Hardware Co-simulation of Anaglyph 3D image Using Xilinx System Generator

Rakesh Y[1], Dr.K.Sri Rama Krishna[2]
ECE Dept Assistant Professor[1], Professor[2]
[1]SRK Institute of Technology, [2]Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh

Abstract— **In modern era the trend of 3D movies and 3D projection took a very intensive attention. Stereoscopic 3D images are elementarily obtained by overlapping left and right eye images in different color planes of a single image for successive viewing through colored glasses. Here we present a novel FPGA based reconfigurable architecture for 3D image color space conversion RGB to YCbCr for implementing real-time DSP applications using Xilinx System Generator (XSG) for MATLAB. The CSC design is implemented in hardware which is realized using the Xilinx Block Sets presented in Simulink. From the hardware software co-simulation block VHDL/Verilog code is generated and is tested on Atlys FPGA Spartan 6 (xc6slx45-2csg324).**

Keywords— **CSC, FPGA, XSG, Simulink, MATLAB**

Introduction

In today's world there is enormous increase in 3D demand which is quite natural, as market is growing rapidly due to huge requirement of electronics media Humans will discern the depth illusion in a 3D image from two non-identical perspectives. The two viewpoints are from both the eyes provides an excellent immersive vision. The two different images are collectively called as a stereoscopic image and the whole process is called stereo image method. As the bandwidth requirement is very high when transmitting images in RGB color space there is a need to convert images into different color spaces such as YUV, YIQ and YCbCr and then transmitted. Depending upon the application and requirements the choice of the color space is chosen. In this paper we present a novel architectural module for efficient implementation of RGB to YCbCr color space conversion using an FPGA based system. Due to their low power dissipation per unit computation, high performance and re-configurability FPGAs are an attractive choice to implement in the system hardware. Xilinx System Generator provides a set of Simulink blocks (models) for several hardware operations that could be implemented on various Xilinx FPGAs. These blocks can be used to simulate the functionality of the hardware system using Simulink environment. System Generator is used as a high level suited design tool in order to create a custom data path in FPGA. Designs are captured in the DSP friendly Simulink modeling environment using a Xilinx specific blockset. All of the downstream FPGA implementation steps including synthesis and place and route are automatically performed to generate an FPGA programming file. In addition, the software provides for the hardware simulation and hardware-in-the-loop verification, referred to as hardware co-simulation [2, 3], from within this environment. For easier hardware verification and implementation compared to HDL based approach we use hardware co-simulation methodology. When compared in terms of cost efficiency the usage of Simulink simulation and hardware-in-the loop approach is far better than other methodologies.

XSg design flow

The integration of Simulink and MATLAB from The Math Works [2] and the Xilinx FPGA design suite of tools [3], now allow system development from a model-based view point which targets an FPGA. Xilinx System Generator (XSG) [5] is an Integrated Design Environment (IDE) for FPGAs within the ISE 14.1 development suite, which uses Simulink [5], as a development environment and is presented in the form of model based design. Here Designs are modeled and simulated using MATLAB, Simulink and Xilinx library and the tool automatically generates the HDL code that is to be mapped into pre-optimized Xilinx blocks in the design. There is a connection between XSG blocks and Simulink blocks i.e., gateway blocks. XSG automatically generates simulation results, RTL synthesis, VHDL/Verilog code, User Constraint File (UCF) and mapping hardware. It was created primarily to deal with complex Digital Signal Processing (DSP) applications but now it is intensively used for the implementation of many image processing applications.

The XSG Design flow is as shown in Figure 1.

