

FPGA implementation of high throughput DA based DCT using System Generator

SIDDALA PAVANI S.NAGI REDDY, D.NAGA PURNIMA

1Research Scholar, Dept of ECE TKR Engineering College, Hyderabad, India.

2Ass Prof, Dept of ECE, TKR Engineering College, Hyderabad, India.

3Application engineer at Uni String Tech Solutions Pvt Ltd, Hyderabad, India.

Abstract—Images form the significant part of data, particularly in remote sensing, biomedical and video conferencing applications. A digital image bitmap can contain considerably large amounts of data causing exceptional overhead in both computational complexity as well as data processing. Storage media has exceptional capacity; however, access speeds are typically inversely proportional to capacity. A compressed image consumes less storage space. The bandwidth required is also less compared to uncompressed data. Hence there is a need of image compression mechanism. To meet the differing needs of many applications, the JPEG standard includes lossy compression (DCT based). The acronym JPEG stands for the Joint Photographic Experts Group. Grayscale image compression uses 2-D DCT [1], quantization and entropy coding. In this paper architecture for 2D-DCT is implemented. The filtering process of DCT is done by using VHDL coding by using MATLAB DCT coefficients. Input and output images can be observed through system generator. All these modules for JPEG compression [2] process are developed using VHDL language and system generator tool.

Tools & h/w used: MATLAB [6] and Xilinx ISE, Xilinx system generator, Xilinx FPGA board (Spartan 6).

Index Terms — DCT, System generator, VHDL, FPGA.

I. INTRODUCTION

A. Image Compression

In digital image processing [3] the image compression is having more importance in today's world. To meet a given target bit-rate for storage (and transmission), the compression ratio of loss less methods (e.g. Huffman, Arithmetic, and LZW) is not high enough for image and video compression, especially when the distribution of pixel values is relatively flat.

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. The 2D-DCT architecture is mentioned in below fig.1. 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: compact representation, fast transmission, memory savings, and so on.

B. Compression techniques

Image compression methods build redundancies both in the data and the non-linearity of the human vision. They exploited the correlation in space for still images and in both space and time for video signals. Compression in space is known as intra frame compression while compression in time e is known as inter frame compression. Generally compression methods with a high ratio are lossy in that the reconstructed data will not be identical to the original. The lossy algorithms also generally exploit the aspects of the human vision system. For instance the eye is more perceptive to find the details in the luminance (brightness) signals than to those in the chrominance (color) signals. Consequently the luminance signal is usually sampled at a higher spatial resolution.

The encoded (or compressed) representation of luminance signal receives more bits (a higher dynamic range) than that of a chrominance signals does. Also the eye is more sensitive to energy with low spatial spectral frequency than to energy high spatial frequency.

C. 2-Dimensional DCT architecture:

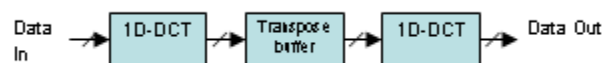


Fig.1: 2D-DCT architecture

The general equation for a 2D (N by M image) DCT is defined by the following equation:

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i) \cdot \Lambda(j) \cdot \cos \left[\frac{\pi \cdot u}{2 \cdot N} (2i + 1) \right] \cos \left[\frac{\pi \cdot v}{2 \cdot M} (2j + 1) \right] \cdot f(i, j)$$

II. Xilinx System Generator

Recently, Field Programmable Gate Array (FPGA) technology has become a viable target for the implementation of algorithms suited to Digital Signal Processing applications. Field-

programmable gate arrays (FPGAs) are nonconventional processors built primarily out of logic blocks connected by programmable wires. Each logic block has one or more lookup tables (LUTs) and several bits of memory. As a result, logic blocks can implement arbitrary logic functions (up to a few bits). Therefore FPGAs, as a whole can implement circuit diagrams, by mapping the gates and registers onto logic blocks.

