

High Speed Arithmetic Coder Architecture used in SPIHT

Sukhi S¹, Rafeekha M J²

¹PG scholar, Dept of Electronics and Communication Engineering, TKM Institute Of Technology, Kollam, Kerala, India,

²Assistant Professor, Dept of Electronics and Communication Engineering, TKM Institute Of Technology, Kollam, Kerala, India

Abstract— Arithmetic coding (AC) method can obtain optimal performance for its ability to generate codes with fractional bits. It is widely used in various image compression algorithms. The set partitioning in hierarchical trees (SPIHT) uses an AC method to improve its peak signal-to-noise ratio (PSNR) about 0.5 db. SPIHT is a very powerful image compression technique, very fast among the best image compression algorithm. In this algorithm three ordered lists are used to store the significant information during set partitioning. As the algorithm proceeds, data are manipulated between three lists namely, the list of insignificant pixels (LIP), the list of insignificant sets (LIS) and the list of significant pixels (LSP). The LIP contains the pixels which on comparison with the threshold fall below it and are considered insignificant during that pass. LIS contains sets of wavelet coefficients which are defined by tree structures, and which had been found to have magnitude smaller than a threshold. Set Partitioning in Hierarchical Trees (SPIHT) algorithm for image compression is proposed with an arithmetic coder thereby it compresses the Discrete Wavelet Transform decomposed images. SPIHT has straightforward coding procedure and requires no tables which make a SPIHT algorithm an appropriate one for low cost hardware implementation. In order to avoid rescanning the wavelet transformed coefficients, a breadth first search SPIHT without lists is used instead of SPIHT with lists. With the help of Breadth First search high speed architecture is achieved. The coding is done in VHDL language, synthesized using Xilinx ISE 13.2 and simulated using ISim.

Keywords— Arithmetic coding, Common bit detection (CBD) circuit, Discrete wavelet transform (DWT), Set Partitioning in Hierarchical Trees (SPIHT)

I. INTRODUCTION

Nowadays, the size of storage media increases day by day. Although the largest capacity of hard disk is about two Terabytes, it is not enough large if we store a video file without compressing it[1]. For example, if we have a colour video file stream, that is, with three 720x480 sized layer, 30 frames per second and 8 bits for each pixel. Then the storage space can be obtained as $720 * 480 * 3 * 8 * 30 = 249\text{Mbit/s}$. This equals about 31.1MB per second. For a 650MB CD-ROM, we can only store a video of about 20 seconds long. That is why we want to do image and video compression though the capacity of storage media is quite large now. The motivation of data compression is using less quantity of data to represent the original data without distortion.

Consider the system in Fig. 1. when the encoder receives the target image, it converts the image into bit stream. On the other hand, the decoder receives the bit stream and then con-

verts it back to the image. If the quantity of bit stream is less than that of the original image then we call this process as

Image Compression Coding.

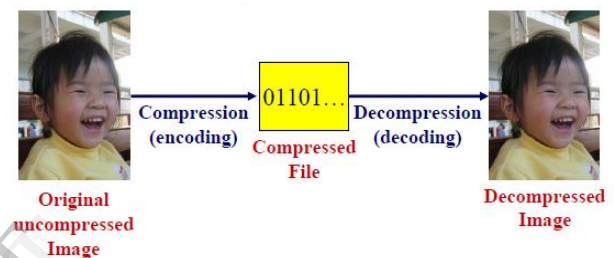


Fig.1.Fundamental of image compression

Now, we want to know how and why we can make an image compressed. Generally speaking, the neighbouring pixels in an image have highly correlation to each other. That is why images can be compressed in a high compression ratio. Compression ratio is the ratio of the original uncompressed image file to the compressed file[2].

$$\text{Compression Ratio} = \frac{\text{Uncompressed file Size}}{\text{Compressed File Size}} \quad (1)$$

Two fundamental components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image/video). Irrelevance reduction omits only those parts of the signal that are not noticed by the signal receiver. In general, three types of redundancies that can be identified namely; Spatial Redundancy or correlation between neighboring pixel values. Spectral redundancy or correlation between different color-planes or spectral-bands. Temporal redundancy or correlation between adjacent frames in a sequence of images especially in video applications. Compression techniques are classified into two namely lossy compression and lossless compression. Lossless compression technique guarantees the exact duplication of input in compress/decompress cycle. This is mainly used in medical applications[3]. Methods for lossless image compression are: Run-length encoding, DPCM, Predictive Coding and Entropy encoding. With Lossy compression technique higher levels of data reduction is possible but it results in less perfect reproduction of the original image. This is useful in applications such as broadcast television, videoconferencing, and facsimile transmission, in which a certain amount of error is acceptable trade-off for increased compression per-

formance. Methods for lossy image compression are: Reducing the color space to the most common colours in the image, chroma sub-sampling, transform coding, and fractal compression. Block diagram of image compression technique is shown in figure.2.

