Video and Display Controller Using FPGA

Abdul Raed Bin Rosli

Department of Electronics, Faculty of Electrical Engineering
Universiti Teknologi MARA, 40450 Shah Alam, Selangor
raed_fast88@yahoo.com

***Abstract*—This document describes the implementation of video capture and display system FPGA (Field Programmable Gate Array). The system captures video using CMOS image sensor camera to be processed and displayed on the monitor through the VGA (Video Graphic Array) port. The digital data from the camera consists of YUV colour space was processed and then converted to RGB (Red Green Blue) colour space to be displayed on a VGA monitor. The processor and the VGA controller architecture were described using VHDL (VHSIC Hardware Description Language). The system was implemented on a Xilinx Virtex 4 FPGA.**

Key words: FPGA, YCrCb, VGA, RGB, VHDL.

I. INTRODUCTION

Video capturing and display is commonly used in surveillance and machine intelligence. Recently, surveillance systems with image recording functions become vital devices both in private and public places [1]. The key components of this system are video camera, controller and display. The camera and the display operate in two different video color models, YCrCb and RGB. YCrCb and RGB are two most popular color models used in video and image processing [2]. YCrCb, is a scale and offset version of the YUV color space that commonly used in video system, including cameras. In other word, YCrCb is the digital version of YUV. It consist of luminance (Y) information, chrominance red (Cr) and chrominance blue (Cb) information. The RGB is the common choice for computer graphics because color display user red, green and blue to create desired color often displayed on CRT or LCD monitor [2]. The camera has output in form of YUV. To be able to process the signal in FPGA, the camera output signal is converted to YCrCb color space. Video signals with a microprocessor processing is not a good choice because a CPU is usually too slow for video processing. It is decided that Xilinx VIRTEX 4 FPGA as a suitable platform for video signal processing. The FPGA, act as an interface between the camera and the display to control and process the signal flow in the system pipeline, thus involving digital signal processing. The selection of VIRTEX 4 FPGA as processor is based on several factors. VIRTEX 4 FPGA is an off-the-shelf programmable device with the best mix of logic, memory, I/O, clock management, and digital signal processing speed. Furthermore, the FPGA selection was dependent on availability. The available FPGA board in the lab was the VIRTEX 4 FPGA Evaluation Board.

II. METHODOLOGY

The block diagram of the video processing pipeline is shown in figure 1. Components of the system are described in next sub-sections. The most important part is the FPGA device where the VHDL source code was implemented and synthesized.

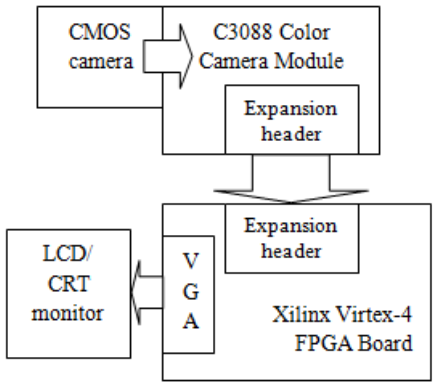

Figure 1: Video camera and display system dataflow.

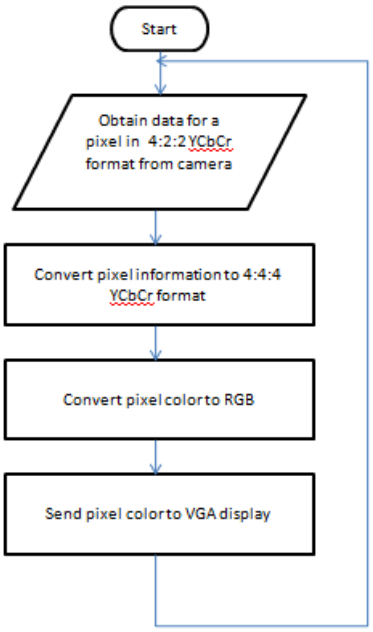

Figure 2: System flowchart

Figure 2 shows the flowchart of the system. After startup, the camera obtains pixel information in form of video signal. The camera then transmits the signal pixel by pixel. The FPGA converted the format of the pixel from 4:2:2 to 4:4:4 format then later convert the color space from YCbCr to RGB. RGB signal was sent to the VGA display to be display. The system continuously obtains pixel information, means that whenever a pixel is being displayed, another is being converted. Thus, timing management is very important.

