

Performance Analysis of 2D DCT for Ultra High Definition Video Coding

G.Latha¹, B.Suresh²

M.Tech [Scholar], Department of ECE, Amrita Sai Institute of Science And Technology, Vijayawada, AP, India¹

Assistant Professor, Department of ECE, Amrita Sai Institute of Science And Technology, Vijayawada, AP, India²

Abstract: Data compression is the reduction or elimination of redundancy in data representation in order to achieve savings in storage and communication costs. Data compression techniques can be broadly classified into two categories: Lossless, Lossy schemes. In lossless methods, the exact original data can be recovered while in lossy schemes a close approximation of the original data can be obtained. The lossless method is also called entropy coding schemes since there is no loss of information content during the process of compression. Digital images require an enormous amount of space for storage. This work is to design VLSI architecture for the JPEG Baseline Image Compression Standard. The architecture exploits the principles of pipelining and parallelism to the maximum extent in order to obtain high speed the architecture for discrete cosine transforms and the entropy encoder are based on efficient algorithms designed for high speed VLSI. In recent years, a working group known as Joint Photographic Expert Group (JPEG) has defined an international standard for coding and compression of continuous tone still images. This standard is commonly referred to as the JPEG standard. Two dimensional DCT takes important role in compression. Architecture and VERILOG design of 2-D DCT, combined with quantization and zig-zag arrangement, is implemented in this project. DCT calculation used in this project is made using scaled DCT. The output of DCT module needs to be multiplied with post-scaler value to get the real DCT coefficients. Post-scaling process is done together with quantization process. 2-D DCT is computed by combining two 1-D DCT that connected by a transpose buffer. In this project the Modelsim is used for simulation.

Keywords: Model Sim, DCT, Verilog

I. INTRODUCTION

Despite the many advantages of digital representation of signals compared to the analog counterpart, they need a very large number of bits for storage and transmission. For example, a high-quality audio signal requires approximately 1.5 megabits per second for digital representation and storage. A television-quality low-resolution color video of 30 frames per second with each frame containing 640 x 480 pixels (24 bits per color pixel) needs more than 210 megabits per second of storage. As a result, a digitized one-hour color movie would require approximately 95 gigabytes of storage. The storage requirement for upcoming high-definition television (HDTV) of resolution 1280 x 720 at 60 frames per second is far greater. A digitized one-hour color movie of HDTV-quality video will require approximately 560 gigabytes of storage. A digitized 14 x 17 square inch radiograph scanned at 70 pm occupies nearly 45 megabytes of storage. Transmission of these digital signals through limited bandwidth communication channels is even a greater challenge and sometimes impossible in its raw form. Although the cost of storage has decreased drastically over the past decade due to significant advancement in microelectronics and storage technology, the requirement of data storage and data processing applications is growing explosively to outpace this achievement.

Interestingly enough, most of the sensory signals such as still image, video, and voice generally contain significant amounts of superfluous and redundant information in their canonical representation as far as the human perceptual

system is concerned. By human perceptual system, we mean our eyes and ears. For example, the neighboring pixels in the smooth region of a natural image are very similar and small variations in the values of the neighboring pixels are not noticeable to the human eye. The consecutive frames in a stationary or slowly changing scene in a video are very similar and redundant. There are many other secondary advantages in data compression. For example, it has great implications in database access. Data compression may enhance the database performance because more compressed records can be packed in a given buffer space in a traditional computer implementation. This potentially increases the probability that a record being searched will be found in the main memory. Data security can also be greatly enhanced by encrypting the decoding parameters and transmitting them separately from the compressed database files to restrict access of proprietary information. An extra level of security can be achieved by making the compression and decompression processes totally transparent to unauthorized users.

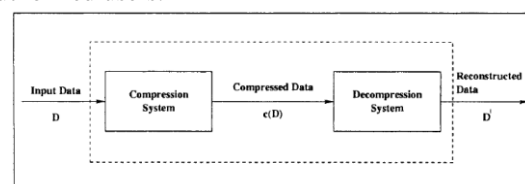


Fig.1: CODEC Block Diagram

The rate of input-output operations in a computing device can be greatly increased due to shorter representation of

data. In systems with levels of storage hierarchy, data compression in principle makes it possible to store data at a higher and faster storage level (usually with smaller capacity), thereby reducing the load on the input-output channels. Data compression obviously reduces the cost of backup and recovery of data in computer systems by storing the backup of large database files in compressed form.

