ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (162-169), Month: October - December 2014, Available at: www.researchpublish.com

# VLSI Implementation of Image Compression and Encryption Using SPIHT and Stream Cipher Method

Miss Sampada Dange[1], Mrs. Sumedha Borde[2]

[1] Student of ME communication, MIT, Aurangabad, India
[2]Lecturer at Dept of Electronics and communication, MIT, Aurangabad, India

***Abstract:*** **This technique is proposed for compression and encryption. Here an image is first compressed using SPIHT compression algorithm combined with Huffman coding and then the compressed image is encrypted using stream cipher. This technique is mainly used for highly efficient encryption and good compression. The Stream cipher method is chosen for their high encryption rate and they are mainly of bit-by-bit encryption. They don't simply mean encryption but provides authenticity and integrity. Then for compression we prefer SPIHT for their unique factors such as larger bit depth, progress from lossy to lossless compression, unrestricted dimensions in image, etc. In this paper Code was simulated using Xilinx and proposed architecture is implemented using FPGA. This paper focuses mainly on highly secure encryption and high compression rate using SPIHT method in Larger Images.**

***Keywords:*** **Huffman coding, SPIHT, EZW, Stream cipher, symmetric key encryption.**

## I. INTRODUCTION

An exponentially growing amount of data, image, video, music, voice, virtual reality, etc., is being transmitted around the world. No matter how much we increase telecommunication bandwidth and disk storage capacity, a vital practical need remains to compress data so that multimedia can be transmitted faster and stored more efficiently. Data compression reduces the size of multimedia by reducing an object's redundancy. Images are very important documents nowadays; to work with them in some applications they need to be compressed, more or less depending on the purpose of the application. There are some algorithms that perform this compression in different ways; some are lossless and keep the same information as the original image, some others loss information when compressing the image.

This paper focuses on two classifications one, the compression and the other is encryption. Image Compression is used to minimize the amount of memory needed to represent an image. Image are often require a large number of bits to represent them, and if the image needs to be transmitted or stored. It is impractical to do so without somehow reducing the number of bits. The choice of compression algorithm involves several conflicting considerations. These include degree of compression required, and the speed of operation. Obviously if one is attempting to run programs direct from their compressed state, decompression speed is paramount.

The other consideration is size of compressed file versus quality of decompressed image. This enables to save more space and time consumption is reduced. There comes the role of SPIHT algorithm which satisfies all the requirements that one needs. In the other end an encryption technique is carried out on compressed image. In cryptography, it is an ocean where various algorithms are present. Usually cryptography is represented as the scrambling of data at the transmitter side using the encryption algorithm and then unscrambling them at the receiver side using the decryption algorithm. A secret key is used to scramble the data and then the same is used to unscramble them. One method is that using different keys at both the transmitter and the receiver side. This type of encryption falls under asymmetric key encryption On the other hand; the secret key is same at both the on transmitter and receiver side, so it falls under the category of symmetric key encryption.

Thus among them the stream cipher is chosen, which comes under the category of symmetric key encryption. Normally in symmetric key encryption, the same key is used at the decryption section to reveal the secret data at the receiver side.

## II. METHODOLOGY

### *A. Compression Technique*

The SPIHT algorithm was designed and introduced by Said and Pearlman (Amir Said and William A. Pearlman. 1996.). SPIHT exploit the spatial dependence by partitioning the pixel values into parent-descendent groups. The coder starts with a threshold value that is the largest integer power of two that does not exceed the largest pixel value. Pixels are evaluated in turn to see if they are larger than the threshold; if not, these pixels are considered insignificant. If a parent and all of its descendents are insignificant, then the coder merely records the parent's coordinates. Since the children's coordinates can be inferred from those of the parent, those coordinates are not recorded, resulting in a potentially great savings in the output bit stream.

In the compression section, best suitable algorithm is SPIHT (Set Partitioning In Hierarchical Trees) is a wavelet based algorithm. Normally the SPIHT algorithm is decomposed using hierarchical wavelet decomposition to form many sub bands. The wavelet transformation is decomposed depending on the frequency, where they are sub sampled by horizontal and vertical channels using sub band filters. Here the reconstruction is done by applying using the inverse discrete wavelet transform. SPIHT is computationally very fast among the image compression algorithms known today. This method saves lots of bits during transmission compared with the Embedded zero tree wavelet transforms. The structure below shows the decomposition of an image into various sub bands namely LL, LH, HL, and HH. The input image is thus first encrypted using stream cipher and then compressed using SPIHT algorithm. In SPIHT, the partitioning decisions are binary decisions that are transmitted to the decoder. Providing a significance map encoding, which is efficient than the EZW algorithm.

