ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

Efficient Image Compression Using Two Dimensional Discrete Cosine Transformation Technique

N.A.V. Kumar[1], **G.A. Rao**[2]

Department of ECE, Avanthi Institute of Engineering & Technology, Visakhapatnam, India [1]

Department of ECE, Avanthi Institute of Engineering & Technology, Visakhapatnam, India [2]

Abstract: The technique of compression reduces the data storage requirements while offering an attractive approach to reduce the communication cost in transmitting high volumes of data over long-haul links via higher effective utilization of the available bandwidth in the data links. In this work we report an efficient method utilising the advantages of 2D Discrete Cosine Transformation (DCT). The proposed pipelined 2D DCT exploits the principles of pipelining and parallelism to obtain high speed the architecture designed for high speed VLSI.

Keywords: 2D DCT, VLSI, compression.

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 neighbouring pixels in the smooth region of a natural image are very similar and small variations in the values of the neighbouring 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.

Some audio data beyond the human audible frequency range are useless for all practical purposes. This fact tells us that there are data in audio-visual signals that cannot be perceived by the human perceptual system. We call this perceptual redundancy. In English text files, common words (e.g., “the”) or similar patterns of character strings (e.g., “ze”, “th”) are usually used repeatedly. It is also observed that the characters in a text file occur in a well-documented distribution, with letter e and “space” being the most popular. In numeric data files, we often observe runs of similar numbers or predictable interdependency among the numbers. We have mentioned only a few examples here. There are many such examples of redundancy in digital representation in all sorts of data.

Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and hence communication costs. Reducing the storage requirement is equivalent to increasing the capacity of the storage medium and hence communication bandwidth. Thus the development of efficient compression techniques will continue to be a design challenge for future communication systems and advanced multimedia applications'.

II. IMAGE COMPRESSION

Image compression is an important topic in the digital world, whether it can be commercial photography, industrial imagery, or video [3]. A digital image bitmap can contain considerably large amounts of data causing exceptional overhead in both computational complexity as well as data processing. Compression is important to manage large amounts of data for network, Internet, or storage media.

Data compression itself is the process of reducing the amount of information into a smaller data set that can be used to represent, and reproduce the information. Types of image compression include loss less compression, and lossy compression techniques that are used to meet the needs of specific applications [3]. JPEG compression can be used as a loss less or a lossy process depending on the requirements of the application both lossless and lossy compression techniques employ reduction of redundant data.

The Joint Photographic Experts Group produced the well-known image format JPEG, a widely used image format. JPEG provides solid baseline compression algorithm that can be modified numerous ways to fit any desired application. The JPEG specification was released initially in 1991, although it does not specify a particular implementation.

A. Lossless Compression

Loss less compression techniques work by removing redundant information as well as removing or reducing information that can be recreated during decompression. Loss less compression is ideal, as source data will be recreated without error. However, this leads to small compression ratios and will most likely not meet the needs of many applications. Compression ratios are highly dependent on input data, thus loss less compression will not meet the requirements of applications requiring a constant data rate or data size. Loss less techniques employs entropy encoders such as Huffman encoders. Huffman produced an efficient variable length-coding scheme.The main benefit of lossy compression is that the data rate can be reduced. This is necessary as certain applications require high compression ratios along with accelerated data rates. Of course significant compression can be achieved simply by removing a lot of information and hence quality from the source image, but this is often inappropriate.

Lossy compression algorithms attempt to maximize the benefits of compression ratio and bit rate, while minimizing loss of quality. Finding optimal ways to reach this goal is a severely complicated process as many factors must be taken into account. Variables such as a quality factor which is used to scale the quantization tables used in JPEG can either reduce the resulting image quality with higher compression ratios, or conversely improving image quality with lower compression ratios [8]. JPEG, although lossy, can give higher compression ratios than GIF while leaving the human observer with little to complain of loss in quality. Compression induced loss in images can cause both missing image features as well as artifacts added to the picture. Artifacts are caused by noise produced by sources such as quantization, and may show up as blocking within the image. Blocking artifacts within an image can become apparent with higher compression ratios. The edges are representative of blocks used during compression, such as 8x8 blocks of pixels, used in JPEG.