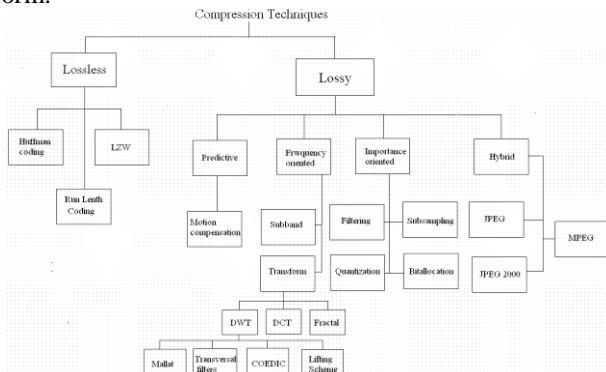


Fig.2: Classification of compression techniques

The advantages of data compression will enable more multimedia applications with reduced cost and hence aid its usage by a larger population with newer applications in the near future.

II. DISCRETE COSINE TRANSFORMATION

After each input 8x8 block of pixels is transformed to frequency space using the DCT, the resulting block contains a single DC component, and 63 AC components. The DC component is predictive encoded through a difference between the current DC value and the previous. This mode only uses Huffman coding models, not arithmetic coding models which are used in JPEG extensions. This mode is the most basic, but still has a wide acceptance for its high compression ratios, which can fit many general applications very well.

Loss less Mode

Quite simply, this mode of JPEG experiences no loss when comparing the source image, to the reproduced image. This method does not use the discrete cosine transform, rather it uses predictive, differential coding. As it is loss less, it also rules out the use of quantization. This method does not achieve high compression ratios, but some applications do require extremely precise image reproduction.

Base Line Jpeg Compression

The baseline JPEG compression algorithm is the most basic form of sequential DCT based compression. By using transform coding, quantization, and entropy coding, at an 8-bit pixel resolution, a high-level of compression can be achieved. However, the compression ratio achieved is due to sacrifices made in quality. The baseline specification assumes that 8-bit pixels are the source image, but extensions can use higher pixel resolutions. JPEG assumes that each block of data input is 8x8 pixels, which are serially input in raster order.

Baseline JPEG compression has some configurable portions, such as quantization tables, and Huffman tables.. By studying the source images to be compressed, Huffman codes and quantization codes can be optimized to reach a higher level of compression without losing more quality than is acceptable. Although this mode of JPEG is not highly configurable, it still allows a considerable amount of compression. Furthermore compression can be achieved by sub sampling chrominance portions of the input image, which is a useful technique playing on the human visual system.

Level Shift

In order to make the data fit the discrete cosine transform, each pixel value is level shifted by subtracting 128 from its value. The result of this is 8-bit pixels that have the range of -127 to 128, making the data symmetric across 0. This is good for DCT as any symmetry that is exposed will lead toward better entropy compression. Effectively this shifts the DC coefficient to fall more in line with value of the AC coefficients. The AC coefficients produced by the DCT are not affected in any way by this level shifting.

Discrete Cosine Transform (DCT):

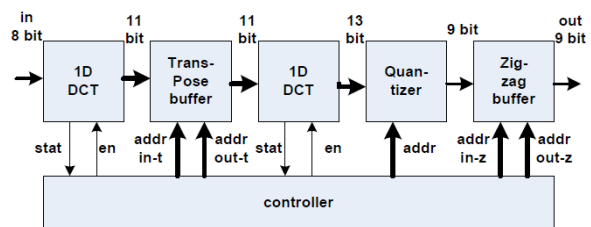


Fig.3: DCT Implementation

The discrete cosine transform is the basis for the JPEG compression standard. For JPEG, this allows for efficient compression by allowing quantization on elements that is less sensitive. The DCT algorithm is completely reversible making this useful for both loss less and lossy compression techniques.

