International Journal of Engineering Research and Development
ISSN: 2278-067X, Volume 1, Issue 3 (June 2012), PP.15-22
www.ijerd.com

RGB to YCbCr Color Conversion using VHDL approach

E. Prathibha[1], Dr.A.Manjunath[2], Likitha.R[3]
[1,2,3]*Department of EEE Sri Krishna Institute of Technology, Bangalore, Karnataka*

***Abstract*—**In this paper, we present the architecture and design of a color space conversion module. The color space converter module is used for changing the image from RGB color space to YCbCr color space. The color space conversion module was designed using VHDL and was implemented on an FPGA. This design methodology helped us to achieve faster the time to market and also the ability to reuse one physical device across multiple functions.

***Keywords*—**VLSI, Fuzzy Logic, FPGA, VHDL, RGB, YcbCr

I. INTRODUCTION

Color space conversion has become an integral part of image processing and transmission. Real time images and video are stored in RGB color space [1]. Transmitting images in RGB color space is not practical as their bandwidth requirement is very high. To overcome this problem and minimize the bandwidth requirement images in RGB color space are converted into other color space such as YUV, YIQ and YCbCr and then transmitted. The choice of the color space is dependent on the application and their requirement such as less storage, bandwidth or computation in analog or digital domains [2]. In this paper we present the architecture of a color conversion module for efficient implementation of color space conversion from RGB to YCbCr color space using an FPGA based system.

II. BLOCK DIAGRAM

The block diagram for the proposed system is shown in Fig. 1. The proposed system consists of a color conversion module designed using VHDL and implemented on a FPGA. The color conversion module is interfaced with the real world image through MATLAB. For color space conversion, the RGB image is taken and converted into pixels using MATLAB. The generated pixels are given to the color converter module. The color conversion module generates new set of pixels, YCbCr pixels which are given back to the MATLAB which recreates the image in YCbCr color space. This design helps us to reduce the overall hardware requirement of the system.

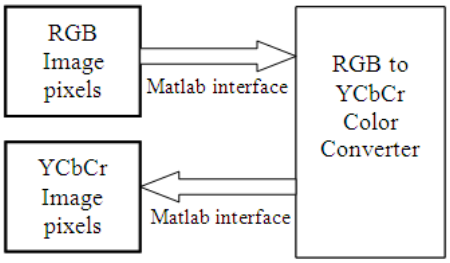

Figure1. Block diagram of the proposed system.

III. INTRODUCTION TO COLOR SPACES

A color space is a method by which we can specify, create and visualize color. As humans, we may define a color by its attributes of brightness, hue and colorfulness. A computer may describe a color using the amounts of red, green and blue phosphor emission required to match a color. A printing press may produce a specific color in terms of the reflectance and absorbance of cyan, magenta, yellow and black inks on the printing paper [3].

A color is thus usually specified using three co-ordinates, or parameters. These parameters describe the position of the color within the color space being used. They do not tell us what the color is, that is dependent on what color space is being used.

A color space is a mathematical representation of a set of colors.

The most popular color models are

1.RGB (used in computer graphics);

2.YIQ, YUV, or YCbCr (used in video systems);

3. CMYK (used in color printing).

However, none of these color spaces are directly related to the intuitive notions of hue, saturation, and brightness. All color spaces can be derived from the RGB information supplied by devices such as cameras and scanners [1] [4].

III.1 RGB Color Space

The red, green and blue (RGB) color space is widely used throughout computer graphics. Red, green and blue are three primary additive colors: individual components are added together to form a desired color and are represented by a three dimensional, Cartesian coordinate system as shown in Figure 2.

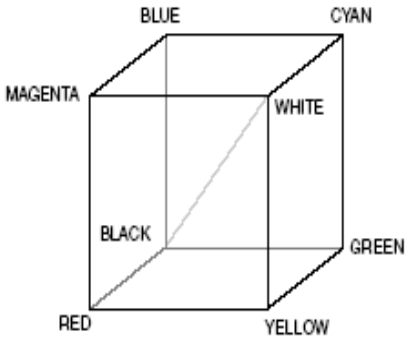

Figure 2. RGB Color Cube.

The RGB color space is the most prevalent choice for computer graphics, because color displays uses red, green, and blue to create the desired color [1].

The RGB color model colors, as shown in Fig. 3 can be written in several different ways.

- The color values may be written in the range 0.0 (minimum) to 1.0 (maximum). Full intensity red is 1.0, 0.0, 0.0.
- The color values may be written as percentages, from 0% (minimum) to 100% (maximum). Full intensity red is 100%, 0%, 0%.

