Color Space Converter IP Core - Lattice Radiant Software

User Guide

FPGA-IPUG-02085-1.1

June 2020

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 and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. 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. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CSC	Color Space Converter
FPGA	Field Programmable Gate Array
IP	Intellectual Property
RTL	Register Transfer Level

1. Introduction

Color Space Converters (CSC) convert signals from one color space to another color space. Color space conversion is often required to ensure compatibility with display devices or to make the image data amenable for compression or transmission. CSCs are widely used in video and image display systems including televisions, computer monitors, color printers, video telephony and surveillance systems. They are also used in many video/image compression and processing applications, and in the implementation of NTSC/PAL/SECAM television standards, JPEG and MPEG systems.

The Lattice Color Space Converter IP Core is widely parameterizable and can support any custom color space conversion requirement. Furthermore, several commonly used color space conversion methods are provided as ready-to-use configurations.

This design is implemented in Verilog. It can be targeted to CrossLink™-NX and Certus™-NX FPGA devices and implemented using the Lattice Radiant® Software Place and Route tool integrated with the Synplify Pro® synthesis tool.

1.1. Quick Facts

Table 1.1 presents a summary of the Color Space Converter IP Core.

Table 1.1. Quick Facts

IP Requirements	Supported FPGA Families	CrossLink-NX, Certus-NX
Resource Utilization	Targeted Devices	LIFCL-40, LIFCL-17, LFD2NX-40
	Resources	See Table A.1
Design Tool Support	Lattice Implementation	IP Core v1.0.x – Lattice Radiant Software 2.1
	Synthesis	Lattice Synthesis Engine
		Synopsys® Synplify Pro for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant Software 2.1 User Guide.

1.2. Features

Key features of the Color Space Converter IP Core include:

- Input data width from 8 to 16 bits
- Output data width from 8 to 16 bits
- Signed or unsigned input and output data
- Supports standard configurations as well as custom configurations
- Parameterized coefficients precision from 9 to18 bits
- Full precision as well as limited precision output
- Programmable precision and rounding options for the output
- Optional sequential or parallel architecture for area or throughput optimization
- Registered input option available for input setup 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:

- _*n* are active low
- _*i* are input signals
- _*o* are output signals

1.3.3. Attribute Names

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

2. Functional Descriptions

2.1. Overview

A color space is a three-dimensional representation of the color and intensity of an image's pixel. An example of a color space is RGB wherein each pixel's color is represented by the constituent red, green, and blue components. This color space is used in computer displays where the CRT uses red, green, and blue to display a multi-colored pixel. However, an RGB color space may not be ideal for image processing, efficient image transmission, or human interpretation of color information. A color space that represents a color pixel using the characteristics of hue, saturation, and brightness is more akin to the way humans interpret color information. HIS and HSV are examples of such color spaces.

It is known that human vision is more sensitive to brightness than color. In an image, the color green carries more of the brightness information than the red and blue components. Therefore, some of the information from the red and blue color components can be reduced in order to compress the signal for more efficient processing. It is useful to deploy a color space representing brightness (luminance) and color components (chrominance) for processing applications. Common examples of such color spaces are YUV, YIQ and YCbCr, which are part of many video standards.

The following are some commonly used color spaces:

- **RGB** – Red, Green, Blue. This color space is used in computer displays.
- **YIQ, YUV, YCbCr** – Luminance, Chrominance. These color spaces are used in television systems. YIQ is used in NTSC systems, YUV is used in PAL systems and YCbCr is used in digital television systems.
- **CMY(K)** – Cyan, Magenta, Yellow, (Black). This color space is used in printing applications. The fourth component, black, is used to improve both the density range and color range. This removes the need to generate a good black color from CMY components.

Figure 2.1 shows a functional diagram for the Color Space Converter IP Core.

