International Journal of Research Studies in Science, Engineering and Technology
Volume 2, Issue 8, August 2015, PP 20-25
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

Novel Design of 2D-DCT Using System Generator for Compression of Images

Kurella Priyanka[1], Konda Kiran Kumar[2], CH. Madhan Mohan[3]

M.Tech VLSI & System Design
[1]*kurellapriyanka@gmail.com*, [2]*kiran3712@gmail.com*, [3]*madhanmohan.vlsi@gmail.com*

Abstract: *The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form. Compressing an image is significantly different than compressing raw binary data. In this paper we are implementing the JPEG image compression by using the 2D-DCT process. A DCT (Discrete Cosine Transform) based method is specified for 'lossy' compression, and a predictive method for 'Lossless' compression. Discrete Cosine Transform (DCT) is a mathematical tool that has a lot of electronics applications, from audio filters to video compression hardware. DCT transforms the information from the time or space domains to the frequency domain, such that other tools and transmission media can be run or used more efficiently to reach application goals. And here the design can be implemented by using the system generator environment. So that at sample level representation at the output level can't be a effective method for the image processing [1] applications. The efficiency of the design can be increased when we have the image representation at both input and output level. For this we are going for system generator environment for the better quality of the images and the output representations on FPGA. Here we are using mainly system generator as the tool which will perform both simulation and synthesis operations of the design. The system generator tool will have the backend as Xilinx environment and the editor which will appear to us as matlab.*

Keywords: *VHDL, System Generator (MATLAB, XILINX), Image compression, 2D-DCT, Simulink.*

1. INTRODUCTION

Desktop computers communicate information primarily via their screens, so for computer programmers and designers graphics are a major concern. Researches have been made to improve the way programs display data. One of the results of these researches was a vast array of computers capable of displaying complex graphical images which quality approaching television or magazines. In many areas of computing applications including games, education, desktop publishing, graphical design and most recently the World Wide Web programs using complex graphics are used. Although graphics do a great deal to enhance the usability and visual aesthetics of such applications, they consume prodigious amounts of disk storage. The answer to this problem is image compression.

2. DISCRETE COSINE TRANSFORM (DCT)

This section describes the brief about what is the purpose and advantage of DCT. It explains how to apply DCT. The key to the JPEG compression is a mathematical transformation known as the Discrete Cosine Transform (DCT). It includes also the well-known fast Fourier Transform (FFT). Its basic operation is to take a signal and transform it from one type of representation to another.

In this case the signal is a graphical image. The concept of this transformation is to transform a set of points from the spatial domain into an identical representation in frequency domain. It identifies pieces of information that can be effectively thrown away without reducing the image's quality. First table the DCT breaks the source image into N x N matrix or block down. In practice, N most often equals 8 because a larger block, though would probably give better compression, often takes a great deal of time to perform DCT calculations, creating an unreasonable tradeoff. As a result, DCT implementations typically break the image down into more manageable 8 x 8 blocks then we apply the discrete cosine transform on the matrix.

The mathematical function for a two-dimensional DCT is:

$$DCT(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixol(x,y) Cos\left[\frac{(2x+1)i\pi}{2N}\right] Cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

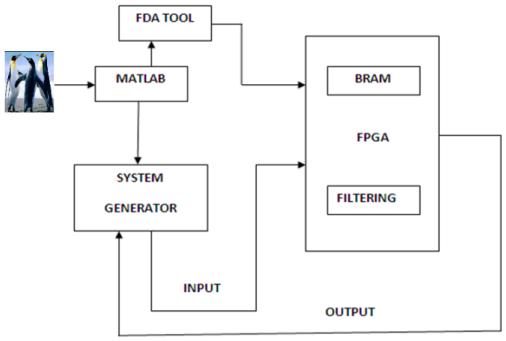

Fig1. *Architecture of the 2D-DCT design for image compression.*

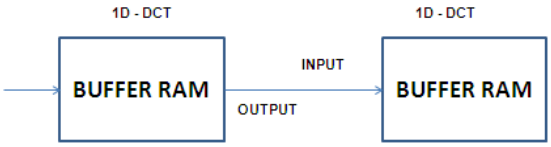

Fig2. *Buffer operation in the design.*

Fig3. *Input image for the design.*

3. SYSTEM GENERATOR

Now a day's Field Programmable Gate Array (FPGA) technology is having more importance for the algorithms which are suited to DSP applications. Field-programmable gate arrays (FPGAs) are non conventional processors built CLB's (configurable logic blocks) and these are connected by programmable interconnectors. Each CLB is having one or more lookup tables (LUTs) and memory. Hardware can be implemented on FPGA by interconnecting the CLB's, internally logic gates and registers form the circuit diagrams onto CLB's. CLB's having slices, slices having LUT's, flip-flops, logic gates, muxes.

MATLAB Simulink environment is having libraries which are having the built in blocks. By using this built in block we can have a design depends on application for modeling, simulating, and analyzing the dynamical systems. For DSP operations a tool called Xilinx System Generator [2] (XSG) which offers block libraries that plugs into Simulink tool. This tool containing accurate models of the FPGA's of particular math, logic, and DSP functions.

