Design and Implementation of Haar Wavelet Transform and Stripe Logic Based Modified SPIHT Algorithm for Medical Imaging

Mr. Ravi Kiran H K[1], Dr. Paramesha[2]
[1]Research scholar,[2] Asst. Professor and Head
[1] Dept of ECE, ,
[1] RIT,Hassan,Karnataka,India
[1] ravikiranhsn@gmail.com

***Abstract* - Image compression has got wide variety of techniques. Among them Set Partitioning in Hierarchical Trees (SPIHT) algorithm is one of the efficient and simpler techniques in the compression field. The basic SPIHT technique has got two main limitations; one of its limitations is in the memory space and other in the speed of the process. To overcome these limitations, listless SPIHT algorithm has been improved, and a Stripe logic based parallel SPIHT (SPS) technique is implemented in this work that uses three passes with no lists in order to reduce required memory space and to enhance the processing speed. A test bench is developed in Verilog so that the design can be functionally verified using ModelSim software. Initially a MATLAB code has been written in order to obtain text file from the image and then the images are first compressed using Haar wavelet transform (HWT) where the image is decomposed into sub-images. After completing the Haar wavelet process another entropy coding technique known as Stripe logic based parallel SPIHT has been applied where it disintegrates a picture into block of stripes and encodes each and every bit in a bit-plane of that block of stripe simultaneously. The proposed method provides better performance parameters such as Efficiency, Compression Ratio, MSE and PSNR values for medical images.**

***IndexTerms* - Haar Wavelet Transform (HWT), Stripe logic, SPIHT, Compression Ratio and PSNR**

I. INTRODUCTION

While transferring the medical images from one place to another place, reducing memory space and the speed of transmission are the two vital factors. Medical images are generally larger in size and the main challenge is to deal with the storage and transmission requirements of huge data. Compression is the technique to overcome this challenge. In medical image compression we needs to preserve the exact data of the original image as much as possible. Hence, an efficient, high-speed/high-resolution image compression technique became the focus in the research field of medical image compression [5].

In case of remote diagnosis and treatment of patients by means of telecommunication technology such as telemedicine where the images of patients acquired are to be transmitted in compressed form on a constant bit rate channel from one site to the other the accuracy becomes very much important because tumours in the CT scan, fractures in X-rays if wrongly diagnosed may result in the adverse effect on the patients health. In such cases, the need for rapid and precise diagnosis and the drawbacks of bandwidth of the communications medium may need the use of efficient compression technique [4].

Compression is of two types, one is the **Lossless compression** that has the ability to reconstruct the exact original data from the data that is compressed and the image quality is not much decreased and the other is the **Lossy compression** where the exact original image cannot be obtained back from the compressed data and the image quality is somewhat decreased. Some of the Lossless compression techniques are: Huffman coding, Arithmetic coding, Run Length Encoding, Dictionary coders etc., and some of the Lossy compression techniques are: JPEG coding, DCT coding, KLT coding etc.,

HWT is a simplest transform that is hand calculative. It is considered to be a prototype for studying more sophisticated wavelets that divides a image into sub-bands through averaging and deducting method. Compression techniques like EZW and SPIHT are considered to be the benchmark techniques in the field of compression. Besides, better compression and better image quality, the bit stream can be truncated at any point [3]. In SPIHT algorithm aspect many researchers proposed various modifications to improve performance of SPIHT. In this algorithm it mainly aims for better PSNR values [2].

Hybridizing the compression techniques provides still good results. HWT and SPIHT algorithm techniques are the most efficient algorithms in the field of image compression. The proposed paper describes algorithm for implementation of techniques like HWT and Modified Stripe logic based Parallel SPIHT.