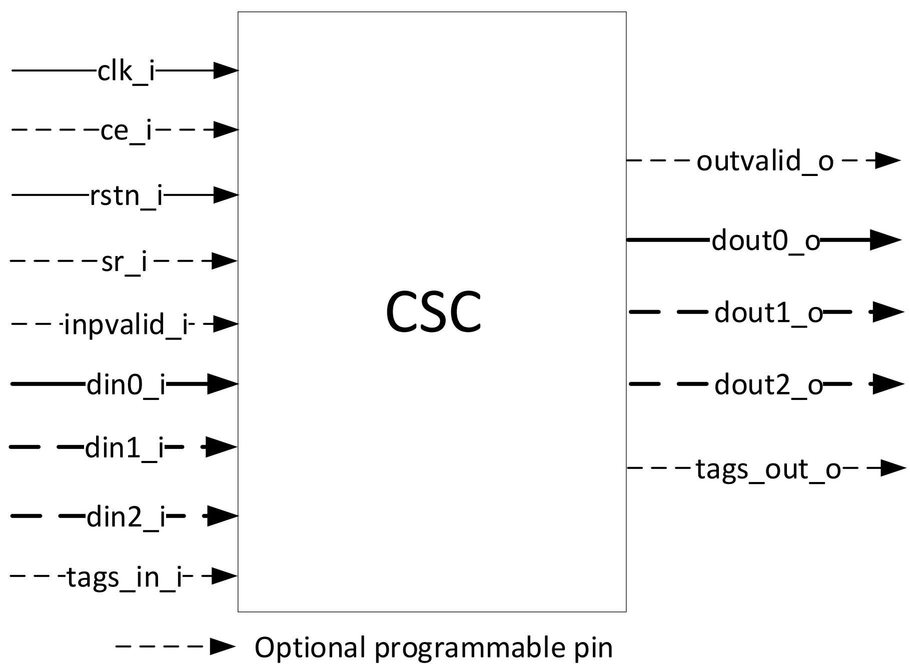

Figure 2.1. Functional Block Diagram

2.2. Signal Description

Table 2.1. Color Space Converter IP Core Signal Description

Port Name	I/O	Width	Description
Clock and Reset			
clk_i	In	1	Reference clock for input and output data.
rstn_i	In	1	System-wide asynchronous active-low reset signal.
I/O Ports			
din0_i[1]	In	*Input data width*[3]	Input data. When the sequential architecture is selected, this port is used to give input data for all the three input color planes in sequence. When the parallel architecture is selected, this port is used to give input data for the first input color plane.
dout0_o[1]	Out	*Output data width*[3]	Output data. When the sequential architecture is selected, this port is used to give output data for all the three output color planes in sequence. When the parallel architecture is selected, this port is used to give output data for the first output color plane.
tags_in_i	In	*Input tags width*[3]	User-defined tag input, up counter.
tags_out_o	Out	*Input tags width*[3]	User-defined tag output, up counter which is delayed by *Latency* with respect to tags_in_i.
Parallel Architecture			
din1_i[1 2]	In	*Input data width*[3]	Input data for second color plane.
din2_i[1 2]	In	*Input data width*[3]	Input data for third color plane.
dout1_o[1 2]	Out	*Output data width*[3]	Output data for second color plane.
dout2_o[1 2]	Out	*Output data width*[3]	Output data for third color plane.
Valid Signals			
inpvalid_i	In	1	Input data valid. Indicates valid data is present on din0_i (also on din1_i and din2_i when present). When the parallel architecture is selected, this port is optional. In this case, this port is not used directly in the core but is used to generate the outvalid_o signal after initial core latency. When the sequential architecture is selected, this port is always enabled. In this case, this port is used inside the core and also used to generate the outvalid_o signal after initial core latency. Also, when the sequential architecture is selected, this signal should be asserted high for one clock cycle when valid data for the first input color plane is present on the din0_i port. For the second and third input color planes data, this signal should be low. Input data for all the three input color planes should be applied at successive clock cycles without any gap.
outvalid_o	Out	1	Output data valid. Indicates valid data is present on dout0_o (also on dout1_o and dout2_o when present). When the parallel architecture is selected, this port is optional. When the sequential architecture is selected, this port is always enabled and asserted high when the valid data is present for the first output color plane. During output data of second and third color planes outvalid_o is low.

Optional I/O			
ce_i	In	1	Clock Enable. While this is de-asserted, the core ignores all other synchronous inputs and maintain its current state.
sr_i	In	1	Synchronous Reset. Asserted for at least one clock period duration to re-initialize the core. After synchronous reset, all internal registers are cleared and outvalid_o goes low.

Notes:

1. For RGB to YCbCr: din0_i/din1_i/din2_i/dout0_o/dout1_o/dout2_o stands for R/G/B/Y/Cb/Cr accordingly.
   For YCbCr to RGB: din0_i/din1_i /din2_i /dout0_o /dout1_o /dout2_o stands for Y/Cb/Cr/R/G/B accordingly.
   For YUV to RGB: din0_i/din1_i /din2_i /dout0_o /dout1_o /dout2_o stands for Y/U/V/R/G/B accordingly.
   For RGB to YUV: din0_i/din1_i /din2_i /dout0_o /dout1_o /dout2_o stands for R/G/B/Y/U/V accordingly.
   For YIQ to RGB: din0_i/din1_i /din2_i /dout0_o /dout1_o /dout2_o stands for Y/I/Q/R/G/B accordingly.
   For RGB to YIQ: din0_i/din1_i /din2_i /dout0_o /dout1_o /dout2_o stands for R/G/B/Y/I/Q accordingly.
   For YIQ to YUV: din0_i/din1_i /din2_i /dout0_o /dout1_o /dout2_o stands for Yin/I/Q/Yout/U/V accordingly.
2. These ports are available only when the selected *Architecture* is Parallel.
3. The bit width of some signals is set by the attribute. Refer to Table 2.3 for the description of these attributes.

