YCbCr 8-Bit to RGB565 Converter

April 2013 Reference Design RD1152

Introduction

YCbCr 8-bit to RGB565 converter converts YCbCr 4:2:2 8-bit color space information to RGB565 color space. To facilitate easy insertion to practical video systems, this design example takes video stream control signals (H_SYNC, V_SYNC, FID, and DEN) and delays them appropriately, so that control signals can be easily synchronized with the output video stream. This document provides a brief description of YCbCr 8-bit to RGB565 Converter and its implementation.

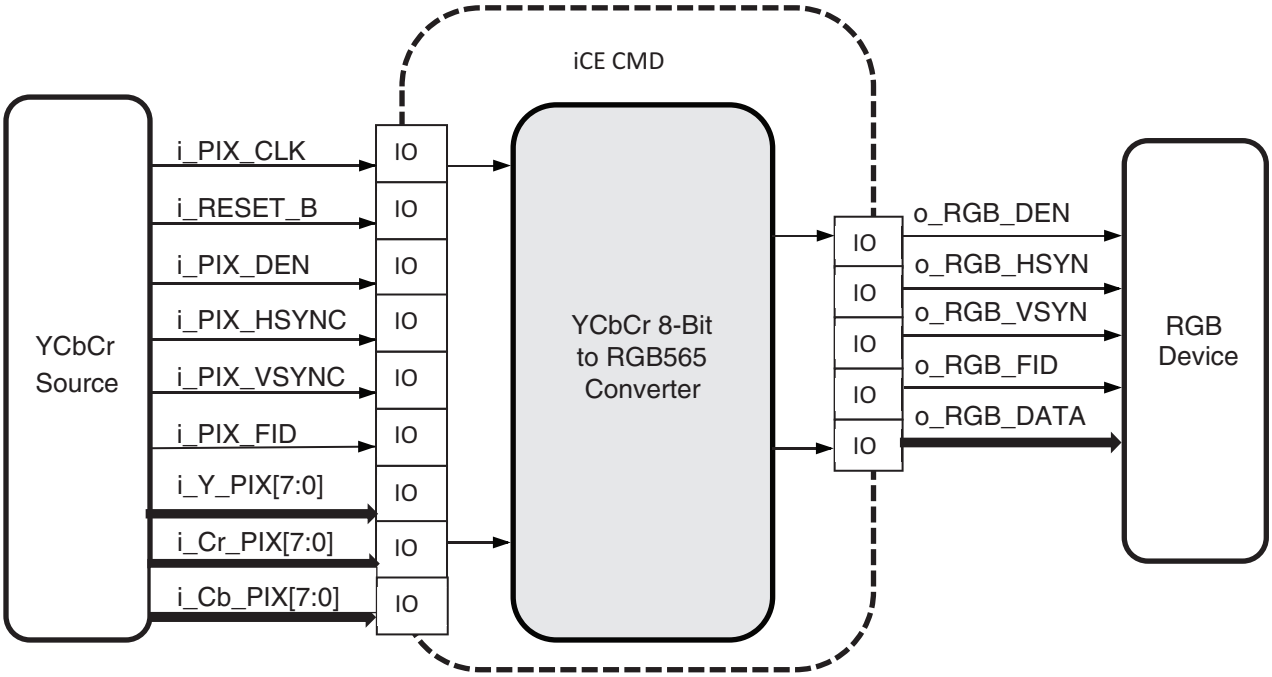
The design is implemented in VHDL. The Lattice iCEcube2™ Place and Route tool integrated with the Synopsys Synplify Pro® synthesis tool is used for the implementation of the design. The design can be targeted to other iCE40™ FPGA product family devices.

Features

- 8-bit YCbCr 4:2:2 input and RGB565 output
- Pipelined implementation
- Latency of 4 cycles
- H_SYNC, V_SYNC, FID and DEN control signals for video synchronization
- VHDL RTL and functional test bench