II. LITERATURE SURVEY

The authors B. Priti Bramhankar and Sachin B [4] in their paper have provided an approach for compression of medical images by using lifting scheme coupled with SPIHT. Youngseop Kim and William A Pearlman [6] provide a low memory implementation by utilizing stripe based SPIHT efficiently. Yin -hua Wu [5] in his paper has aimed at providing high speed and high resolution in the compression of images. V Vipin [7] has presented a design to perform real time image compression using SPIHT and arithmetic coding technique. Ping lui [8] in his paper has aimed at shortage of the SPIHT algorithm, by adopting

lifting wavelet transform in order to over-come the shortcomings of decoding image quality and coding time. Raghunad K Bhattar, K. R. Ramakrishnan and Dasgupta K S [3] have proposed a paper in which they gives brief theory of Image Compression to reduce the memory requirements using EZW and Stripe based SPIHT algorithm. Initially inorder to reduce the storage space required, many listless algorithms have been proposed. In their paper Kai Liu, Evgeniy Belyaev, and Jie Guo [2] avoids rescanning of image by using Arithmetic coding technique coupled with modified SPIHT without lists algorithm and thereby reducing the memory consumption. Though it efficiently reduces the required memory size, the coefficients are not processed in parallel and hence the processing speed may be low. In order to increase the speed Yongseok Jin and Hyuk- Jea Lee [1] in their paper have described a Block based Parallel SPIHT (BPS) technique for compression that increases the speed of the process. In this paper both the reduced memory space and enhanced processing speed has been implemented by utilizing the HWT and Stripe Logic Based Parallel SPIHT algorithm..

III. METHODOLOGY

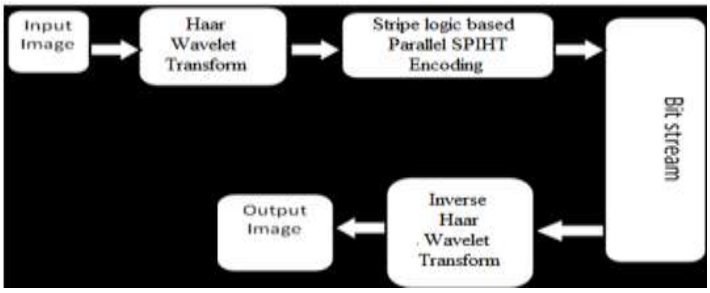

Fig 1: Block diagram of the entire process

Initially any medical image from the database is converted into text file which can either be hex file or the binary file utilizing the MATLAB source code. The converted image pixels are then stored in the memory for further process. After the pixels are read from the memory, it is applied to HWT and the Modified Stripe Logic based Parallel SPIHT algorithm for the compression of image. The compressed image will be decompressed using Inverse HWT in order to get the image that is almost similar to the original image. Then the decompressed image pixel values are stored in memory again. These values are then converted back to text file so that they can be converted into image using MATLAB. Important performance parameters such as PSNR, MSE, CR and Efficiency are calculated.

A. Wavelet Transform

Wavelet transforms are considered to have an extremely important role in the field of image compression because it allows simultaneous analysis of both the time and frequency. For image compression applications, the wavelet transform is a most suitable method when compared to the Fourier transform and Z transform.

Fourier transform is not practical for spectral data computation because it requires all the previous and the future signal data and it cannot observe the frequencies that varies along with time because the function after Fourier Transform is time independent. On the other hand, wavelet transforms relies on wavelets that are differs frequency in certain limited time duration. The Wavelet Transform uses wavelets of finite energy. It gives good time resolution and poor frequency resolution at high frequencies, whereas it provides good frequency resolution and poor time resolution at low frequencies.

Fourier Transform and Z transform can be applied to either cosine or sine parts of the signal whereas WT considers both. And the result will be less redundant in the Wavelet Transform and is efficient for the image compression technique.
Wavelet Transform is of two types,

- **Continuous Wavelet Transform** results in higher number of samples than the input and gives more redundant information
- **Discrete Wavelet Transform** results in lesser or same number of samples as that of the input and it gives the less redundant data

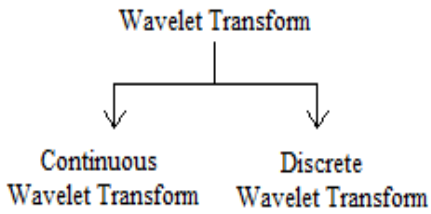

Fig 2: Types of Wavelet Transform

There are different sorts of wavelets such as,

- Haar wavelet
- Morlet wavelet
- Daubechies wavelet

The wavelet transform utilized in this work is Haar Wavelet Transform (HWT).

a) Haar Wavelet Transform

In the year 1910 Alfred Haar introduced the first wavelet system. HWT is famous for its simplicity, straightforwardness and speed of computation. Haar wavelet is considered to be inconsistent and hence not differentiable. Two sorts of coefficients are obtained from the Haar Wavelet Transform,

