LatticeCORE

Color Space Converter IP Core User Guide

April 2015
IPUG63_2.0

Table of Contents

Introduction

Color Space Converters (CSC) are used in video and image display systems including televisions, computer monitors, color printers, video telephony and surveillance systems. CSCs 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.

A CSC converts 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.

The Lattice CSC 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.

Quick Facts

Table 1-1 through Table 1-3 give quick facts about the CSC IP core for LatticeECP3™, MachXO2™ and ECP5 devices. The configurations indicated in Table 1-1 through Table 1-3 correspond to the parameter selections given in Table A-1.

Table 1-1. CSC IP Core for LatticeECP3 Devices – Quick Facts

		CSC Configuration			
		Config1	**Config2**	**Config3**	**Config4**
Core Requirements	FPGA Families Supported	LatticeECP3			
	Minimum Device Needed	LFE3-17EA-6FN484CES			
Resource Utilization	Targeted Device	LFE3-150EA-6FN1156C			
	Data Path Width	8	8	16	8
	LUT4s	259	98	835	493
	sysMEM™ EBRs	0	0	0	0
	Registers	336	185	487	300
	MULT18x18C	9	3	5	9
	MULT9x9C	0	0	0	0
Design Tool Support	Lattice Implementation	Lattice Diamond® 3.4			
	Synthesis	Synopsys Synplify Pro for Lattice J-2014.09L			
	Simulation	Aldec Active-HDL 9.3 SPI Lattice Edition			
		Mentor Graphics ModelSim SE6.6e or later			

Table 1-2. CSC IP Core for MachXO2 Devices – Quick Facts

		CSC Configuration			
		Config1	**Config2**	**Config3**	**Config4**
Core Requirements	FPGA Families Supported	MachXO2			
	Minimum Device Needed	LCMXO2-1200HC-4TG100C			
Resource Utilization	Targeted Device	LCMXO2-4000HE-5FG484C			
	Data Path Width	8	8	16	8
	LUT4s	1633	643	2645	1663
	sysMEM EBRs	0	0	0	0
	Registers	798	368	835	625
	MULT18x18	0	0	0	0
	MULT9x9	0	0	0	0
Design Tool Support	Lattice Implementation	Lattice Diamond 3.4			
	Synthesis	Synopsys Synplify Pro for Lattice J-2014.09L			
	Simulation	Aldec Active-HDL 9.3 SPI Lattice Edition			
		Mentor Graphics ModelSim SE6.6e or later			

Table 1-3. CSC IP Core for ECP5UM Devices – Quick Facts

		CSC Configuration			
		Config1	**Config2**	**Config3**	**Config4**
Core Requirements	FPGA Families Supported	ECP5U/UM			
	Minimum Device Needed	LFE5UM-25F-6MG285C			
Resource Utilization	Targeted Device	LFE5UM-85F-7BG756C			
	Data Path Width	8	8	16	8
	LUT4s	259	251	835	493
	sysMEM EBRs	0	0	0	0
	Registers	336	185	487	300
	MULT18x18	9	3	5	9
	MULT9x9	0	0	0	0
Design Tool Support	Lattice Implementation	Lattice Diamond 3.4			
	Synthesis	Synopsys Synplify Pro for Lattice J-2014.09L			
	Simulation	Aldec Active-HDL 9.3 SPI Lattice Edition			
		Mentor Graphics ModelSim SE6.6e or later			

Features

- Input data width of 8, 10, 12, and 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
- Configurable DSP block based or look-up-table (LUT) based multiplier implementations
- Registered input option available for input set-up time improvement.

Functional Description

This chapter provides a functional description of the CSC IP core.

Color Spaces

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 a RGB color space where each pixel's color is represented by the constituent red, green and blue components. This color space is a natural choice for computer displays where the CRT uses these colors to display a multi-colored pixel. However, a RGB color space may not be the ideal one for image processing or 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, green color 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.

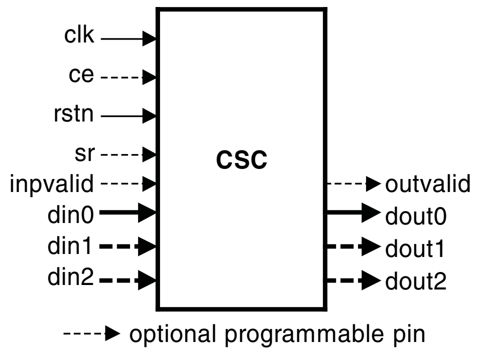
The following are some commonly used color spaces:

- **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.

Block Diagram

The top-level interface diagram for the CSC IP is given in Figure 2-1.

Figure 2-1. Top-level Interface Diagram for the CSC IP Core



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 given by 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 given by the following equations.

$$R' = 1.164 * Y' +0.0 * Cb + 1.596 * Cr -222.912$$
$$G' = 1.164 * Y' -0.392 * Cb -0.813 * Cr + 135.616$$
$$B' = 1.164 * Y' + 2.017 * Cb + 0.0 * Cr -276.8$$

Example applications that use CSC for R’G’B’ to Y’CbCr Conversion and Y’CbCr to R’G’B’ conversion are shown in Figure 2-2 and Figure 2-3.

Figure 2-2. JPEG Encoding Application

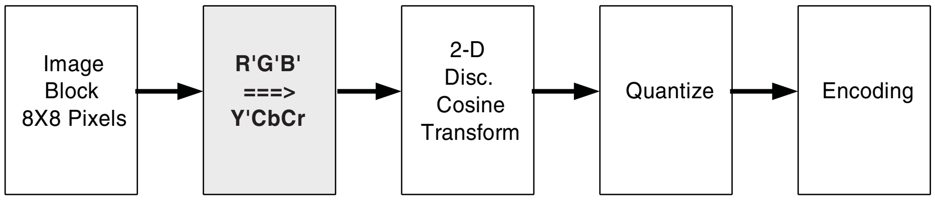
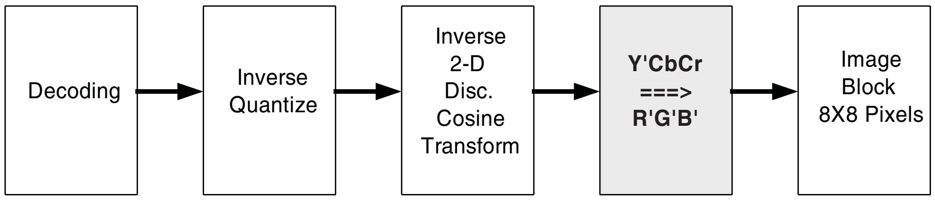


Figure 2-3. JPEG Decoding Application



CSC Implementation

The Color Space Converter is implemented using multipliers and adders operating on input pixel data and a set of coefficients defined for that conversion. The general color space conversion equations can be expressed by the following matrix multiplication.

$$\begin{bmatrix} dout0 \\ dout1 \\ dout2 \end{bmatrix} = \begin{bmatrix} cMH & cMI & cMJ & cMK \\ cNH & cNI & cNJ & cNK \\ cPH & cPI & cPJ & cPK \end{bmatrix} = \begin{bmatrix} din0 \\ din1 \\ din2 \\ 1 \end{bmatrix}$$

The pixel values of the input color space, din0, din1 and din2 are read through the input ports. The constants denoted by cMH,cMI, ... , cPK, are the coefficients used for the color space conversion. These coefficients are either provided by the user or automatically determined by the IP GUI for standard conversions. The values of the pixel components in the converted color space are available through the output ports: dout0, dout1 and dout2.

The CSC IP offers the choice of two different architectures: parallel and sequential. In the parallel architecture, all three 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 few clock cycles. In the sequential architecture, the input data for the three 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 few clock cycles.

Standard and Custom Core Types

Table 2-1 lists the standard configurations available in the CSC IP GUI and their coefficient values.