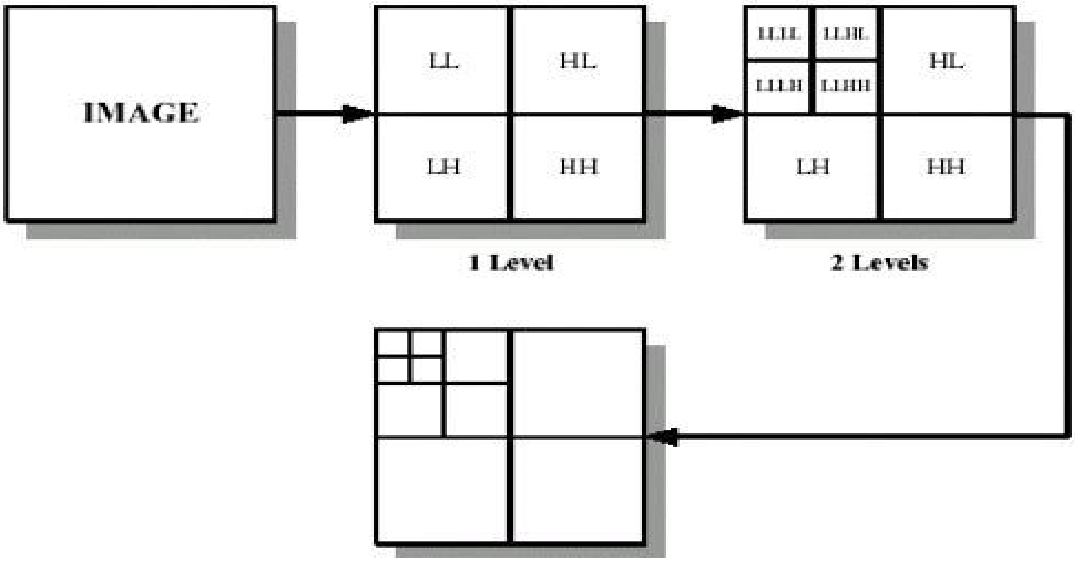


**Fig.1** Discrete Wavelet Transform

### *B. Encryption Technique*

Encryption is the conversion of data into a form, called a cipher text, which cannot be easily understood by unauthorized people. Decryption is the process of converting encrypted data back into its original form, so it can be understood. Encryption is used to protect the confidentiality of information when it must reside or be transmitted through unsafe environments. Encryption is also used for "digital signatures" to authenticate the origin of messages or data. Among the various encryption methods stream cipher belonging to the category of symmetric key encryption emerges the best encryption technique. Stream cipher actually is the simplest of all algorithms. It encrypts one bit at a time, unlike the block ciphers which encrypt the whole block at a time. Usually the sizes, the bit accommodate 16 or 32 bits [8]. The encryption key is generated by a pseudorandom key generator.

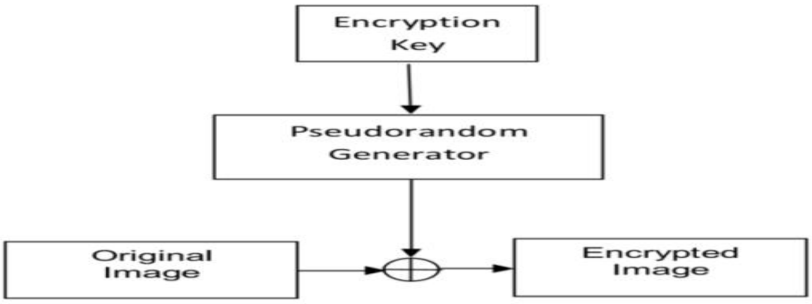


**Fig.2.** Generation of Key and Encrypted Image using Stream Cipher

SPIHT is a wavelet based algorithm. Normally the SPIHT algorithm is decomposed using hierarchical wavelet decomposition to form many sub bands. The wavelet transformation is decomposed depending on the frequency, where they are sub sampled by horizontal and vertical channels using sub band filters. Here the reconstruction is done by applying using the inverse discrete wavelet transform.