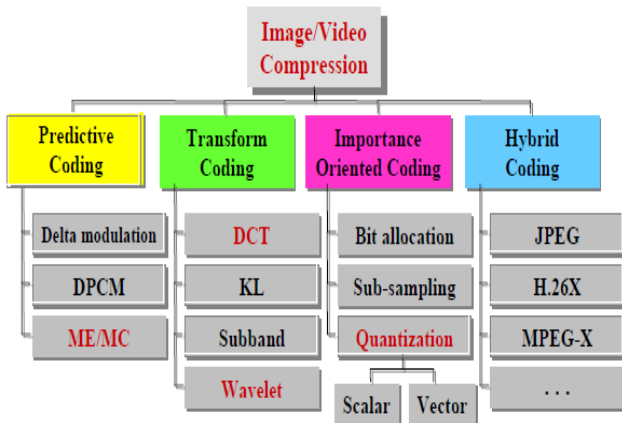


Fig.2. Block diagram of image compression technique

Generally a compression system consists of encoder and decoder stages. Encoder consists of the mapper, quantizer and encoder. Mapper transforms the input image into a format by reducing the inter-pixel redundancies. Quantizing refers to a reduction of the precision of the floating point values of the wavelet transform. A symbol encoder further compresses the quantized values to give a better overall compression. It uses a model to accurately determine the probabilities for each quantized value and produces an appropriate code based on these probabilities so that the resultant output code stream will be smaller than the input stream. An arithmetic coder or Huffman coder is used as symbol encoders. Decoder part of the compression model consists of symbol decoder and inverse mapper[4][5]. Among these FPGAs are best suited to implement image processing algorithms because they offers features like infinite reprogram ability, high parallelism, flexibility, computational power and short development time.

II. LITERATURE REVIEW.

A. Set Partitioning In Hierarchical Trees Algorithm

SPIHT is an image compression algorithm with lists to store the significant information of wavelet coefficients for image coding purpose[6]. It mainly uses three different lists, namely List of Significant Pixels(LSP),List of Insignificant sets(LIS) and list of Insignificant pixels(LIP). In SPIHT algorithm initially all wavelet coefficients are considered as insignificant sets and these sets are placed in the list called LIS. Wavelet coefficients in LIS are compared with a predefined threshold if the wavelet coefficient value is greater than the predefined threshold then they are placed in the list called LSP and if the value is lesser than the predefined threshold then the wavelet coefficients are placed in the list called LIP. All the pixel values in the list LIS are tested for there significant state. Significant state of the wavelet coefficients are tested using the equation (2) which is as given below. So, this the reason why SPIHT algorithm less bits to code after the discrete wavelet transform [7].

$$S_n(\tau) = \begin{cases} 1 & \max_{i,j \in \tau} \{C_{i,j}\} \geq 2^n \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

Where $C_{i,j}$ represents the coefficient value for the position i,j in the wavelet sub-band. Γ represents set of coefficients and $S_n(\Gamma)$ represents the significant state of the set. Flow diagram of SPIHT is shown in figure.3.

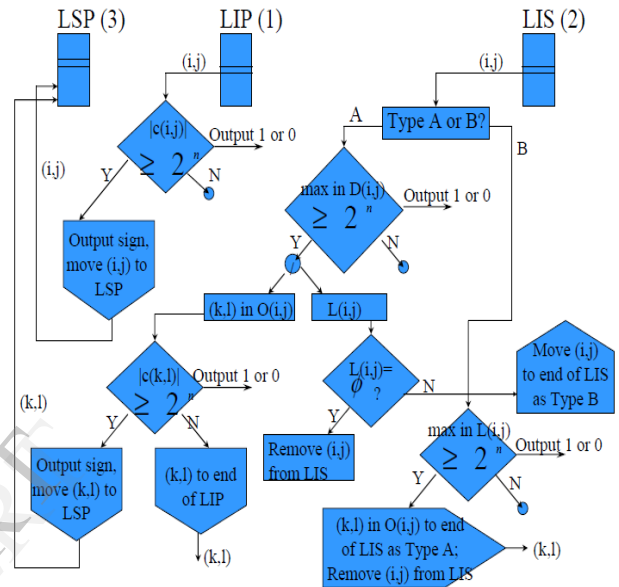


Fig.3. Flow diagram of SPIHT

B. Discrete Wavelet Transform

A Discrete Wavelet Transform is a wavelet transform in which wavelets are discretely sampled. If Discrete Wavelet Transform is compared with the Fourier Transform it has an advantage of temporal resolution which captures both frequency and location information. A wave is an oscillating function of time or space and is periodic. In contrast, wavelets are localized waves. Wavelets have their energy concentrated in time and they are used for analysis of transient signals. Wavelet transform uses wavelets of finite energy whereas the Fourier Transform and short Time Fourier Transform uses wavelets of finite energy.