Functional Description

Figure 1. Functional Description



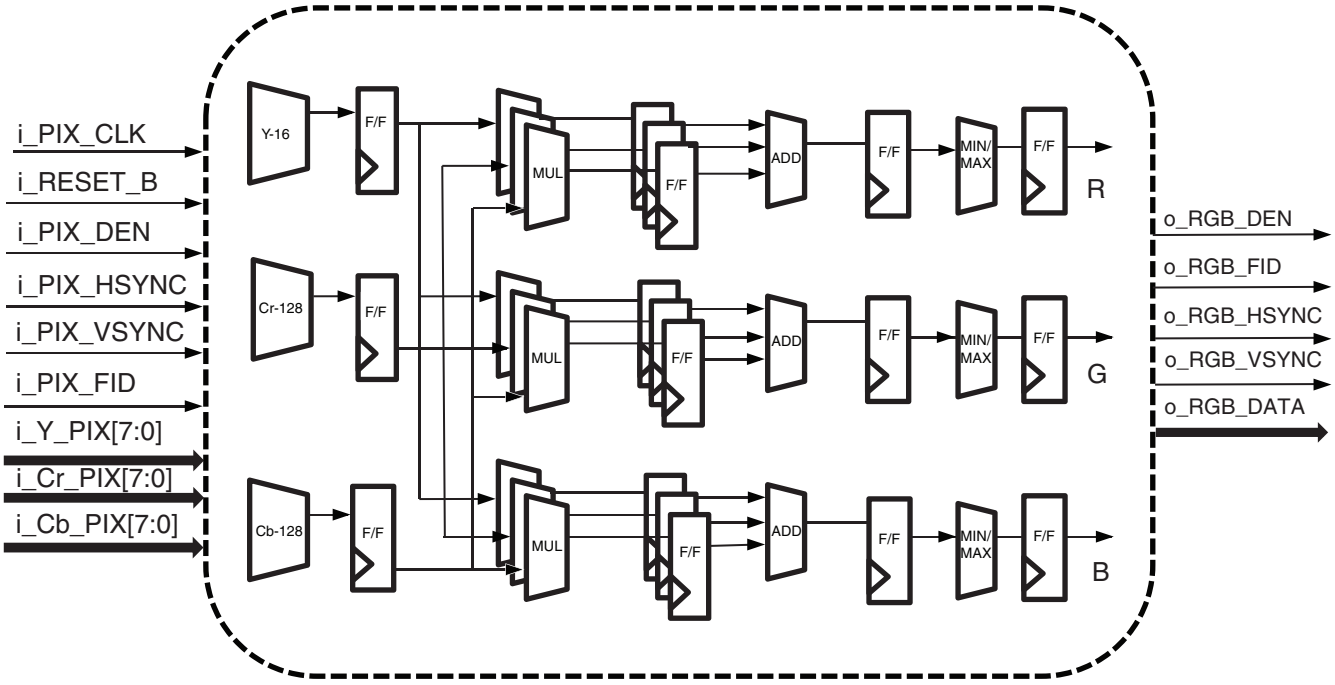
Signal Description

Table 1. Signal Description

Signal	Width	Type	Description
i_PIX_CLK	1	Input	Input Pixel Clock
i_RESET_B	1	Input	Asynchronous Active high reset
i_PIX_DEN	1	Input	Synchronous Data Enable
i_PIX_HSYNC	1	Input	Horizontal Sync
i_PIX_VSYNC	1	Input	Vertical Sync
i_PIX_FID	1	Input	Input Frame ID (odd/even field indicator)
i_Y_PIX	8	Input	Y component of pixel
i_Cr_PIX	8	Input	Cr component of pixel
i_Cb_PIX	8	Input	Cb component of pixel
o_RGB_DEN	1	Output	RGB Data valid
o_RGB_HSYNC	1	Output	Pipelined Horizontal Sync
o_RGB_VSYNC	1	Output	Pipelined Vertical Sync
o_RGB_FID	1	Output	Output odd/even field indicator (pipelined)
o_RGB_DATA	16	Output	Converted RGB Component

Design Module Description

Figure 2. Functional Block Diagram



Configurable parameter

None

Register Map

This design does not have any user accessible registers or memory.

Design Details

This module converts 8-bit YCbCr 4:2:2 to RGB565 as per the following conversion expressions:

- R = 1.164(Y-16) + 1.596(Cr-128)
- G = 1.164(Y-16) - 0.813(Cr-128) - 0.392(Cb-128)
- B = 1.164(Y-16) + 2.017(Cb-128)

