IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 1, January 2016

Implementation of RGB to YCbCr Color Space Transformation for ARINC Transceiver

Hamsavahini R[1], Rashmi N[2]

Assistant Professor, Electronics and Communication Engineering, BMSIT &Management, Bengaluru, India[1, 2]

Abstract: This paper presents an accurate model for inter color space conversion (RGB to YCbCr), for transmission through HDMI specification, for ARINC 818 transceiver, which is based on the simplified shift and sub sampling recovery. Real-time video processing heavily relies on the color space conversion. Due to the real-time requirement, the traditional conversion methods often suffer from the moderate conversion speed, inaccuracy and low video quality.[5]

Keywords: HDMI, ARINC, video conversion, RGB, YCbCr.

I. INTRODUCTION

Colour space, also known as colour model has been widely used in many fields of digital video/image processing. A colour space is a useful method for users to understand the colour capabilities of a particular digital device or file. It represents what a camera can see, what a monitor can display or what a printer can print, etc. Certain colour spaces are also used to provide encoding formats for high speed video transmission [1].

There are a variety of colour spaces, such as RGB, CMYK, YCbCr, HSV, HSI, HSL etc[2]. Each colour system has different applications, for example RGB is the standard design of computer graphics systems[1]; CMYK is mainly used in printing and hard copy output; YCbCr[3] is associated with colour display of some hardware and also for high speed video transmission protocols such as HDMI.

As we all know, other spaces are usually defined by transformation of the RGB colour space. The inter-conversion of RGB and YCbCr 4:2:2 model cannot accomplish directly. Thus it can be divided into four steps: the conversion of RGB to YCbCr 4:4:4 YCbCr 4:4:4 to YCbCr 4:2:2. YCbCr 4:2:2 to YCbCr 4:4:4, YCbCr 4:4:4 to RGB. The whole process of conversion are extensively required to meet the time requirements.

II. DESIGN AND IMPLEMENTATION

This paper has been modularised and broken down to two different blocks, each performing a certain task. These two block will be implemented at the transmitting end of a video transmission system. The implementation is aimed to make the video conversion system universal or compatible with any video monitors of different specifications, perform with minimum latency of 7 to 8 clock cycles which is the least recorded till date and a standalone device not conformed to external influences.

The following are the two modules of the video conversion system implemented:

Module 1: RGB to YCbCr (4:4:4) conversion and

Module2: YCbCr (4:4:4) to YCbCr(4:2:2) conversion

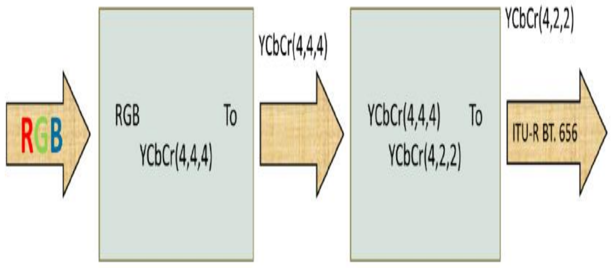


Fig. 1 Module Block Diagram

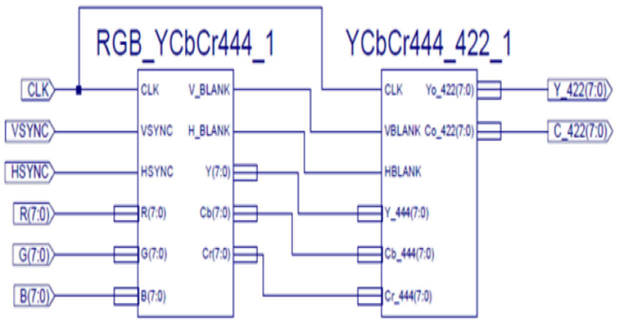


Fig. 1 Schematic Diagram

RGB to YCbCr (4,4,4)Module :

This module takes in N-bit RGB data (N=8, in this case), Sync signals: horizontal sync and vertical sync and an external clock. It gives out converted N-bit YCbCr (4:4:4) equivalent data along with the blanking signals that is adjusted for the output of the next module. It has a propagation delay of only 1 clock cycle between the input and the output that is an unforced delay.