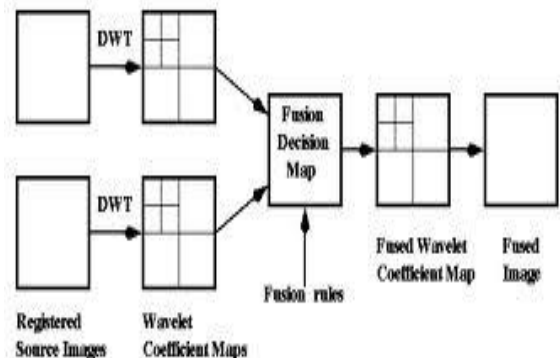


Fig.4. Block diagram of DWT

III .PROPOSED METHOD

The main objective of this method is to designed and implemented a SPIHT Encoding Coder for Lossy virtual Images. Embedded Zero Tree Wavelet algorithm (EZW) used here is simple, specially designed for wavelet transform and effective image compression algorithm. This algorithm is devised by Shapiro and it has property that the bits in the bit stream are generated in order of importance, yielding a fully embedded code. SPIHT stands for Set Partitioning in Hierarchical Trees. The SPIHT coder is a highly refined version of the EZW algorithm and is a powerful image compression algorithm that produces an embedded bit stream from which the best reconstructed images. The SPIHT algorithm was powerful, efficient and simple image compression algorithm. By using these algorithms, the highest PSNR values for given compression ratios for a variety of images can be obtained.

A. System architecture

The block diagram of the SPIHT encoding is as shown in the Fig.5. The input image is gray scale image of size 512x512 each pixel of 8 bits. The transformed wavelet coefficients are placed in a buffer and they are accessed in a SPIHT-Breadth First Search way. Processor dispatcher dispatches the transformed wavelet coefficients to the arithmetic coder through internal bus.

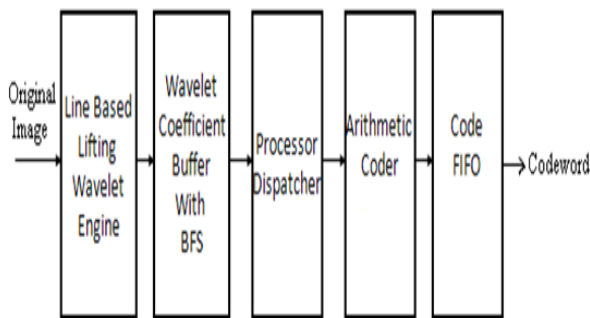


Fig.5. Block diagram of SPIHT encoding

Output of arithmetic coder is provided to the internal bus and sent to the code FIFO. The Read FIFO and Truncate module are responsible for the final code stream formation, which reads each code FIFO from top to bottom and truncates the code stream according to the bit rate requirement.

B. Line Based Lifting Wavelet Engine

Lifting step scheme consists of three simple phases: the first step, or Lazy wavelet, splits the data into two subsets: even and odd; the second step calculates the wavelet coefficients (high-pass) as the failure to predict the odd set based on

the even set; finally the third step updates the even set using the wavelet coefficients to compute the scaling function coefficients (low-pass). The predict phase ensures polynomial cancellation in the high-pass, and the update phase ensures preservation of moments in the low-pass. In folded architecture the output of processing element is fed back through the delay registers. By adding different number of delay registers and coefficients of processing elements folded architecture based lifting scheme can be used to design different wavelets. Folded architecture for (9, 7) wavelet with line based lifting wavelet engine is as shown in Fig.3. Coefficients for (9, 7) are 0.5 and 0.25 which are represented by a, b, c, d. Two delay registers (D) are used in order to schedule the data in each phase. Based on the phase of interleaved computation, the coefficient for multiplier M1 is either a or c or similarly the coefficient for multiplier M2 is b or d. Critical path in this folded architecture is reduced[8].