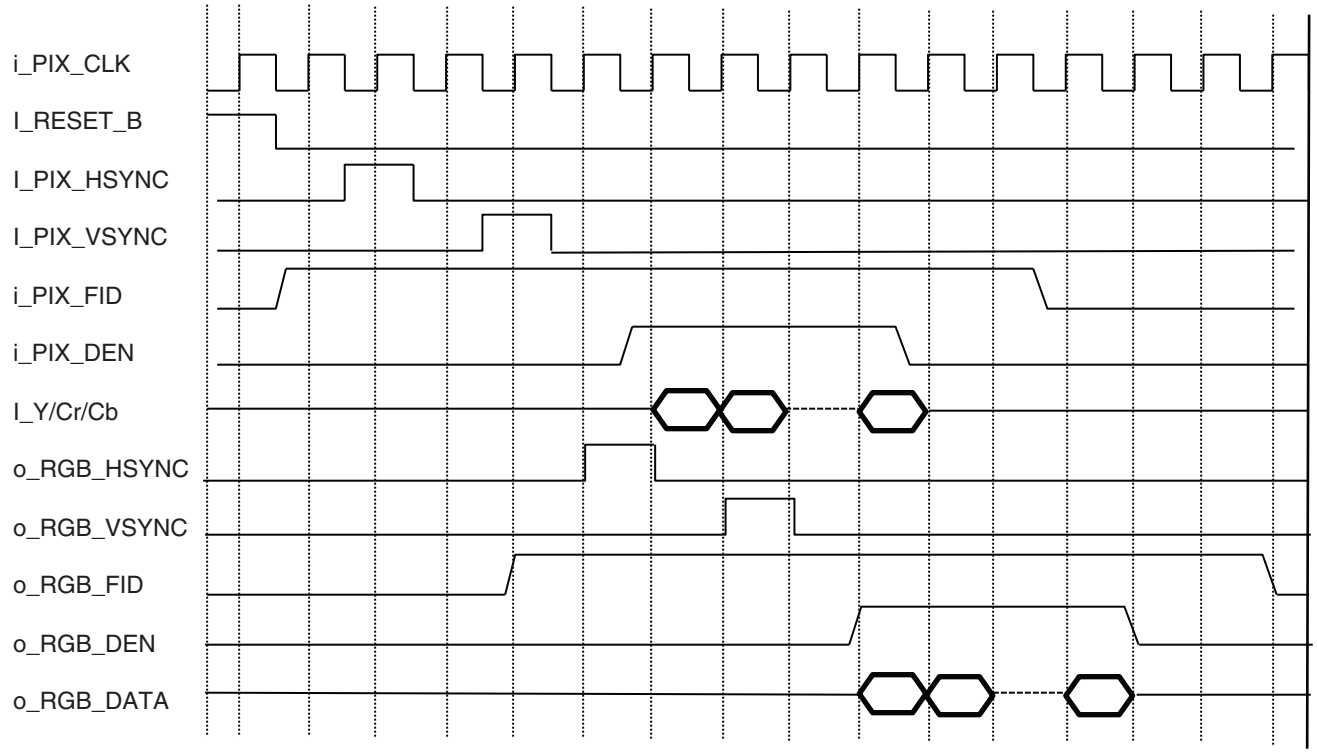
The implementation comprises of a set of constant coefficient multipliers and add/sub logic arranged in pipelined fashion. Considering the large amount of data path involved here, a pipelined implementation is provided here to improve the performance. If the converted R, G and B values falls outside the allowed range, then the values are clipped and limited to the maximum/minimum possible range. This is a fully synchronous design and all the modules listed in the block diagram generates registered output through input pixel clock. To facilitate easy insertion to practical video systems, the design makes use of video synchronization signals pixel clock (i_PIX_CLK), valid data indicator (i_PIX_DEN) and generates a delayed version of i_PIX_HSYNC, i_PIX_VSYNC, i_PIX_DEN and i_PIX_FID so that control signals synchronized with the output RGB565 stream.

Initialization Conditions

This design does not have any user specific initialization conditions.