The DCT is a special case of the well-known Fourier transform. Essentially the Fourier transform in theory can represent a given input signal with a series of sine and cosine terms. The discrete cosine transform is a special case of the Fourier transform in which the sine components are eliminated. For JPEG, a two-dimensional DCT algorithm is used which is essentially the one-dimensional version evaluated twice. By this property there are numerous ways to efficiently implement the software or hardware based DCT module. The DCT is operated two dimensionally taking into account 8 by 8 blocks of pixels. The resulting data set is an 8 by 8 block of frequency space components, the coefficients scaling the series cosine terms, known as basis functions. The First element at row 0 and column 0, is known as the DC term, the average frequency value of the entire block. The other 63 terms are AC components, which represent the spatial frequencies that compose the input pixel block, by scaling the cosine terms within the series. There are two useful products of the DCT algorithm. First it has the

ability to concentrate image energy into a small number of coefficients. Second, it minimizes the interdependencies between coefficients. These two points essentially state why this form of transform is used for the standard JPEG compression technique. By compacting the energy within an image, more coefficients are left to be quantized coarsely, impacting compression positively, but not losing quality in the resulting image after decompression. Taking away inter-pixel relations allows quantization to be non-linear, also affecting quantization positively. DCT has been effective in producing great pictures at low bit rates and is fairly easy to implement with fast hardware based algorithms.

An orthogonal transform such as the DCT has the good property that the inverse DCT can take its frequency coefficients back to the spatial domain at no loss. However, implementations can be lossy due to bit limitations and especially apparent in those algorithms in hardware. The DCT does win in terms of computational complexity as there are numerous studies that have been completed in different techniques for evaluating the DCT. The discrete cosine transform is actually more efficient in reconstructing a given number of samples, as compared to a Fourier transform. By using the property of orthogonality of cosine, as opposed to sine, a signal can be periodically reconstructed based on a fewer number of samples. Any sine based transform is not orthogonal, and would have to take Fourier transforms of more numbers of samples to approximate a sequence of samples as a periodic signal.

As the signal we are sampling, the given image, there is actually no real periodicity. If the image is run through a Fourier transform, the sine terms can actually incur large changes in amplitude for the signal, due to sine not being orthogonal. DCT will avoid this by not carrying this information to represent the changes. In the case of JPEG, a two-dimensional DCT is used, which correlates the image with 64 basis functions.

The DCT equation can be represented in matrix format. The matrix of DCT is

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 & 0 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 & 0 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 & 0 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 & 0 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 & 0 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 & 0 \end{bmatrix}$$

The first row (i: j) of the matrix has all the entries equal to 1/8 as expected from The columns of T form an orthogonal set, so T is an orthogonal matrix. When doing the inverse DCT the orthogonality of T is important, as the inverse of T is Tr, which is easy to calculate.

III. IMPLEMENTATION

The generalized implementation flow diagram of the project is represented as follows. Initially the market research should be carried out which covers the previous version of the design and the current requirements on the design. Based on this survey, the specification and the architecture must be identified. Then the RTL modeling

should be carried out in VHDL with respect to the identified architecture. Once the RTL modelling is done, it should be simulated and verified for all the cases. The functional verification should meet the intended architecture and should pass all the test cases.

Once the functional verification is clear, the RTL model will be taken to the synthesis process. Three operations will be carried out in the synthesis process such as

- a) Translate
- b) Map
- c) Place and Route

The developed RTL model will be translated to the mathematical equation format which will be in the understandable format of the tool. These translated equations will be then mapped to the library that is, mapped to the hardware. Once the mapping is done, the gates were placed and routed. Before these processes, the constraints can be given in order to optimize the design. Finally the BIT MAP file will be generated that has the design information in the binary format which will be dumped in the FPGA board.