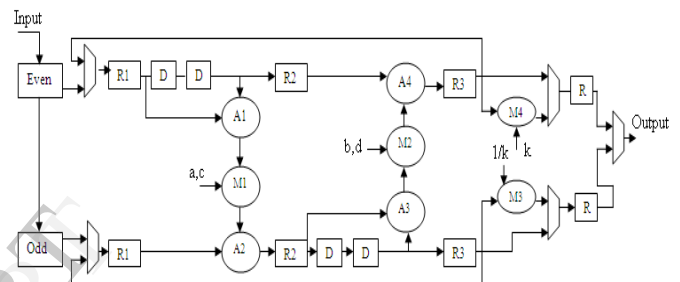


Fig.6. Architecture of Lifting based Wavelet Engine

Fig.7. illustrates the steps for performing a two-level DWT on an image. The 1-D DWT is first performed on the rows of the image producing low-frequency L1 and high-frequency components H1. After performing a 1-D DWT again on the columns of and the first level of decomposition is completed LL1, LH1, HL1 and HH1 are obtained.

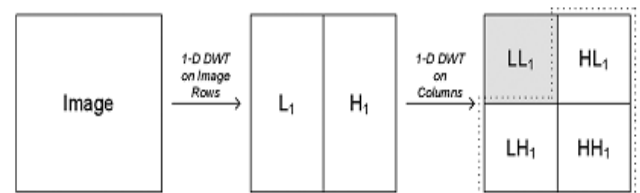


Fig.7. Applying DWT on an image

C. 2-D DWT Architecture

The computation style of the entropy coder after DWT will affect the optimal scheduling of the 2-D DWT computation. Fig.8. shows the simplified JPEG2000 functional block diagram. Embedded Block Coding with Optimized Truncation (EBCOT) is a block-coding engine. Images after DWT are decomposed into many sub-bands. Every sub-band is then partitioned into code-blocks. EBCOT processes these quantized wavelet coefficients code-block by code block. After

Tier- 1 compression of EBCOT every code-block will generate a sub-bitstream.

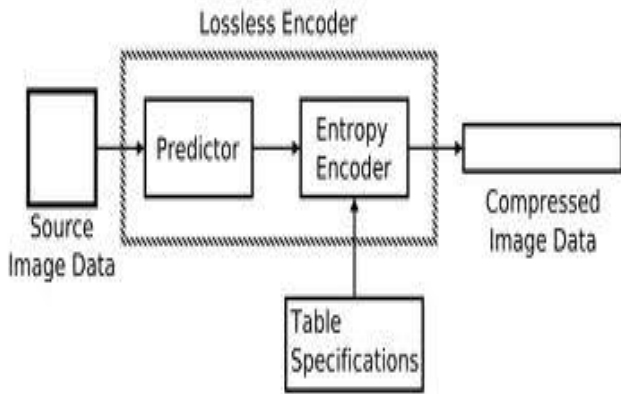


Fig.8. functional block diagram of Simplified JPEG2000

To extend the 1-D DWT core to compute 2-D DWT in JPEG2000 two cases are considered. First, if a frame memory is necessary and has already existed before DWT operation. The data of the whole image are assumed to be stored in the memory. Although 2-D DWT can be scheduled to calculate all rows first (horizontal 1-D DWT) and then all columns (vertical 1-D DWT), it is possible to start the EBCOT computation once there is a complete code-block data available. Due to the in-place computing capability of lifting scheme the original samples can be replaced directly by the calculated coefficients. Hence, the original frame-size memory is enough. The advantage of this implementation is the ease of data flow control. Due to the interleaving characteristics of the output, i.e. one low pass sample followed by, one high pass sample. The interleaving storage arrangement is illustrated by an example of a 4 X 4 image show in Fig. 8. An address generator (AG) is needed to provide the proper access addresses to read samples for next level wavelet decomposition and then write back. The block diagram of the JPEG 2000 system is shown in Fig. 10. The frame memory is used for the storage of the data for DWT, and also for the entropy coded sub-bitstreams of each code-block after EBCOT.

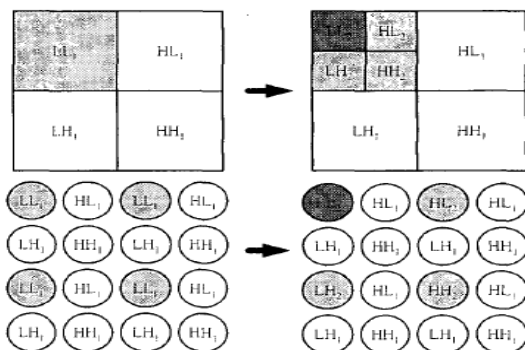


Fig.9. Example of 2-level wavelet transform shows the in-place interleaving organization of wavelet coefficients. where a circle represents a pixel.

Second, if a frame memory is not available or not allowed due to the constraint on the cost memory size. Then the concept of line-based DWT can be adopted. Since EBCOT is not line-based, the height of the line buffer will depend on the height of the code-block. The required buffer size for DWT will be smaller than the frame memory.