2.3. Attribute Summary

The configurable attributes of the Color Space Converter IP Core are shown in Table 2.2.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
General			
Core Type	Custom, Computer RGB to YCbCr: SDTV, Computer RGB to YCbCr: HDTV, Studio RGB to YCbCr: SDTV, Studio RGB to YCbCr: HDTV, YCbCr: SDTV to Computer RGB, YCbCr: HDTV to Computer RGB, YCbCr: SDTV to Studio RGB, YCbCr: HDTV to Studio RGB, YUV to Computer RGB, Computer RGB to YUV, YIQ to Computer RGB, Computer RGB to YIQ, YIQ to YUV	Computer RGB to YCbCr:SDTV	—
Coefficient Width	9 – 18	14	—
Input data width	8 – 16	14	—
Input data type	Signed, Unsigned	Signed	—
Input Tags width	9 – 18	9	Active if *Support VSS IP Suite* == True
Provide Clock Enable	Checked, Unchecked	Checked	—
Provide Synchronous Reset	Checked, Unchecked	Checked	—
Provide Inpvalid/Outvalid	Checked, Unchecked	Checked	Active if *Architecture* == Parallel
Registered Input	Enable, Disable	Enable	—
Keep data at blank time	Checked, Unchecked	Unchecked	—
Support VSS IP Suite	Checked, Unchecked	Checked	—
Architecture	Serial, Parallel	Parallel	—

Attribute	Selectable Values	Default	Dependency on Other Attributes
din0_i coefficient for dout0_o	Calculated	0.257	*Core Type*
din1_i coefficient for dout0_o	Calculated	0.504	*Core Type*
din2_i coefficient for dout0_o	Calculated	0.098	*Core Type*
free coefficient for dout0_o	Calculated	16.0	*Core Type*
din0_i coefficient for dout1_o	Calculated	-0.148	*Core Type*
din1_i coefficient for dout1_o	Calculated	-0.291	*Core Type*
din2_i coefficient for dout1_o	Calculated	0.439	*Core Type*
free coefficient for dout1_o	Calculated	128.0	*Core Type*
din0_i coefficient for dout2_o	Calculated	0.439	*Core Type*
din1_i coefficient for dout2_o	Calculated	-0.368	*Core Type*
din2_i coefficient for dout2_o	Calculated	-0.071	*Core Type*
free coefficient for dout2_o	Calculated	128.0	*Core Type*
Custom Coefficients			
Custom din0_i coefficient for dout0_o	-100–100	0.257	Active if *Core Type* == Custom
Custom din1_i coefficient for dout0_o	-100–100	0.257	Active if *Core Type* == Custom
Custom din2_i coefficient for dout0_o	-100–100	0.257	Active if *Core Type* == Custom
Custom free coefficient for dout0_o	-1000–1000	16.0	Active if *Core Type* == Custom
Custom din0_i coefficient for dout1_o	-100–100	-0.148	Active if *Core Type* == Custom
Custom din1_i coefficient for dout1_o	-100–100	-0.291	Active if *Core Type* == Custom
Custom din2_i coefficient for dout1_o	-100–100	0.439	Active if *Core Type* == Custom
Custom free coefficient for dout1_o	-1000–1000	128.0	Active if *Core Type* == Custom
Custom din0_i coefficient for dout2_o	-100–100	0.439	Active if *Core Type* == Custom
Custom din1_i coefficient for dout2_o	-100–100	-0.368	Active if *Core Type* == Custom
Custom din2_i coefficient for dout2_o	-100–100	-0.071	Active if *Core Type* == Custom
Custom free coefficient for dout2_o	-1000–1000	128.0	Active if *Core Type* == Custom
Output			
Output data width	8–16	12	—
Output data type	Signed, Unsigned	Signed	—
Latency	Calculated	—	—
Rounding	None, Rounding up, Rounding away from zero, Rounding forwards zero, Convergent rounding	None	—
Overflow	Saturation, Wrap-around	Saturation	—