A. Digital Camera Interface

The camera used was C3088 1/4" Color Camera Module with digital output. It uses OmniVision's CMOS image sensor OV6620 with output of 5.0 Vp-p digital YCrCb 4:2:2 format signal. The signal carries pixels with a total of 356x292 pixels. The I^2C port was connected with the expansion pin of the FPGA board. The output signals from the camera that were taken were output Y bus, Horizontal Sync, Vertical Sync, Pixel clock output and output UV bus. Figure 3 shows the camera device.

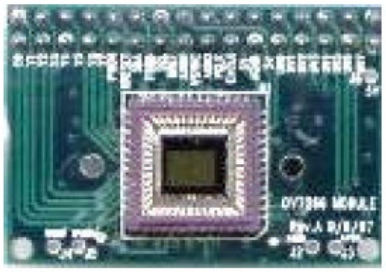

Figure 3: Camera Module with CMOS image sensor

B. Color Space Conversion

Many color models have been proposed, each oriented towards supporting a specific task or solving a particular problem [3]. YCrCb and RGB were the two models that involved in this system, each with different identity. Color space is a method by which we can specify, create and visualize color [4]. In this section, the color space conversion was implemented using VHDL language

The camera supplied a continuous 8 bit-wide digital image data stream in the form of luminance and chrominance, YCrCb. The chroma format of the data is 4:2:2. Figure 4 shows the 4:2:2 format. To convert the video information to RGB, a chroma resampling method was implemented using VHDL. The system uses color space conversion module from Xilinx IP Core convert422to444. The 4:2:2 format contains horizontally subsampled chroma. For every two luma samples, there is an associated pair of Cr and Cb samples [4]. The subsampled chroma locations are co sited with alternate luma samples. Delivery of this format involves interleaving Cr and Cb on a single bus, and running this bus at full sample rate.

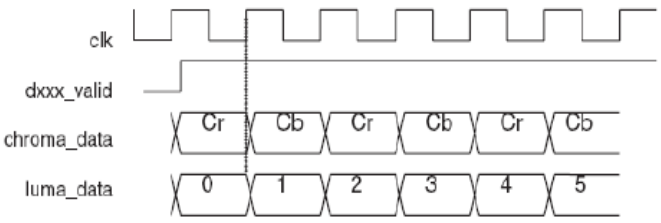

Figure 4: camera output in form of 4:2:2 chroma format.

Luma and chroma 4:2:2 format 8-bit wide output bus of the camera was converted to 4:4:4 format, which have three bus of the same width, Y, Cr and Cb. This format is the same as RGB format, thus it is an important step to be done before converting the color to RGB color.

A VHDL model source code was developed to convert YCrCb 4:4:4 format data into RGB 4:4:4 format. Y has the range of 16 to 235 and Cb and Cr have the range of 16 to 240 [5]. All inputs and outputs bus are 8-bit wide. The implementation of the source code was based on the matrix equation shown in equation 1 below

$$\begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix} = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix} \begin{bmatrix} x1 + b1x \\ x2 + b2x \\ x3 + b3x \end{bmatrix} \quad (1)$$

Each variable have standard values base on type of color conversion. For this system the values for the variable are shown in equation 2.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \mathbf{1.164} & \mathbf{0} & \mathbf{1.6} \\ \mathbf{1.164} & \mathbf{-0.392} & \mathbf{-0.813} \\ \mathbf{1.164} & \mathbf{2.017} & \mathbf{0} \end{bmatrix} \begin{bmatrix} Y - \mathbf{16} \\ Cb - 128 \\ Cr - 128 \end{bmatrix} \quad (2)$$

The source code was translated in to pipeline diagram shown in figure 5. The diagram shows the bus width of the input and output and the implementation of adders and multipliers in the design. X1, X2 and X3 are the input signal while B1x, B2x and B3x are the shift factor. Y1, Y2 and Y3 are the output. B1y, B2y and B3y are the post-shift factor which in this case equal to zero. A simple way is to describe the VHDL behavior is using equations such as addition and multiplication and ISE tool synthesized the hardware according to the design objective setting. In this project, the design objective was set to balance. DATA_ENA signal to enable the input to be process while DATA_RDY signals the arrival of the output.