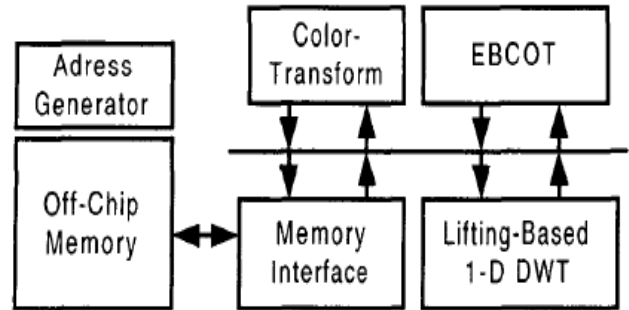


Fig.10. JPEG 2000 system Block diagram

D. Principle of arithmetic coding

Arithmetic coding is a variable-length source encoding technique [7]. In traditional entropy encoding techniques such as Huffman coding, each input symbol in a message is substituted by a specific code specified by an integer number of bits. In arithmetic coding a sequence of input symbols is represented by an interval of real numbers between 0.0 and 1.0. More probable symbols reduce the interval less than the less probable symbols and hence add fewer bits in the encoded message. As a result, the coding result can reach to Shannon's entropy limit for a sufficiently large sequence of input symbols as long as the statistics are accurate. In arithmetic coding uses three registers namely low, high and range. Cumulative frequency is defined as the cumulative counts of the symbol "i". If current interval is given by (low, high) then the values of range, low and high are calculated using the formula as given in equation.

$$\begin{aligned} \text{Range} &= \text{High} - \text{Low} + 1 \\ \text{High} &= \text{Low} + \text{Range} * [\text{cum_freq}[i-1] / \text{cum_freq}[0]] \\ \text{Low} &= \text{Low} + \text{Range} * [\text{cum_freq}[i] / \text{cum_freq}[0]] \end{aligned} \quad (3)$$

Where cum_freq[i] represents the cumulative frequency of the symbol. Arithmetic coding is a well-known method for lossless data compression. Arithmetic coding is a compression technique that can give compression levels at very near entropy.

E. SPIHT Algorithm

SPIHT is a wavelet-based image compression coder. It first converts the image into its wavelet transform and then transmits information about the wavelet coefficients. The decoder uses the received signal to reconstruct the wavelet and performs an inverse transform to recover the image. We selected SPIHT because SPIHT and its predecessor, the embedded zerotree wavelet coder, were significant breakthroughs in still image compression in that they offered sig-

nificantly improved quality over vector quantization, JPEG, and wavelets combined with quantization, while not requiring training and producing an embedded bit stream. SPIHT displays exceptional characteristics over several properties all at once [8] including:

- Good image quality with a high PSNR
- Fast coding and decoding
- A fully progressive bit-stream
- Can be used for lossless compression
- May be combined with error protection
- Ability to code for exact bit rate or PSNR

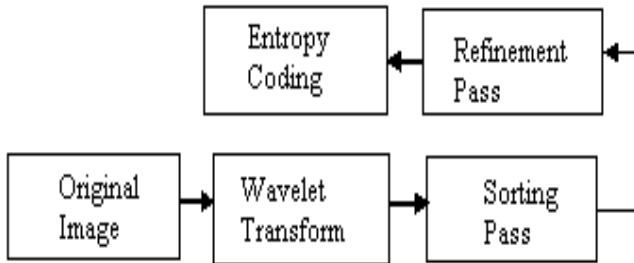


Fig.11. Flow Chart of SPIHT algorithm

SPIHT codes a wavelet by transmitting information about the significance of a pixel. By stating whether or not a pixel is above some threshold, information about that pixel's value is implied. Furthermore, SPIHT transmits information stating whether a pixel or any of its descendants are above a threshold. If the statement proves false, then all of its descendants are known to be below that threshold level and they do not need for computation. In addition to transmitting wavelet coefficients in a implementation, for both the encoder and decoder since the order must be calculated for each image. Bit plane ordering, the SPIHT algorithm develops an individual order to transmit information within each bit plane. The ordering is implicitly created from the threshold information discussed above and by a set of rules which both the encoder and decoder agree upon. Thus each image will transmit wavelet coefficients in an entirely different order. Slightly better peak signal to noise ratio (PSNR) are achieved by using this dynamic ordering of the wavelet coefficients[9]. The trade-off for the improvement is increased run-times, in a hardware considered during the rest of the current pass. At the end of each pass the threshold is divided by two and the algorithm continues. By proceeding in this manner, information about the most significant bits of the wavelet coefficients will always precede information on lower order significant bits, which is referred to as bit plane ordering. Within each bit plane data is transmitted in three lists: the list of insignificant pixels (LIP), the list of insignificant sets (LIS) and the list of significant pixels (LSP).