III. LITERATURE REVIEW ON JPEG

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.

A. 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.

B. 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. 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.

IV. DISCRETE COSINE TRANSFORMATION

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 [1]. The DCT algorithm is completely reversible making this useful for both loss less and lossy compression techniques.

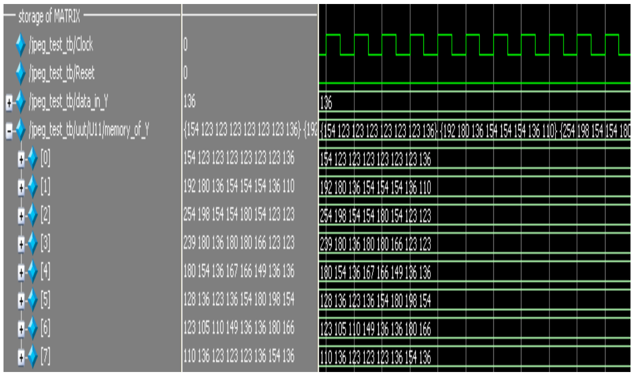
The DCT is a special case of the well-known Fourier transform. Essentially the Fourier transform in theory can represent a given input signal with a series of sine and cosine terms. The discrete cosine transform is a special case of the Fourier transform in which the sine components are eliminated. For 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 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 [3]. 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 [6]. 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.

V. IMPLEMENTATION OF THE TECHNIQUE

The JPEG process and the developed architecture for the required functionality were discussed in the previous chapters. Now this chapter deals with the simulation and synthesis results of the JPEG process. Here Modelsim tool is used in order to simulate the design and checks the functionality of the design. Once the functional verification is done, the design will be taken to the Xilinx tool for Synthesis process and the netlist generation.
The Appropriate test cases have been identified in order to test this modelled JPEG architecture. Based on the identified values, the simulation results which describes the operation of the process has been achieved. This proves that the modelled design works properly as per its functionality.

A. Simulation

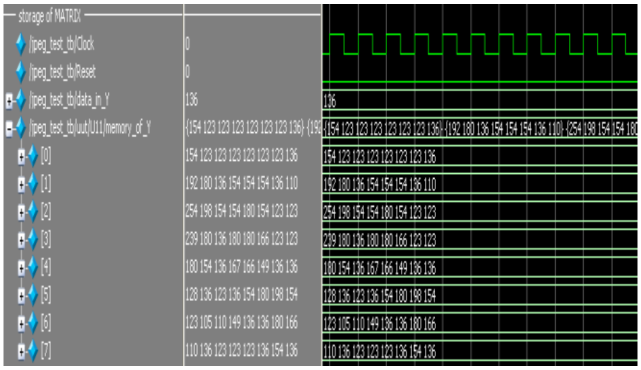
The test bench is developed in order to test the modeled design. This developed test bench will automatically force the inputs and will make the operations of algorithm to perform.



storage of MATRIX

/jpeg_test_tb/Clock	0
/jpeg_test_tb/Reset	0
/jpeg_test_tb/data_in_Y	136
/jpeg_test_tb/uut/U11/memory_of_Y	{154 123 123 123 123 123 123 136} {192
[0]	154 123 123 123 123 123 123 136
[1]	192 180 136 154 154 154 136 110
[2]	254 198 154 154 180 154 123 123
[3]	239 180 136 180 180 166 123 123
[4]	180 154 136 167 166 149 136 136
[5]	128 136 123 136 154 180 198 154
[6]	123 105 110 149 136 136 180 166
[7]	110 136 123 123 123 136 154 136