SPIHT is computationally very fast among the image compression algorithms known today [3]. This method saves lots of bits during transmission compared with the Embedded zero tree wavelet transforms. The structure below shows the decomposition of an image into various sub bands namely LL, LH, HL, and HH. The input image is thus first encrypted using stream cipher and then compressed using SPIHT algorithm [5]. In SPIHT, the partitioning decisions are binary decisions that are transmitted to the decoder, providing a significance map encoding, which is efficient than the EZW algorithm.

## III. PROPOSED SCHEME

### A. *SPIHT Compression*

The SPIHT algorithm is unique in that it does not directly transmit the contents of the sets, the pixel values, or the pixel coordinates. What it does transmit is the decisions made in each step of the progression of the trees that define the structure of the image. Because only decisions are being transmitted, the pixel value is defined by what points the decisions are made and their outcomes, while the coordinates of the pixels are defined by which tree and what part of that tree the decision is being made on. The advantage to this is that the decoder can have an identical algorithm to be able to identify with each other decisions and create identical sets along with the encoder.

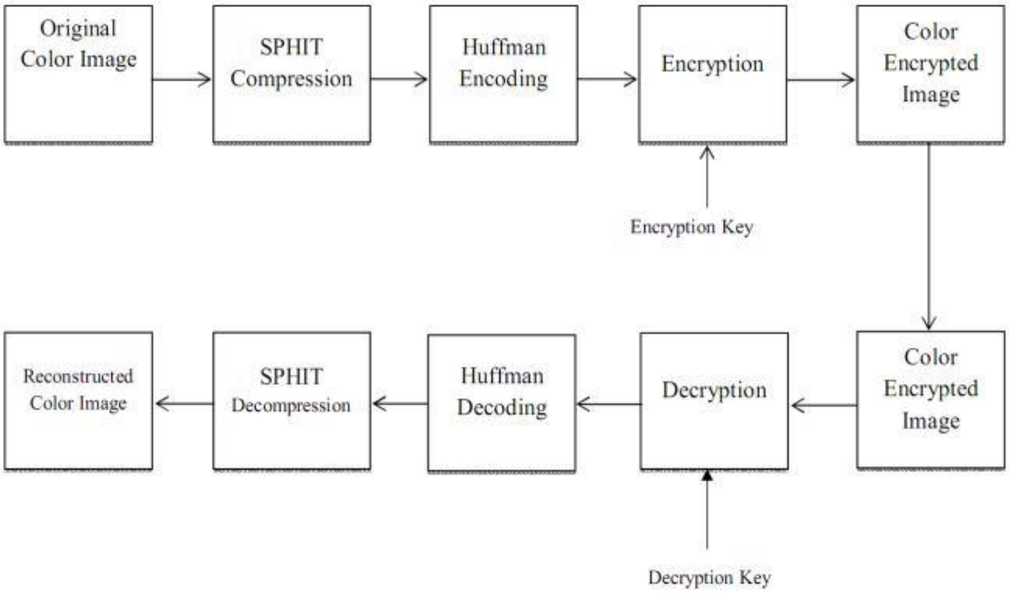


**Fig.3.** Block Diagram of Proposed System

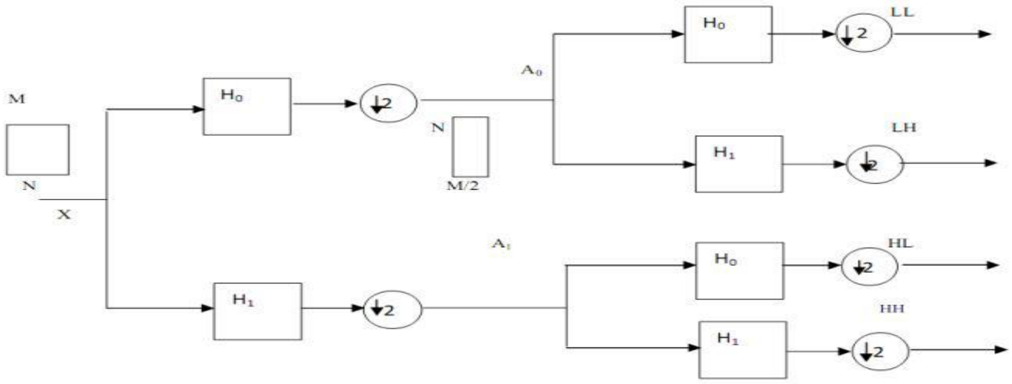


**Fig.4.** Sub Band Decomposition of an MxN image

The part of the SPIHT that designates the pixel values is the comparison of each pixel value to $2n \leq | ci,j | < 2n+1$ with each pass of the algorithm having a decreasing value of n. In this way, the decoding algorithm will not need be passed the pixel values of the sets but can get that bit value from a single value of n per bit depth level. This is also the way in which the magnitude of the compression can be controlled. By having an adequate number for n, there will be many loops of information being passed but the error will be small, and likewise if n is small, the more variation in pixel value will be tolerated for a given final pixel value. A pixel value that is $2n \leq | ci,j |$ is said to be significant for that pass By sorting through the pixel values, certain coordinates can be tagged at "significant" or "insignificant" and then set into partitions of sets. The trouble with traversing through all pixel values multiple times to decide on the contents of each set is an idea that is inefficient and would take a large amount of time. Therefore the SPIHT algorithm is able to make judgments by simulating a tree sort and by being able to only traverse into the tree as much as needed on each pass. This works exceptionally well because the wavelet transform produces an image with properties that this algorithm can take advantage of. This "tree" can be defined as having the root at the very upper left most pixel values and extending down into the image with each node having four (2 x 2 pixel group) offspring nodes.