Table 2-1. Coefficients for Standard Configurations

Core Type		*din0	*din1	*din2	+
Computer R_G_B__to Y_CbCr:SDTV	dout0	0.257	0.504	0.098	16.0
	dout1	-0.148	-0.291	0.439	128.0
	dout2	0.439	-0.368	-0.071	128.0
Computer R_G_B__to Y_CbCr:HDTV	dout0	0.183	0.614	0.062	16.0
	dout1	-0.101	-0.338	0.439	128.0
	dout2	0.439	-0.399	-0.04	128.0
Studio R_G_B__to Y_CbCr:SDTV	dout0	0.299	0.587	0.114	0.0
	dout1	-0.172	-0.339	0.511	128.0
	dout2	0.511	-0.428	-0.083	128.0
Studio R_G_B__to Y_CbCr:HDTV	dout0	0.213	0.715	0.072	0.0
	dout1	-0.117	-0.394	0.511	128.0
	dout2	0.511	-0.464	-0.047	128.0
Y_CbCr:SDTV to Computer R_G_B_	dout0	1.164	0.0	1.596	-222.912
	dout1	1.164	-0.391	-0.813	135.488
	dout2	1.164	2.018	0.0	-276.928
Y_CbCr:HDTV to Computer R_G_B_	dout0	1.164	0.0	1.793	-248.128
	dout1	1.164	-0.213	-0.534	76.992
	dout2	1.164	2.115	0.0	-289.344
Y_CbCr:SDTV to Studio R_G_B_	dout0	1.0	0.0	1.371	-175.488
	dout1	1.0	-0.336	-0.698	132.352
	dout2	1.0	1.732	0.0	-221.696
Y_CbCr:HDTV to Studio R_G_B_	dout0	1.0	0.0	1.54	-197.12
	dout1	1.0	-0.183	-0.459	82.176
	dout2	1.0	1.816	0.0	-232.448
Y_UV to Computer R_G_B_	dout0	1.0	0.0	1.14	0.0
	dout1	1.0	-0.395	-0.581	0.0
	dout2	1.0	-2.032	0.0	0.0
Computer R_G_B__to Y_UV	dout0	0.299	0.587	0.114	0.0
	dout1	-0.147	-0.289	0.436	0.0
	dout2	0.615	-0.515	-0.1	0.0
Y_IQ to Computer R_G_B__	dout0	1.0	0.956	0.621	0.0
	dout1	1.0	-0.272	-0.647	0.0
	dout2	1.0	-1.107	1.704	0.0
Computer R_G_B__to Y_IQ	dout0	0.299	0.587	0.114	0.0
	dout1	0.596	-0.275	-0.321	0.0
	dout2	0.212	-0.523	0.311	0.0

Table 2-1. Coefficients for Standard Configurations (Continued)

Core Type		*din0	*din1	*din2	+
Y_IQ to Y_UV	dout0	1.0	0.0	0.0	0.0
	dout1	0.0	-0.544639	0.838671	0.0
	dout2	0.0	0.838671	0.544639	0.0

Notes:
For RGB to YCbCr: din0/din1/din2/dout0/dout1/dout2 stands for R/G/B/Y/Cb/Cr accordingly.
For YCbCr to RGB: din0/din1/din2/dout0/dout1/dout2 stands for Y/Cb/Cr/R/G/B accordingly.
For YUV to RGB: din0/din1/din2/dout0/dout1/dout2 stands for Y/U/V/R/G/B accordingly.
For RGB to YUV: din0/din1/din2/dout0/dout1/dout2 stands for R/G/B/Y/U/V accordingly.
For YIQ to RGB: din0/din1/din2/dout0/dout1/dout2 stands for Y/I/Q/R/G/B accordingly.
For RGB to YIQ: din0/din1/din2/dout0/dout1/dout2 stands for R/G/B/Y/I/Q accordingly.
For YIQ to YUV: din0/din1/din2/dout0/dout1/dout2 stands for Yin/I/Q/Yout/U/V accordingly.

When a Core type is selected as Custom, the user must manually enter the coefficient values in the GUI.

Full Precision Outputs

The full precision output width is given by the following sum.

Fullwidth = Input Data Width + Coefficient Width + 2

When output data width Owidth is configured as Fullwidth, then the binary point position Bpoint in this case tells the scale factor for all the coefficients. Coefficients are multiplied by 2^Bpoint before quantization.

When output data width Owidth is less than (Fullwidth-Bpoint), the LSB of the output data is the (Bpoint-1)th bit of the full width output data, and the MSB of the output data is the (Owidth+Bpoint-1)th bit of the full width output data. In this case, binary point position is zero.

When output data width Owidth is larger than (Fullwidth-Bpoint), the LSB of the output data is the (Fullwidth-Owidth-1)th bit of the full width output data, and the MSB of the output data is the (Fullwidth-1)th bit of the full width output data. In this case, binary point position is (Bpoint+Owidth-Fullwidth).

Limited Precision Outputs

The output setting can be used to configure the output data type and width. The output data type can be either configured as “Signed” or “Unsigned”.

The following options are supported to deal with LSBs rounding:

- **None** – Discards all bits to the right of the output least significant bit and leaves the output uncorrected.
- **Rounding up** – Rounds up if the fractional part is exactly one-half (e.g. 2.5 will be rounded to 3, -2.5 will be rounded to -2).
- **Rounding away from zero** – Rounds away from zero if the fractional part is exactly one-half (e.g. 2.5 will be rounded to 3, -2.5 will be rounded to -3).
- **Rounding towards zero** – Rounds towards zero if the fractional part is exactly one-half (e.g. 2.5 will be rounded to 2, -2.5 will be rounded to -2).
- **Convergent rounding** – Rounds to the nearest even value if the fractional part is exactly one-half (e.g. 2.5 will be rounded to 2, -2.5 will be rounded to -2, 3.5 will be rounded to 4, -3.5 will be rounded to -4).

The following options are supported to deal with MSBs overflow:

- **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.

Signal Descriptions

The I/O port definitions are given in Table 2-2.

Table 2-2. Interface Signal Description

Port	Bits	I/O	Description
All Configurations			
clk	1	I	Reference clock for input and output data.
rstn	1	I	System-wide asynchronous active-low reset signal.
din0	8 - 16	I	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	5 - 35	O	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.
When Parallel Architecture is Selected			
din1	8 - 16	I	Input data for second color plane.
din2	8 - 16	I	Input data for third color plane.
dout1	5 - 35	O	Output data for second color plane.
dout2	5 - 35	O	Output data for third color plane. This port is always enabled when the parallel architecture is selected.
Valid Signals			
inpvalid	1	I	Input data valid. Indicates valid data is present on din0 (also on din1 and din2 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 used to generate the outvalid 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 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 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	1	O	Output data valid. Indicates valid data is present on dout0 (also on dout1 and dout2 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 is low.
Optional I/Os			
ce	1	1	Clock Enable. While this is de-asserted, the core will ignore all other synchronous inputs and maintain its current state.
sr	1	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 goes low.

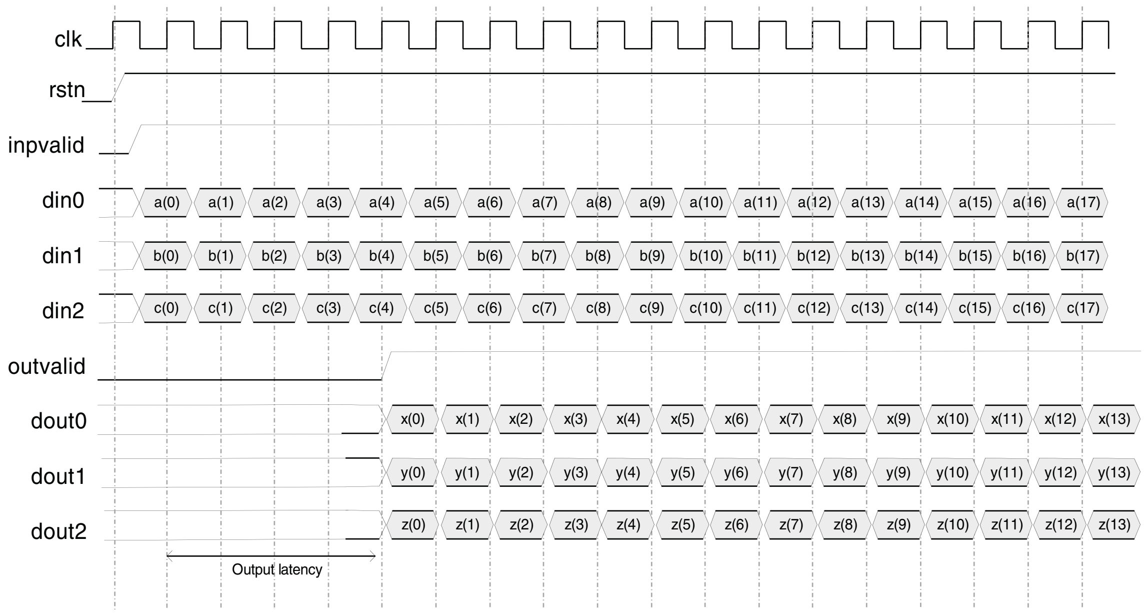
Timing Diagrams

Parallel Architecture Timing

Figure 2-4 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, din1 and din2.