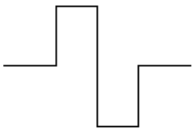

Fig 3: Haar Wavelet

- Coarse approximation (Average of two samples)
- Finer details (Subtraction of two samples)

It has both the forward and reverse transforms

- Forward Transform
  - Computation of scaling elements - adding two adjacent samples and divide by 2
  - Computation of wavelet elements - subtract two adjacent samples and divide by 2
- Inverse Transform
  - Computation requires simply addition and subtraction

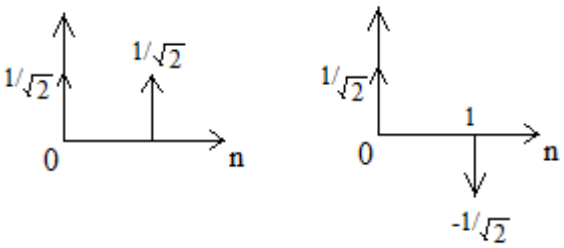

Fig 4: Waveform for the Haar filters

The entire picture is disintegrated into sub images called low-low(LL), low-high(LH), high-low(HL) and high-high(HH) in the first level. And in the second level the low-low(LL) sub band is further sub-divided. LL is considered for further division since it contains more information that is useful.

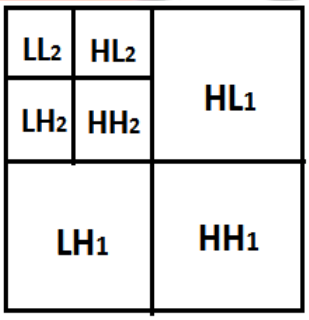

Fig 5: Second level decomposition

B. **SPIHT Technique**

In 1996, Pearlman and Said proposed a wavelet based image compression technique called as SPIHT algorithm. It is an efficient technique for image compression that produces an embedded stream of bits from which the best images in the MSE and PSNR sense can be obtained. It is a powerful implementation of EZW (Embedded Zero Wavelet) method. After the application of wavelet transform to an image, the SPIHT technique partitions the disintegrated wavelet into significant and insignificant pixel values based on the threshold value.

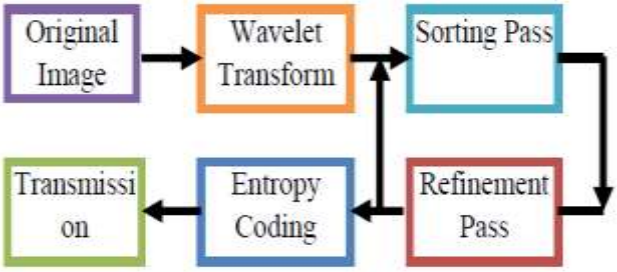

Fig 6: Flowchart of SPIHT

The SPIHT encoding process utilizes three lists

- ***LIP*** (List of Insignificant Pixels) contains elements that have magnitudes lesser than the threshold value
- ***LIS*** (List of Insignificant Sets) contains set of wavelet elemets that have magnitudes lesser than the threshold value and are obtained by tree structures.
- ***LSP*** (List of Significant Pixels) is a list of elements that have magnitudes greater than the threshold.

Each pass is subdivided into two,

Sorting Pass: It encodes the coefficients that are not significant in higher threshold value and when the threshold value is reduced then it may become significant.

Refinement Pass: In the refinement pass, the nth MSB of the coefficients in the significant pixel list is the final output. Coefficients which are already found to be significant in the previous pass are refined in magnitude by a process of successive approximation.

The image is partitioned into approximation and detail sub images. One filter is applied along the columns and then applied along the rows. Thus the operation results in four bands namely low-low(LL), low-high(LH), high-low(HL) and high-high(HH). Each column of an G*H image is filtered and then down sampling will be done which provides two G*H/2 images. Then each row is filtered and sub sampled which gives four G/2*H/2 images of these four sub images, the image obtained by low pass filtering rows and columns is referred as LL image, the image obtained by low pass filtering rows and high pass filtering the columns is referred to as LH image, the one obtained by high pass filtering rows and low pass filtering the columns is referred to as HL image and the other one obtained by high pass filtering rows and columns is referred to as HH image.