***SPIHT Algorithm***

```
1. Initialization: output n= |log2 (max (i, j) {c (i, j)}|;
set the LSP as an empty list, and add the coordinates
(i,j) € H to the LIP, and only those with descendants
also to the LIS, as type A entries.
2. Sorting pass:
2.1 For each entry (i,j) in the LIP do:
2.1.1 Output Sn (i,j)
2.1.2 if Sn (i,j) then move (i,j) to the LSP and output
the sign of c(i,j)
2.2 For each entry (i,j) in the LIS do:
2.2.1 if the entry is of type A then
 output Sn(D(i,j));
 if Sn(D(i,j)) then
for each (k,l) € O(i,j)do: output Sn(k,l);
if Sn(k,l) then add (k,l) to the LSP and output the sign
of ck,l;
```

2.2.1 if the entry is of type B then

if Sn (k,l) = 0 then add (k,l) to the end of the LIP;

if L(i,j)≠0then move (i,j) to the end of the LIS, as an

entry of type B and go to Step2.2.2 else, remove entry

(i,j) from the LIS

Output Sn(L(i,j));

if Sn (L(i,j)) = 1 then add each (k,l) € O(i,j) to the end of the LIS as an entry of type A; remove (i,j) from the LIS.

3. Refinement pass for each entry (i,j) in the LSP, except those included in the last sorting pass (with same n), output the n-th most significant bit of |ci,j|;

4. Quantization-step update: decrement n by 1 and go to Step 2.

### *B. Huffman Encoding*

Huffman coding is to find a way to compress the storage of data using variable length codes. Our standard model of storing data uses fixed length codes. For example, each character in a text file is stored using 8 bits. There are certain advantages to this system. When reading a file, we know to Always read 8 bits at a time to read a single character. But as you might imagine, this coding scheme is inefficient. The reason for this is that some characters are more frequently used than other characters. Let's say that the character 'e' is used 10 times more frequently than the character 'q'. It would then be advantageous for us to use a 7 bit code for e and a 9 bit code for q instead because that could shorten our overall message length. Huffman coding finds the optimal way to take advantage of varying character frequencies in a particular file. On average, using Huffman coding on standard files can shrink them anywhere from 10% to 30% depending to the character distribution.

### *C. Image Encryption*

Encryption is the process of scrambling information. It makes the original content into an unintelligible manner, unfit to be read by any attacker. This process of encrypting data can be done either on a block by block basis or on a bit by bit basis. The former method produces Block Cipher output and the latter produces Stream Cipher output. The input image is given to a stream cipher encryption block. Assume the original uncompressed image with gray value range [0,255]. The pixels of this image is represented as pi,j and the bits of these pixels are eight in number and are represented as bi,j,0, bi,j,1,......bi,j,7.