3.1. Xilinx System Generator

Xilinx System Generator is a DSP design tool from Xilinx. By using this tool we can have a model-based design for the FPGA hardware design. By using the Xilinx specific block set we can capture DSP friendly simulink modeling environment by this system-level modeling tool. Xilinx block set extends Simulink in many ways to provide a modeling environment that is well suited for hardware design on FPGA. The tool provides high-level abstractions that are automatically compiled into an FPGA at the push of a button. All the operations means the implementation steps which are synthesis, mapping, place & route will perform automatically to generate a bit stream file of programming file. Xilinx block set contains 90 DSP building blocks for simulink. By these Xilinx DSP blocks the design can be optimized for the selected device. Below figure4 and 5 shows the system generator symbol and system generator black box.

Fig4. *System generator symbol*

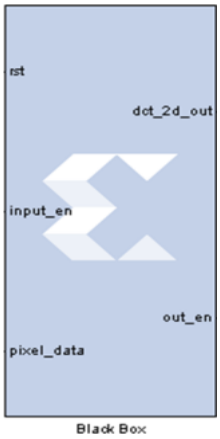

Fig5. *Black Box Symbol*

In system generator we have the black box environment. The black box is a Xilinx system generator block which it incorporates hardware description language (HDL) models into System Generator. In system generator this block provides both the simulation behavior in Simulink and the implementation files to be used during code generation. The ports which are mentioned in HDL file and attached to the black box produce and consume the same sorts of signals as other System Generator blocks. When a black box is translated into hardware, the associated HDL entity is automatically incorporated and wired to other blocks in the resulting design.

Either VHDL or Verilog can be incorporated with the black box into a Simulink model. The HDL which is incorporated with the Black box can be co-simulated with Simulink using the System Generator interface to either ISE simulator or the ModelSim simulation software.

The below figure shows the hardware co simulation environment on FPGA.

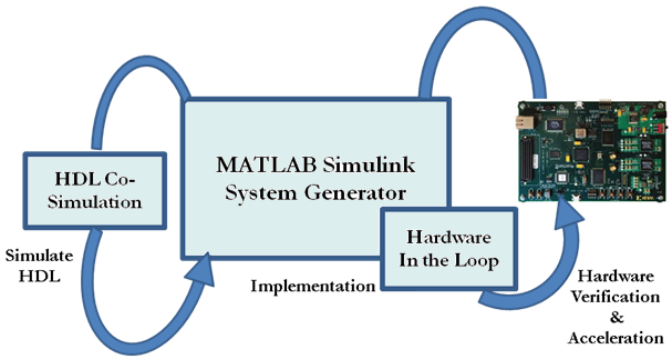

Fig6. *FPGA Based Hardware-Software Co-Simulation Environment*

3.2. Hardware Co-Simulation

The word Hardware co-simulation will comes at system generator environment. Hardware co-simulation is a hardware design technique. This technique offers that the simulation of the design into a hardware form. The design can be used to loaded and verified by this technique whether it is correctly working or not on the FPGA. ISim is used for complementary flow to the software-based HDL simulation.

3.3. Hardware-In-The-Loop

Hardware-in-the-loop (HIL) simulation is a technique which is used for the development and test of complex real-time embedded systems. In general the simulation process refers to the testing the design by applying different test inputs. In the same way in this hardware-in-the-loop concept we are testing our design before going to the hardware implementation on FPGA means hardware co-simulation process. The below figure shows the block level representation of the hardware in loop process using system generator on FPGA.

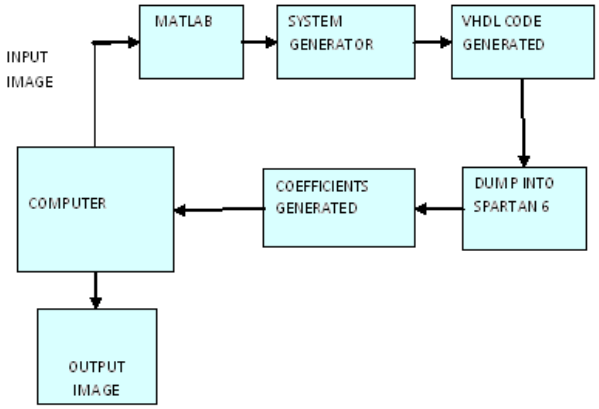

Fig7. *Hardware-In-Loop Implementation*

3.4. Advantages of DCT

- Less memory storage for the DCT format data or image.
- High correlation for images.
- DCT compresses the image easily and will give high quality image.
- The output image will have least amount of data.

4. SPARTAN 601 BOARD

Fig8. *SP601 Board*

The above figure shows SP601 FPGA kit. The SP601 board enables hardware and software developers to create or evaluate designs targeting the Spartan-6 XC6SLX16-2CSG324 FPGA. The SP601 provides board features which include DDR2 memory controller, a parallel linear flash, a tri-mode Ethernet PHY, general-purpose I/O (GPIO), and a UART. Here we have used matlab 2010ra, Xilinx 13.2i version for to get the system generator configuration. Xilinx system generator modelsim block provided the waveforms. By the Simulink block Image viewer, we have observed the output images according the implementation of 2D-DCT logic.