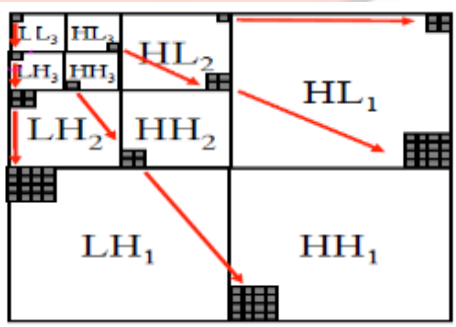

Fig 7: Spatial Orientation trees

Then the bits are separated as significant and insignificant lists based on the conditions. Then sorting and refinement pass are applied on the lists to get the compressed image.

C. Stripe Logic Based Parallel SPIHT Technique (SPS)

In the SPIHT technique, though the utilization of the lists is efficient, it is very memory consuming and the passes are not processed simultaneously and hence the processing speed will be relatively low. Hence in order to overcome these problems, Stripe based Parallel SPIHT technique is proposed in this work.

Wavelet coding requires the buffering of the entire image. For medical images, the requirements to store the elements are very high. Hence to overcome this problem, in this paper Stripe buffer is used that stores few lines of wavelet elements that belongs to the same spatial location where only some bits of the image are processed at once instead of entire image. Figure 8 shows Embedded wavelet coding based stripe logic and Figure 9 shows the Formation of stripes.

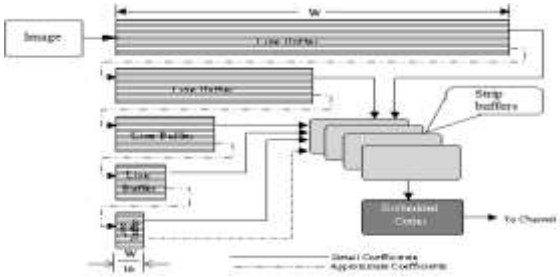

Fig 8: Wavelet coding based on Stripe Logic

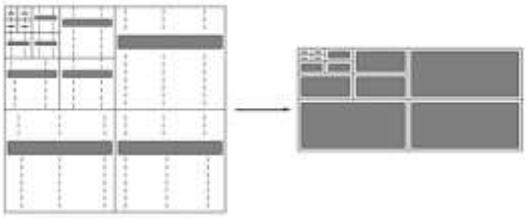

Fig 9: Formation of Stripes

SPS partitions the entire image into stripes and encodes every bit in the bit plane of that stripe. It utilizes three passes namely refinement pass (RP), sorting pass (SP) and first refinement pass (FRP). It disintegrates bit-plane by bit-plane and processes them simultaneously from MSB to LSB plane. For example, the wavelet coefficients of size 8×8 are disintegrated into four elements of 4×4.

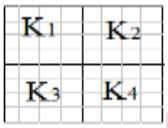

Fig 10: Disintegrated Block

The technique starts from the Non-Zero bit plane. If the highest pixel value in the orientation tree is lesser than 2^6 then the coding starts from the fifth non zero bit-plane.

$\mathcal{K}$: the 4×4 block to be processed.
$\mathcal{L}$: the 2×2 sub-block of $\mathcal{K}$.
n: the bit-plane number

Refinement Pass (RP):
for each $\mathcal{L} \in \mathcal{K}$ such that $S_{n+1}(\mathcal{L}) = 1$ **do**
  for each $w(i) \in \mathcal{L}$ **do**
    output $S_n(w(i))$ (= the nth bit of $|w(i)|$); # magnitude bit
    if $S_{n+1}(w(i)) = 0 \wedge S_n(w(i)) = 1$ **then**
      output the sign of $w(i)$; # sign bit
  set $S_n(\mathcal{L})$ to 1

Sorting Pass (SP):
if $S_{n+1}(\mathcal{K} \cup D(\mathcal{K})) = 0 \wedge \sim (parent(\mathcal{K}) \wedge S_n(parent(\mathcal{K})) = 0)$ **then**
  output $S_n(\mathcal{K} \cup D(\mathcal{K}))$; # sorting bit
  if $S_n(\mathcal{K} \cup D(\mathcal{K})) = 1$ **then**
    for each $\mathcal{L} \in \mathcal{K}$ such that $S_{n+1}(\mathcal{L}) = 0$ **do**
      output $S_n(\mathcal{L} \cup D(\mathcal{L}))$; # sorting bit
      if $S_n(\mathcal{L} \cup D(\mathcal{L})) = 0$ **then**
        set $S_n(\mathcal{L})$ to 0
      else
        set $S_n(\mathcal{L})$ to 1
else
  for each $\mathcal{L} \in \mathcal{K}$ such that $S_{n+1}(\mathcal{L}) = 0$ **do**
    set $S_n(\mathcal{L})$ to 0