Table 2.3. Attributes Descriptions

Attribute	Description
General	
Core Type	Selects between Custom and pre-defined standard configurations. When the *Core Type* is selected as Custom, you must manually enter the coefficient values in the Custom Coefficients tab.
Coefficient Width	Specifies the coefficient precision width
Input data width	Specifies the bit width for the input color planes
Input data type	Signed or Unsigned Input data type
Input Tags width	Specifies the bit width for the tag ports: tags_in_i and tags_out_o
Provide Clock Enable	If enabled, input port ce_i is added to the core
Provide Synchronous Reset	If enabled, input port sr_i is added to the core
Provide Inpvalid/Outvalid	Configurable depending on the value of *Architecture*. If enabled, inpvalid_i and outvalid_o ports are added to the core
Registered Input	If enabled, inputs are registered. The core inputs' setup time improves by registering the inputs. This attribute is useful when the input data is provided on the device pins
Keep data at blank time	This attribute keeps the auxiliary data of the video stream unchanged during blank time.
Support VSS IP Suite	If enabled, input port tags_in_i and output port tags_out_o are added to the core.
Architecture	Selects between parallel and sequential implementation architectures
Output	
Output data width	Provides the latency for the selected core configuration
Output data type	Specifies the bit width of the output color planes
Latency	Provides the output latency for the selected core configuration
Rounding	Allows you to specify the rounding method when there is a need to drop one or more LSBs from the true output. **None** – discards all bits to the right of the output lease significant bit and leaves the output uncorrected. **Rounding up** – Rounds up if the fractional part is exactly one-half (for example, 2.5 is rounded to 3, -2.5 is rounded to -3). **Rounding away from zero** – Rounds away from zero if the fractional part is exactly one-half (for example, 2.5 is rounded to 3, -2.5 is rounded to -3). **Rounding towards zero** – Rounds towards zero if the fractional part is exactly one-half (for example, 2.5 is rounded to 2, -2.5 is rounded to -2). **Convergent rounding** – Rounds to the nearest even value if the fractional part is exactly one-half (for example, 2.5 is rounded to 2, -2.5 is rounded to -2, 3.5 is rounded to 4, -3.5 is rounded to -4).
Overflow	This attribute is available whenever there is a need to drop some of the MSBs from the true output. **Saturation** – The output is made equal to the maximum positive or negative value based on the sign bits. **Wrap-around** – The MSBs are discarded without making any corrections.

2.4. Operations Details

2.4.1. Timing Specifications

This section contains operational timing diagrams applicable to the Color Space Converter IP Core interfaces.

2.4.1.1. Parallel Architecture

Figure 2.2 shows the input and output signal timing diagram for the parallel architecture. The input data for all the three 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 three color planes appears on the output ports dout0_o, dout1_o and dout2_o. The signal outvalid_o is asserted to indicate valid output data present on the output ports.

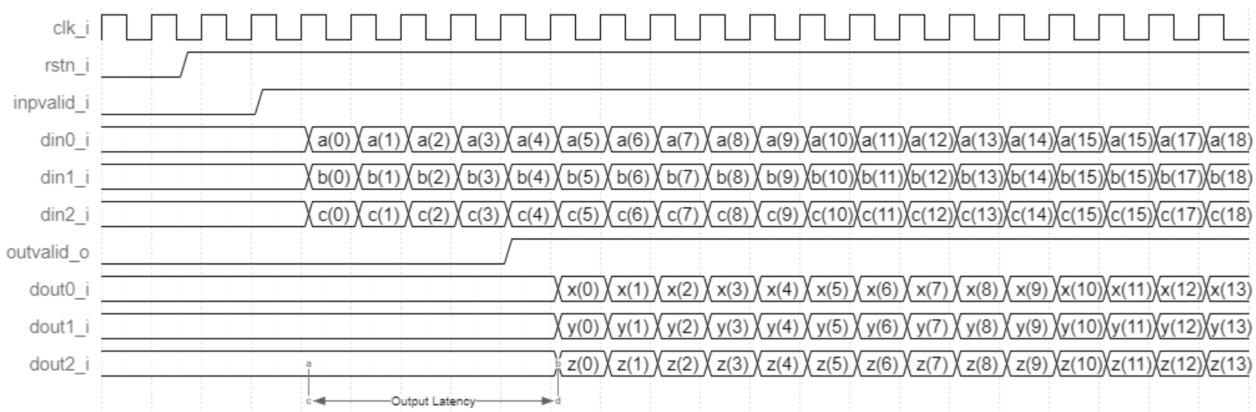

Figure 2.2. Parallel Architecture Timing Diagram

2.4.1.2. Sequential Architecture

Figure 2.3 shows the input and output signal timing for the sequential architecture. The input data for all three color planes are applied in sequence on the input port din0_i. The signal inpvalid_i is asserted to indicate the first color plane data on din0_i. In the following two cycles, the second and third color plane data are applied on din0_i. After a latency of a few cycles, the output data for the first color plane appears on the output port dout0_o. The signal outvalid_o is asserted to indicate the first color plane data on dout0_o. In the following two cycles, the second and third color plane data appear on dout0_o.