The signal inpvalid 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, dout1 and dout2. The signal outvalid is asserted to indicate valid output data present on the output ports.

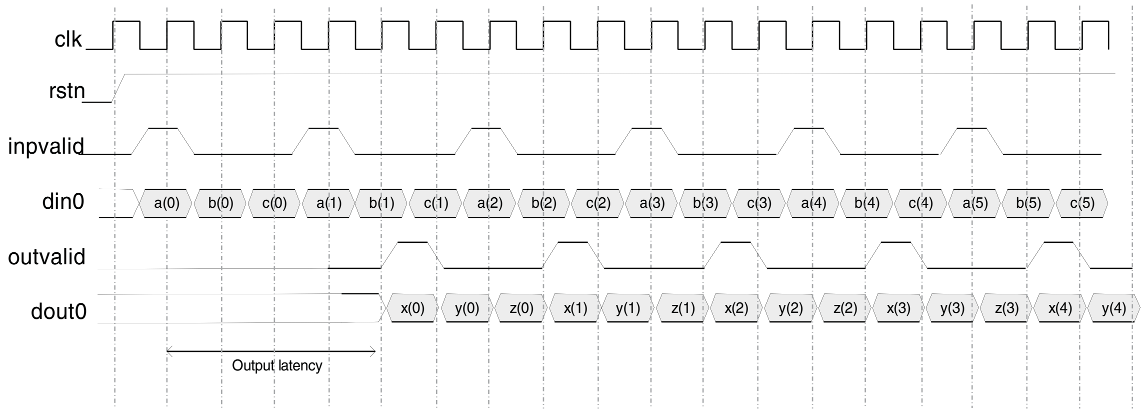
Figure 2-4. Parallel Architecture


clk
rstn
inpvalid
din0
a(0) a(1) a(2) a(3) a(4) a(5) a(6) a(7) a(8) a(9) a(10) a(11) a(12) a(13) a(14) a(15) a(16) a(17)
din1
b(0) b(1) b(2) b(3) b(4) b(5) b(6) b(7) b(8) b(9) b(10) b(11) b(12) b(13) b(14) b(15) b(16) b(17)
din2
c(0) c(1) c(2) c(3) c(4) c(5) c(6) c(7) c(8) c(9) c(10) c(11) c(12) c(13) c(14) c(15) c(16) c(17)
outvalid
dout0
x(0) x(1) x(2) x(3) x(4) x(5) x(6) x(7) x(8) x(9) x(10) x(11) x(12) x(13)
dout1
y(0) y(1) y(2) y(3) y(4) y(5) y(6) y(7) y(8) y(9) y(10) y(11) y(12) y(13)
dout2
z(0) z(1) z(2) z(3) z(4) z(5) z(6) z(7) z(8) z(9) z(10) z(11) z(12) z(13)
Output latency

Sequential Architecture Timing

Figure 2-5 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. The signal inpvalid is asserted to indicate the first color plane data on din0. In the following two cycles, the second and third color plane data are applied on din0. After a latency of a few cycles the output data for the first color plane appears on the output port dout0. The signal outvalid is asserted to indicate the first color plane data on dout0. In the following two cycles, the second and third color plane data appear on dout0.

Figure 2-5. Sequential Architecture


clk
rstn
inpvalid
din0
a(0) b(0) c(0) a(1) b(1) c(1) a(2) b(2) c(2) a(3) b(3) c(3) a(4) b(4) c(4) a(5) b(5) c(5)
outvalid
dout0
x(0) y(0) z(0) x(1) y(1) z(1) x(2) y(2) z(2) x(3) y(3) z(3) x(4) y(4)
Output latency

Parameter Settings

The IPexpress™ or Diamond is used to create IP and architectural modules in the Diamond software. Refer to the IP Core Generation section for a description of how to generate the IP.

Table 3-1 provides the list of user-configurable parameters for the CSC IP core. The parameter settings are specified using the CSC IP core Configuration GUI in IPexpress.

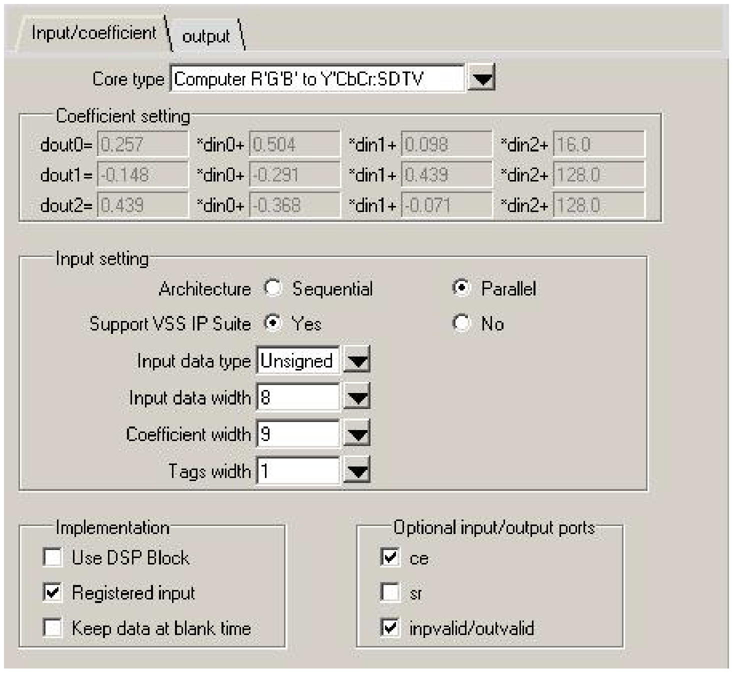
Table 3-1. Parameter Descriptions

Parameters	Range/Options	Default Value
Core Type		
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
Input Settings		
Architecture	{Sequential, Parallel}	Sequential
Input data type	{Signed, Unsigned}	Unsigned
Input data width	{8, 10, 12, 16}	8
Coefficient precision width	9-18	18
Implementation		
Registered input	Yes or No	Yes
Optional Input/Output Ports		
ce	Yes or No	No
sr	Yes or No	No
inpvalid/outvalid	Yes or No	No
Output Settings		
Output Data Type	Signed, Unsigned	Unsigned
Output Data Width	5-36	8
Overflow	Saturation, Wrap-around	Saturation
Rounding	None, Rounding up, Rounding away from zero, Rounding toward zero, convergent rounding	None

Input/Coefficient Tab

Figure 3-1 shows the content of the Input/Coefficient tab.

Figure 3-1. Input/Coefficient Tab



Input/coefficient | output

Core type: Computer R'G'B' to Y'CbCr:SDTV

Coefficient setting

dout0=	0.257	*din0+	0.504	*din1+	0.098	*din2+	16.0
dout1=	-0.148	*din0+	-0.291	*din1+	0.439	*din2+	128.0
dout2=	0.439	*din0+	-0.368	*din1+	-0.071	*din2+	128.0

Input setting

Architecture: Sequential [✓ Parallel]

Support VSS IP Suite: [✓ Yes] No

Input data type: Unsigned

Input data width: 8

Coefficient width: 9

Tags width: 1

Implementation

- ☐ Use DSP Block
- ✓ Registered input
- ☐ Keep data at blank time

Optional input/output ports

- ✓ ce
- ☐ sr
- ✓ inpvalid/outvalid

Core Type

Selects between custom and pre-defined standard configurations.