First Refinement Pass (FRP):
for each $\mathcal{L} \in \mathcal{K}$ such that $S_n(\mathcal{L}) = 1$ and $S_{n+1}(\mathcal{L}) = 0$ **do**
  set $S_n(\mathcal{L})$ to 1
  for each $w(i)$ in $\mathcal{L}$ **do**
    output the nth bit of $|w(i)|$ # magnitude bit
    if $S_n(w(i)) = 1$ **then**
      output the sign of $w(i)$ # sign bit

Fig 11: SPS Technique for a 4 × 4 block

SPS method overcome the drawback of memory requirement by implementing listless algorithm and the speed will be enhanced by parallel processing of the three passes.

The SPS method for a single stripe block of 4 × 4 is shown in Fig 11. The 4 × 4 stripe block is denoted by K and that is divided again into four 2 × 2 blocks denoted by L n denotes the bit-plane number.

RP visits the blocks that are significant in the previous bit-plane. It provides the nth magnitude bit of the significant block if the pixel is significant in the present bit-plane then it outputs the sign bit. In the SP the block is partitioned as either a

significant or an insignificant stripe block and the sorting bits are transmitted. SP initially checks the significance of a block, and if that block is found to be significant then it will be divided into sub-blocks.

The sub-block that is not significant will be handled by the SP and the sub-block that is significant will be handled by the FRP pass.

IV. IMPLEMENTATION OF HWT AND STRIPE BASED PARALLEL SPIHT TECHNIQUE

A. Software Implementation

In this work, simulation results are seen in ModelSim that is a HDL simulation environment,. This is invoked by MATLAB through the socket server. ModelSim simulator 6.3g_p1 is used in this work.

MATLAB and ModelSim linking can be done in two ways. One through the shared memory server and the other through TCP/IP links using socket calls. If MATLAB and ModelSim are running on two different machines then TCP/IP link is mandatory. For same machine, shared memory will be more useful. For inter process communication, MATLAB starts a server called Hdldaemon.

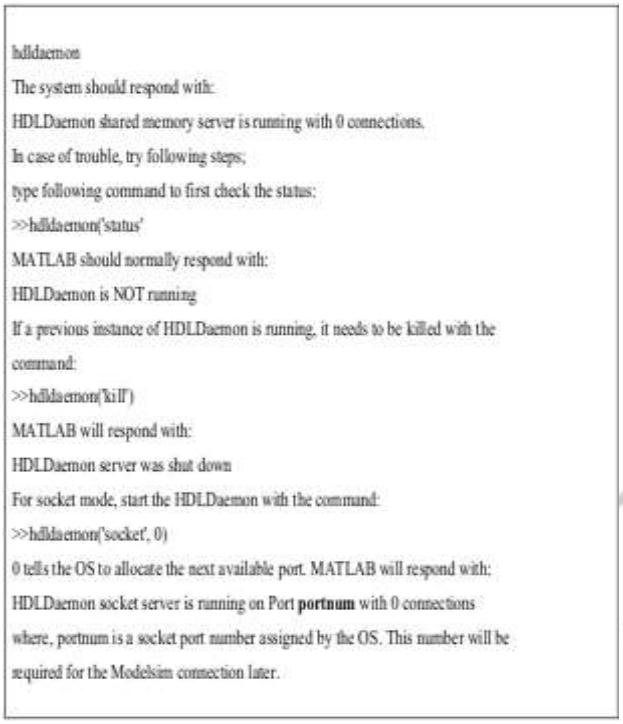
hdldaemon
The system should respond with:
HDLDaemon shared memory server is running with 0 connections.
In case of trouble, try following steps;
type following command to first check the status:
>>hdldaemon('status'
MATLAB should normally respond with:
HDLDaemon is NOT running
If a previous instance of HDLDaemon is running, it needs to be killed with the command:
>>hdldaemon('kill')
MATLAB will respond with:
HDLDaemon server was shut down
For socket mode, start the HDLDaemon with the command:
>>hdldaemon('socket', 0)
0 tells the OS to allocate the next available port. MATLAB will respond with:
HDLDaemon socket server is running on Port **portnum** with 0 connections
where, portnum is a socket port number assigned by the OS. This number will be required for the Modelsim connection later.