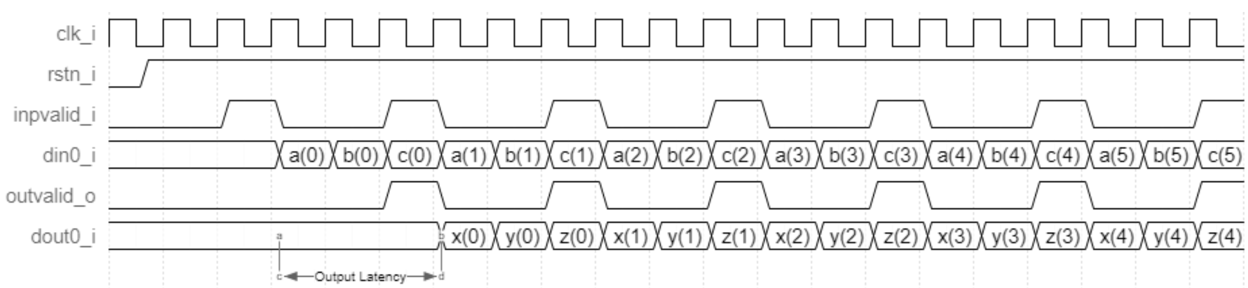

Figure 2.3. Sequential Architecture Timing Diagram

2.5. Color Space Conversion

Color space conversion is required when transferring data between devices that use different color space models. For example, RGB to YCbCr color space conversion is required when displaying a computer image on a television. Similarly, YCbCr to RGB color space conversion is required when displaying television movies on a computer monitor. As a color can be represented completely using three dimensions, a color space is a three dimensional space. Color space conversion is a one-to-one mapping from one color space to another color space.

R'G'B' to Y'CbCr color space conversion is provided in the following equations. The prime notations are used to denote gamma-corrected values.

Y' = 0.257 * R' + 0.504 * G' + 0.098 * B' + 16

Cb = -0.148 * R' -0.291 * G' + 0.439 * B' + 128

Cr = 0.439 * R' - 0.368 * G' -0.071 * B' + 128

Y'CbCr to computer R'G'B' conversion is provided in the following equations.

R' = 1.164 * Y' +0.0 * Cb + 1.596 * Cr -222.912

G' = 1.164 * Y' -0.392 * Cb -0.813 * Cr + 135.616

B' = 1.164 * Y' + 2.017 * Cb + 0.0 * Cr -276.8

Examples of applications that use CSC for R'G'B' to Y'CbCr Conversion and Y'CbCr to R'G'B' conversion are shown in Figure 2.4 and Figure 2.5.

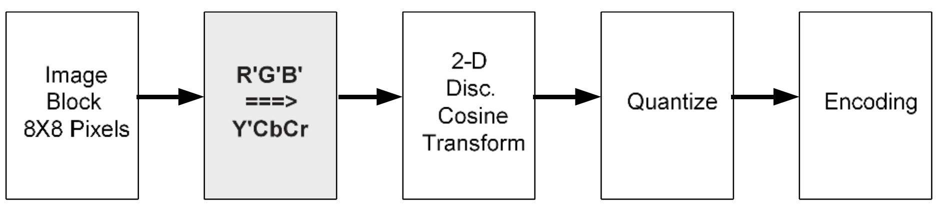

Figure 2.4. JPEG Encoding Application

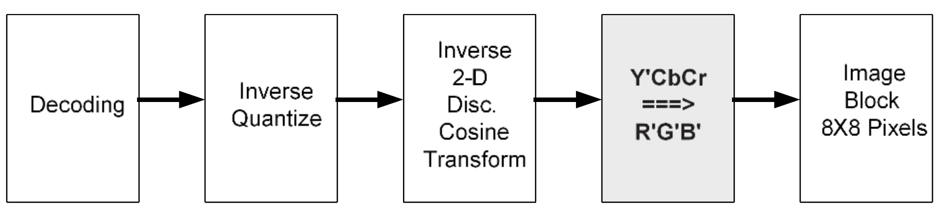

Figure 2.5. JPEG Decoding Application

3. IP Generation and Evaluation

This section provides information on how to generate the Color Space Converter 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 2.1 User Guide.

3.1. Licensing the IP

An IP core-specific license string is required enable full use of the Color Space Converter IP Core in a complete, top-level design. You can fully evaluate the IP core through functional simulation and implementation (synthesis, map, place and route) without an IP license string. This IP core supports Lattice's 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 string. See Hardware Evaluation section for further details. However, a license string is required to enable timing simulation and to generate bitstream file that does not include the hardware evaluation timeout limitation.

3.2. Generation and Synthesis

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 Color Space Converter IP Core in Lattice Radiant Software is described below.

To generate the Color Space Converter IP Core:

1. Create a new Lattice Radiant Software project or open an existing project.
2. In the **IP Catalog** tab, double-click on **Color Space Converter** under **IP, DSP** category. The **Module/IP Block Wizard** opens as shown in Figure 3.1. Enter values in the **Component name** and the **Create in** fields and click **Next**.