F. Architecture of arithmetic coder

The architecture of arithmetic coder is as shown in Fig.5. Arithmetic coder consists of three main parts, upper and lower bound update, and common bit detector. The tree construction part visits the wavelet coefficients by the breadth first search order. For speedup, all valid bit-planes are scanned in parallel. Significant wavelet coefficients are provided as input to the arithmetic coder.

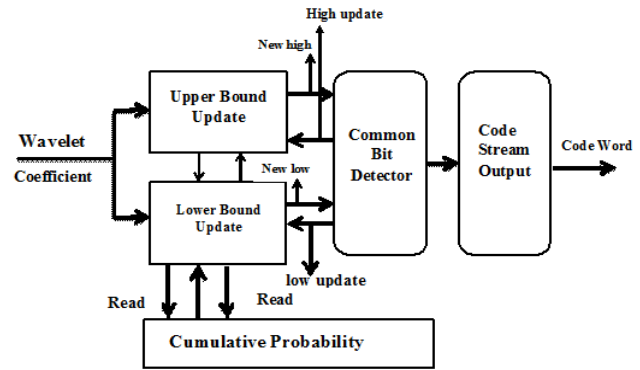


Fig.12. Architecture of arithmetic coder

Upper bound and lower bound update units in arithmetic coder are used for updating the initial intervals to 0 and 1. High and low values are multiplied with cumulative probabilities and updated high and low values are given as input to common bit detector unit. Output is then passed to the code stream output unit in arithmetic coder. Cumulative probability part will hold the frequency of occurrence of wavelet coefficient. Upper bound and lower bound update units in arithmetic coder are implemented with carry look ahead adder and floating point multiplier.

F. Upper Bound and Lower Bound Update

Upper bound and lower bound update registers consists of Look ahead adder and Fast array multiplier. The calculation units for upper and lower bound update units are as shown in the Fig.13. Output of upper and lower update units are new_low and new_high. For speed up purpose, a Carry Look Ahead adder and a floating point multiplier are employed to reduce the delay of critical path. The cumulative probabilities cum_freq [i] and cum_freq [i-1] are provided by the cumulative probability module and this is accompanied by high and new values to compute new bounds for the probability interval these are calculated using the formula (3).

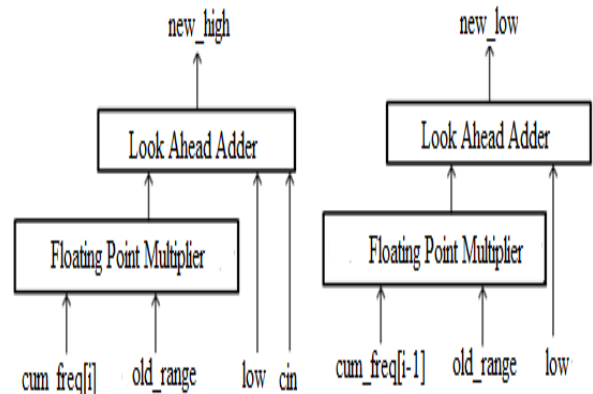


Fig.13. Calculation units for upper and lower update unit

G. Common bit detector

The internal structure of common bit detector is as shown in the Fig.7. The new bound values are then registered and connected to the common bit detector (CBD) part which unrolls the internal loop and records the same bits from the MSB to the LSB between two registers.

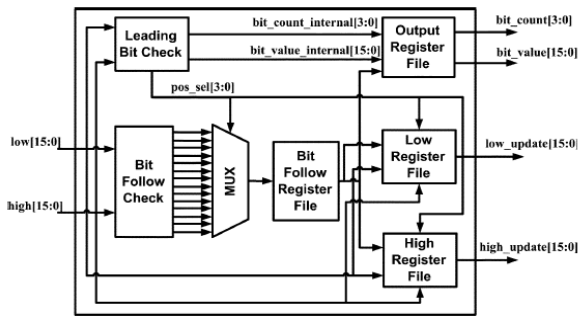


Fig.14. Internal Structure of common Bit Detector

IV. SIMULATION RESULTS

The design of SPIHT algorithm with arithmetic coder is described in VHDL. Design and testing of individual module has been carried out. Fig.15.depicts the simulation results of lifting based DWT. The input to this module is the pixel values stored in 1-D array. Output of this module is a transformed coefficient.

