

FPGA implementation of newLM-SPIHT colored image compression with reduced complexity and low memory requirement compatible for 5G

Yasmine M. Tabra, Bayan Mahdi Sabbar

Department of Information & Communication, Information Engineering College, Al-Nahrain University, Iraq

Article Info

Article history:

Received Nov12, 2018

Revised Jan 30, 2019

Accepted Feb 8, 2019

Keywords:

5G

Runlength

SPIHT

VHDL

Wavelet Transform

ABSTRACT

The revolution in 5G mobile systems require changes to how image is handled. These changes are represented by the required processing time, the amount of space for uploading and downloading. In this paper, a development on WT (Wavelet Transform) along with LM-SPIHT (Listless-Modified Set Partitioning in Hierarchical tree) coding and with additional level of Runlength encoding for image compression has been proposed. The new implementation reduces the amount of data needed to be stored in several stages, also the amount of time required for processing. The compression has been implemented using VHDL (Very High Descriptive Language) on netFPGA-1G-CLM Kintex-7 board. The new implementation results show a reduction in the complexity as processing time.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Yasmine M. Tabra,

Department of Information & Communication, Information Engineering College,

Al-Nahrain University, Iraq.

E-mail: yasminetabra@yahoo.com

1. INTRODUCTION

With the fast development of mobile systems, techniques need to be changed to cope with these developments; among the modifications that need to take place such as image compression. Image compression can be achieved by removing one or more of the three basic data redundancies as briefly outlined: [1]

- Coding redundancy: if the number of bits per pixel that is required to represent the image is higher than is necessary.
- Inter pixel redundancy: The correlations among image pixels, which result from the structural or geometrical relationships between the objects in the image lead to inter pixel redundancy.
- Psycho visual redundancy: The less important information is considered to be redundant since it is ignored by the human vision system and hence omitted.

Many researchers implemented JPEG2000, but the problem was always the high decoding complexity and the amount of memory required to store data.

JPEG 2000 image compression is based on Discrete Wavelet Transform (DWT) which is considered a Lossy compression method. The objective of DWT coding is to divide the spectrum of one image into the Low-pass and the High-pass components. JPEG 2000 is a 2-dimension DWT based Image Compression standard [2].

A coding algorithm developed for DWT transformed images is the Set Partitioning in Hierarchical Trees Algorithm (SPIHT). The SPIHT algorithm can be applied to grey-scale and colored images. It ensures that important information is restored first to make it an effective use in networks[3]. SPIHT algorithm is applied on the wavelet transformed image to reduce the correlation between neighboring pixels especially when original image is concentrated in the lowest frequency band of the transformed image [3].

In this paper a modified version of SPIHT is based on reduction of three processing lists in sorting pass into single processing list including refinement passes. Also to improve compression performance and further reduction in memory requirement, Listless SPIHT has been proposed. In L-SPIHT, coefficients are extracted from the whole wavelet-transformed image and encoded separately to reduce the memory requirement without the need to store lists.

Another modification to SPIHT was introduced is the use of three trees instead of continuous scanning. This will help reduce the processing time by performing parallel processing to the three trees. As a final compression step, Runlength Encoding is a very simple form of lossless data compression by storing a single data value and its repetition count. This is most useful on data containing many such runs such as icons, line drawings, and animations [4].

From the parameters that need to be calculated to test the quality of the reconstructed image are Mean Square Error (MSE) and peak signal to noise ratio (PSNR) ratio. Let's assume that the original image is 'A' and the reconstructed image is 'B'.

$$MSE = \frac{1}{N} \sum_{i,j} [A(i,j) - B(i,j)]^2 \quad (1)$$

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right] \quad (2)$$

The original and the reconstructed images are considered indistinguishable by human eyes if the PSNR value is 40 dB or greater [5]. Another performance factor is the compression ratio of the image is given by

$$CR = \frac{\text{No of bits in original image}}{\text{No of bits in compressed image}} \quad (3)$$

Yashaswini P R1, and Mr. Ravi Kiran [5] used a modified SPIHT that utilizes only a single list instead of three lists. Thomas W. Fry, and Scott Hauck [6] used a modification to the original SPIHT algorithm to reduce the computation.

2. THE PROPOSED METHOD

2.1. Discrete Wavelet Transform

The DWT has become so popular that it provides an efficient method for sub band decomposition of signals. The process starts by passing signal through filters with different cut-off frequencies [7]. These filters are the low pass and high-pass decomposition filters to generate four lower resolution components: one low-low (LL1) sub-image, which is the approximation of the original image and three detailed sub-images, which represent the horizontal (LH1), vertical (HL1) and diagonal directions (HH1) of the original image [8,9].

Firstly, the rows of the array are passed to the filters to divide the array into two vertical halves, with the first half represents the average coefficients, while the second vertical half represents the detail coefficients. Secondly, the process is repeated again with the columns, resulting in four sub-bands [10].

Figure 1 shows the output of the two-dimensional wavelet transform for two decomposition steps in a conceptual way. L denotes a low-pass and H a high-pass band. The first letter stands for the transforming of the rows while the second one relates to the columns and the number to the decomposition step [11].