5. SIMULATION & HARDWARE RESULTS

The following figure9 and figure 10 shows the simulation and the hardware results of the entire system generator design and model file for the 2D-DCT [3] for the image compression. Figure 11

represents the sample level representation in the system generator by using Wave scope simulation environment. Figure 12 shows the output image of the design after compression.

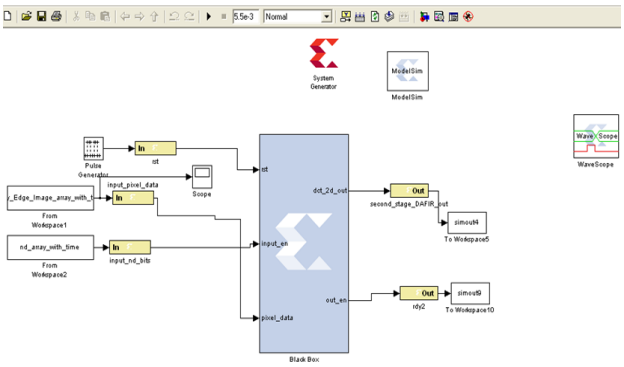

Fig9. *System generator model for simulation of the image compression by using DCT*

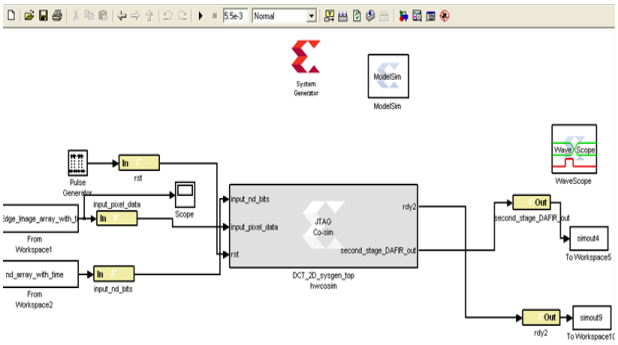

Fig10. *System generator model for hardware co-simulation of the image compression by using DCT*

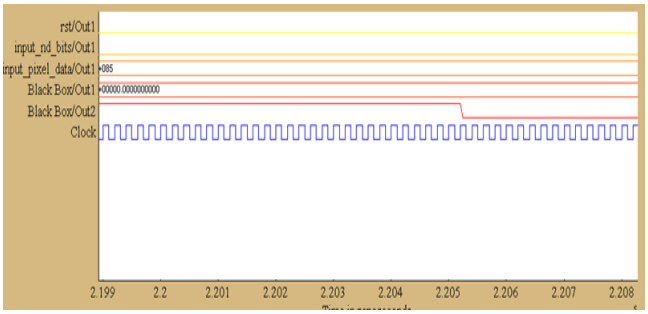

Fig11. *Modelsim Wave scope output.*

Fig12. *Output image*

6. TABLE

The following table represents the resource utilization and Speed representation on the SP601 FPGA of the current design of the image compression.

IP and configuration	Resource utilization		Speed (max freq)	Suitable usage
	Slices(reg) Used/available-%	DSP48 Used/available -%		
Image-2D-DCT filtering Device:Spartan-6	615/18224 - 3%	16/32 - 50%	100Mhz	Image processing application

7. CONCLUSION

Image compression is an extremely important part of modern computing. By having the ability to compress images to a fraction of their original size, valuable (and expensive) disk space can be saved. In addition, transportation of images from one computer to another becomes easier and less time consuming.

In this paper we have concluded that the image compression is done with the 2D-DCT by using Xilinx system generator. By using this system generator environment efficient analysis of the both input and output images will increase when compared to the sample level representation of the image. At VHDL level implementation we don't have GUI environment and we can not display the image for the both input and output. So that the image can be displayed by using the Xilinx system generator environment by using the black box environment.

REFERENCES

[1] Rafael C.Gonzalez, Richard E. Woods "Digital image processing" Prentice Hall publication, second edition.

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

[3] L. Agostini, S. Bampi, "Pipelined Fast 2-D DCT Architecture for JPEGTmage Compression". IEEE Computer Society, 2001.

AUTHORS' BIOGRAPHY

Kurella Priyanka, is presently pursuing final semester M. Tech in VLSI at Nagarjuna Institute of Technology & Sciences, Srinivasa Nagar (vill), V.T.Palem, Miryalguda, Nalgonda Dist, A.P., India.

Konda Kiran Kumar, working as an Assistant Professor at Nagarjuna Institute of Technology & Sciences. Technology & Sciences, Srinivasa Nagar (vill), V. T. Palem, Miryalguda, Nalgonda Dist, A.P., India.

CH. Madhan Mohan is presently working as Application Engineer in Unistring Tech Solutions Pvt Ltd, Hyderabad, Telangana, India.