The main goal of the module is to convert the incoming RGB data to an encoding technique called as YCbCr (4:4:4) by following a matrix conversion table. This module has to also be in sequence with the incoming sync signals and generate the blanking signals.

The module begins only if a new video frame has arrived. This will be signalled by the arrival of the vertical sync

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 1, January 2016

pulse, this sync pulse activates the module to start and send the vertical blanking signal.

During the time there won't be any data transmitted in the RGB buses (i.e. will be sending only 0's). Once the sync ends the front porch begins indicating the start of active video once it gets over. During the whole time of the sync and the front and back porch the vertical blanking signal will be in the 'ON' condition.

Likewise the horizontal sync occurs for every new line in the video. When the active video ends the horizontal back porch followed by the sync to indicate new line and the front porch for the preparation of the new occurs. During this time the horizontal blanking signals remains in the 'ON' condition until the blanking is over and the new line of the active video is fed to the RGB buses but until then we will be sending only 0's in the RGB bus during blanking.

In order to order to perform decimal operations on 8-bit vectored values, shift operation has been performed to the 8-bit data, for example, if a data, say, 10100011 has to be multiplied by 0.504, instead of performing the multiplication operation, we just shift it by the number equivalent to the decimal value (i.e. 0.504 is equal to the sum of 2-1 and 128-1, 0.500 + 0.007= 0.507). This operation is performed accordingly to other values of the matrix and the RGB is converted to YCbCr (4:4:4).

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.256 & 0.504 & 0.098 \\ -0.148 & -0.292 & 0.441 \\ 0.441 & -0.369 & -0.071 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

Fig. 3 RGB to YCbCr conversion Matrix

The above conversion matrix was used to convert RGB to YCbCr (4:4:4) format, this matrix cannot be synthesised. So, this matrix has been converted to a synthesisable form shown below (Fig 4) whose values are almost equivalent to the above matrix.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} \frac{1}{4}+\frac{1}{128} & \frac{1}{2}+\frac{1}{128} & \frac{1}{16}+\frac{1}{32} \\ \frac{1}{8}+\frac{1}{128}-\frac{1}{4}-\frac{1}{32} & \frac{1}{4}-\frac{1}{2}-\frac{1}{32}-\frac{1}{128} & \frac{1}{2}+\frac{1}{128}-\frac{1}{16} \\ \frac{1}{2}-\frac{1}{16}+\frac{1}{128} & \frac{1}{8}-\frac{1}{2}+\frac{1}{128} & \frac{1}{16}-\frac{1}{8}-\frac{1}{128} \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

Fig. 4 RGB to YCbCr conversion synthesisable Matrix

The above matrix in fig 4. when converted back to decimal form shown in fig 5. the values are almost equivalent to the original conversion matrix shown in fig 4.. Showing that not much of difference has occurred and the shifting operation was also done successfully.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.258 & 0.508 & 0.0937 \\ -0.148 & -0.289 & 0.445 \\ 0.445 & -0.367 & -0.071 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

Fig. 5 synthesisable Matrix after converting to decimal form

YCbCr(4,4,4) to YCbCr(4,4,2) module :

This module takes in N-bit YCbCr (4:4:4) data (N=8, in this case), blanking signals: hblank and vblank signals that is adjusted for subsampling the input to give the desired output and an external clock. It gives out subsampled N-bit Y and C (i.e. YCbCr (4:2:2) data). It has a propagation delay of only 2 clock cycle between the input of the previous module and the output of the present one.

The main goal of the module is to subsample the incoming converted YCbCr (4:4:4) data to YCbCr (4:2:2), this is done by sending the luma component and selectively sending the croma component, this perception was experimented because of the human perception of vision being more sensitive to the luma component than the croma component. This module has to also be in sequence with the incoming blanking signals.

This section will explain subsampling of the converted YCbCr (4:4:4) data to YCbCr (4:2:2) data. The technique of subsampling has already been introduced in the previous chapter. This subsampling procedure is performed according to the International Telecommunication Union Recommendations (ITU-R BT.601)[4-7].

The details of the subsampling process will be explained below.

This module, like the previous module, waits for the vertical blanking to end and then sends its output. This helps to maintain the synchronization between the modules. As the YCbCr (4:4:4) data arrives, the luma component (Y) is sent without being subsampled but the chroma components (Cr, Cb) are selectively sampled (i.e. every odd interval only Cb is sent and every even interval only Cr is sent).