Fig 12: Invoking the socket server

Once the MATLAB and ModelSim are linked then the simulation results can be seen directly. The synthesis report and RTL view can be seen in the XILINX window.

B. Hardware Implementation

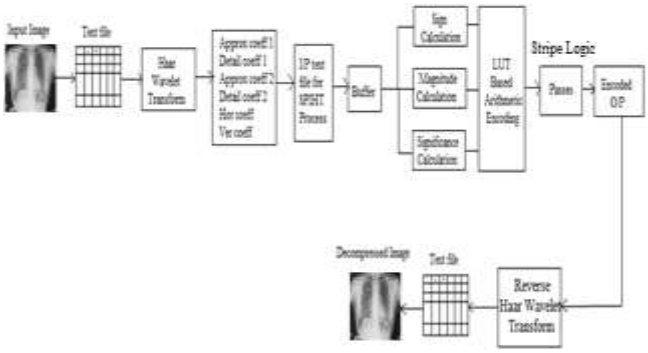

Fig 13: Implementation of HWT and SPS Technique

Using the MATLAB source code initially any medical image from the database is converted into txt.file format. For further process the converted file is stored in the memory. Write enable (we) signal is used for write and read operations.

Once the full memory is read, the counter is reset and starts counting from the beginning. In this work the process is done on the 256×256 image size. Hence, primarily the image of any size is resized to 256×256 and the array of same is assigned. Then the even and odd pixels are separated and stored in different memory locations by using a counter. Then the pixels are stored in a register and applied on Haar Wavelet Transform technique that provides the output of 6 files. Then that is given to the stripe logic based Modified SPIHT coding technique for further process where the sign, magnitude and significance bits are calculated.

The compressed image will be decompressed using Inverse Haar Wavelet Transform in order to get the original image. The decompressed image pixel values are stored in the memory again. These values are then converted to text file so that they can be converted back into image in MATLAB. Specifications like PSNR, MSE, CR etc are calculated. The flow diagram of the proposed process is as shown in the Figure 14.

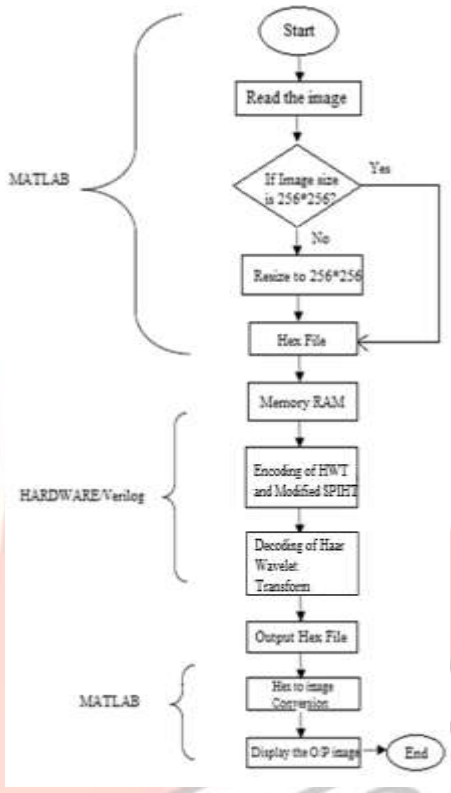

Fig 14: Flow Diagram

V. QUALITY MEASURES

The Reconstructed image quality is measured in terms of Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Compression Ratio (CR) and Efficiency.