The color values may be written as simply by multiplying the range 0.0 to 1.0 by 255. This is commonly found in computer science. Full intensity red is 255, 0, 0.

III.2 YCbCr Color Space

YCbCr is a family of color spaces used in video systems. Y is the luma component and Cb and Cr are the chorma components. The YCbCr color space is a scaled and an offset version of the YUV color space. Y is defined to have a range of 16– 235; Cb and Cr are defined to have a nominal range of 16–240 [5].
The basic equations to convert between RGB and YCbCr are:

$Y = 0.257R + 0.504G + 0.098B + 16$

$Cb = -0.148R - 0.291G + 0.439B + 128$

$Cr = 0.439R - 0.368G - 0.071B + 128$

The above equations have been used to generate the logic used in the color conversion module.

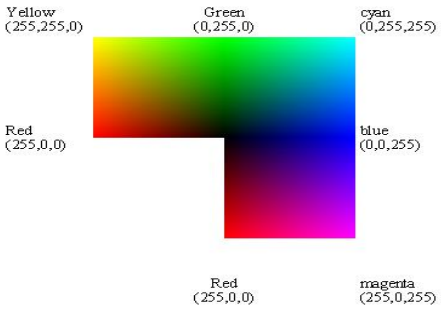

Figure3. RGB Color Model

III.3 Color Bars

Tables 1, 2 gives RGB, and YCbCr values for 100% amplitude, 100% saturated color bars for the different colors,(i.e., for White, Yellow, Cyan, Green, Magenta, Red, Blue, Black)[6].

	White	**Yellow**	**Cyan**	**Green**	**Magneta**	**Red**	**Blue**	**Black**
R	255	255	0	0	255	255	0	0
G	255	255	255	255	0	0	0	0
B	255	0	255	0	255	0	255	0

Table 1. 100% RGB Color Bars

	White	**Yellow**	**Cyan**	**Green**	**Magneta**	**Red**	**Blue**	**Black**
Y	235	210	170	145	107	82	41	16
Cb	128	16	166	54	202	90	240	128
Cr	128	146	16	34	221	240	110	128

Table 2. 100% YCbCr Color Bars

IV. DESIGN AND IMPLEMENTATION

The design and implementation of the color conversion module is explained in this section. The color conversion module was designed using VHDL and then synthesized to a target FPGA [7].

IV.1 Functional Description

The functional diagram of the proposed color space converter module is shown in Fig. 4

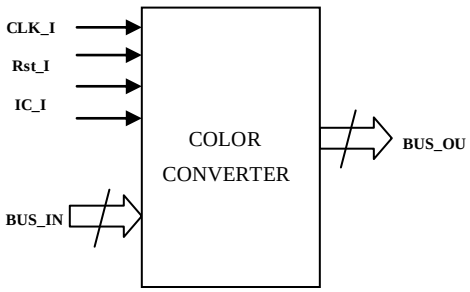

Figure 4.Color Space Converter pin-out.

The reference design has a CLK_I and RST_I Ports, which act as Clock and Reset ports to the module. Ports IC_I and OC_I are the Input and Output conversion types, which specifies the Input Color Space and Output Color Space depends on the application. BUS_IN are the pixels Inputs of the particular image to the color space converter and BUS_OUT gives the converted pixels output of the image.

IV.2 Block Diagram

The block diagram of the implemented color space converter module is as shown in Fig. 5. The input to the color space converter module in the form of pixels is given to BUS_IN. The two logical inputs IC_I and OC_I help us to define the color space conversion. If IC_I is 0 (zero) and OC_I is 1 (one) then the color space converter converts the pixels in RGB color space into pixels in YCbCr color space. If IC_I is 1 (ne) and OC_I is 0 (zero) then the color space converter converts the pixels in YCbcr color space into pixels in RGB color space[8].

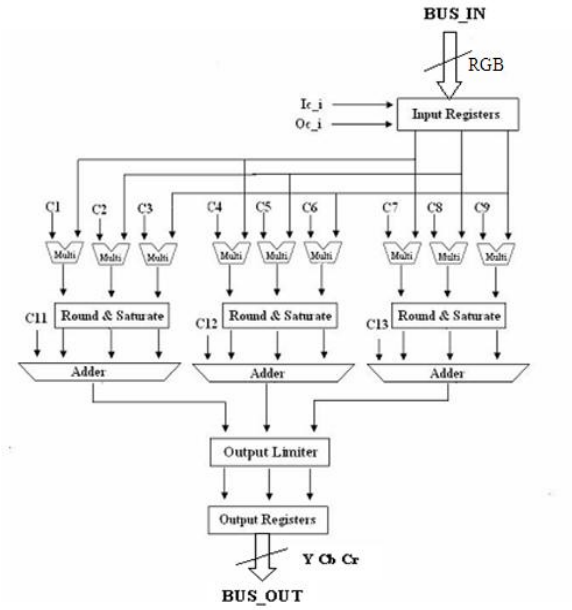

Figure5. Color Space Converter Block Diagram

V. SIMULATION AND SYNTHESIS REPORT

The project was implemented using Model Sim XE simulator and was synthesized using Xilinx ISE 10.1i. The FPGA used for implementation was 2v250fg256-6.