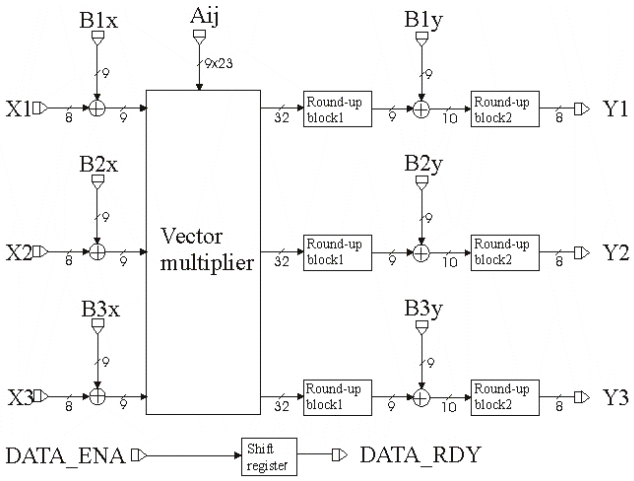

Figure 5: Pipeline diagram of color converter

C. VGA Display Interface Design

VGA is one kind of industrial standard, this standard has defined many parameters of VGA, such as display resolution, refreshing rates, synchronization signal timing, signal polarity and RGB signals electrical level, etc. The reason that VGA is called Video Graphics Array is the monitor displays a frame of image data finally is an array which is composed of M column and N row pixel spot [6]. M×N is said on the display resolution. VIRTEX 4 FPGA board provided an on board ADV7125 DAC to convert the digital RGB signal to 0.7Vp analog signal where it will be recognized by the monitor to display the information.

In VGA display protocol, the timing control is crucial. The timing signals that control the VGA display are Horizontal Synch (HS) and Vertical Synch (VS). The Horizontal Synch (*HS*) is responsible for indicating that a new line is starting, while the Vertical Synch (*VS*) is responsible for indicating that a new screen is starting [7]. Figure 6 shows the representation of the control signals.

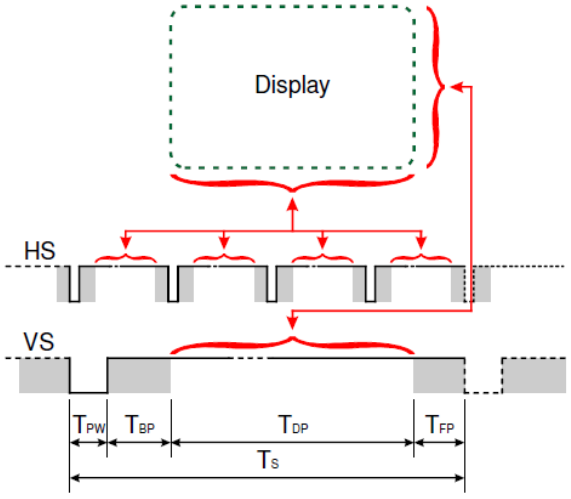

Figure 6: Synchronization signal HS and VS.

A typical PC monitor has a 60 Hz refresh rate. For a 640x480 pixel display mode, it is calculated that a pixel need at most 40 ns to be painted on the screen. Therefore, the clock frequency was configured to be at 25MHz. The front and back porch intervals, TFP and TBP are the pre- and post-sync pulse times. Information cannot be displayed during these times.

III. Simulations And Results

A. VGA Experiment

A source code was developed to display colors on the VGA display. The code included instructions to control the RGB signal sent to the VGA port by row and column. Figure 7 show the display result. To generate HS and VS, a simple row and column counter were implemented. The VS and HS signal is normally high. When the row counter hits 494 VS will go low for one clock cycle. HS signal goes low at column counter between 660 and 756. Column counter counts until 800 while row counter counts 525. Row counter increases by 1 after column counter reached 800.

