An Accelerated Approach for Image compression based on FPGA

Shefin F[1]
[1]PG Scholar ECE Department,
TKM Institute of Technology, Karuvelil
P.O, Kollam, Kerala-691505, India,
shefinspyker@gmail.com

Abhilash R V[2]
[2]Assistant professor: ECE Department,
TKM Institute of Technology, Karuvelil P.O, Kollam,
Kerala-691505, India

***Abstract*— Transmission and storage of uncompressed images and videos are costly and impractical. So data compression methods are developed for this purpose. There is various transformation techniques used for these data compression methods. Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels. The Joint Photographic Expert Group (JPEG) standard was based on the Discrete Cosine Transform (DCT). It is one of the most popular and widely used compression standards. Two dimensional Discrete Cosine Transform module (2D –DCT) is a major module in lossy sequential JPEG image compression method along with other modules like quantization and entropy encoder. The present work is DWT based image compression based on FPGA. Later EZW algorithm will be developed which is fairly general and performs remarkably well with most types of images.**

Keywords—JPEG,DCT,DWT,EZW.FPGA

I. INTRODUCTION

In many different fields, digitized images are replacing conventional analog images as photograph or x-rays. The volume of data required to describe such images greatly slow transmission and make storage prohibitively costly. The information contained in images must, therefore, be compressed by extracting only the visible elements, which are then encoded. The quantity of data involved is thus reduced substantially. The fundamental goal of data compression is to reduce the bit rate of transmission or storage while maintaining an acceptable fidelity or image quality. Compression can be achieved by transforming the data, projecting it on a basis of function, and then encoding this transform. Because of the nature of image signal and mechanism of human vision, the transform used must accept non stationary and be well localized in both the space and frequency domains. To avoid redundancy, which hinders compression, the transform must be at least bi orthogonal and lastly, in order to save CPU time, the corresponding algorithm must be fast.

II. LITERATURE SURVEY

The discrete wavelet transform (DWT) has gained wide popularity due to its excellent decorrelation property, many modern image and video compression systems embody the DWT as the transform stage. It is widely recognized that the 9/7 filters are among the best filters for DWT-based image compression. In fact, the JPEG2000 image coding standard employs the 9/7 filters as the default wavelet filters for lossy compression and 5/3 filters for lossless compression. The performance of a hardware implementation of the 9/7 filter bank (FB) depends on the accuracy with which filter coefficients are represented. Lossless image compression techniques find applications in fields such as medical imaging, preservation of artwork, remote sensing etc. Day-by-day Discrete Wavelet Transform (DWT) is becoming more and more popular for digital image compression. Biorthogonal (5, 3) and (9, 7) filters have been chosen to be the standard filters used in the JPEG2000 codec standard. After DWT was introduced, several codec algorithms were proposed to compress the transform coefficients as much as possible. Among them, Embedded Zero tree Wavelet (EZW), Set Partitioning in Hierarchical Trees (SPIHT) and Embedded Bock Coding with Optimized Truncation (EBCOT) are the most famous ones.

III. PROPOSED SYSTEM

The architecture of DWT comprises of Memory, Control unit, DWT processor and an External memory which is not included during the design since it is already there in the VIRTEX board. The memory used is a dual port ram which stores both the pixel values of an image which is called as text file and also stores the DWT coefficients after the DWT. The memory is also required for the storing of encoded coefficients. Control unit produces the control signals for the DWT processor to function. Figure 1 shows the basic block diagram of the DWT architecture.

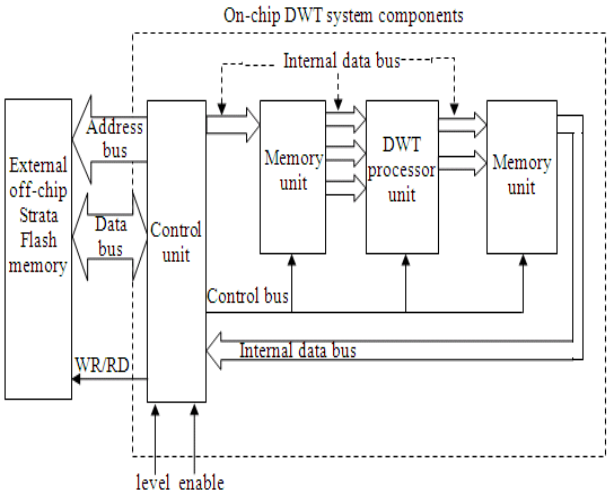


Figure 1. DWT Block Diagram

A. Data Representation and Word Length

At first, before designing the 2-D DWT processor, data type representation and image pixels word length must be taken into consideration. There are two ways for data representation with hardware design, either floating point or fixed point. By representing data in floating point method the results will be more accurate due to the greater range of numbers (32 bits or 64 bits), but this accuracy requires more silicon area on FPGA and needs more complex design . Fixed point representation has the advantages of less silicon area on FPGA, and easier to implement a design. So the fixed point method is used to represent the data. The most common representation for fixed point numbers is the 2´s complement, this type is appropriate with hardware design for arithmetic computations. Another important consideration is the word length, which means, the number of bits per pixel. The original image data word length is 8-bits per pixel, this amount of bits is not enough for wavelet transform coefficients. It is observed that the retrieved image in inverse discrete wavelet transform has distortion, because of overflow condition.