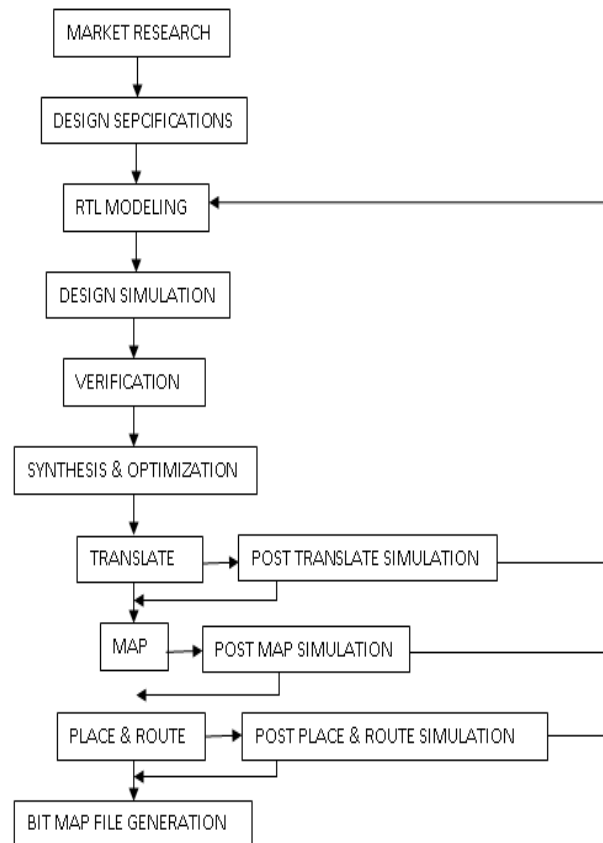


Fig.4: Flow Chart of Implementation

IV. RESULTS

The test bench is developed in order to test the modeled design. This developed test bench will automatically force the inputs and will make the operations of algorithm to perform. The results are presented in the fig.5 through fig.7.

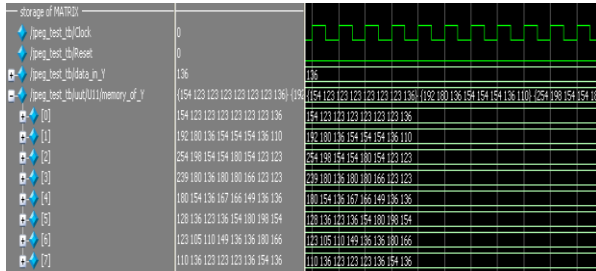


Fig.5: Storage of all Pixel Values

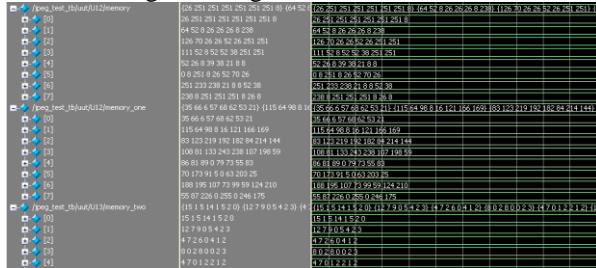


Fig.6: Discrete Cosine Transform

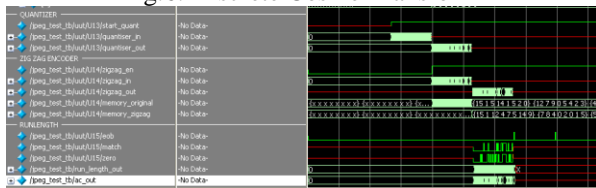


Fig.7: Quantization and Encoding

V. CONCLUSION

Basing on the results presented in fig.5 through fig.7 the following conclusion can be arrived. Two high speed FPGA architectures for the 2D DCT computation are presented in this paper. Both architectures use Distributed Arithmetic in order to replace the multipliers that are needed by the DCT algorithm. In the 2D_DCT_2ROM architecture two ROMs are used in order to double the performance, while in the 2D_DCT_4ROM four ROMs are used to quadruple the performance. The synthesis results prove that both architectures are very good choices for applications of high time performance requirements.

REFERENCES

- [1] J. Lee and H. Kalva, "Video Coding Techniques and Standards", In Furht B. (Ed.) Encyclopedia of Multimedia, Springer-Verlag Berlin Heidelberg, 2008.
- [2] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, Issue 7, pp. 560-576, July 2003.
- [3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp.1649-1668, December 2012.
- [4] S. A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review", IEEE ASSP Magazine, Vol. 6, Issue. 3, pp. 4-19, July 1989.