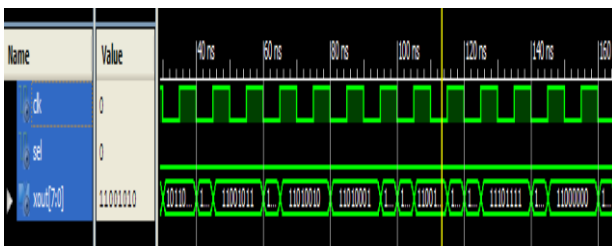


Fig.15. Simulation result of lifting based DWT

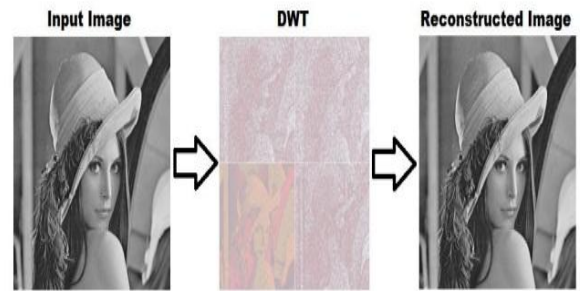


Fig.17. Simulation result of lifting based DWT in MATLAB

Output of DWT is provided as input to SPIHT algorithm so the transformed wavelet coefficients are initially stored in the List of Insignificant Sets (LIS) now the wavelet coefficients in the LIS are compared with the threshold value if the wavelet coefficients are greater than the threshold then they are placed in List of significant pixels otherwise they are placed in List of Insignificant Pixels.

Output of SPIHT-Breadth First Search algorithm is as shown in the Fig.18.

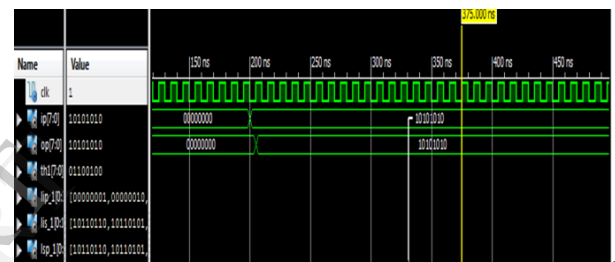


Fig.18. Simulation result of SPIHT algorithm

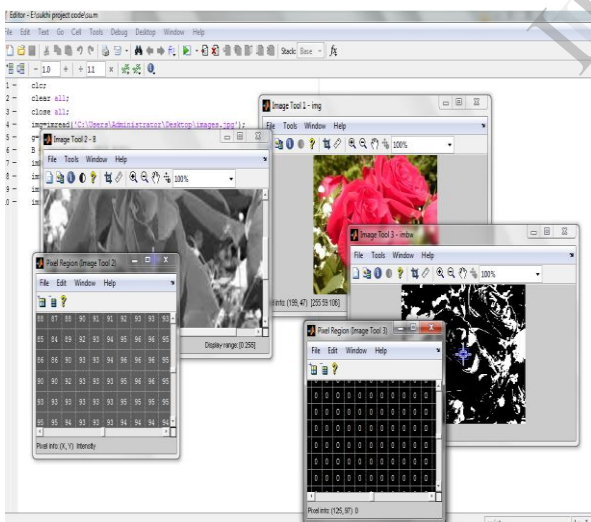


Fig.16. Simulation result of DWT in MATLAB

Fig.16. shows the simulation output of DWT in MATLAB software. Gray scale image of 512X512 is given as input.1-D DWT is first applied on columns and then on the rows.

RTL Schematic view of SPIHT-Breadth First Search algorithm is as shown in the Fig.19. and Fig.20.