B. Internal Memory Units

There are two internal on-chip memory units in the system design, each one has size (2N) for (N×N) image, these memory units are used as a cache memory, to speed up the DWT system assignments because the internal memory is faster than external memory. Each of these internal memory units is a single dual port block ram, therefore only one clock cycle is needed for reading or writing two words of data in two random memory locations. The memory units are also used to make an agreement between the 12 bits system word length and the 16 bits external memory word length.

The first memory unit works as a split stage for the lifting scheme wavelet transform, this unit works on separating the even indexed pixels from the odd indexed pixels. The separation process can be done by putting out even memory locations from (port A) output port, and odd memory locations from (port B) output port. This unit consists of a dual port block ram with two registers connected to the block ram output ports as shown in the Fig. 3.4. This arrangement delivers three outputs (X[2n], X[2n+1], and X[2n+2]) form the memory unit at each clock cycle to the DWT processor unit. Since, the DWT processor has two outputs for each clock cycle the (LPF, and HPF) coefficients, a second memory unit is necessary. This memory unit is also a dual port block ram used as dual input ports and single output port. This unit brings out the wavelet coefficients toward the external memory.

The internal memory unit must be designed in such a manner that it must store and load the pixel values faster and precise. The pixel values is scanned and stored in a particular manner. The text file reading is done here and the read pixel values are stored in specific locations. The image pixels are obtained using Matlab and it is written to a text file. Later these pixel values are arranged in column wise on the text file and this modified text file is called.

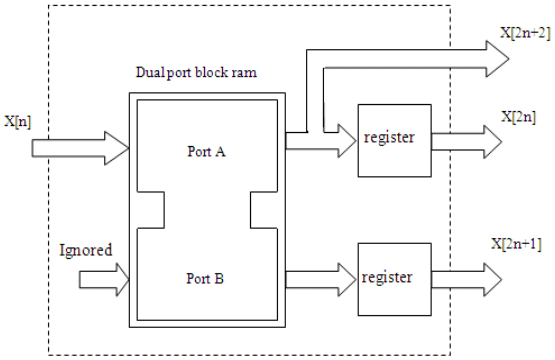


Figure 2. Internal Memory Block

C. DWT Processor Unit

This unit represents the hardware design of the (5/3) lifting scheme filter, it contains predict stage and update stage. The same unit is used for row operations and then for column operations. This unit has three input ports connected to the three output ports of the first memory unit, and has two output ports connected to the input ports of the second memory unit as shown in the Fig. 4. In the hardware implementation of the filter design there are three registers to store incoming data from the first memory unit (three input data per clock cycle). Four adders and two shifters are used instead of multipliers, also a register is used to store the previous value of the details coefficients (D [2n-1]) for the next computation according to the lifting equations. The DWT Processor can be built using filters and down samplers as shown in figure 3.

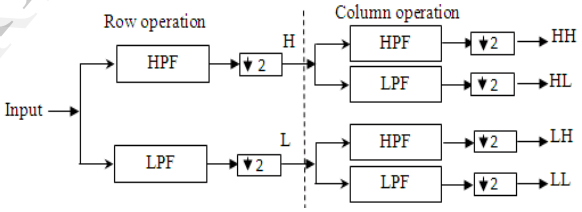


Figure 3.Block Diagram of DWT Processor

D. Control Unit

There are two duties for the control unit. The first, it controls the on-chip DWT system components, by providing control signals (read, write, status, and enable), also it gives the appropriate addresses for memories units, and controls the data flow in the proposed design. The second, it provides complete interface signals and buses with external memory (read, write, enable, address bus, and data bus). The control unit is designed with finite state machine (FSM) method. There are two input signals that are connected to the switches of the FPGA kit, these control signals must be asserted by the user before starting the system operations. First one is the DWT level signal, used to select the wanted decomposition levels. The Second is the enable signal, used to enable the system components to perform DWT operations. After setting the (enable and level) signals, the control unit is responsible for all system operations starting with reading image data from external memory, enable DWT processor unit operations, and writing DWT sub bands to the external memory.