V.1 RGB Color Space to YCbCr Color Space Conversion

The simulate results of individual color conversion from RGB color space to YCbCr color space is presented in this section. Fig. 6 shows the simulated results of white color conversion from RGB color space to YCbCr color space.

White color:

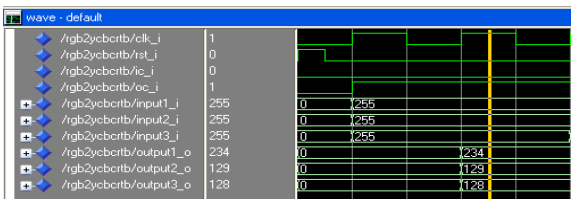

Figure 6. Simulated Result for White Color.

From Table 1 and Table 2, we can observe that white color is represented as (255,255,255) in RGB color space and as (235,128,128) in YCbCr color space. From Fig. 6 we can see that the input pixels corresponding to white color in RGB color space (255,255,255) are given to the conversion module and pixels corresponding to white color in YCbCr color space are generated by the color conversion module at the output. The pixels generated at the output matches with the theoretical values, validating the designed conversion module.

Similar exercise was done for all the basic colors tabulated in Table 1 and Table 2 and the observed results are shown in Fig. 7 to Fig. 13. From the figures it can be observed that the simulated output matches the theoretical output.

Yellow Color:

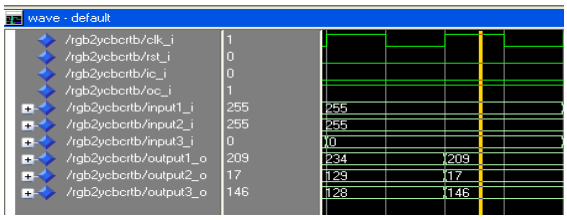

Figure 7. Simulated Result for Yellow Color

Cyan Color:

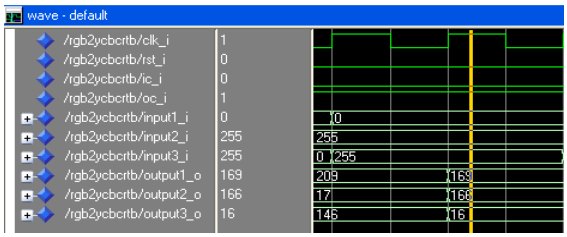

Figure 8. Simulated Result for Cyan Color

Green Color:

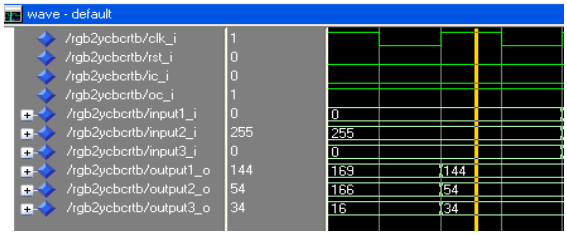

Figure 9. Simulated Result for Green Color

Magenta Color:

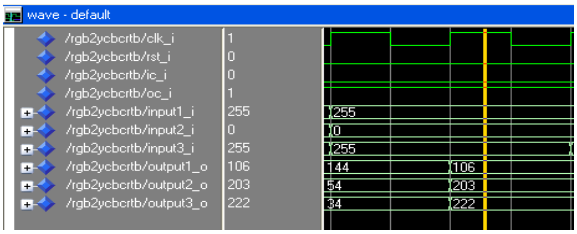

Figure 10. Simulated Result for Magenta Color

Red Color:

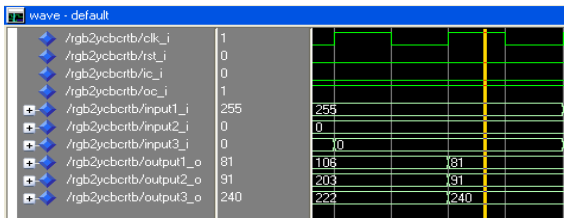

Figure 11. Simulated Result for Red Color

Blue Color:

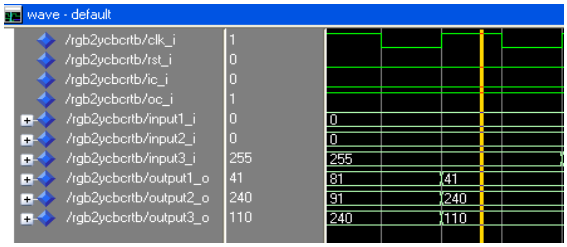

Figure 12. Simulated Result for Blue Color

Black Color:

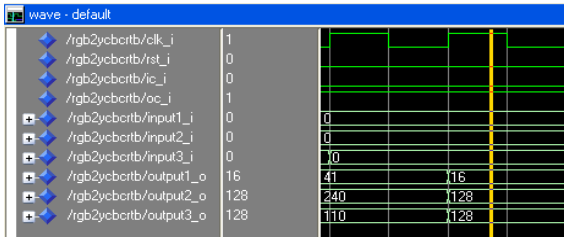

Figure 13. Simulated Result for Black Color

VI. VERIFICATION USING MATLAB

The authenticity of the designed color conversion module is verified for real time application by comparing its output with the output of color conversion module available in MATLAB. For the verification purpose a real time image in RGB color space as shown in Fig. 14 was selected.

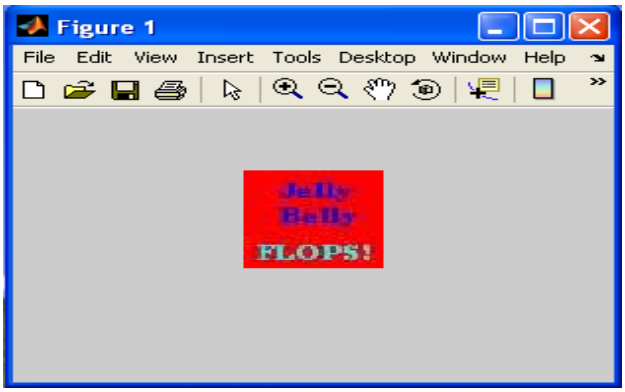

Figure 14 RGB color Image.

The selected picture is read using MATLAB and is converted into pixels in RGB color space using inbuilt MATLAB function. The RGB color space pixels generated are then converted into YCbCr color space pixels using MATLAB color conversion module. Then the same RGB color space pixels generated using MATLAB from the selected picture are given to the designed color conversion module to generate YCbCr color space pixels. Comparing YCbCr pixels generated by MATLAB as well as the designed color conversion module we verify the authenticity of the designed color module. It was observed that both MATLAB and the designed color module generate the same pixels values in the YCbCr color space authenticating the designed color conversion module. The first ten pixel values generated in YCbCr color space using MATLAB are shown in Table 4 and the corresponding pixel values generated by the designed color conversion module are shown in Table 5. The same procedure was followed for YCbCr color space to RGB color space conversion. For that purpose the image shown in Figure 15 was used and then the pixels generated by the color conversion module was compared with those generated by MATLAB for verifying the designed color conversion module.

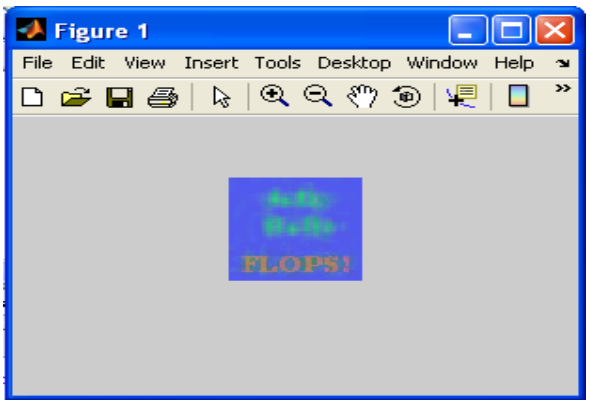

Figure 15 YCbCr color Image.