This is done only during the active video duration and can be seen clearly in Fig 6.

During the blanking period, the first four clock pulses according to the ITU recommendation (ITU-R BT.656) will be used to send the End of Active Video (EAV) (FF 00 00 XY) delimiter through the outputs named Yo and Co. And this XY that is being sent is the preamble. During the rest of the blanking duration Yo and Co will be transmitting hexadecimal equivalent of decimal values 128 and 16.

Four clock cycles before the end of the blanking we will be sending Start of Active Video (SAV) during this duration the outputs Yo and Co will be transmitting FF 00 XY again. XY, or the preamble differs for different functions. It has different values for both the beginning and the end of the vertical blanking and horizontal blanking signals respectively.

The preamble can be constructed according to the table shown in Fig.7.

There are certain criteria in the table, such as V and H in the control byte section, which will be manipulated according to the position[4] (i.e. beginning or the end of horizontal or vertical blanking).

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 1, January 2016

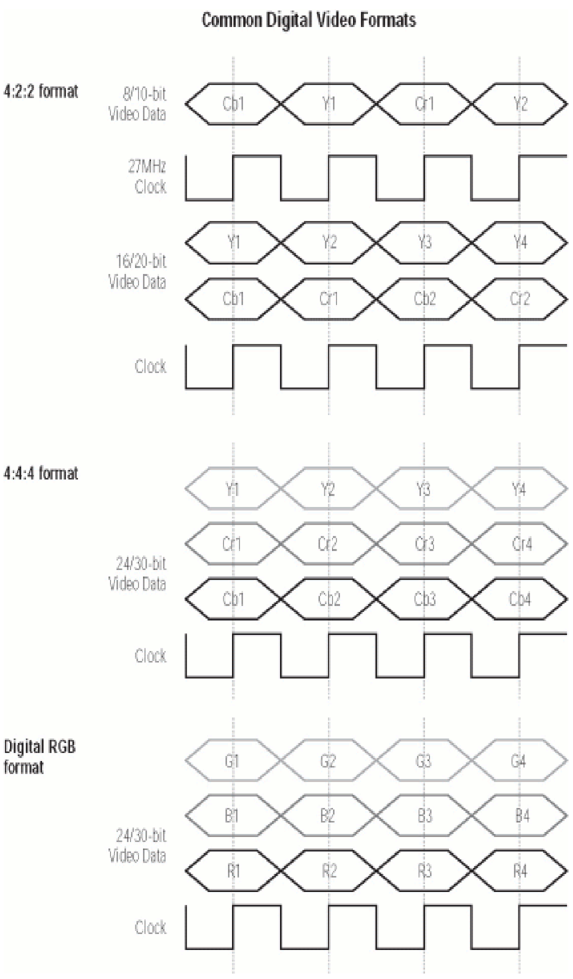
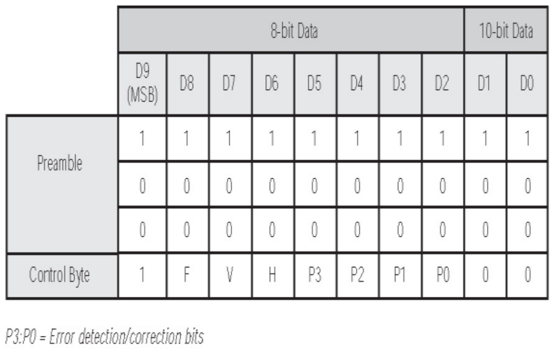


Fig. 6 Process of Subsampling



	8-bit Data								10-bit Data	
	D9 (MSB)	D8	D7	D6	D5	D4	D3	D2	D1	D0
Preamble	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
Control Byte	1	F	V	H	P3	P2	P1	P0	0	0

P3:P0 = Error detection/correction bits

Fig. 7 SAV/EAV Preamble codes

• V = 1 during Vertical Blanking
• V = 0 when not in Vertical Blanking
• H = 0 at SAV
• H = 1 at EAV

So, during the beginning of vertical blanking XY should be 0XB6 (182 decimal), during the end of vertical blanking XY should be 0XAB (171 decimal), during the beginning of horizontal blanking XY should be 0X9D (157 decimal), and during the end of horizontal blanking XY should be 0X80 (128 dec). The preamble is very important as it helps in reconstructing the sync pulses for both horizontal and vertical blanking[3].