With more than 1,000 built-in functions as well as toolbox extensions, MATLAB is an excellent tool for algorithm development and data analysis. An estimated 90% of the algorithms used today in DSP originate as MATLAB models. Simulink is a graphical tool, which lets a user graphically design the architecture and simulate the timing and behavior of the whole system. It augments MATLAB, allowing the user to model the digital, analog and event driven components together in one simulation.

MATLAB Simulink is an interactive tool for modeling, simulating, and analyzing dynamical systems. Using Simulink one can quickly build up models from libraries of pre-built blocks. Xilinx System Generator (XSG) for DSP is a tool which offers block libraries that plugs into Simulink tool (containing bit-true and cycle-accurate models of their FPGAs particular math, logic, and DSP functions). The AccelDSP tool allows DSP algorithm developers to create HDL designs from MATLAB and export them into System Generator for DSP.

Xilinx System Generator[4] is a DSP design tool from Xilinx that enables the use of the Math works model-based Simulink environment for FPGA design. It is a system-level modeling tool in which designs are captured in the DSP friendly Simulink modeling environment using a Xilinx specific Block set. It extends Simulink in many ways to provide a modeling environment that is well suited to hardware design. The tool provides high-level abstractions that are automatically compiled into an FPGA at the push of a button. All of the downstream FPGA implementation steps including synthesis and place and route are automatically performed to generate an FPGA programming file. Over 90 DSP building blocks are provided in the Xilinx DSP block set for Simulink. These blocks leverage the Xilinx IP core generators to deliver optimized results for the selected device.

System Generator[5] provides many features such as System Resource Estimation to take full advantage of the FPGA resources, Hardware Co-Simulation and accelerated simulation through hardware in the loop co-simulation; which give many orders of simulation performance increase. It also provides a system integration platform for the design of DSP FPGAs that allows the RTL, Simulink, MATLAB and C/C++ components of a DSP system to come together in a single simulation and implementation environment. Below Fig.2 presents the design flow of XSG.

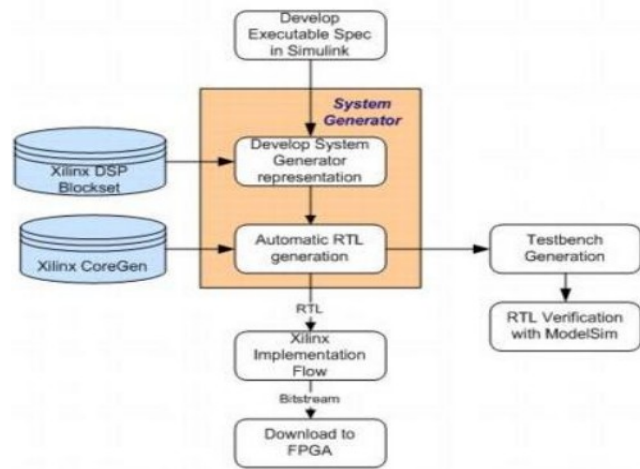


Fig.2: System Generator Flow

III. BRIEF EXPLANATION

A. Implementation:

The design was synthesized into an Xilinx Spartan 6 family FPGA. System is tested with real photographic image. The standard grey-scale image shown in below fig.3 is used as test bench to verify the design.



Fig.3: Image input

The proposed design is mentioned in below fig.4. The complete synthesis results of the design on Spartan-6 (architecture of sp-6 is shown in fig.5) FPGA are presented in below Table I, whose hardware was fit in device. System designed in uses Sparta-6 FPGA and straight forward multiplication without special algorithm.

IV. PROPOSED DESIGN

FPGA implementation for JPEG

Image compression is an important topic in the digital world. Whether it is commercial photography, industrial imagery, or video. A digital image bitmap can contain considerably large amounts of data causing exceptional overhead in both computational complexity as well as data processing. Storage media has exceptional capacity; however, access speeds are typically inversely proportional to capacity. To meet the differing needs of many applications, the JPEG standard includes two basic compression methods, each with various modes of operation. A DCT (Discrete Cosine Transform) based method is specified for 'lossy' compression, and a predictive method for 'lossless' compression.