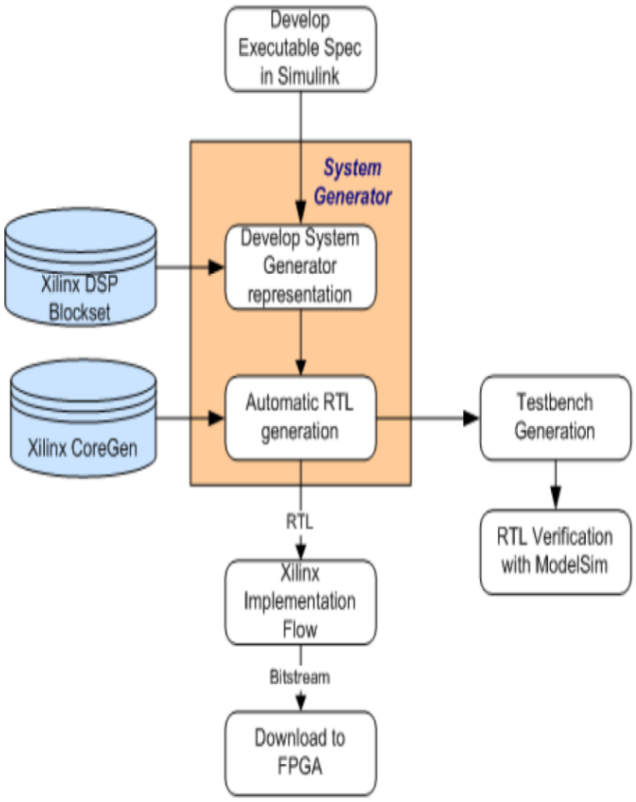

Fig.1. XSG Design flow

ANAGLYPH 3D IMAGE

Stereoscopic 3D illusion accomplished by concealing each eye's image using filters of dissimilar colours, usually red and cyan. The two differently filtered coloured images in anaglyph 3D image maps one for each eye. Having two photos capturing at an instant is the vital part in creating an anaglyph image. Both photos must be focused on the same object, sliding the camera horizontally between 3 and 5 cm for the next picture.

Anaglyph delivers a marginally distinct perspective to individual eyes. From the variance between the two viewpoints and other visual indications, the human optical system can provoke the stereoscopic depiction of spatial

correlation in the scene. To generate an anaglyph image, the left and right images of a stereo image pair are superimposed in discrete colour planes. The two images will be isolated from the amalgamated picture by colour filtering and fed to each eye. The red channel of the left image and the blue and green channel of the right image are fused to produce a red-cyan colour anaglyph image. It is essential to have coloured glasses which is devised by two unlike colours red and cyan. These glasses act as filters and permit each eye to see only what it deserves, thus, creating an illusion.

COLORSPACE CONVERSION

Color Space Conversion (CSC) [6] is the mathematical translation of the numerical representation of a color from one color cube definition to the other. In most of the videos there is a full bandwidth of RGB signals that were not determined to be to be economically efficient enough as a means for storage and broadcasting. That's why RGB signals are encoded to YCbCr , where primary colors red, blue, green are proceesed into images that closely resemble the original RGB image , but at a much lower bandwidth for transmission. In YCbCr space, an image is represented by one luma (Y) and two chroma (Cb, Cr) components. The luma channel contains brightness information; it is essentially a grey scale version of the image and the chroma values are color offsets. In YCbCr space, the bandwidth of an image tends to be concentrated in the Y Channel. This leaves the Cb and Cr channels with less information, so they can be represented with fewer bits. The human eye doesn't actually see equally well in the different color bands with our human-vision [9] system optimized for the red, green bands but not quite as sensitive to changes in blues. Scientist and engineers looking for ways to reduce the bandwidth and/or bit rate of a video system have created other color spaces (and sampling spaces) that reduce the amount of blue information in a system while maintaining a subjectively high picture quality. Therefore, many video systems sub-sample the color information [9] (chrominance) while transmitting the black and white (luminance) in full resolution. This subsampling is often applied to luminance-chrominance color space systems such as YCbCr where Y represents the luminance information and Cr and Cb are color difference signals that represent the chrominance information. In these systems all of the Y samples are used but every other color sample is dropped. These systems are referred to as 4:2:2 sampling. The 4:2:2 nomenclatures signify that for every 4 Y samples only 2 Cr and 2 Cb samples are saved. Owing to the bandwidth saving benefits of these different image formats different video equipment will adopt different color space encodings. Interoperability between such equipment often requires a device to convert the output of one video device in a given color space to the color space needed as input for the downstream device.