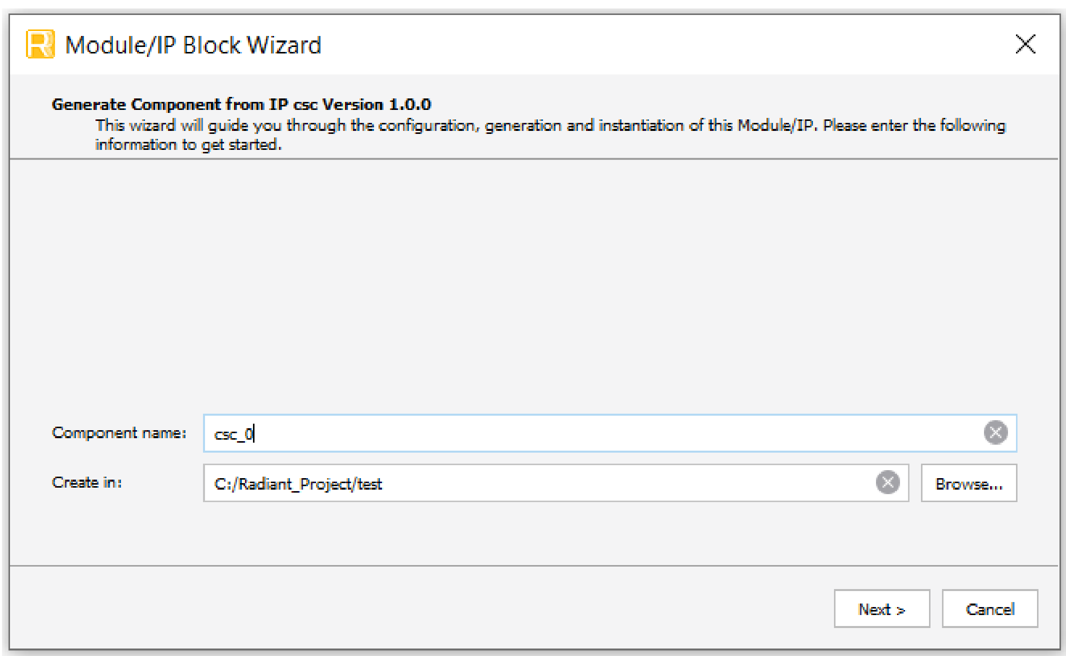

Figure 3.1. Module/IP Block Wizard

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

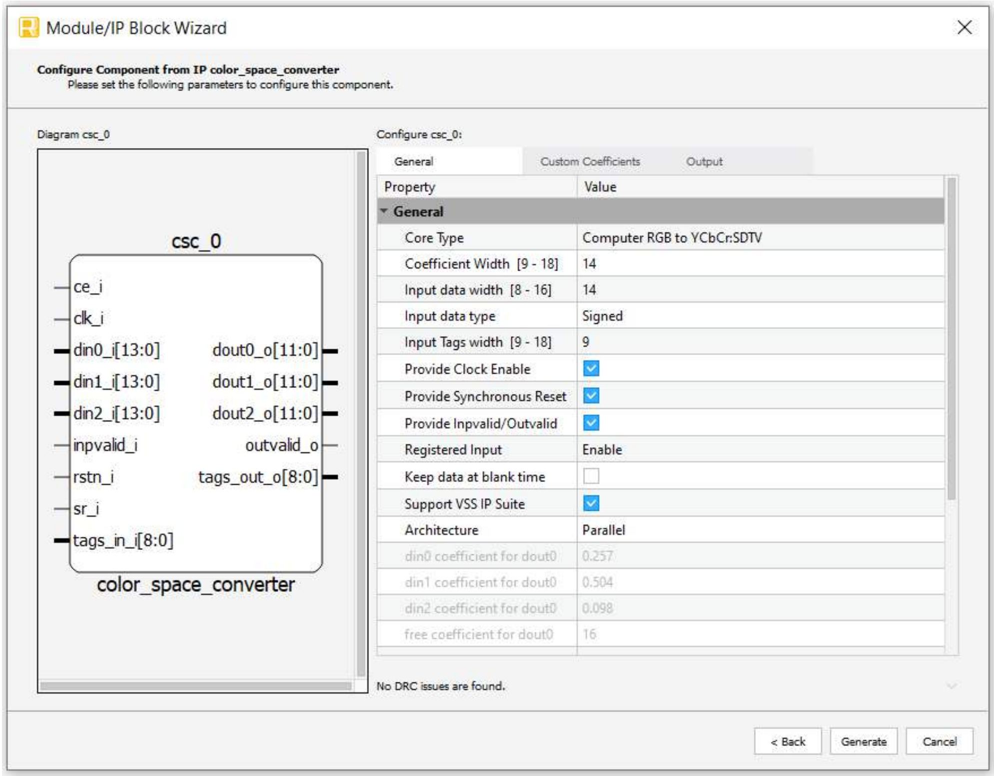

Figure 3.2. Configure User Interface of Color Space Converter IP Core

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

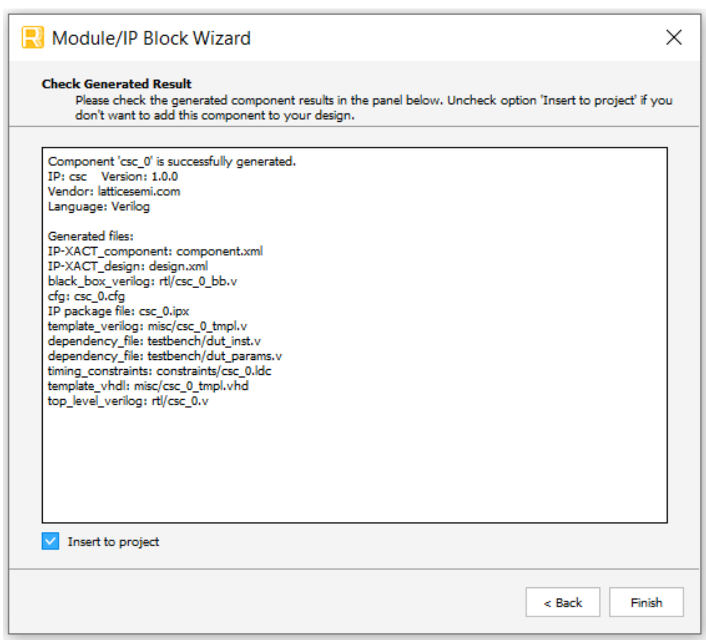

Figure 3.3. Check Generating Result

5. Click the **Finish** button. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 3.1.

The generated Color Space Converter IP Core package includes the black 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 attribute values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration attributes 