DCT (Discrete Cosine Transform) 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: compact representation, fast transmission, memory savings, and so on. The JPEG compression procedure involves five steps: down sampling, 2-D DCT, quantization and entropy coding. Grayscale image compression uses only last three steps. This is shown in Fig.4.

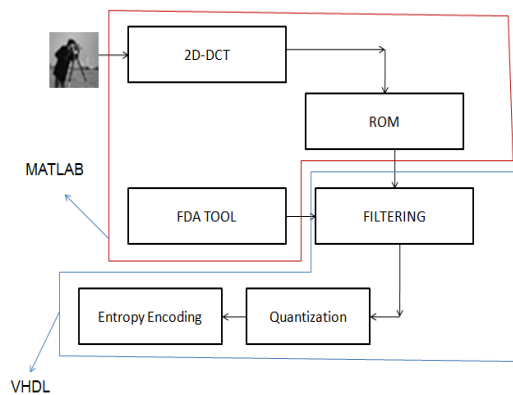


Fig.4: Block Diagram of the proposed Solution

By using the above implementation, an image of size 32X32 can be compressed. If the size of the image increases, then automatically the memory requirements will also increase. So, I am trying to implement the proposed solution in SPARTAN 6 Xilinx FPGA tool kit. By using the Intellectual Property of Xilinx, the BRAM is created for storing the coefficients. An image of size 256X256 is taken for the experiment.

A. Hardware-in-the-loop

Hardware-in-the-loop (HIL) simulation is a technique that is used in the development and test of complex real-time embedded systems. HIL simulation provides an effective platform by adding the complexity of the plant under control to the test platform. The complexity of the plant under control is included in test and development by adding a mathematical representation of all related dynamic systems. These mathematical representations are referred to as the "plant simulation". The embedded system to be tested interacts with this plant simulation. How HIL works. An HIL simulation must include electrical emulation of sensors and actuators. These electrical emulations act as the interface between the plant simulation and the embedded system under test. The value of each electrically emulated sensor is controlled by the plant simulation and is read by the embedded system under test (feedback). Likewise, the embedded system under test implements its control algorithms by outputting actuator control signals. Changes in the control signals result in changes to variable values in the plant simulation.

In many cases, the most effective way to develop an embedded system is to connect the embedded system to the real plant. In other cases, HIL simulation is more efficient. The metric of development and test efficiency is typically a formula that includes the following factors: 1. Cost 2. Duration, 3. Safety, 4. Feasibility.

The cost of the approach is be a measure of the cost of all tools and effort. The duration of development and testing affects the time-to-market for a planned product. Safety factor and development duration are typically equated to a cost measure. Specific conditions that warrant the use of HIL simulation include the following:

- Enhancing the quality of testing
- Tight development schedules
- High-burden-rate plant
- Early process human factor development

B. Hardware in loop implementation process:

The Hardware In-Loop in the proposed work is implemented as follows: here, the loop is formed between the hardware and the software. First, the coding is designed in MATLAB, will be executed and an MDL file will be created by using the system generator. Then, by using this system generator, the VHDL file will be created. Then this will be dumped into the hardware SPARTAN6, where the coefficients of the input image will be decoded. The output from the SPATTAN6 is again sent to the MATLAB where the output can be seen which is the compressed version of the original input image. Hence, forming a loop, justifying the name, Hardware-In-Loop. The below block diagram shows the HIL implementation.

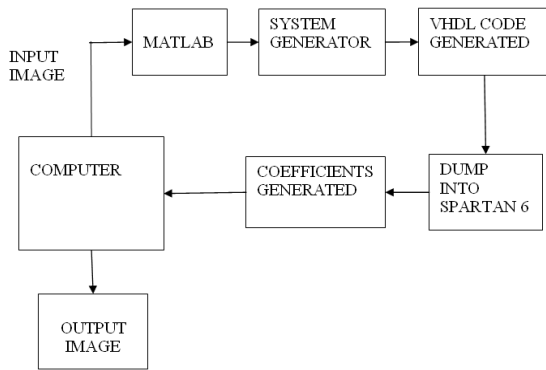


Fig.6. Hardware in loop process.