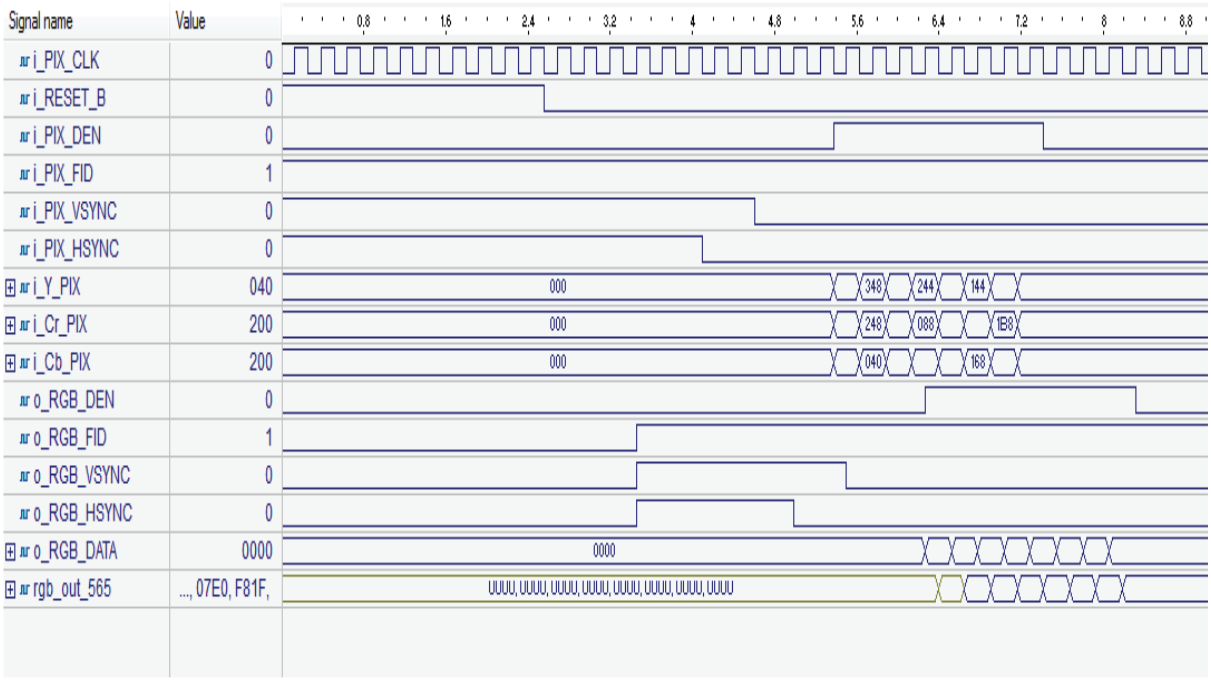
Timing Diagram

Figure 3. Timing Diagram



Simulation Waveforms

Figure 4. Simulation Waveforms

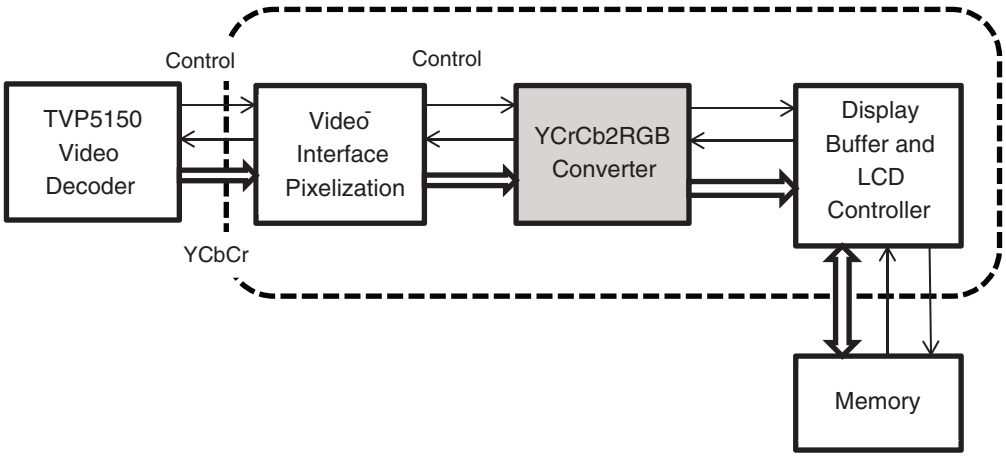


Usage Examples

Common YCbCr video sources include NTSC video decoders, MPEG decoders, and cameras. YCbCr to RGB converters are useful in applications like Video Surveillance, Display Systems, Image/Video processing applications, Image decompression systems etc…

Example usage of this module is illustrated in the block diagram below, which interfaces TVP5150 NTSL/PAL/SECAM Video Decoder to a Display system. Video Pixelization module decodes ITU-R BT.601 YCbCr stream and generates HSync, VSync, DEN and FID control signals.

Figure 5. Usage Example



Simulation setup comprises of a testbench which provides input Y, Cb, Cr values for various colors like red, blue, green, white etc… The DUT generated output RGB565 values are compared against the corresponding known RGB565 values.

Implementation

This design is implemented in VHDL. When using this design in a different device, density, speed or grade, performance and utilization may vary.

Performance and Resource Utilization

Table 2. Performance and Resource Utilization

Family	Language	Utilization (LUTs)	f_{MAX} (MHz)	I/Os	Architecture Resources
iCE40[1]	VHDL	273	>50	49	(63/160) PLBs

1. Performance and utilization characteristics are generated using iCE40-LP1K-CM121 with iCEcube2 design software.

References

- iCE40 Family Handbook

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2013	01.0	Initial release.