RESEARCH ARTICLE OPEN ACCESS

FPGA based JPEG Encoder

Krupali Lanjewar*, Prof. Dr. R. S. Kawitkar**
**(Department of Electronics &Telecommunication S.C.O.E.Vadgaon, Pune, India*
*** (Department of Electronics &Telecommunication S.C.O.E.Vadgaon, Pune, India*

ABSTRACT
Compression is playing a vital role in data transfer. Hence, Digital camera uses JPEG standard to compress the captured image. Hence, it reduces data storage requirements. Here, we proposed FPGA based JPEG encoder. The processing system is coupled with DCT and then it is quantized and then it is prepared for entropy coding to form a JPEG encoder.
Keywords: DCT, Entropy coding, FPGA, JPEG, Quantization.

I. INTRODUCTION

JPEG 2000 standard is designed to meet the needs of variety of applications such as multimedia, medical imaging etc. He proposed an optimized architecture of bit plane coder for Embedded Block Coding with Optimal Truncation (EBCOT) algorithm targeting its FPGA implementation. Although several speed up techniques exist, we present architecture whose performance is improved based on detailed analysis of data path used to obtain context windows. His proposed design works at 67 MHz after post placement and routing on Xilinx XC2V1000 device. [1]

Digital image compression is the most important part in the multimedia applications which aims to reduce the number of bits in an image data for its efficient storage (less storage area). Discrete cosine transform (DCT) is usually used in JPEG based image transform coding. [2] He proposed separable 2-D discrete Hartley transform (SDHT) and its Distributed Arithmetic (DA) based hardware architecture as an alternate to DCT in transform based coding of image compression. The proposed DA architecture for 1-D DHT has very less computations as compared to existing 1-D DCT. The proposed DHT architecture implemented in FPGA indicates a significant hardware savings as compared to FPGA resources used in an efficient memory based DA approach. The additional advantage of SDHT is that its inverse transform is same as forward transform with a constant division. This is demonstrated through a Xilinx FPGA XC2VP30 device. [2]

II. PROPOSED SYSTEM DESIGN

The Proposed system consists of three main steps as DCT, Quantization and entropy encoding. A summary of the characteristics of the baseline coding processes is given as:

1. DCT-based process
2. Source image : 8-bit samples within each component
3. Sequential process
4. Huffman coding : 2AC and 2DC tables

Figure 1 shows the main procedures for all encoding processes based on the DCT.

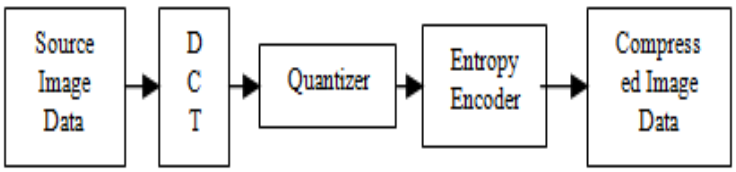

Figure1. Block Diagram of the system

2.1. Source Image Data

Figure 2 indicates the orientation of an image component by the terms top, bottom, left, and right. The order by which the data units of an image component are input to the compression encoding procedures is defined to be left-to-right and top-to-bottom within the component. [3]
Samples

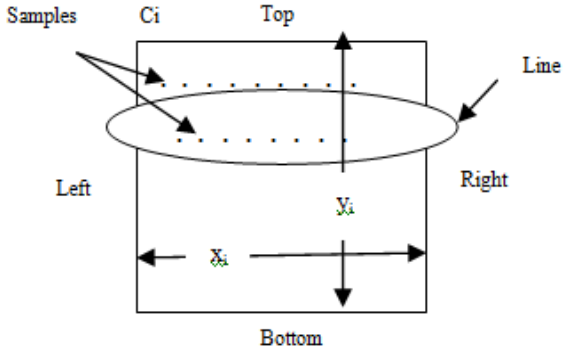

Figure2. Characteristics of an image component

The input image is bitmap file. It must be store as "image.bmp". Windows bitmap files are stored in a device-independent bitmap (DIB) format that allows Windows to display the bitmap on any type of display device. The term "device independent" means that bitmap specifies pixel color in a form independent of the method used by a display to represent color. The default filename extension of a Windows DIB file is .BMP. Here, Image resolution is not limited. It takes an RGB input (row-wise). The header is the widely employed JFIF. JPEG File Interchange Format is a minimal file format which enables JPEG bit streams to be exchanged between a wide variety of platforms and applications. Although any JPEG process is supported by the syntax of the JPEG File Interchange Format (JFIF) it is strongly recommended that the JPEG baseline process be used for the purposes of file interchange. This ensures maximum compatibility with all applications supporting JPEG. JFIF conforms to the JPEG Draft International Standard (ISO DIS 10918-1).

2.1.1. Image Orientation

In JFIF files, the image orientation is always top-down. This means that the first image samples encoded in a JFIF file are located in the upper left hand corner of the image and encoding proceeds from left to right and top to bottom.