Spartan 601 board:

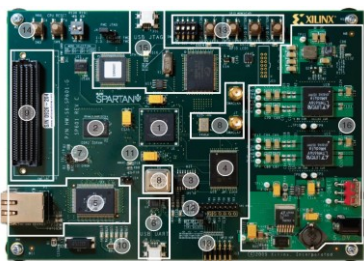


Fig.7. SP601 Board

Spartan 6 FPGA architecture:

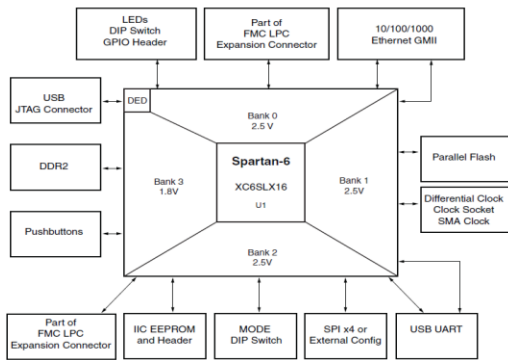


Fig.8. Block Diagram of SP601

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 for evaluating the Spartan-6 family that are common to most entry-level development environments.

B. Methodology:

This compression technique reduces the number of bits required to represent an image. The implementation of this technique considers the principle which states that the human eye is more sensitive to low frequencies than high frequencies.

Hence we neglect high frequency components in the image using the non-uniform quantization technique.

So by applying different coding techniques we reduce the number of bits to represent an image. This project makes use of system generator, hardware co-simulation, and hardware in-loop to compress the image. The same will be tested by using SPARTAN 6 Xilinx tool kit.

C. Significance of the work:

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. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This also means that lossy compression techniques can be used in this area. Xilinx 13.2 Edition, MATLAB 10.2 Version inter-related with system generator is used for simulation and synthesis respectively. Spartan 6.1 FPGA board will be used for testing and demonstration of the implemented system.

Applications of data compression are primarily in transmission and storage of information. Image transmission applications are in broadcast television, remote sensing via satellite, military communications via aircraft, radar and sonar, teleconferencing, computer communications, facsimile transmission, and the like.

Image storage is required for educational and business documents, medical images that arise in Computer Tomography (CT), Magnetic Resonance Imaging (MRI) and digital radiology, motion pictures, satellite images, weather maps, geological surveys, and so on. Application of data compression is also possible in the development of fast algorithms where the number of operations required implementing and algorithm is reduced by working with the compressed data.

V. RESULTS

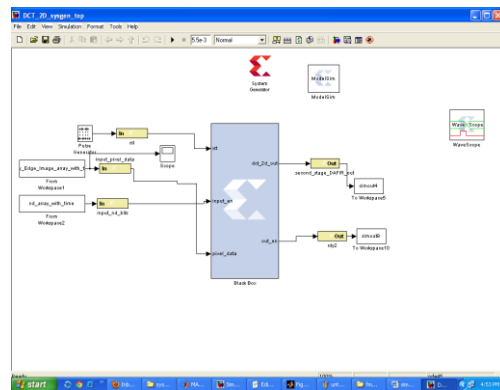


Fig.9. System generator model for simulation of the image compression by using DCT