Fig.1. Simulation of storage pixel values



storage of MATRIX

/jpeg_test_tb/Clock	0
/jpeg_test_tb/Reset	0
/jpeg_test_tb/data_in_Y	136
/jpeg_test_tb/uut/U11/memory_of_Y	{154 123 123 123 123 123 123 136} {192
[0]	154 123 123 123 123 123 123 136
[1]	192 180 136 154 154 154 136 110
[2]	254 198 154 154 180 154 123 123
[3]	239 180 136 180 180 166 123 123
[4]	180 154 136 167 166 149 136 136
[5]	128 136 123 136 154 180 198 154
[6]	123 105 110 149 136 136 180 166
[7]	110 136 123 123 123 136 154 136

Fig.2. Simulation of DCT of JPEG in compression.

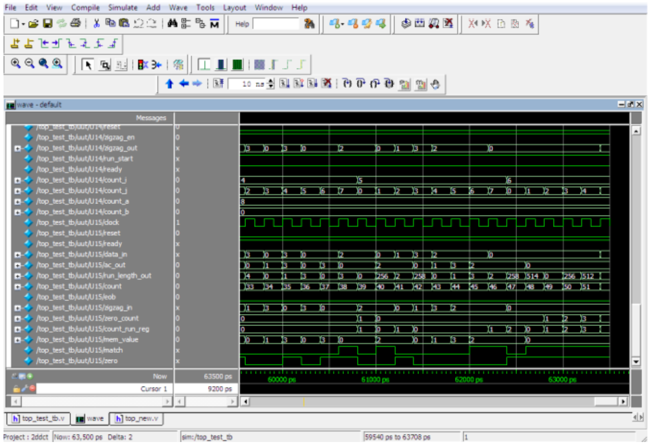


Fig.3. Quantization and Encoding data.

VI. CONCLUSION

In this work data compression using lossless method where JPEG data remains retrieved without losing the information. This simulated, synthesized and implemented by VERILOG language using XILINX ISE Tool. We have used Discrete Cosine transform (DCT) for the data compression and quantization. This data is encoded and which are shown in the simulation results in VERILOG. The data compression and transmission is obtained in parallel and pipelining process with different algorithms used. We can further extend the implementation for proposed data compression using different techniques.

.

REFERENCES

[1]. L. Agostini, S. Bampi, "Pipelined Fast 2-D DCT Architecture for JPEG Image Compression" Proceedings of the 14th Annual Symposium on Integrated Circuits and Systems Design, Pirenopolis, Brazil. IEEE Computer Society 2001. pp 226-231.

[2]. Y. Arai, T. Agui, M. Nakajima. "A Fast DCT-SQ Scheme for Images". Transactions of IEICE, vol. E71, nÂ. 11, 1988, pp. 1095-1097.

[3]. D. Trang, N. Bihn, "A High-Accuracy and High-Speed 2-D 8x8 Discrete Cosine Transform Design". Proceedings of ICGCRCICT 2010, vol. 1, 2010, pp. 135-138

[4]. I. Basri, B. Sutopo, "Implementasi 1D-DCT Algoritma Feig-Winograd di FPGA Spartan-3E (Indonesian)". Proceedings of CITEE 2009, vol. 1, 2009, pp. 198-203

[5]. E. Magli, "The JPEG Family of Coding Standard," Part of "Document and Image Compression", New York: Taylor and Francis, 2004.

[6]. Wallace, G. K. ,"The JPEG Still Picture Compression Standard", Communications of the ACM, Vol. 34, Issue 4, pp.30-44. 1991.

[7]. Sun, M., Ting C., and Albert M., ''VLSI Implementation of a 16X 16 Discrete Cosine Transform'', IEEE Transactions on Circuits and Systems, Vol. 36, No. 4, April 1989.

[8]. Xilinx, Inc., "Spartan-3E FPGA Family : Data Sheet ", Xilinx Corporation, 2009.

[9]. Omnivision, Inc., "OV9620/9120 Camera Chip Data Sheet ", Xilinx Corporation, 2002