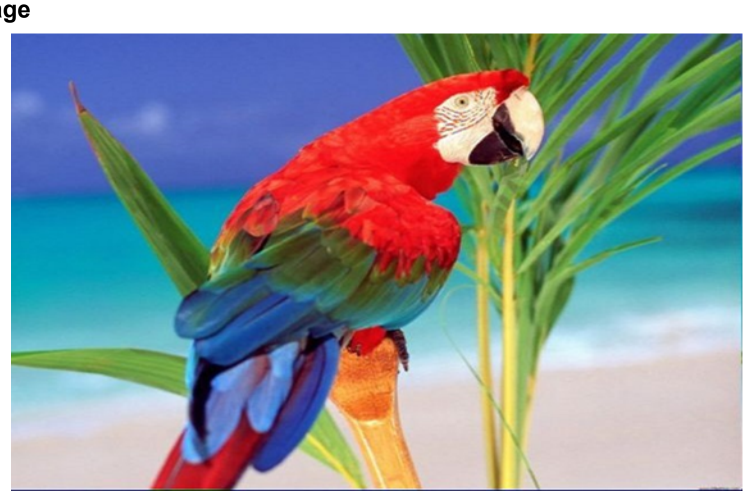
The following is the timing diagram for White Balance IP showing video data and output image.

Figure 11 • **White Balance IP**



4.2 Input Image

Figure 12 • **Input Image**



4.3 Output Image

Figure 13 • **Output Image**

5 Resource Utilization

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

Table 5 • **Resource Utilization**

Resource	Usage
DFFs	10
4LUTs	241
RAM1K20	0
MACC	3

Note: G_DATA_WIDTH = 8