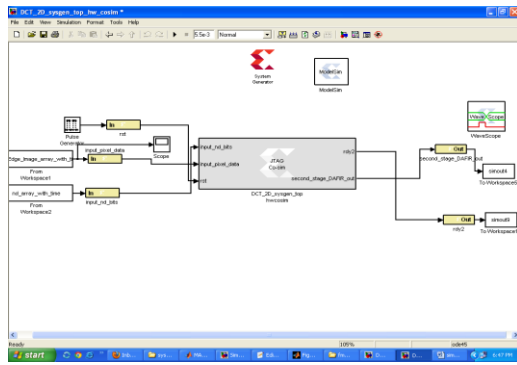


Fig.10. Final system generator model for the proposed design

Advantages of DCT

1. The DCT, transform the data or image into format that can be easily compressed. Reducing digitized image into the least amount of data.
2. The DCT format data or image required less memory storage.
3. For image coding, the DCT works well because of high correlation among adjacent data samples (pixel values). Because of this correlation, the DCT provides near optimal reduction while retaining high image quality.
4. It has a mathematical simplicity, which enables its rapid computation both in software and state of the art hardware, and its outstanding ability to accurately describe second-order statistics of digital audio and video signals.

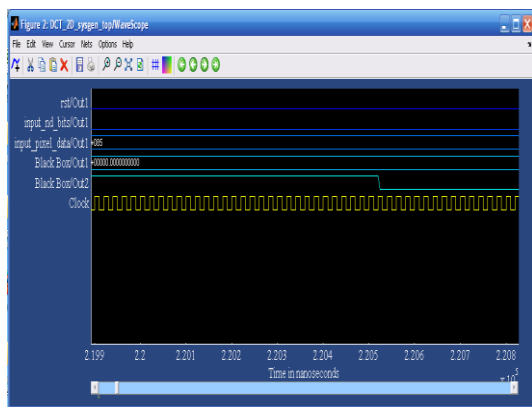


Fig.11. Simulation results of model file

VI. CONCLUSIONS

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 (this is why image compression has played such an important role in development of the internet).

In this work, the implementation of JPEG encoder architecture for JPEG image compression standard is described. The architectures for the various stages are based on efficient and high performance designs suited for VLSI implementation. The implementation was tested for functional correctness using VHDL with Xilinx tool and on MATLAB. The design is tested with grey scale image by using System Generator Hardware Co-Simulation and Hardware In-Loop. Maximum frequency can be achieved by this system is 200MHz. The design takes less device resources and suitable for FPGA like Xilinx SP601. The latency produced by design is less compared to previous works. Finally it is designed as a balanced architecture to previous works.

TABLE I

DEVICE UTILISATION USING XILINX SPARTAN-6

LOGIC UNIT	USED	AVAILABLE
Number of slice	394	2,278
Number of LUT FF's	765	1,198
Number of slice LUT's	1,052	9,112
Number of bonded IOB'S	2	232

REFERENCES

- [1] Enas Duhri Kusuma, Thomas Sri Widodo "FPGA Implementation of Pipelined 2D-DCT and Quantization Architecture for JPEG Image Compression" IEEE, 2010.
- [2] L. Agostini, S. Bampi, "Pipelined Fast 2-D DCT Architecture for JPEGImage Compression". IEEE Computer Society, 2001.
- [3] Rafael C.Gonzalez, Richard E. Woods "Digital image processing" Prentice Hall publication, second edition.
- [4] Xilinx System Generator User's Guide, www.xilinx.com.
- [5] S. Mittal, S. Gupta, and S. Dasgupta, "System Generator: The State-Of-Art FPGA Design Tool for DSP Applications", Third International Innovative Conference on Embedded Systems, Mobile Communication and Computing (ICEMC2 2008), August 11-14, 2008, Global Education Center, Infosys.



SIDDALA PAVANI received her B.Tech Electronics and Communication Engineering in 2012 and presently pursuing M.Tech ECE in TKR Engineering college Her research interests are low power VLSI design, Circuit Design.



S.NAGI REDDY asst Prof, ECE Dept, TKR Engineering College, Hyderabad, INDIA.



D.NAGA PURNIMA received her B.Tech ECE in 2010 .now working as application Eng at **Uni String Tech Solutions Pvt Ltd**, Hyderabad -INDIA.