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 black box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	These files provide instance templates for the IP core.

3.3. Running Functional Simulation

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

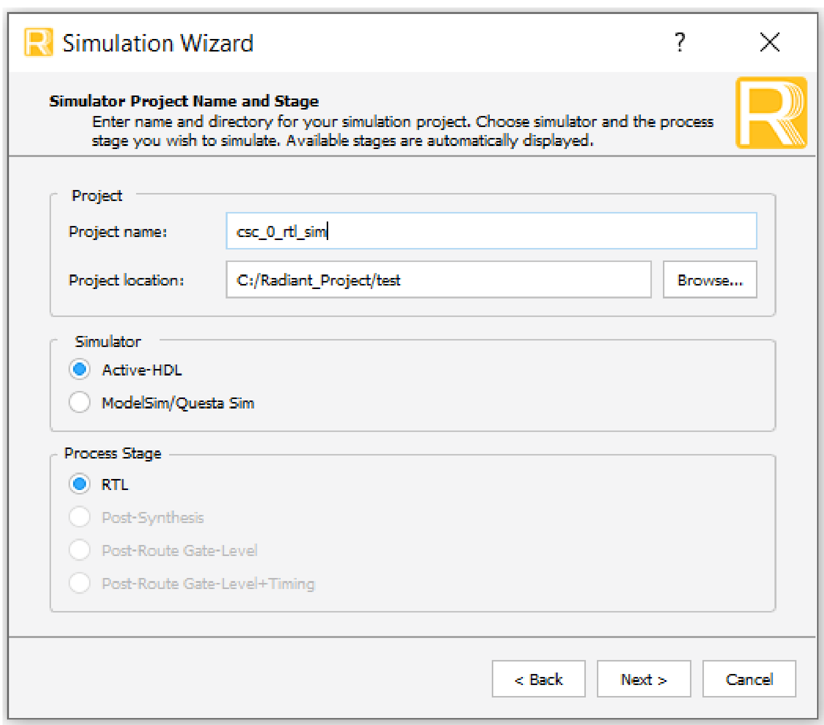

Figure 3.4. Simulation Wizard

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

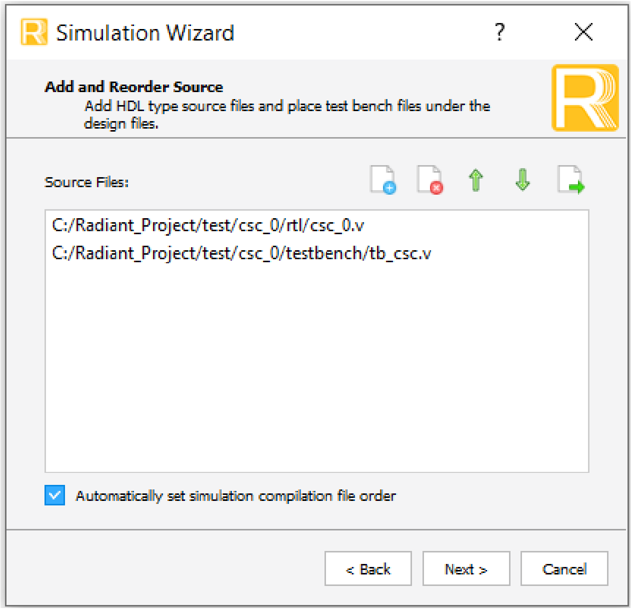

Figure 3.5. Adding and Reordering Source

7. Click **Next**. The **Summary** window is shown. Click **Finish** to run the simulation.

Note: It is necessary to follow the procedure above until it is fully automated in the Lattice Radiant Software Suite.