III.RESULTS

In this section, waveforms are presented, obtained from post synthesis simulation. The waveforms will clearly depict the behaviour of the signals, as shown below.

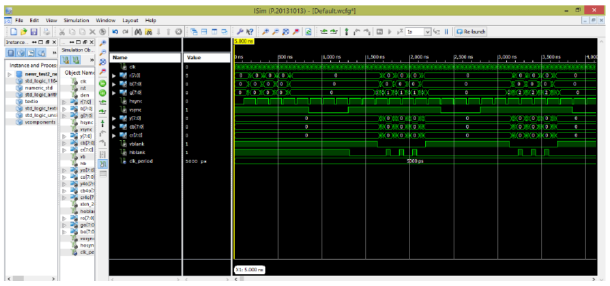


Fig. 8 output of sync and Blanking signals

In Fig.8 it can be seen that the horizontal sync (hsync) and the vertical sync (vsync) are active low signals, we have synced the horizontal blanking (hblank) and the vertical blanking (vblank) such a way that it acts as an active high signal, so whenever the sync is 'ON' the blanking signal would have turned 'ON' for the duration of back porch. Also to be noticed is that during the blanking the RGB is transmitting 0's.

During the vertical blanking notice that RGB is transmitting only 0's. And the vertical blanking signal (vblank) remains 'ON' for a longer duration than the vertical sync pulse (vsync) accounting for the back porch and front porch. As mentioned above the vertical sync pulse stays active for the duration of 3 lines, or 3 horizontal sync pulses.

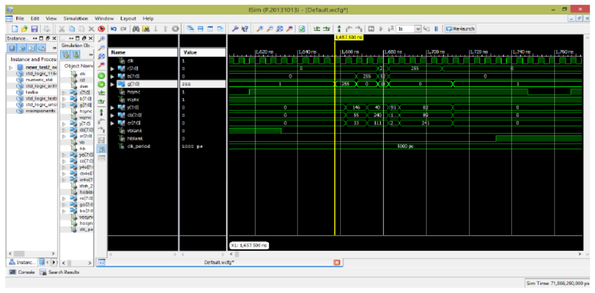


Fig. 9 Conversion of RGB to YCbCr (4:4:4)

In Fig.9, it can be seen that the conversion of the RGB to YCbCr (4:4:4) according to the code conversion techniques that were introduced above, are working perfectly. And it can be seen that there is a propagation delay of 1 clock cycle between the input data and output data.

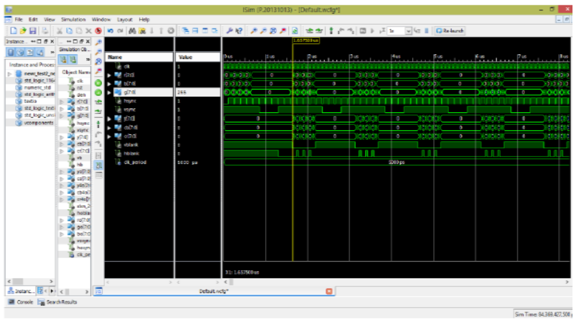


Fig. 10 Post Synthesis Simulation

In Fig.10, it can be seen that in the post synthesis simulation, the front porch, sync, back porch and the active data are arriving and in sync.

One can see that the position and clock duration of the sync pulses and the porches are exact as required.

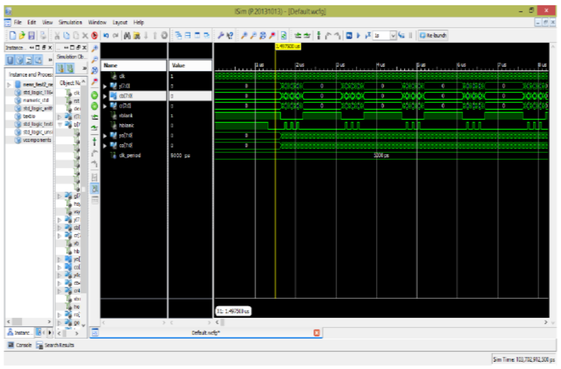


Fig. 11 output of sync and Blanking signals

In Fig.11, one can notice that as the previous module, this module also waits for the fall in the vertical blanking (vblank) input.