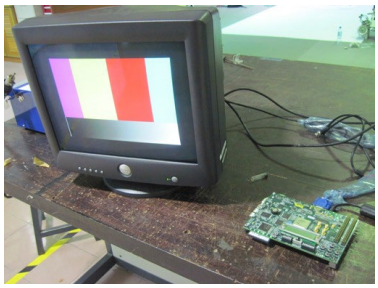

Figure 7: 640x480 display

B. VHDL Testbench

To verify the coding behavior before downloading it on board, a testbench code was written. The simulation was done using ALTERA ModelSim 6.5b software. The stimulus or test input was generated from an ASCII converted image data. Figure 9 shows the waveform for the testbench simulation. Refer back to equation 1 and 2. The input data are x1, x2 and x3 while the output data are y1, y2, y3. Inputs for this system are YCbCr signals and outputs are RGB signals.

The controller was implemented using VHDL and compiled using Xilinx ISE 12.1. ISE tool was used to synthesis the code. The target device was VIRTEX 4 technology XC4VSX-10ff668. ISE tool also generated the block model and RTL schematic. Table 1 shows the synthesis results. Figure 7 shows the RTL schematic. The design has been tested on Xilinx VIRTEX 4

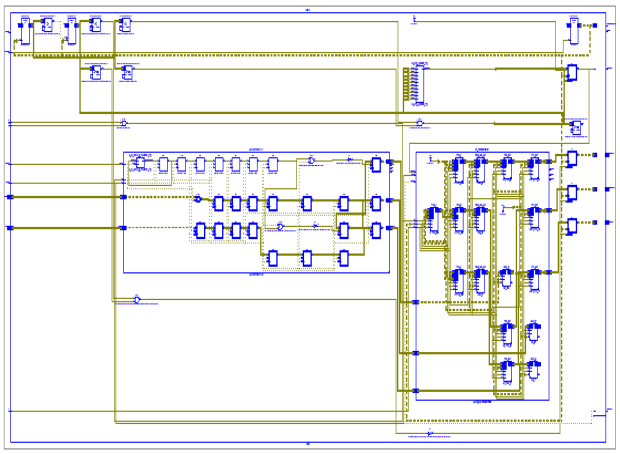

Figure 7: RTL schematic synthesis generated

TABLE 1
LOGIC UTILIZATION OF XC4VSX-10FF668

Logic utilization	Used	Available
Slice Flip Flops	148	30,720
LUTs	165	30,720
Bonded IOBs	50	448
BUFG/BUFGCTRLs	3	32
DSP48s	3	192

The signal radix is in the form of hexadecimal. Note that the output signals were delayed by 8 clock cycle. This shows that 8 clock cycle needed for the color conversion process. For the first batch, Y = 3F, Cb = 23, Cr = 40. The conversion results are R = 00, G = 87, B = EC. The signal y1c, y2c, y3c, y1bv, y2bv, y3bv are internal signal and can be neglected.

IV. CONCLUSION

This paper introduced the concept of video camera and display controller implemented on FPGA that could capture live video and display it on a VGA display. FPGA is the base for video and image processing. The concept can be apply in many areas and can benefit individual and industries. In the future, proposal will be made to optimize the system performance by trading off between speed, area and power consumption and improve the image quality of the display.

ACKNOWLEDGMENTS

The author is grateful to supervisor, Madam Fadzliana Saad and co-supervisor Dr Azilah Saparon of Department of Electronics, Faculty of Electrical Engineering UiTM Shah Alam for their support, guidance, criticism and advices in completing this project.