Architecture

Selects between parallel and sequential implementation architectures.

Input Data Type

Signed or unsigned input data type

Input Data Width

The bit width for the input color planes.

Coefficient Width

The bit width for the input color planes.

Use DSP Block

If this option is checked, the core will use the DSP Block for implementation (not applicable for LatticeSC/M).

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.

Keep Data at Blank Time

This option is to keep the auxiliary data of the Video stream unchanged during blank time.

Connect Reset Port to GSR

If this option is checked, the GSR is instantiated and used to route the CSC's rstn input. Using GSR improves the utilization and performance of the CSC IP. However, if GSR is used an active input in rstn will reset most of the FPGA components as well. This option must be checked to enable the hardware evaluation capability for this IP.

ce

Input port ce is added to the core when checked.

sr

Input port sr is added to the core when checked.

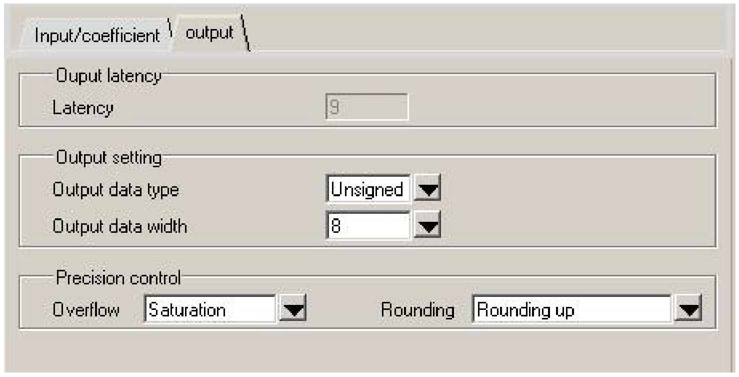
inpvalid/outvalid

If Architecture is selected as Sequential, then this is always checked. If architecture is selected as Parallel, then this is optional. If checked, this option will add inpvalid and outvalid ports to the core.

Output Tab

Figure 3-2 shows the content of the Output/Coefficient tab.

Figure 3-2. Output/Coefficient Tab



Ouput latency
Latency 9

Output setting
Output data type Unsigned
Output data width 8

Precision control
Overflow Saturation
Rounding Rounding up

Output Latency

This provides the output latency for the selected core configuration.

Output Data Type

Signed or unsigned output data type

Output Data Width

The bit width for the output color planes

Overflow

This option allows the user to specify what kind of overflow control is to be used. This parameter is available whenever there is a need to drop some of the MSBs from the true output. If the selection is Saturation, the output value is clipped to the maximum, if positive or minimum, if negative, while discarding the MSBs. If the selection is Wrap-around, the MSBs are simply discarded without making any correction.

Rounding

This option allows the user to specify the rounding method when there is a need to drop one or more LSBs from the true output.

IP Core Generation

This chapter provides information on how to generate the Lattice CSC IP core using the Diamond software 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 CSC IP core in a complete, top-level design. Instructions on how to obtain licenses for Lattice IP cores are given at:

http://www.latticesemi.com/products/intellectualproperty/aboutip/isplevercoreonlinepurchas.cfm

Users may download and generate the CSC IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The CSC IP core also supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core that operate in hardware for a limited time (several hours) without requiring an IP license. See the Hardware Evaluation section for further details. However, a license is required to enable timing simulation, to open the design in the Diamond EPIC tool, and to generate bitstreams that do not include the hardware evaluation timeout limitation.

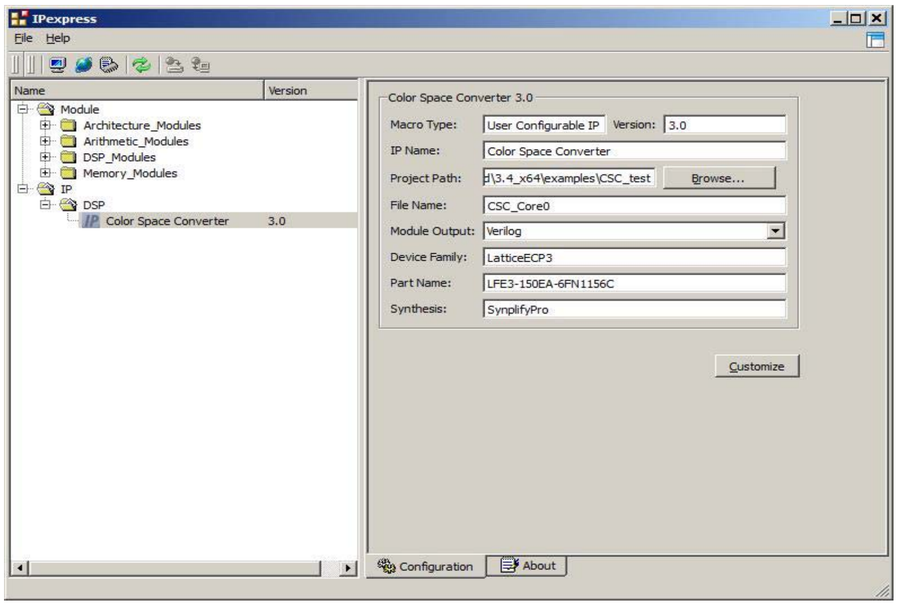
Getting Started

The CSC IP core is available for download from the Lattice IP server using the IPexpress tool. The IP files are automatically installed using InstallShield 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.

The Diamond IPexpress tool GUI dialog box for the CSC IP core is 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 loaded.
- **File Name** – "username" designation given to the generated IP core and corresponding folders and files.
- **(Diamond) Module Output** – Verilog or VHDL.
- **Device Family** – Device family to which IP is to be targeted (e.g. MachXO2, ECP5, LatticeECP3, etc.). Only families that support the particular IP core are listed.
- **Part Name** – Specific targeted part within the selected device family.

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



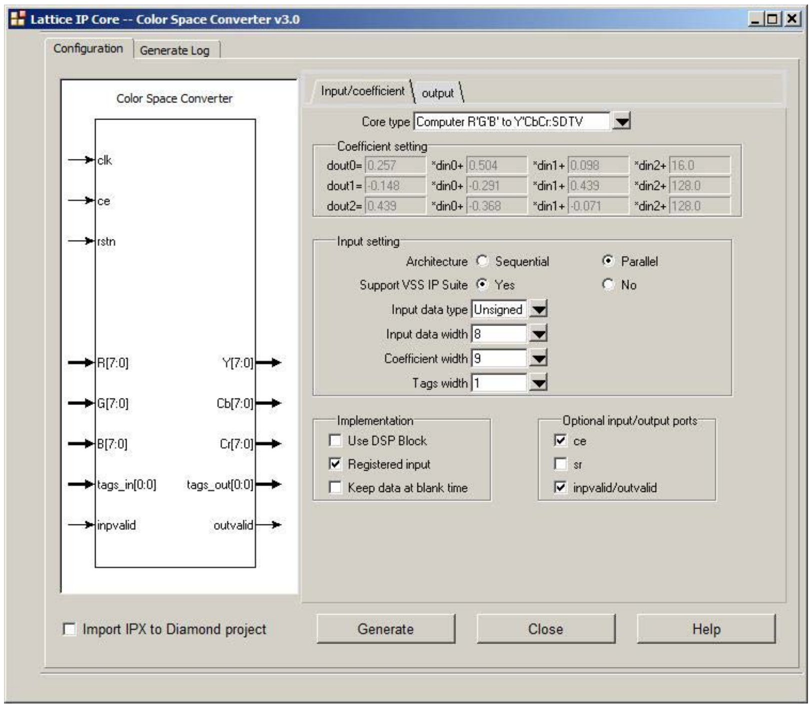
Color Space Converter 3.0

Macro Type: User Configurable IP Version: 3.0
IP Name: Color Space Converter
Project Path: d\3.4_x64\examples\CSC_test
File Name: CSC_Core0
Module Output: Verilog
Device Family: LatticeECP3
Part Name: LFE3-150EA-6FN1156C
Synthesis: SynplifyPro

Note that if the IPexpress tool is called from within an existing project, Project Path, Module Output, 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 CSC 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 the Parameter Settings section for more information on the CSC IP core parameter settings.