E. External Memory

The external memory is the on board 128Mbit Intel strata flash (parallel NOR flash PROM). It is used for storing the original image and later the DWT coefficients. This memory is configured as 8 word (128Mbit), each memory location is of 16 bit word size, the strata flash memory has 23 bit memory address bus width, 16 bit data bus width, and three important control signals that must be taken in to consideration (chip select, output enable, and write enable). The strata flash memory controller is designed by using the VHDL language to be appropriate for the proposed design. This controller is included in the proposed DWT control unit, already mentioned. There are some configurations on FPGA starter kit board that must be set properly to deal with strata flash, such as, the on board components that shares connections with strata flash memory which must be disabled, to ensure that only one data source is active at a time . Another configuration is setting the on board FPGA mode jumpers, also see . In this proposed design, the FPGA chip configured to the BPI Up (byte peripheral interface) mode, this means that the memory address starts at address (0) and increments through the address space. The original image data is converted to hexadecimal format before storing it in the memory, the conversion operation is achieved by using a MATLAB program. In order to store the original image in the on board memory, the strata flash controller is used for writing image data file, in hexadecimal format, in the addressed memory locations. Then the DWT proposed architecture is downloaded on the FPGA chip to perform DWT operations and store wavelet coefficients in the strata flash memory.

IV. EZW ENCODING

When searching through wavelet literature for image compression schemes it is almost impossible not to note Shapiro's Embedded Zerotree Wavelet encoder or EZW encoder for short. An EZW encoder is an encoder specially designed to use with wavelet transforms, which explains why it has the word wavelet in its name. The EZW encoder was originally designed to operate on images (2D-signals) but it can also be used on other dimensional signals. The EZW encoder is based on Progressive encoding to compress an image into a bit stream with increasing accuracy. This means that when more bits are added to the stream, the decoded image will contain more detail, a property similar to JPEG encoded images. It is also similar to the representation of a number like every digit we add increases the accuracy of the number, but we can stop at any accuracy we like. Progressive encoding is also known as embedded encoding, which explains the E in EZW. This leaves us with the Z. This letter is a bit more complicated to explain, but we will give it a try in the next paragraph. Coding an image using the EZW scheme, together with some optimizations results in a remarkably effective image compressor with the property that the compressed data stream can have any bit rate desired. Any bit rate is only possible if there is information loss somewhere so that the compressor is lossy. However, lossless compression is also possible with an EZW encoder, but of course with less spectacular results.

The EZW encoder is based on two important observations:

1. Natural images in general have a low pass spectrum.

When an image is wavelet transformed the energy in the sub bands decreases as the scale decreases (low scale means high resolution), so the wavelet coefficients will, on average, be smaller in the higher sub bands than in the lower sub bands. This show that progressive encoding is a very natural choice for compressing wavelet transformed images, since the higher sub bands only add detail;

2. Large wavelet coefficients are more important than small wavelet coefficients.

These two observations are exploited by encoding the wavelet coefficients in decreasing order in several passes. For every pass a threshold is chosen against which all the wavelet coefficients are measured. If a wavelet coefficient is larger than the threshold it is encoded and removed from the image, if it is smaller it is left for the next pass. When all the wavelet coefficients have been visited the threshold is lowered and the image is scanned again to add more detail to the already encoded image. This process is repeated until all the wavelet coefficients have been encoded completely or another criterion has been satisfied (maximum bit rate for instance). The trick is now to use the dependency between the wavelet coefficients across different scales to efficiently encode large parts of the image which are below the current threshold. It is here where the zerotrees enter. So, let me now add some detail to the foregoing. (As most explanations, this explanation is a progressive one.)

A wavelet transform transforms a signal from the time domain to the joint time-scale domain. This means that the wavelet coefficients are two-dimensional. If we want to compress the transformed signal we have to code not only the coefficient values, but also their position in time. When the signal is an image then the position in time is better expressed as the position in space. After wavelet transforming an image we can represent it using trees because of the sub sampling that is performed in the transform. A coefficient in a low sub band can be thought of as having four descendants in the next higher sub band. The four descendants each also have four descendants in the next higher sub band and we see a quad-tree emerge: every root has four leafs.

V. SIMULATION RESULTS

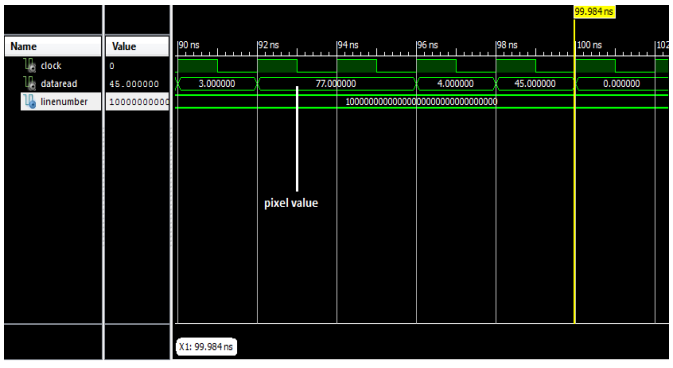


Figure 4. Text File Reading

The above figure shows the reading of the text file. The image is loaded to Matlab and the corresponding pixel values are obtained using Matlab function. The obtained pixel value

is then stored as a text file in the computer hard-disk. By using VHDL program the text file is called to the Xilinx IDE and the reading of the text file is done. The dataread in the simulation result shows the pixel value which is being read out from a text file.