One can see that as the vblank falls the output Yo and Co change from 0's to certain data. Here the blank signals horizontal and vertical are active high signals.

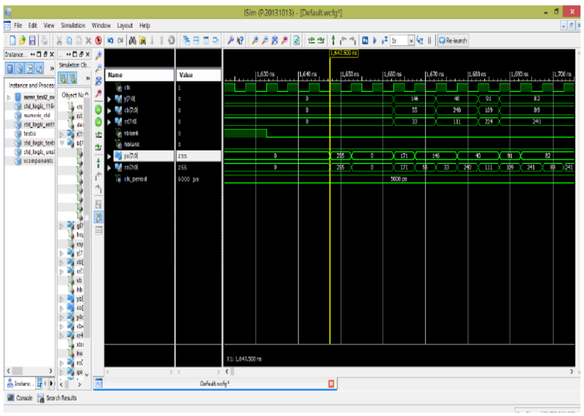


Fig. 12 Output of subsampling

In Fig.12, one can see the detailed view of the subsampling that has been performed. If one looks at the 1,670ns position carefully it can be noticed that the luma component (Y) is transmitted, and the croma component (Cr,Cb) is transmitted one after the other in intervals, or is subsampled. The same thing can be seen in 1,680ns position also.

The module is subsampling according to the recommendations and the no of bits transmitted is reduced from 26 bits to 16 bits without compromising much on the quality of the video. It is evident that there is propagation delay of two clock cycles.

Notice that four clock cycles (without accounting for delay) after the vertical blanking has gone low we are transmitting 0XFF (255) 0X00 0X00 0XAB (171 in decimal, this resembles the end of the vertical blanking as mentioned above).

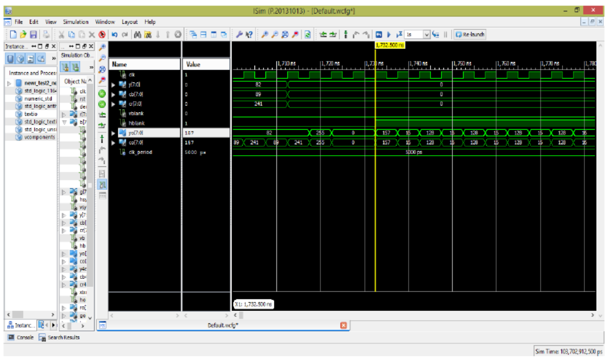


Fig. 13 output of EAV

In Fig.13, one can see that four clock pulses before the horizontal blanking we are transmitting 0xFF (255), 0x00, 0x00 and 0x9D (157 in decimal, this resembles the start of the horizontal blanking as mentioned above). The quadruplets make up the End of Active Video delimiter (EAV) for the end of a line.

It can also be seen that, during the blanking period we will be transmitting hexadecimal values 0x80 and 0x10 (i.e. 128 and 16 in decimal). This is sent alternatively, I.E. during even clock cycle of blanking, we will be transmitting 0x80 and during odd cycles of blanking, we will be transmitting 0x10.

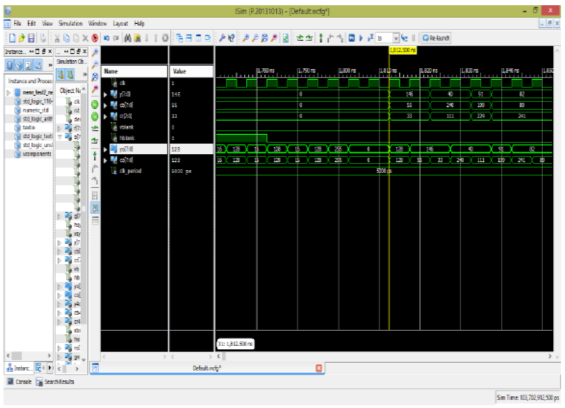


Fig. 14 Output of sub sampling at the SAV

In Fig.14, we can see that four clock pulses after the horizontal blanking we are transmitting 0xFF (255), 0x00, 0x00 and 0x80 (128 in decimal, this resembles the end of the horizontal blanking as mentioned above) right after the blanking active video data gets transmitted. Notice the two clock pulse delay, that's due to the delay of the previous module combined with the present module. Right after the Start of Active Video (SAV) the subsampling continues during the active video.