Figure 4-2. Configuration GUI (Diamond Version)



Core type Computer R'G'B' to Y'CbCr:SDTV

Coefficient setting

dout0= 0.257	*din0+ 0.504	*din1+ 0.098	*din2+ 16.0
dout1= -0.148	*din0+ -0.291	*din1+ 0.439	*din2+ 128.0
dout2= 0.439	*din0+ -0.368	*din1+ -0.071	*din2+ 128.0

Input setting

Architecture: Sequential [✓ Parallel]
Support VSS IP Suite: [✓ Yes] No
Input data type: Unsigned
Input data width: 8
Coefficient width: 9
Tags width: 1

Implementation

- ☐ Use DSP Block
- ☑ Registered input
- ☐ Keep data at blank time

Optional input/output ports

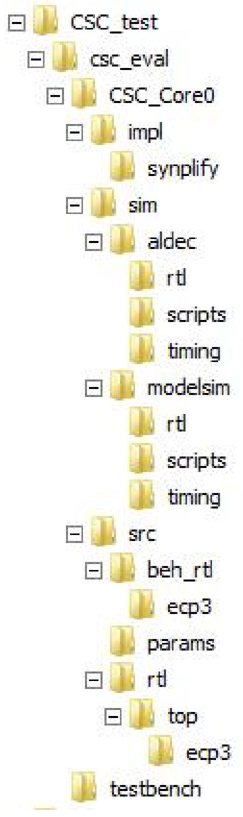
- ☑ ce
- ☐ sr
- ☑ inpvalid/outvalid

☐ Import IPX to Diamond project

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. LatticeECP3 CSC IP Core Directory Structure



```
CSC_test
  csc_eval
    CSC_Core0
      impl
        synplify
      sim
        aldec
          rtl
          scripts
          timing
        modelsim
          rtl
          scripts
          timing
      src
        beh_rtl
          ecp3
        params
        rtl
          top
            ecp3
    testbench
```

The design flow for IP created with the IPexpress tool uses a post-synthesized module (NGO) for synthesis and a protected model for simulation. The post-synthesized module is customized and created during the IPexpress tool generation. The protected simulation model is not customized during the IPexpress tool process, and relies on parameters provided to customize behavior during simulation. 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	Sim	Synthesis Diamond	Description
*<username>*_inst.v			This file provides an instance template for the IP.
<username>.v	Yes		This file provides the CSC core for simulation.
*<username>*_beh.v	Yes		This file provides a behavioral simulation model for the CSC IP core.
*<username>*_bb.v		Yes	This file provides the synthesis black box for the user’s synthesis.
<username>.ngo *.ngo		Yes	The ngo files provide the synthesized IP core used by Diamond. This file needs to be pointed to by the Build step by using the search path property.

Table 4-1. File List

File	Sim	Synthesis Diamond	Description
<u*username*>.lpc			This file contains the IPexpress tool options used to recreate or modify the core in the IPexpress tool.
<*username*>_top.[v,vhd]	Optional	Optional	This file provides a module which instantiates the CSC core. This file can be easily modified for the user's instance of the CSC core. This file is located in the csc_eval/<username>/src/rtl/top/ directory.

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

Table 4-2. Additional Files

File	Description
<*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 CSC IP core package includes black-box (<*username*>_bb.v) and instance (<user-name>_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>\csc_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 CSC IP core is provided for Aldec Active-HDL (Verilog and VHDL) simulator, Mentor Graphics ModelSim simulator. The functional simulation includes a configuration-specific behavioral model of the CSC 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>\csc_eval\<username>\sim\modelsim\scripts`. The simulation script supporting Aldec evaluation simulation is provided in `\<project_dir>\csc_eval\<username>\sim\aldec\scripts`. Both ModelSim and Aldec simulation is supported via test bench files provided in `\<project_dir>\csc_eval\testbench`. Models required for simulation are provided in the corresponding \models folder. Users may run the Aldec evaluation simulation by doing the following:

1. Open Active-HDL.
2. Under the Tools tab, select **Execute Macro**.
3. Browse to folder `\<project_dir>\csc_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>\csc_eval\<username>\sim\modelsim\scripts`.
3. Under the Tools tab, select **Execute Macro** and execute the ModelSim "do" script shown.

***Note**: When the simulation completes, a pop-up window will appear asking “Are you sure you want to finish?” Answer **No** to analyze the results. Answering **Yes** closes ModelSim.*

Synthesizing and Implementing the Core in a Top-Level Design

The CSC IP core itself is synthesized and provided in NGO format when the core is generated through IPexpress. You may combine the core in your own top-level design by instantiating the core in your top-level file as described in the Instantiating the Core section and then synthesizing the entire design with either Synplify or Precision RTL Synthesis.

The following text describes the evaluation implementation flow for Windows platforms. The flow for Linux and UNIX platforms is described in the Readme file included with the IP core.

The top-level file <*userame*>_top.v is provided in `\<`*`project_dir`*`>`**`\csc_eval\`**`<`*`username`*`>`**`\src\rtl\top`**. Push-button implementation of the reference design is supported via the project file <*username*>.ldf located in `\<`*`project_dir`*`>`**`\csc_eval\`**`<`*`user-name`*`>`**`\impl\(synplify or precision)`**.

To use this project file in Diamond:

1. Choose **File** > **Open** > **Project**.
2. Browse to `\<`*`project_dir`*`>`**`\csc_eval\`**`<`*`username`*`>`**`\impl\synplify (or precision)`** in the Open Project dialog box.
3. Select and open <*username*>_.ldf. 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.

Hardware Evaluation

The CSC IP core supports Lattice’s 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 (several 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.

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 T**arget** 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 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**.

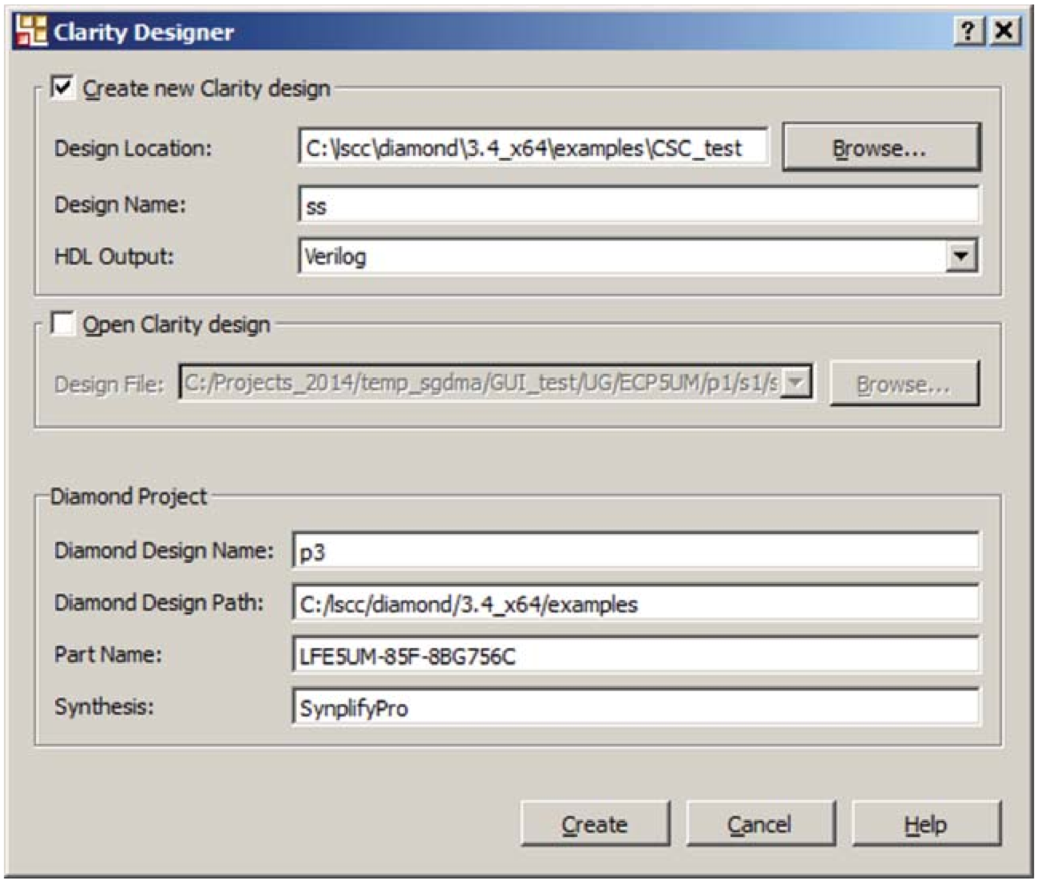
The IPexpress package file (.ipx) supported by Diamond holds references to all of the elements of the generated IP core required to support simulation, synthesis and implementation. The IP core may be included in a user's design by importing the .ipx file to the associated Diamond project. To change 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.