Figure 5 shows the control unit which produces the control signal for the dwt processor. The control unit has several signals such as reset, clk, start and ready in order for the controlling of the DWT processor. The control signals from the control unit controls the flow of data to and from the memory.

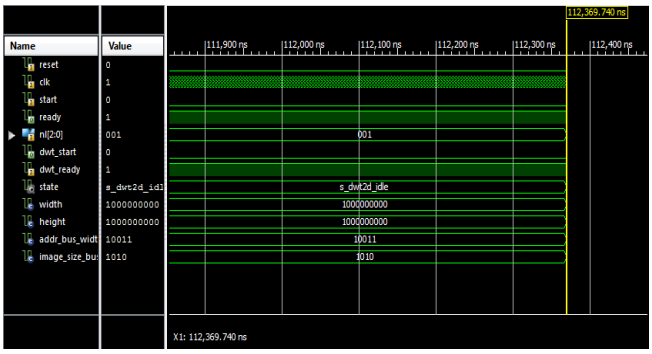
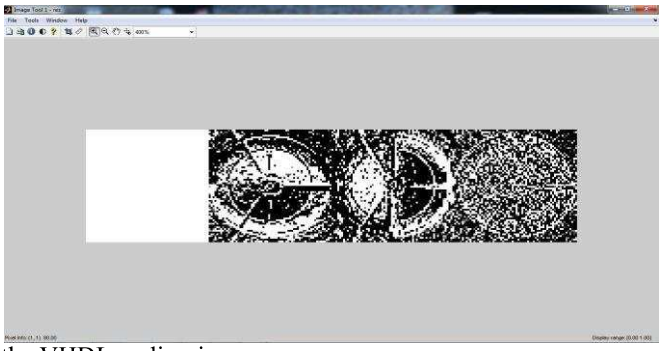


Figure 5. Control unit

Figure 6 shows the Xilinx simulated result of a 3 level DWT of the image (Cambada.jpg) and the coefficients are represented in order. Here the three detail values are represented by h, v, d and the approximate value is a.

Figure 6. 3 Level DWT

Figure 7 shows the Matlab result of the same image which is decomposed 3 times with DWT. The Matlab result is taken in order top check whether the values obtained using



the VHDL coding is same.

Figure 7. 3 Level DWT in Matlab

VI. CONCLUSION

A new Architecture for the Discrete Wavelet Transform is simulated using the XILINX ISE and the total time taken for simulation is noted to be 239ms which is much faster. Later these coefficients can be used for the Encoding Process so that the Image can be compressed.

REFERENCES

[1] M. Mesbahuddin Sarker, "EZW Algorithm and Computation of Its Coefficients for Image Compression by Using "Bottom-Up" Approach", *IJSET* , Vol. 2, Issue 6, pp. 532 – 538 June 2013.

[2] Vikas kumar and sonal,"Architecture for Three-Dimensional(3-D) Discrete Wavelet Transform for Grayscale image," *IJCEM International Journal of Computational Engineering & Management*, Vol. 12, April2011.

[3] V.S. Shingate, T. R. Sontakke & S.N. Talbar, **"**Still Image Compression using Embedded Zerotree Wavelet Encoding", *International Journal of Computer Science & Communication,* Vol. 1, No. 1, January-June 2010, pp.21-24

[4] Koko H. S. and Agustiawan .H,(2009), "Parallel pipelined VLSI architectures for lifting-based two-dimensional forward discrete wavelet transform," in *Proc. Int. Conf. Signal Acq. Process. (ICSAP)*, , pp. 18–25.

[5] Subhasis Saha-"Image Compression - From DCT to Wavelets: A Review", Crossroads, Volume 6 , Issue (March2000) Pages: 12 -21 Year of Publication: 2000 ,ISSN:1528-4972.

[6] Janaki R, Dr.Tamilarasi A, "Still Image Compression by Combining EZW Encoding with Huffman Encoder" ,*International journal of Computer Application* , Vol. 13, Issue 7, January 2011.

[7] Nasir, M. R., Rolan, G. W., François, G. M., Ronald R. C.,adaptive wavelet Packet Basis Selection for Zerotree Image Coding", l*EEE Transactions on Signal Processing,* vol. 12, no. 12,*December 2003.*