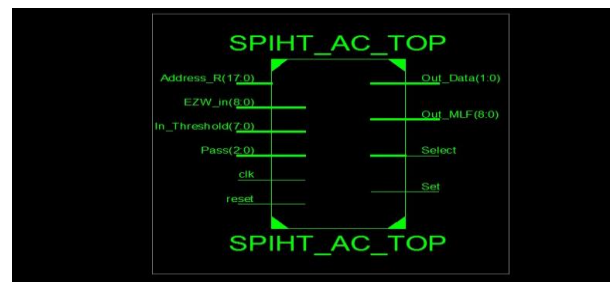


Fig.19. RTL Schematic view of SPIHT

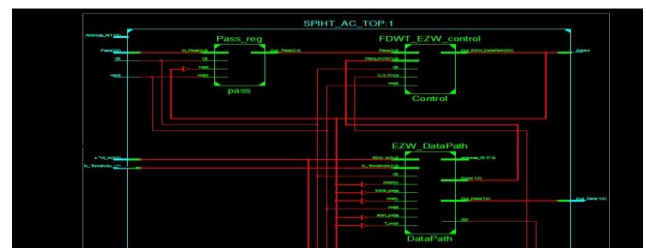


Fig.20. RTL Schematic view of SPIHT

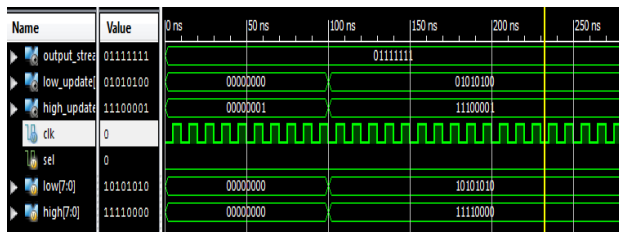


Fig.21. Simulation result of common bit detector

Fig.21. shows the simulation result of common bit detector module which unrolls the renormalization stage of arithmetic coder.

The Simulation result for SPIHT algorithm with arithmetic coder after integrating all the modules is as shown in Fig.22. The final output consists of code word generated which is of 8bits.

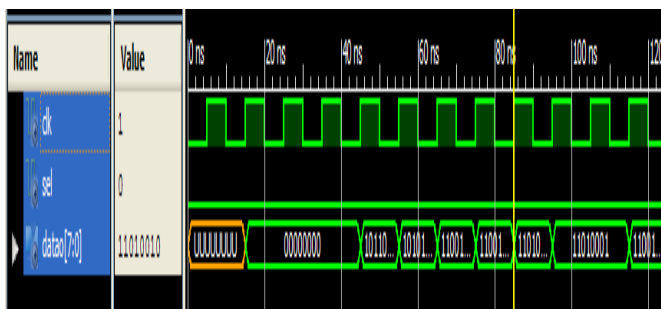


Fig.22. Simulation result of SPIHT algorithm with arithmetic coder after integration.

V. CONCLUSIONS

SPIHT is a very powerful image compression algorithm, widely used in real time applications. In the first phase Gray scale image of 512X512 is given as input. Convert this image into corresponding pixel values by using MATLAB and given to the arithmetic coder architecture. Arithmetic coder mainly consists of three modules .Upper bound modules, Lower bound module and a Common bit detector. Upper bound and lower bound update registers consists of Lookahead adder and Fast array multiplier. Output of this module updated low and high values which are provided as input to common bit detector. Common bit detector module which unrolls the renormalization stage of arithmetic coder. Combine this coder with SPIHT algorithm .Compression of 4:1 is achieved in this architecture so the memory occupied by Code word is reduced by four times when compared to the input with a bit-rate of 8bpp. As a result smaller storage space is needed to store the encoded bit-stream and it is easy to transmit encoded bit-stream in lesser transmission bandwidth.

REFERENCES

1. H. Pan, W.-C. Siu, and N.-F. Law, "A fast and low memory image coding algorithm based on lifting wavelet transform and modified SPIHT," *Signal Process.: Image Commun.*, vol. 23, no. 3, pp. 146–161, Mar. 2008.
2. M. Akter, M. B. I. Reaz, F. Mohd-Yasin, and F. Choong, "A modified-set partitioning in hierarchical trees algorithm for real-time image compression," *J. Commun. Technol. Electron.*, vol. 53, no. 6, pp. 642–650, Jun. 2008.
3. A. A. Kassim, N. Yan, and D. Zonoobi, "Wavelet packet transform basis selection method for set partitioning in hierarchical trees," *J. Electron. Imag.*, vol. 17, no. 3, p. 033007, Jul. 2008.
4. M. Dyer, D. Taubman, and S. Nooshabadi, "Concurrency technique for arithmetic coding in JPEG2000," *IEEE Trans. Circuits Systems I, Reg. Papers*, vol. 53, no. 6, pp. 1203–1213, Jun. 2006.
5. H. J. Ritter, "Wavelet based image compression using FPGAs," Ph.D. dissertation, Martin Luther Univ., Halle-Wittenberg, Germany, 2002.
6. R. Stefo, J. L. Núñez, C. Feregrino, S. Mahapatra, and S. Jones, "FPGA-Based modelling unit for high speed lossless arithmetic coding," *Field-Program. Logic Appl. Lecture Notes Comput. Sci.*, vol. 2147/2001, pp. 643–647, 2001.
7. *JPEG2000 Part 1 Final Draft International Standard*, ISO/IEC JTC1/SC29/WG1 N1890, Sep. 2000.
8. F. W. Wheeler and W. A. Pearlman, "SPIHT image compression without lists," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, Jun. 2000, pp. 2047–2050.
9. A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 6, no. 3, pp. 243–249, Mar. 1996.