IP Core Generation in Clarity Designer

Getting Started

The first step in generating an IP Core in Clarity Designer is to start a project in Diamond software with the ECP5 device. Clicking the Clarity Designer button opens the Clarity Designer tool.

Figure 4-4. Starting a Project in Clarity Designer



[✓] Create new Clarity design

Design Location: C:\lscc\diamond\3.4_x64\examples\CSC_test

Design Name: ss

HDL Output: Verilog

[ ] Open Clarity design

Design File: C:/Projects_2014/temp_sgdma/GUI_test/UG/ECP5UM/p1/s1/s

Diamond Project

Diamond Design Name: p3

Diamond Design Path: C:/lscc/diamond/3.4_x64/examples

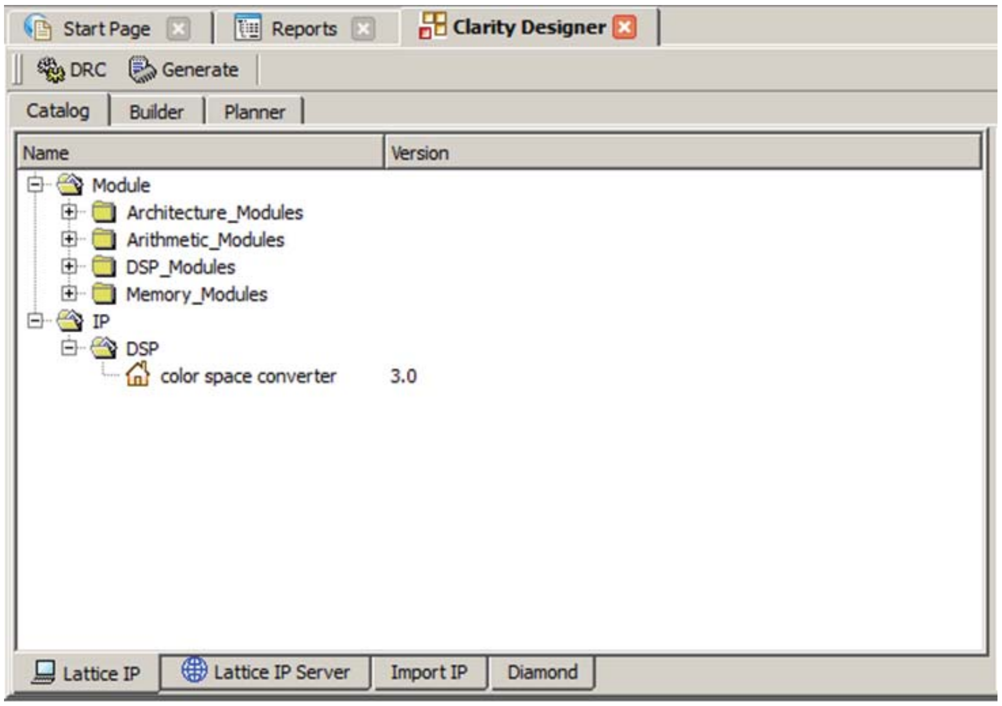
Part Name: LFE5UM-85F-8BG756C

Synthesis: SynplifyPro

As shown in Figure 4-4, you can create a new design or open an existed one. Specify the Design Location, Design Name and HDL Output format. Click **Create** to open the Clarity Designer main GUI window.

The CSC IP core is available for download from the Lattice IP Server using the Clarity Designer tool. The IP files are automatically installed using ispUPDATE technology in any customer-specified directory. After the IP core has been installed, the IP core will be available in the Clarity Designer GUI Catalog window as shown in Figure 4-5.

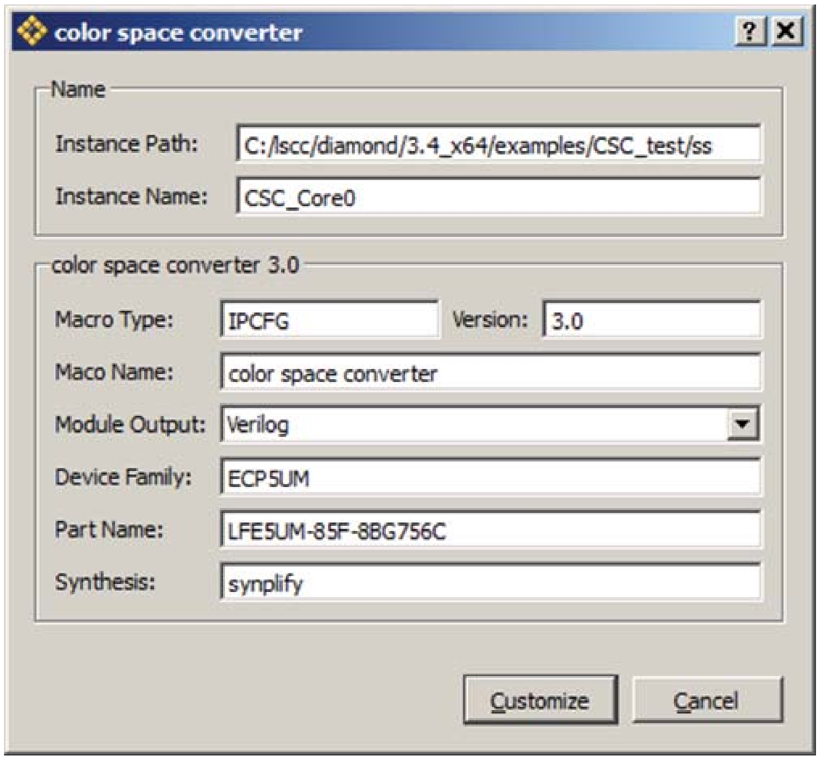
Figure 4-5. Clarity Designer Catalog Window



Name	Version
Module	
Architecture_Modules	
Arithmetic_Modules	
DSP_Modules	
Memory_Modules	
IP	
DSP	
color space converter	3.0

Double-click the IP name to open a dialog box where you can choose configuration options, as shown in Figure 4-6.