In Fig.15, one can see that four clock pulses before the vertical blanking we are transmitting 0xFF (255), 0x00, 0x00 and 0Xx6 (182 in decimal, this resembles the start of the vertical blanking as mentioned above) and during the blanking we are sending hexadecimal 0X80 and 0X10 (i.e. 128 and 16 in decimal). Notice the two clock pulse delay, that's due to the delay of the previous module combined with the present module, this blanking starts right after the End of Active Video (EAV) and at the end of the frame.

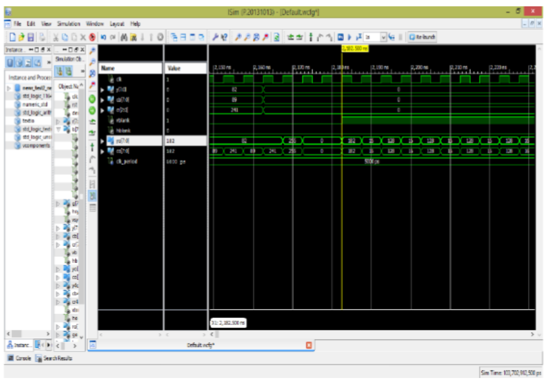


Fig. 15 output of subsampling after the EAV

IV. CONCLUSION

The conversion of color models such as RGB to YCbCr, is extensively used in efficient video transmission. This requires a robust design which caters for many needs such as accuracy, computational delay, and platform and specification independence. Therefore, such a design is proposed in this paper, which can be used for robust inter-conversion between YCbCr and DRGB.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Management and staff, BMS Institute of Technology for providing the facilities.

REFERENCES

[1] Payette, Benoit. "Color Space Converter: R'G'B'to Y'CbCr." *XAPP637 (v1. 0) September* 12 (2002).
[2] Tkalcic, Marko, and Jurij F. Tasic. "Colour spaces: perceptual, historical and applicational background." *Eurocon*. 2003. Marko Tkalcic, Jurij F. Tasic, "Colour spaces - perceptual, historical and application background"
[3] ITU-R Recommendation BT.656-5: Studio encoding parameters of digital television
[4] ITU-R Recommendation BT.601-5: Studio encoding parameters of digital television
[5] Yang, Yang, Peng Yuhua, and Liu Zhaoguang. "A fast algorithm for YCbCr to RGB conversion." Consumer Electronics, IEEE Transactions on 53.4 (2007): 1490-1493.
[6] Bensaali, Faycal, and Abbes Amira. "Accelerating colour space conversion on reconfigurable hardware." *Image and Vision Computing* 23.11 (2005): 935-942.
[7] Ahirwal, Balkrishan, Mahesh Khadtare, and Rakesh Mehta. "FPGA based system for color space transformation RGB to YIQ and YCbCr." *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*. IEEE, 2007.
[8] Wang, Rui, et al. "VLSI design of universal color conversion circuit." *Radio Science Conference, 2004. Proceedings. 2004 Asia-Pacific*. IEEE, 2004.
[9] Bensaali, F., A. Amira, and A. Bouridane. "Design and implementation of efficient architectures for color space conversion." *International Journal on Graphics, Vision and Image Processing* 5.1 (2004): 37-47.
[10] Szedo, Gabor. "Color-Space Converter: RGB to YCrCb." *Xilinx Corp* (2006).
[11] B. Payette, "Color Space Converter: RGB to YCrCb," Xilinx Application Note, XAPP637, V1.0, September 2002.
[12] Goslin, Gregory R. "Using Xilinx FPGAs to design custom digital signal processing devices." *Proc. of*. 1995.
[13] Guanqun, Xiang Shoukun Huang Qijun Jiang, and Ma Jianwei Cheng Fangmin. "The digital video conversion interface system of ITU-R BT. 656 based on FPGA [J]." *Electronic Measurement Technology* 4 (2009): 034.