Thus bi,j,k is given as

bi,j,k=pi,j/2^k mod2 k= 0,1,….,7 (1)

pi,j= $\sum_{u=0}^{7} bi,j,u.2^{\wedge}u7$ (2)

In encryption phase, the exclusive-or results of the original bits and pseudo-random bits are calculated

Bi,j,u=bi,j,u $\oplus$ ri,j,u (3)

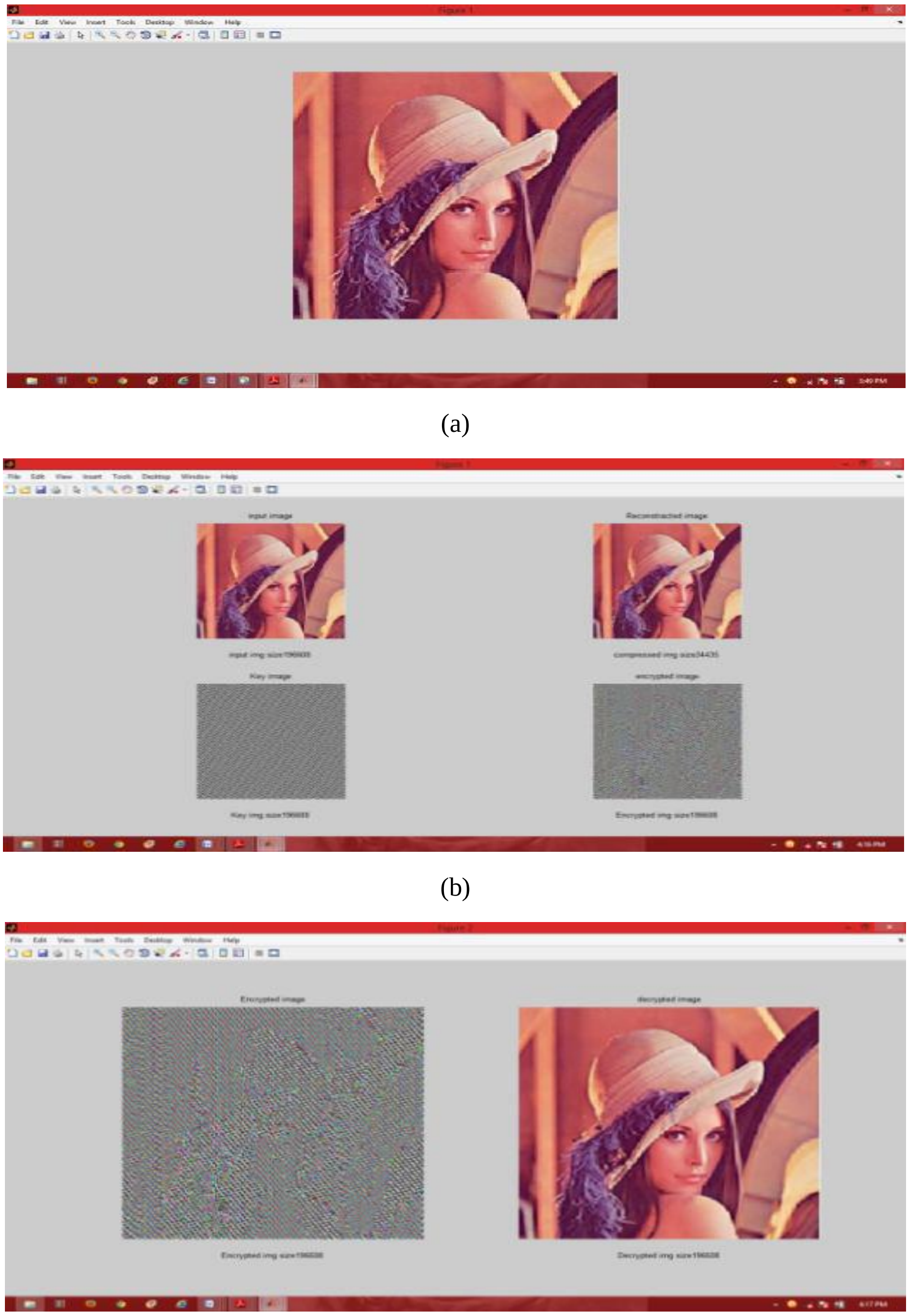
Here ri,j,u are determined by an encryption key using a standard stream cipher [8]. For example consider bi,j,u = 10101010 and ri,j,u = 00001111.Then the output Bi,j,u will be 10100101. This will pseudo-randomly permute the input pixels and produce an encrypted output p’i,j. Advantages of such stream cipher lies in the act that the encrypted output is dependent on the pseudorandom key which offers improved security. They also offer implementation simplicity and higher speed.