YCbCr Color Space was developed as part of the Recommendation ITU-R BT.601 [1] (International Telecommunication Union) for worldwide digital component video standard and is used in television transmissions. In this color model, the luminance component is separated from the color components. Component (Y) represents luminance, and chrominance information is stored as two color difference components. Color component Cb represent the difference between the blue component and a reference value and the color component Cr represents the difference between the red component and a reference value [1]. The basic equations to convert between RGB and YCbCr are:

$$Y = 0.299R + 0.587G + 0.114B + 16$$
$$Cb = -0.169R - 0.331G + 0.5B + 128$$
$$Cr = 0.5R - 0.419G - 0.081B + 128 \quad (1)$$

Among all the color models found, YCbCr seems to be better because humans are sensitive to brightness information than color information which is more in luminance (Y channel) when compared to chrominance (Cb and Cr channels). During video processing the luminance can be removed by converting the image from RGB color model to the YCbCr color model [1].

Implementation resluts and simulation

The model uses the top level HDL module and its Xilinx blockset for RGB to Y, Cb, and Cr components as shown in Figure 2. This model is used for co-simulation i.e., simulation in Matlab and the generated code from the system generator on the hardware. Once the design is verified, a hardware co-simulation block is generated and it will be used to program the FPGA for the CSC design implementation. Figure 2 shows the model with the hardware co-simulation block. The bit-stream download step is performed using a JTAG cable.

The RTL HDL Model generated is synthesized using Xilinx ISE 14.1 and targeted for Atlys FPGA Spartan 6 (xc6slx45-2csg324). The Software and Hardware results of the CSC deign are shown in figures 3 and 4 respectively.