2.2. Discrete Cosine Transform (Dct)

There are four distinct modes of operation under which the various coding processes are defined:

1. Sequential DCT-based,
2. Progressive DCT-based,
3. Lossless, and
4. Hierarchical

The simplest DCT-based coding process is referred to as the baseline sequential process. It provides a capability which is sufficient for many applications. There are additional DCT-based processes which extend the baseline sequential process to a broader range of applications. [3] Among these processes, we have used sequential DCT-based operation. For the sequential DCT-based mode, 8 x 8 sample blocks are typically input block by block from left to right and block-row by block-row from top to bottom.

2.2.1 Level shift

Before encoding process computes the forward DCT for a block of source image samples, the samples will be level shifted to a signed representation by subtracting 2P – 1, where P is the precision parameter specified.

2.3. Quantization

Quantization is done to achieve better compression. Quantization reduces the number of bits needed to store information by reducing the size of the integers representing the information. After the FDCT is computed for a block, each of the 64 resulting DCT coefficients is quantized by a uniform quantizer.

The uniform quantizer is defined by the following equation. Rounding is to the nearest integer:

$$Sq_{vu} = round\ \frac{S_{vu}}{Q_{vu}} \qquad (1)$$

Where,

Sq_{vu} is the quantized DCT coefficient, normalized by the quantizer step size.

2.3.1 Differential DC encoding

After quantization, and in preparation for entropy encoding, the quantized DC coefficient is treated separately from the 63 quantized AC coefficients. The value that shall be encoded is the difference (DIFF) between the quantized DC coefficient of the current block (DC_i) and that of the previous block of the same component (PRED):

$$DIFF = DC_i - PRED \qquad (2)$$

2.4. Entropy Encoding

There are two types of the entropy-coding :

1. Huffman encoding
2. arithmetic encoding

The baseline sequential process uses Huffman coding hence, Huffman encoding is used.

2.4.1 Zig-zag sequence

After quantization, and in preparation for entropy encoding, the quantized AC coefficients are converted to the zig-zag sequence. The quantized DC coefficient (coefficient zero in the array) is treated separately. The zigzag sequence is specified in Figure 3.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Figure 3 Zig-zag sequences of quantized DCT coefficients

2.4.2 Conversion of Huffman table specifications to tables of codes and code lengths

Conversion of Huffman table specifications to tables of codes and code lengths uses three procedures.

1. generates a table of Huffman code sizes.
2. generates the Huffman codes
3. generates the Huffman codes in symbol value order.

The entries in the tables are ordered according to increasing Huffman code numeric value and length.

2.4.3 Bit ordering within bytes

The root of a Huffman code is placed toward the MSB (most-significant-bit) of the byte, and successive bits are placed in the direction MSB to LSB (least-significant-bit) of the byte. In this way, Huffman encoding starts by looking up the tables and stores the final compressed image.

III. IMPLEMENTATION AND METHODOLOGY

The whole system is divided into six parts as follows:

1. Reading an 24- bit uncompressed BMP file.
2. Converting the input signals (Red, Green and Blue, strobed by Process RGB signal) to the YCbCr color space
3. Discrete Cosine Transform
4. Quantisation
5. Entropy coding
6. Compressed image file

Xilinx CORE generator is used in project. The CORE Generator System is a design tool that delivers parameterized cores optimized for Xilinx® FPGAs. It provides you with a catalog of ready-made functions ranging in complexity from simple arithmetic operators such as adders, accumulators, and multipliers, to system-level building blocks such as filters, transforms, FIFOs, and memories. A core cannot be remotely accessed from within a project. Cores must reside inside the project directory in order to be accessible to the project. Coregen is not available for CPLD devices. It is not possible to copy cores from one project to another. You will need to generate new cores if you import a design from another design environment. CoreGen cores are often optimized for a particular device family. To avoid errors in implementation after changing to a new device family, most cores need to be regenerated for the new family.

IV. CONCLUSION

Study of different image interchange format, image header format and design implementation flow of CoreGen is carried out. The synthesis is carried out on Vertex 5 platform. The JPEG Encoder implementation on Vertex 5 infers 13185 gate counts. This design works on a frequency 29.046MHz.JPEG standard is used in digital camera which is used to compress the captured image. Hence less space is needed to store the image. Such images are easily shared on the web.

REFERENCES

[1] A high bit plane coder for JPEG 2000 and its FPGA implementations by Kishor Sarawadekar and Swapna Banerjee Department of E & ECE, I.I.T. Kharagpur
[2] 2-D Separable Discrete Hartley Transform Architecture for Efficient FPGA Resource by Vijay Kumar Sharma, Richa Agrawal, U. C. Pati, K. K. Mahapatra Dept. of Electronics & Comm. Engg. National Institute of Technology, Rourkela, India.
[3] JPEG (ITU – T 81 standard):
[4] http://www/w3.org/Graphics/JPEG/itu-t81.pdf
[5] JFIF (JPEG file headers): http://www.w3.org/Graphics/JPEG/jfif3.pdf