- **MSE:** It is also known as the distortion and is defined as the difference between the original and the reconstructed image and is given as,

$$MSE = \frac{\sqrt{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}[f(x,y)-f'(x,y)^2}}{WH} \quad ..... (1)$$

- **PSNR:** It is one of the important measure for the performance evaluation in the image compression field. Higher PSNR is very much desirable for thr image compression technique to be efficient. It is given by,

$$PSNR = 20log_{10}\frac{255}{MSE} \quad ..... (2)$$

- **CR:** It is the ratio of the original input image to the compressed image and is given as,

$$CR = \frac{Input\ image}{Compressed\ image} \cong 1 \quad ..... (3)$$

- **Efficiency:** It is the ratio of the input image to the decompressed output image and is given as,

$$Efficiency = \frac{Input\ image}{Output\ image} \text{ ..... (4)}$$

VI. RESULTS AND DISCUSSION

A. Verilog simulation results

In this paper, verilog code is written in the Xilinx Window. Then compilation will be done with no errors. A test bench is written in Verilog so that the design can be verified using Modelsim software functionally. First, a Matlab code is written in order to obtain a text file from the input image in'.txt'file format. Design and testing of HWT and SPS algorithm has been carried out and is described in verilog. Figure 15 depicts the simulation results of HWT and SPS technique for compression in ModelSim software.

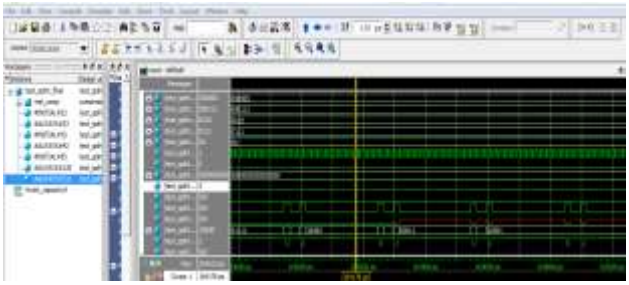

Fig 15: Simulation result of HWT and SPS Technique for compression

Figure 16 depicts the simulation results of Inverse HWT for decompression in ModelSim software.

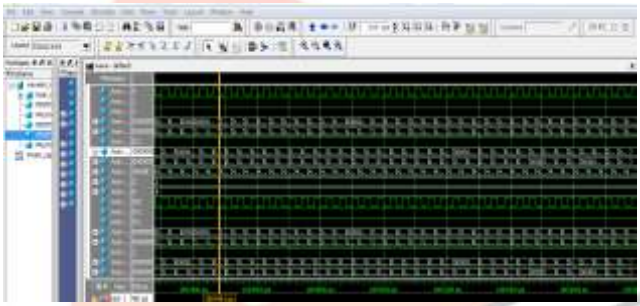

Fig 16: Simulation result of Inverse HWT Technique for decompression

In the hardware synthesis, the RTL view of the entire design can be seen. Figure 17 shows the Register Transfer Logic view of the entire process.

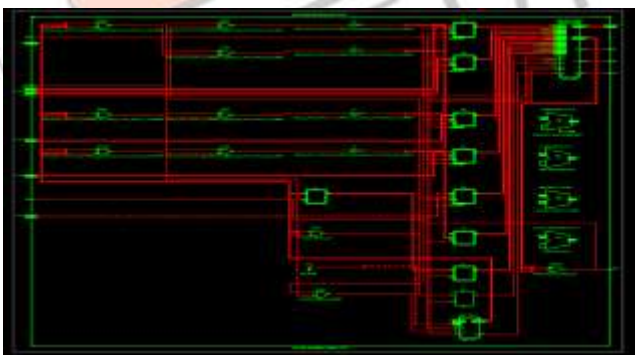

Fig 17: RTL view of the entire process

C. Matlab Simulation Results

Figure 18 shows the the input image and compressed image for the computation of HWT and SPS in MATLAB window.

Input Image

Compressed Image

Fig 18: Simulation result for compression in MATLAB

Figure 19 shows the simulation output of Inverse HWT technique for Decompression in MATLAB software.

Fig 19: Simulation result for decompression in MATLAB

The PSNR Performance for the above image using HWT and SPS technique is **83.6 dB** and MSE is **12.7dB**.

The Table 1 shows different values of CR, Efficiency, Encoding and Decoding time for the entire process.

Table 1: CR, Efficiency, Encoding and decoding time

Sl.no	Parameters	Values
1	CR	72.18dB
2	Efficiency	96.26dB
3	Encoding time	6.941ns
4	Decoding time	7.326ns

VII. ADVANTAGES AND DISADVANTAGES

A. ADVANTAGES

- A better range of compression ratio, PSNR, bit rate and the image quality will be provided
- Most adaptive and best suited for progressive image transmission
- Utilizes less memory space and transmission bandwidth and higher speed can the achieved
- Since lossless compression is reversible, the quality of decompressed image will be almost exact to the original image
- In the Wavelet transforms Blocking artifacts will not be present

B. DISADVANTAGES

- Encoding technique like SPIHT pre-assumes that only dyadic partitioning takes place i.e., only LL sub-band splits iteratively. If we happen to split up any other sub-band for some analysis, then this technique may not work properly.
- Lossless Compression techniques always have got some limitations in amount of compression that can be achieved.
- Bit-plane parallel coding used in this work is not relevant to a decoder. This reduces the decoding speed.