In this 1st we implement this algorithm in matlab when we get sufficient outcome then we proceed to next stage where we translate the code in VHDL. The development of algorithm in VHDL is different in some aspects. The main difference is unlike MATLAB, VHDL does not support many built in functions such as convolution, max, mod, flip and many more. So while implementing the algorithm in VHDL, linear equations of DWT is used. The floating point operations have been avoided here. The VHDL code is compiled and simulated using Xilinx ISE9.2i.

Next, the VHDL codes were synthesized using Xilinx which have produced “gatelevel architecture” for VLSI implementation. Finally, the design codes of SPIHT have been downloaded into FPGA board for verifying the functionality of the design.

## IV. RESULT AND DISCUSSION

The method is experimented with an image in matlab. Using matlab we can get simulation result. The image is first undergoes various stages of compression and then it is encrypted using stream cipher method. The first figure shows the original image, ‘lena.jpg’, the second figure is its compressed and encrypted image , the third image is the reconstructed image. The compression ratio is found to increase as the number of compression steps increases.



(a)

(b)

(c)

**Fig.5.** Implemented Outputs. (a) Original Image (b) Compressed and Encrypted Image Output (c) reconstructed image.

**Table. 1** Compression ratio for different images

| | | Compression ratio | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Image name | Size (MxN) | Level 1 | Level2 | Level 3 | Level 4 | Level 5 | Level 6 | Level 7 | Level 8 |
| Eye | 512 x 512 | 42.91% | 43.42% | 44.06% | 51.13% | 52.85% | 53.22% | 53.31% | 53.33% |
| Lena | 256 x 256 | 40.83% | 42.85% | 45.43% | 50.83% | 52.17% | 52.52% | 52.62% | |

### *Synthesis Results*

Synthesis is performed to transform the VHDL code into logic gate level using Synplify 7.0 by Synplicity. The physical hardware layout is generated using synthesis tools. It is a great design approach to take the VHDL code as a basis and translate it automatically into a netlist. After synthesis, 3 RTL views and 106 Technology views have been achieved. One sample Technology view is given in figure 6. The achieved system frequency is 10.8 MHz. In this design 64 bit registers have been used.