Figure 4-6. Clarity Designer Dialog Box



Instance Path: C:/lscc/diamond/3.4_x64/examples/CSC_test/ss
Instance Name: CSC_Core0
Macro Type: IPCFG
Version: 3.0
Maco Name: color space converter
Module Output: Verilog
Device Family: ECP5UM
Part Name: LFE5UM-85F-8BG756C
Synthesis: synplify

To generate a specific IP core configuration the user specifies:

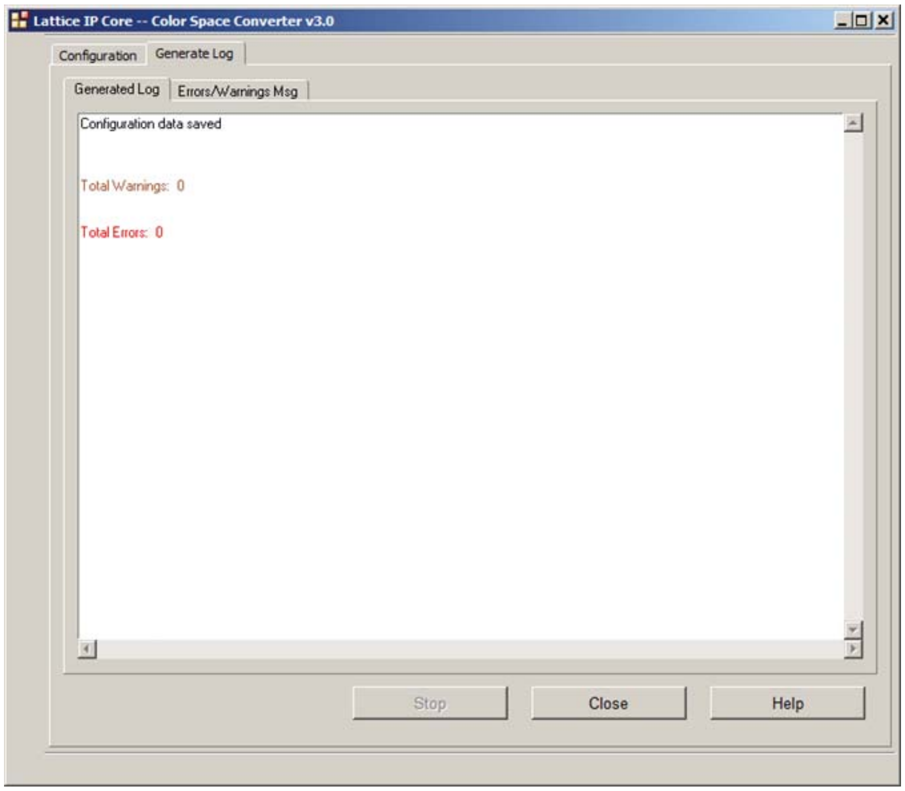
- **Instance Path** – Path to the directory where the generated IP files will be located.
- **Instance Name** – “username” designation given to the generated IP core and corresponding folders and files.
- **Module Output** – Verilog or VHDL.
- **Module Output** – Verilog HDL or VHDL.
- **Device Family** – Device family to which IP is to be targeted.
- **Part Name** – Specific targeted part within the selected device family.

Note that because the Clarity Designer tool must be called from within an existing project path, Module Output, 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:

1. Click the **Customize** button in the Clarity Designer dialog box to display the CSC IP core Configuration GUI, as shown in Figure 4-2.
2. Select the IP parameter options specific to your application. Refer to the Parameter Settings section for more information on the CSC IP core parameter settings.
3. After setting the parameters, click **Configure**.
4. A dialog box, shown in Figure 4-7, displays logs, errors and warnings. Click **Close**.

Figure 4-7. Clarity Designer Generate Log Tab



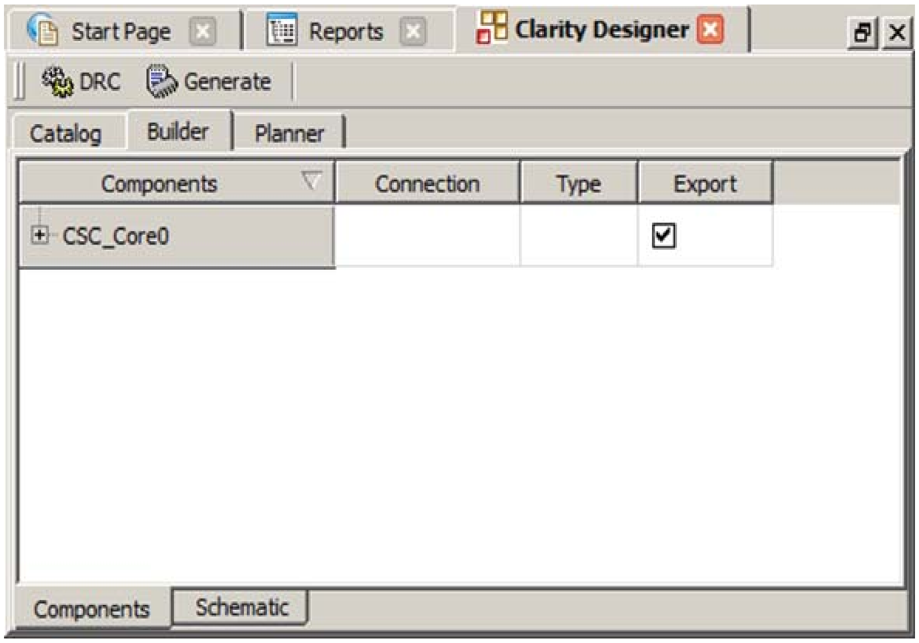
```
Configuration data saved

Total Warnings: 0

Total Errors: 0
```

5. The Clarity Designer Builder tab, shown in Figure 4-8, opens.

Figure 4-8. Clarity Designer Builder Tab

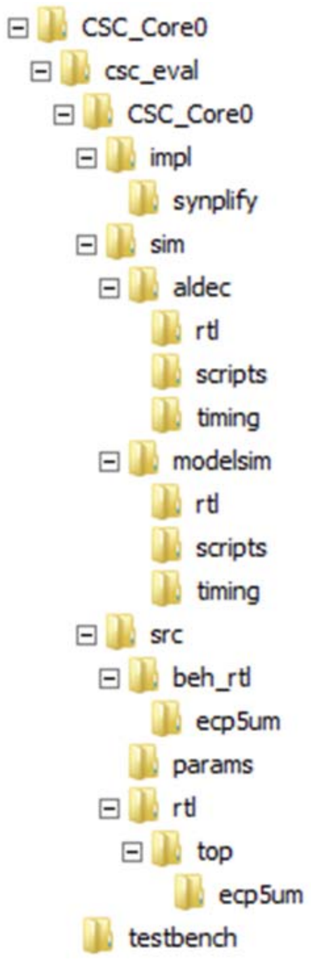


Components	Connection	Type	Export
CSC_Core0			☑

Clarity Designer Created Files and Top Level Directory Structure

The directory structure of the generated files is shown in Figure 4-9.

Figure 4-9. ECP5 SGDMAC IP Core Directory Structure



The design flow for IP created with the Clarity Designer tool uses post-synthesized modules (NGO) for synthesis and a protected model for simulation. The post-synthesized module are customized and created during the Clarity Designer tool generation.

Table 4-3 provides a list of key files and directories created by the Clarity Designer tool and how they are used. The Clarity Designer 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 Clarity Designer tool.

Table 4-3. File List

File	Description
<username>.v	This file provides the CSC core wrapper.
*<username>*_core.v	This file provides the CSC core for simulation.
*<username>*_beh.v	This file provides a behavioral simulation model for the CSC core.
*<username>*_core_bb.v	This file provides the synthesis black box for the user's synthesis.
*<username>*_core.ngo	The ngo files provide the synthesized IP core.
<username>.lpc	This file contains the IPexpress tool options used to recreate or modify the core in the IPexpress tool.

File	Description
<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>*_top.[v,vhd]	This file provides a module which instantiates the CSC core. This file can be easily modified for the user's instance of the CSC core. This file is located in the **`csc_eval/`***`<username>`***`/src`** directory.
generate_core.tcl	This file is created when GUI "Generate" button is pushed. This file may be run from command line.
*<username>*_generate.log	This is the IPexpress scripts log file.
*<username>*_gen.log	This is the IPexpress IP generation log file

Simulation Evaluation

Please refer to the Running Functional Simulation section for details.