Table 4. RGB to YCbCr color space conversion using MATLAB

R	G	B	Y	Cb	Cr
248	3	0	81	90	236
247	4	0	81	90	235
250	5	10	84	94	235
249	5	13	84	95	235
247	4	0	81	90	235
248	4	0	82	90	235
234	9	7	81	94	227
236	8	5	81	93	228
240	7	0	81	90	231
243	7	0	82	90	232

Table 5. RGB to YCbCr color space conversion using the designed color space conversion module

R	G	B	Y	Cb	Cr
248	3	0	81	91	236
247	4	0	81	91	235
250	5	10	83	95	235
249	5	13	83	97	234
247	4	0	81	91	235
248	4	0	81	91	236
234	9	7	81	94	227
236	8	5	80	94	228
240	7	0	81	91	230
243	7	0	79	92	227

The Simulation results from monitor window are shown in the fig 16. This shows the clock, reset, input conversion (Ic_i) is zero and output conversion (Oc_i) is one. Three inputs and corresponding three outputs can be seen. For example, inputs marked are 236, 8, and 5 are RGB values. And next raising edge of clock gives output that are marked in White color are 80, 94 and 228 are YCbCr values.

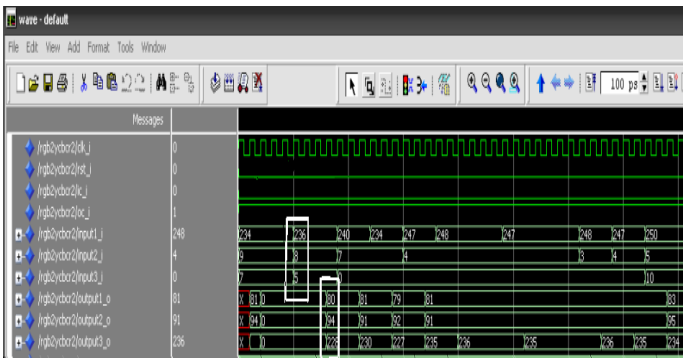

Figure 16: Simulation results from monitor window.

VII. HARDWARE RESULTS

The color space converter is implemented on Virtex II pro FPGA; Table 6 shows the summary of Device utilization summary, and also the Timing summary and power report are presented

Selected Device: **2vp30ff896-7**

Table 6. Device utilization summary.

Logic utilization	**Used**	**Available**	**utilization**
Number of Slices:	967	13696	7%
Number of Slice Flip Flops:	971	27392	3%
Number of 4 input LUTs:	1392	27392	5%
Number of bonded IOBs:	31	556	5%
Number of BRAMs:	30	136	22%
Number of GCLKs	6	16	37%
Number of DCMs	1	8	12%

Timing Summary:

Speed Grade: -7

Minimum period: 9.368ns (Maximum Frequency: 106.741MHz)

Minimum input arrival time before clock: 3.699ns

Maximum output required time after clock: 3.900ns

Maximum combinational path delay: No path found

Power report:

Name	Power (W)	Used	Total Available	Utilization (%)
Clocks	0.000	8	---	---
Logic	0.000	1357	27392	5.0
Signals	0.000	2859	---	---
IOs	0.000	31	588	5.3
BRAMs	0.000	30	136	22.1
DCMs	0.000	1	8	12.5
Total Quiescent Power	0.103			
Total Dynamic Power	0.000			
Total Power	0.103			

VIII. CONCLUSION

Design and architecture of color conversion module designed in VHDL and implemented on FPGA has been presented in this paper. The advantage of the architecture is that the same module can be used for both RGB to YCbCr color space conversion and vice-versa. The designed color space conversion module was verified by comparing the output of the designed module with the output of color conversion module present in MATLAB.

REFERENCES

[1]. Keith Jack, *Video Demystified: A Handbook for the Digital Engineer*, LLH Technology Publishing, Third Edition, 2001.

[2]. F. Bensaali, A. Amira and A. Bouridance, "Accelerating matix product on reconfigurable hardware for image processing applications", IEE Proceedings on Circuits and Devices Systems, vol. 152, No. 3, June 2005.

[3]. SreenivasPatil, 2007 Reconfigurable hardware for color space conversion, Rochester Institute of Technology, Rochester, New York.

[4]. M. Sima, S. Vassiliadis, S. Cotofana and J. T.J. Van Eijndhoven, " Color space conversion for MPEG decoding on FPGA-augmented trimedia processor", Proceedings IEEE International Conference on Application-Specific Systems, Architectures and Processors, pp. 250-259, June 2003.

[5]. R.-L. Hsu, M. Abdel-Mottaleb, and A.K. Jain, "Face Detection in Color Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696-707, May 2002.

[6]. Datasheet (www.alma-tech.com), "High Performance Color Space Converter," ALMA Technologies, May 2002.

[7]. F. Bensaali and A. Amira "Design and Implementation of Efficient Architectures for Color Space Conversion" ICGST-GVIP Journal, Volume 5, Issue1, December 2004.

[8]. B. Payette, "Color Space Converter: RGB to YCrCb," *Xilinx Application Note*, XAPP637, V1.0, September 2002.