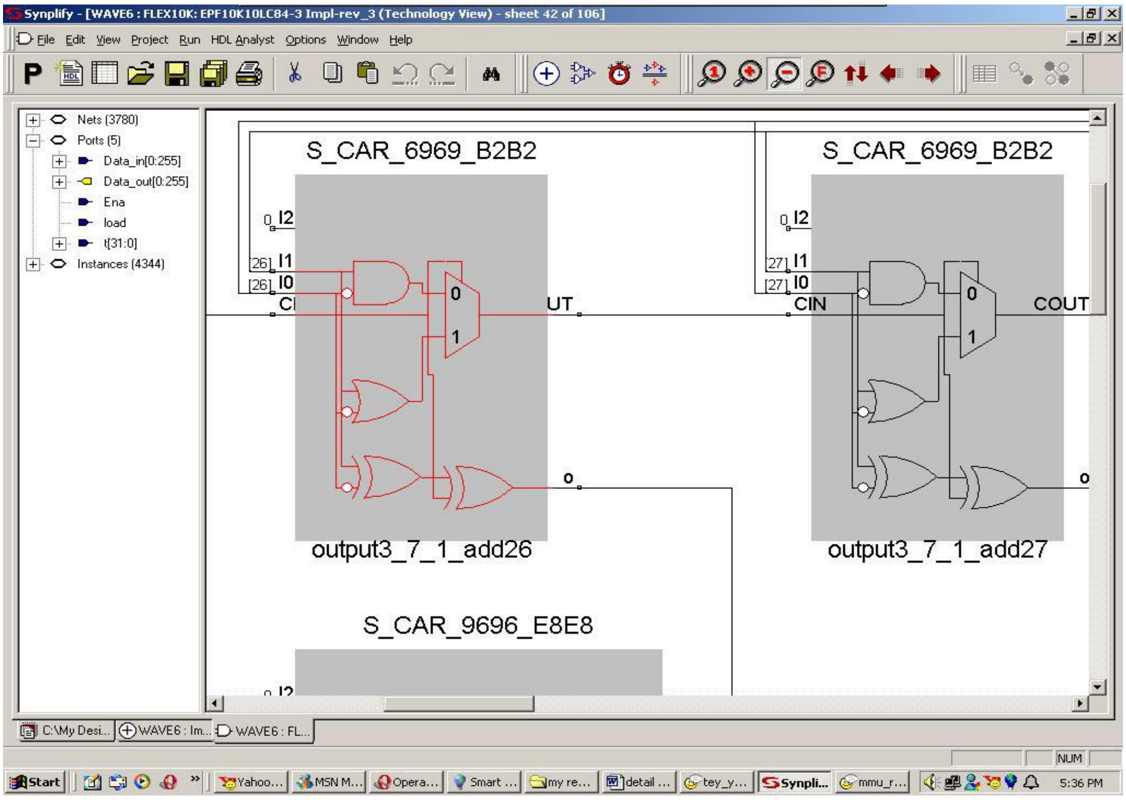


**Fig. 6** Zoomed Technology View of the Design

### *Discussion*

This work focuses on the digital VLSI implementation of SPIHT image compression and encryption algorithm using VHDL. In order to increase the performance and reduce computational complexities many modifications have been made. We have found that higher decomposition is expected to cause higher compression ratio. In order to reduce the complexity and increase computation speed, linear algebra equations of DWT are used while implementing the algorithm in VHDL. This linear algebra approach gives almost the same output. But it takes very less resources. Also in order to balance between the time needed by serial input and resources needed by parallel input, a more suitable approach has been taken where 8 image co-efficients are processed at a time.

## V. CONCLUSION

In this paper, the SPIHT compression combined with Huffman encoding is carried out. And it provides better compression as the size of the larger images can be chosen and can be decompressed with the minimal or no loss in the original image.

Thus on the other hand a stream cipher encryption is carried out to provide best encryption so as to make it highly complicated and secure that other than the arbiter and the receiving party cannot notify or visualize what type of encryption is carried out. Thus high and confidential encryption and best compression rate has been energized to provide better security.

## VI. ACKNOWLEDGEMENT

I am thankful to our principal Dr. C. L. Gogte for his valuable guidance and also thankful to our department head Dr. Ajij sayyed for his encouragement and guidance. Also am thankful to those authors whose paper I have reffered.

## REFERENCES

[1] “Compressing still and moving images with wavelets” by M L Hilton, B D Jawerth and Ayan Sengupta, published in the journal “Multimedia Systems” vol – 2 no-3, 1994.

[2] Rabbani M, Jones P (1991) Digital Image Compression Techniques. (SPIE tutorial texts in optical engineering, vol TT7) SPIE press, Washington.

[3] JPEG official website, www.jpeg.org/jpeg2000.html

[4] Wavelets and Filter Banks by Gilbert Strang and Truong Nguyen,Wellesley-Cambridge Press, 1997

[5] Ripples in Mathematics: the Discrete Wavelet Transform by Arne Jense and Anders la Cour-Harbo, Springer, 2001

[6] “Image Compression Using The Haar Wavelet Transform” by Colm Mulcahy, Spelman College Science & Mathematics Journal, Vol 1, No 1, April 1997, 22-31

[7] K. K. Parkhi, T. Nishitani, “VLSI Architecture for discrete wavwlet trasnsform”, IEEE Trans. On very large scale integration (VLSI) system,vol 1,pp 191-202, Jun 1993.

[8] Abdullah Al Muhit ,Md. Shabibul Islam and Masuri Othman, “ VLSI Implementation of discrete wavwlet transform (DWT) for image compression”' 2$^{nd}$ international conference on autonomous robots and agents dec, 2004.

[9] C. Rengarajaswamy and S. Imaculate Rosoline,”SPIHT Compression on encrypted images”, IEEE, pp.336-341,2013

[10] Sadashivappa, K. V. S. Anand babuand Dr. shrinivas,” colour image compression using SPIHT algorithm”, International journal of computer applications, vol 16, 2011