VIII. CONCLUSION AND FUTURE WORK

The proposed method is based on HWT and SPS technique for medical images is presented for real time applications. This technique reduces the memory requirements, enhances the speed and retains all the advantages of embedded coding. The utilization of this method can reduce the MSE or enhance the PSNR values which are the important performance parameters. HWT makes itself a standard technique for its high efficiency. SPIHT which is a powerful technique for still images offers various good characteristics like, good image quality, high PSNR, fast coding and decoding etc., along with HWT technique

provides an efficient design. Experimental results demonstrate that the PSNR value obtained by this method may reach up to 83.6 dB.

As a future enhancement, the work can be extended for designing higher resolution images of size 512×512, 1024×1024 etc. Number of decomposition levels can be increased that may provide still better compression. Other coding technique can be used instead of bit-plane coding since it is not relevant for decoding to obtain still better results.

REFERENCES

[1] Yongseok Jin, Hyuk- Jea Lee, "A Block -based Pass- Parallel SPIHT algorithm", IEEE, July 2012, PP. 1064-1075.BPS

[2] Kai Liu, Jie Guo and Evgeniy Belyaev, "VLSI Architecture of Arithmetic coder used in SPIHT", IEEE, Vol 20, April 2012, PP. 697-710.neighor

[3] K. Raghunad Bhattar, K.R, Ramakrishnan and Dasguptha K.S, "Stripe based embedded coding of wavelet coefficients for large images", IISC, 2013. [I92]

[4] B. Priti Bramhankar and B. Sachin, "Lifting based wavelet transform along with SPIHT algorithm used for image compression", IJMTER , July 2013, PP. 2033- 2037. [liftin]

[5] Yin-hua Wu, Long-xu Jin and Hong-jiang Tao, "An improved fast parallel SPIHT algorithm and its FPGA implementation", IEEE, 2010, PP. 191-195. [2]

[6] Youngseop Kim and William **A.** Pearlman , "Stripe-Based Spiht Lossy Compression of Volumetric Medical Images for Low Memory Usage and Uniform Reconstruction Quality", IEEE, 2000, PP. 652-655

[7] Mr.Vipin V and Miranda Mathews, "FPGA Implementation of Image Compression using SPIHT technique" , IJAREEIE January 2014, PP. 7001-7007.

[8] Ping Liu, Guanfeng Li, "An Improved SPIHT Algorithm for Image Compression in Low Bit Rate", Scientific Research, September 2013, PP. 245-248.

[9] K. Harika and K.V. Ramana Reddy, "Design and Implementation of Arithmetic Coder used in SPIHT", International Journal of Innovative Technology and Exploring Engineering (IJITEE), August 2013, PP. 119-124.

[10] T. Karthikeyan, S. Nithya, "A Comparative Study of SPIHT and Modified SPIHT Algorithms using ultrasound scan images" , International Journal of Computer Application, August 2015, PP. 115-124.

[11] Rema N.R, Binu Ani Oommen, Mythili P, "Image compression using SPIHT with modified spatial orientation trees" , ICICT, 2014, PP. 1732-1738.

[12] Nikola Sprljan, Sonja Grgic, Mislaw Grgic, "Modified SPIHT algorithm for wavelet packet image coding", ELSEVIER, 2005, PP. 378-388.

[13] I. Venkata Anjaneyulu, P. Rama Krishna, "FPGA Implementation of DWT-SPIHT Algorithm for Image Compression", International Journal of Technology Enhancements and Emerging Engineering Research, 2015, PP. 20-24.

[14] Thomas W. Fry and Scott Hauck, "SPIHT Image Compression on FPGAs" , NASA, 2013.

[15] Hong Yu Chao, Chien Ming Wu, "High Performance low complexity Bit plane coding for MPEG-4 FGS", IEEE, 2005.