The results of the simulation in our example are provided in Figure 3.6.

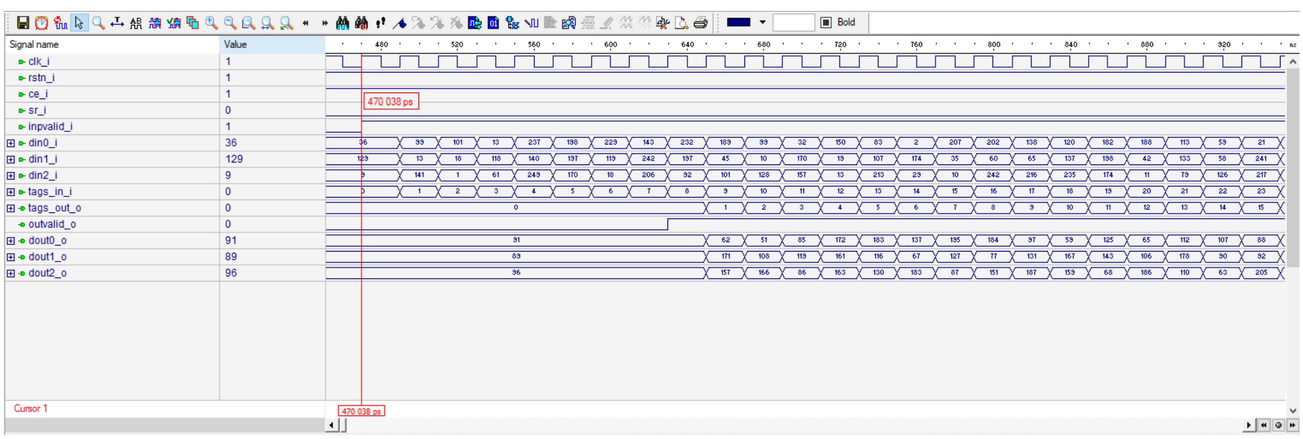

Figure 3.6. Simulation Waveform

3.4. Hardware Evaluation

The Color Space Converter IP Core supports Lattice's IP hardware evaluation capability when used with LIFCL devices. This makes it possible to create versions of the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default. To change this setting, go to Project > Active Strategy > LSE/Synplify Pro Settings.

4. Ordering Part Number

The Ordering Part Number (OPN) for this IP Core are the following:

- CSC-CNX-U – Color Space Converter for CrossLink-NX - Single Design License
- CSC-CNX-UT – Color Space Converter for CrossLink-NX - Site License
- CSC-CTNX-U – Color Space Converter for Certus-NX - Single Design License
- CSC-CTNX-UT – Color Space Converter for Certus-NX - Site License

Appendix A. Resource Utilization

Table A.1 shows the resource utilization of the Color Space Converter IP Core for the LIFCL-40-9BG400I device, using Lattice Synthesis Engine of Lattice Radiant Software. Default configuration is used and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.1. Resource Utilization

Configuration	Clk Fmax (MHz)*	Slice Registers	LUTs	DSP Slices
Default	200 MHz	357	553	3
Architecture = Sequential, Others = Default	200 MHz	273	320	1
Core Type = YUV to Computer RGB, Others = Default	200 MHz	283	393	1
Coefficient Width = 18, *Input Tags Width* = 10, *Provide Synchronous* Reset = Unchecked, Others = Default	200 MHz	336	324	3

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

References

For complete information on Lattice Radiant Project-Based Environment, Design Flow, Implementation Flow and Tasks, as well as on the Simulation Flow, see the Lattice Radiant Software 2.1 User Guide.

Technical Support Assistance

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

Revision History

Document Revision 1.1, June 2020

Section	Change Summary
Introduction	Updated Table 1.1. • Added Certus-NX support. • Updated Table 1.1 to add LFD2NX-40 as targeted device. • Updated Synopsis Synplify Pro version. • Updated Lattice Implementation to Lattice Radiant 2.1.
Attribute Summary	Changed selectable values for *Registered Input.*
Ordering Part Number	Added this section.
Appendix A. Resource Utilization	Updated device to LIFCL-40-9BG400I.

Document Revision 1.0, February 2020

Section	Change Summary
All	Initial release.

SEMICONDUCTOR

www.latticesemi.com