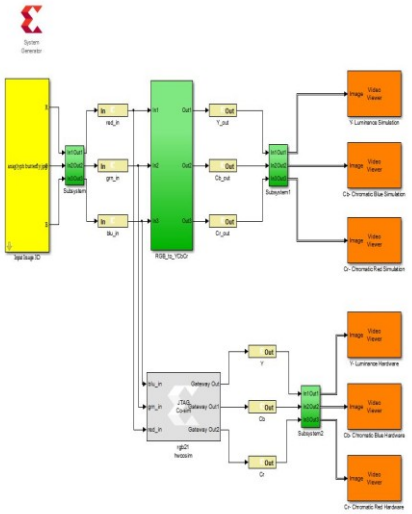

Fig.2. System generator project for H/W Co-Simulation

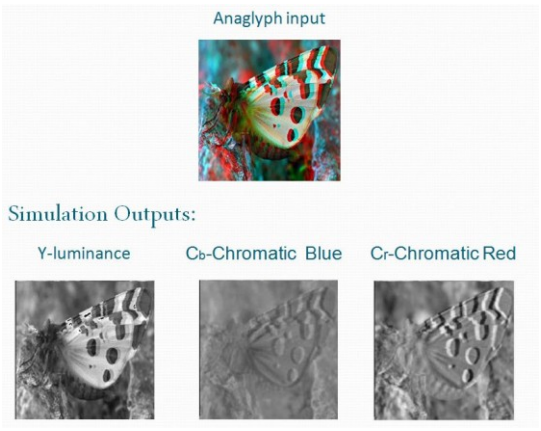

Fig.3. Simulation Results

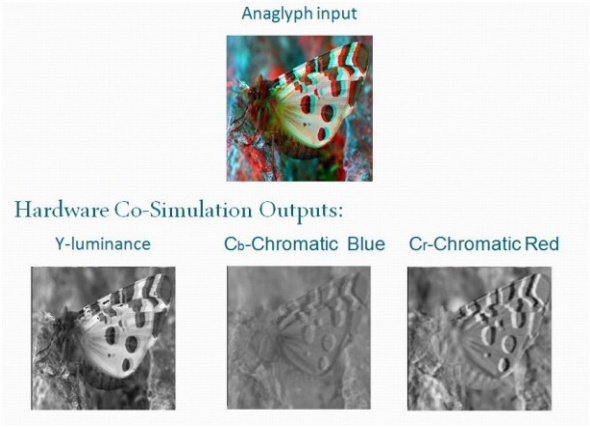

Fig.4. H/W Co-Simulation Results

The optimization setting is for maximum clock speed. Table 1 details the summary report of the resources utilized for the implementation of CSC design. For the implementation of the above model as shown in Fig.2 requires some additional blocks are needed for input/output interfaces, and synchronization. Our architecture spent about 423CLB with a frequency up to 244.573 MHz. Obviously, our proposed architecture has lowered complexity and area is improved efficiently, thus providing a good choice in terms of low-cost hardware.

TABLE I. RESOURCE UTILIZATION REPORT IN THE IMPLEMNTATION OF CSC DESIGN

TABLE II.

Name of the Component	***Resources used***	***Device usage***
Number of Slice Registers	809	1%
Number of Slice LUTs	524	1%
Number of fully used LUT-FF pairs	64	5%
Number of bonded IOBs	119	54%
Maximum Frequency	244.573 MHz	

CONCLUSION

The design is implemented on Atlys FPGA Spartan 6 target board with the help of XSG and Simulink IDE. It uses 64 FFs, 524 LUTs and 119 IOBs. The development time is less. It is less complex and highly flexible for prototyping and modifications. It is implemented with MatLab13.2 version and Xilinx ISE 14.1. The implementation of CSC design of 3D Anaglyph image is very high in terms of accuracy and performance as shown in results

References

[1] T. Saidani, D. Dia, W. Elhamzi, M. Atri and R. Tourki, "Hardware Co-simulation for Video Processing Using Xilinx System Generator". Proceedings of the World Congress on Engineering 2009 Vol I, WCE 2009, July 1 - 3, 2009, London, U.K.

[2] Xilinx System Generator User's Guide, www. Xilinx.com.

[3] T. M.: Embedded Matlab Language User Guide. The MathWorks Inc., 2007. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[4] Yahia Said, Taoufik Saidani, Mohamed Atri, "FPGA-based Architectures for Image Processing using High-Level Design," in WSEAS TRANSACTIONS on SIGNAL PROCESSING, Volume 11, 2015.

[5] S. Allin Christe, M. Vignesh and A. Kandaswamy, "An Efficient FPGA Implementation of MRI Image Filtering and Tumor Characterization Using Xilinx System Generator", International Journal of VLSI design & Communication Systems (VLSICS), vol. 2, no. 4, (2011) December, pp. 95-109 .

[6] M. Bilal and S. Masud, "Efficient color space conversion using custom instruction in a risc processor," in IEEE International Symposium on Circuits and Systems, 2007,pp. 1109 1112.

[7] 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 Digital Object Identifier Volume , Issue , 11-12 Jan. 2008 Page(s):255 – 258

[8] J.C.Moctezuma, S.Sanchez, R.Alvarez, A. Sánchez " Architecture for filtering images using Xilinx system generator" World Scientific Advanced Series In Electrical And Computer Engineering, Proceedings of the 2nd WSEAS International Conference on Computer Engineering and Applications. pages 284-289.2008.

[9] A. Albiol, L. Torres, and E. J. Delp, "An unsupervised color image segmentation algorithm for face detection applications," in Proceedings. 2001 International Conference on Image Processing, vol. 2, 2001, pp. 681–684 vol.2.

[10] F. Bensaali, A. Amira and A. Bouridane "Accelerating matrix product on reconfigurable hardware for image processing applications" IEE Proc.-Circuits Devices Syst., Vol. 152, No. 3, June 2005

[11] ITU-R BT.601-4, 2000, "Parameter Values for the HDTV Standards for Production and International Program Exchange", www.itu.int