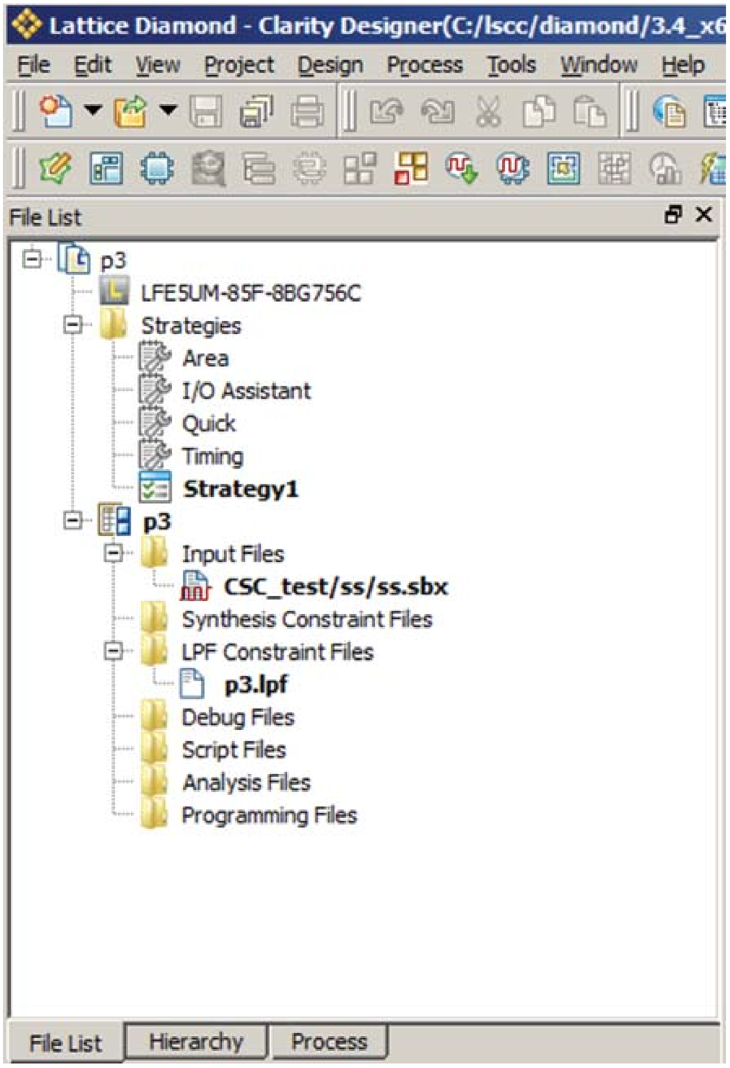
IP Core Implementation

After completing the Configuration step, click the **Generate** button, shown in Figure 4-8, to generate the Clarity Designer file (.sbx).

Clarity Designer (.sbx) files can be used in design projects such as an HDL file or an IPexpress generated (.ipx) file. A key difference between IPexpress generated files and Clarity Designer generated files is that the latter may contain not only a single block but multiple modules or IP blocks and may represent a subsystem. In IPexpress, the process generates a single module or IP. This is a one step process since an IPexpress file can only contain one module or IP. In Clarity Designer, saving a file is a separate step. Modules or IP are configured and multiple modules or IP can optionally be added within the same file. Additionally, since building and planning can also be done, saving the file and generating the blocks may be performed later.

After the Generate step is completed, the ".sbx" file is automatically added to current Diamond Project Input Files list as shown in Figure 4-10.

Figure 4-10. File List in Report Dialog Box



Lattice Diamond - Clarity Designer(C:/lscc/diamond/3.4_x6
File Edit View Project Design Process Tools Window Help
File List

```
p3
  LFE5UM-85F-8BG756C
  Strategies
    Area
    I/O Assistant
    Quick
    Timing
    Strategy1
  p3
    Input Files
      CSC_test/ss/ss.sbx
    Synthesis Constraint Files
    LPF Constraint Files
      p3.lpf
    Debug Files
    Script Files
    Analysis Files
    Programming Files
```

File List
Hierarchy
Process

After this step, click **Process** at the bottom of window, then double-click **Place & Route Design** to Start PAR. This is similar to a standard Diamond PAR flow.

Support Resources

This chapter contains information about Lattice Technical Support, additional references, and document revision history.

Lattice Technical Support

There are a number of ways to receive technical support.

E-mail Support

techsupport@latticesemi.com

Local Support

Contact your nearest Lattice sales office.

Internet

www.latticesemi.com

References

- Keith Jack, "Video Demystified", fourth edition, Elsevier, London, 2005.

LatticeECP3

- DS102, LatticeECP3 Family Handbook

MachXO2

- DS1035, MachXO2 Family Data Sheet

ECP5

- DS1044, ECP5 Family Data Sheet

Revision History

Date	Document Version	IP Version	Change Summary
April 2015	2.0	3.0	Added support for ECP5 and MachXO2 device families.
			Removed references to LatticeXP2™, LatticeECP2M™ and LatticeSC™ device families.
			Added IP Core Generation in Clarity Designer section.
August 2011	01.4	2.0	Utilization data updated to reflect improved coefficient quantization method and more efficient usage of the DSP slice. Added support for unsigned/signed output data type.
			Added support for five LSB rounding methods.
			IP port naming changed to reflect core type selected by user.
December 2010	01.3	1.4	Divided document into chapters.
			Added support for Lattice Diamond design software throughout.
June 2007	01.2	1.3	Updated appendices. Added support for LatticeXP2 FPGA family.
January 2007	01.1	1.2	Updated appendix for LatticeECP2 devices and added appendices for LatticeECP2M and LatticeSC/M FPGA families.
October 2006	01.0	1.1	Initial release.

Resource Utilization

This appendix gives resource utilization information for the CSC IP core in Lattice FPGAs.

IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the Diamond design tool. Details regarding the usage of IPexpress can be found in the IPexpress and Diamond help system. For more information on the Diamond design tool, visit the Lattice web site at: www.latticesemi.com/software.

Table A-1 lists the parameter selections for the example configurations shown in Table A-2 through Table A-4.

Table A-1. Example Configurations Available for CSC

	Config1	Config2	Config3	Config4
Coretype	computerRGB2 YCbCrSDTV	computerRGB2 YCbCrSDTV	YCbCrHDTV2 computerRGB	computerRGB2 YCbCrSDTV
Architecture	P	S	P	P
Input data type	Unsigned	Unsigned	Signed	signed
Input data width	8	8	16	8
Coefficient width	18	18	18	12
Use DSP Block	Y	Y	Y	Y
Registered input	Y	Y	Y	Y
Clk enable	N	N	Y	Y
SyncReset	N	N	Y	Y
output data type	Unsigned	Unsigned	Signed	signed
output data width	8	8	16	8

LatticeECP3 Devices

Table A-2. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	18x18 Multipliers	sysMEM EBRs	f_{MAX} (MHz)
config1	168	259	336	9	0	262
config2	93	98	185	3	0	197
config3	425	835	487	5	0	196
config4	253	493	300	9	0	220

1. Performance and utilization data are generated targeting an LFE3-150EA-6FN1156C device with Lattice Diamond 3.4 design software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP3 family.

Ordering Part Number

The Ordering Part Number (OPN) for the CSC targeting LatticeECP3 devices is CSC-E3-U1.

MachXO2 Devices

Table A-3. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	18x18 Multipliers	sysMEM EBRs	f_{MAX} (MHz)
config1	831	1633	798	0	0	73
config2	326	643	368	0	0	69
config3	1336	2645	835	0	0	54
config4	845	1663	625	0	0	69

1. Performance and utilization data are generated targeting an LCMXO2-4000HE-5FG484C device with Lattice Diamond 3.4 design software. Performance may vary when using a different software version or targeting a different device density or speed grade within the MachXO2 family.

Ordering Part Number

The Ordering Part Number (OPN) for the CSC targeting MachXO2 devices is CSC-M2-U1.

ECP5 Devices

Table A-4. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	18x18 Multipliers	sysMEM EBRs	f_{MAX} (MHz)
config1	168	259	336	9	0	185
config2	131	251	185	3	0	185
config3	425	835	487	5	0	185
config4	253	493	300	9	0	185

1. Performance and utilization data are generated targeting an LFE5UM-85F-8BG756C device with Lattice Diamond 3.4 design software. Performance may vary when using a different software version or targeting a different device density or speed grade within the ECP5 LFE5UM family.

Ordering Part Number

The OPN for the CSC targeting ECP5 devices is CSC-E5-U or CSC-E5-UT.