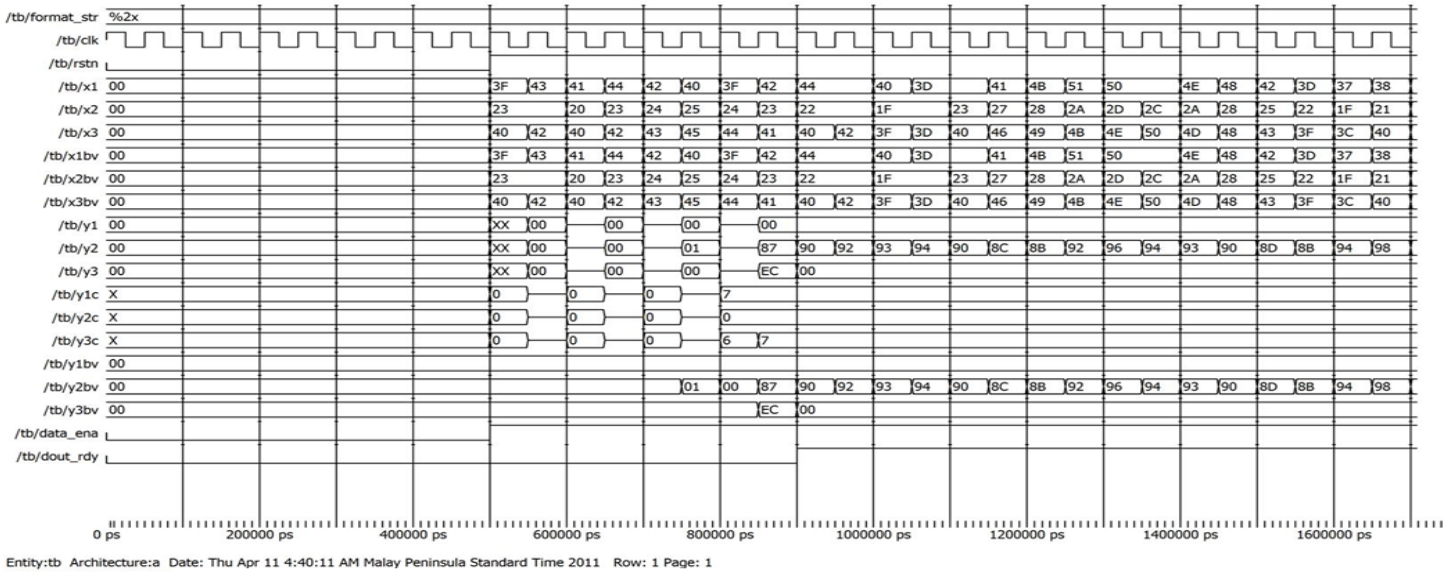

Figure 9: Simulation result of the color converter

REFERENCES

[1] Yuan-Pao Hsu; Hsiao-Chun Miao; Ching-Chih Tsai; , "FPGA implementation of a real-time image tracking system," *SICE Annual Conference 2010, Proceedings of* , vol., no., pp.2878-2884, 18-21 Aug. 2010.

[2] Gabor Szedo, "Color-Space Converter: YCrCb to RGB," XAPP931, v1.2, December 2009.

[3] Sapkal, A.M.; Munot, M.; Joshi, M.A.; , "R'G'B' to Y'CbCr color space conversion Using FPGA," *Wireless, Mobile and Multimedia Networks, 2008. IET International Conference on*, vol., no., pp.255-258, 11-12 Jan 2008.

[4] Clive Walker, "Chroma Resampler," Xilinx, XAPP932 (1.0.1) July 28, 2010.

[5] Wu, Lei; Yang, Shengqi; Wang, Xuzhi; Cheng, Ronghui; Wan, Wanggen; , "Xilinx Virtex-5 FPGA based video input and output interface design," *Wireless Mobile and Computing (CCWMC 2009), IET International Communication Conference on* , vol., no., pp.98-101, 7-9 Dec. 2009.

[6] Wang Ziting; Guo Haili; Sun Yan; , "Design of VGA Image Controller Based on SOPC Technology," *New Trends in Information and Service Science, 2009. NISS '09. International Conference on* , vol., no., pp.825-827, June 30 2009-July 2 2009.

[7] J.L. Russi, "Human-Machine Interface Based on FPGA and VGA Monitor" Federal University of Pampa, Campus Alegrete, 2009