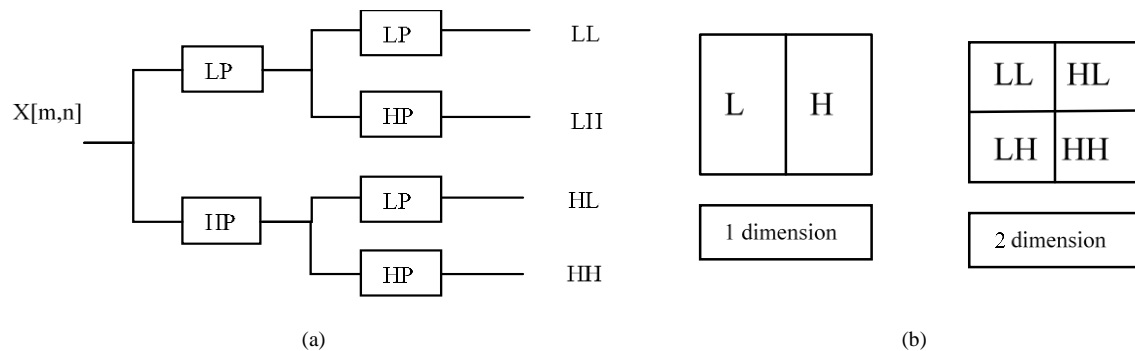


Figure 1. 2-Dimensional DWT

2.2. Set Partitioning in Hierarchical Trees Algorithm (SPIHT)

Embedded coding is based on threshold where values greater or equal to the threshold are called significant. While values less than threshold known as insignificant as in (4)

$$S_n(t) = \begin{cases} 1, & \max\{|c_{i,j}|\} \geq 2^n \\ 0, & \text{otherwise} \end{cases}, \text{ where } 2^n \in |c_{i,j}| < 2^{n+1} \quad (4)$$

This indicates that if the coefficient with maximum magnitude in a set is significant, then the significant test result is 1 [11]. The value of n can be obtained by using (5)

$$n = \lceil \log_2 \max\{|c_{i,j}|\} \rceil \quad (5)$$

The encoder and the decoder perform the same steps and when the decoder receives the comparisons results with threshold from the encoder, it can recover the ordering information from the execution path. [12]. For a given bit rate, the lower bits is used to represent significant values, gives a lower output bit rate [13].

The Wavelet- coefficients can be classified into three sets where each set contains the representative of whole subtree. These trees are List of Significant Set (LIS), List of Significant Pixels (LSP), and List of Insignificant Pixels (LIP) [12].

- LIS includes the location of the coefficients that considered insignificant with respect to selected threshold.
- LSP include the pixels that are significant with respect to selected threshold.
- LIP includes the pixels that are considered insignificant with respect to the selected threshold.

Every pass is divided into two parts: [14]

Sorting pass: in this pass every value that considered insignificant in previous pass and its significant in this pass are encoded.

Refinement passes: the nth MSB of all the significant values are output.

Passes will continue until target data rate is reached or n=0.

2.3. Runlength Encoding

Runlength coding is considered as a one form of lossless compression. It represents a run of same numbers by two values, the number followed by the its run counts [4].

Runlength Encoding provides large compression of data, when the data contains large number of runs. One drawback is that in some times when the data contain small number of runs, it provides no compression. The worst case if there are no runs this will lead to increase the sequence size instead of reducing it [15].

3. THE PROPOSED METHOD

The proposed model is depicted in Figure 2. At first, the input RGB image with size QCIF (176x144) is converted to binary file using MATLAB program to be used as the input to the VHDL program.

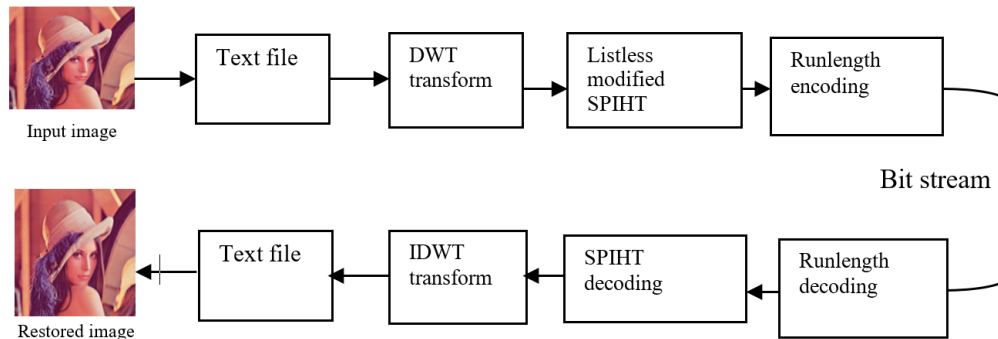


Figure 2. The proposed model

The wavelet filter is carried out for the red, green, and blue image layers in parallel. Each 8x8 block are processed individually to reduce processing time. Hence only 64 pixels (8x64 bits) needed to be stored in buffer before it is passed to the next stage. And further reduction is made by passing only details subset (LL) of the output of wavelet filter to the SPIHT encoder. This is approximately quarter of the original image (76x88) pixel.

A LM-SPIHT is proposed where three trees are used instead of one tree. Each 8x8 block of the detail subset is divided into three trees and without the need to store lists of significant and insignificant. The 8x8 block of image with three trees zigzag scanned is shown in Figure 3.

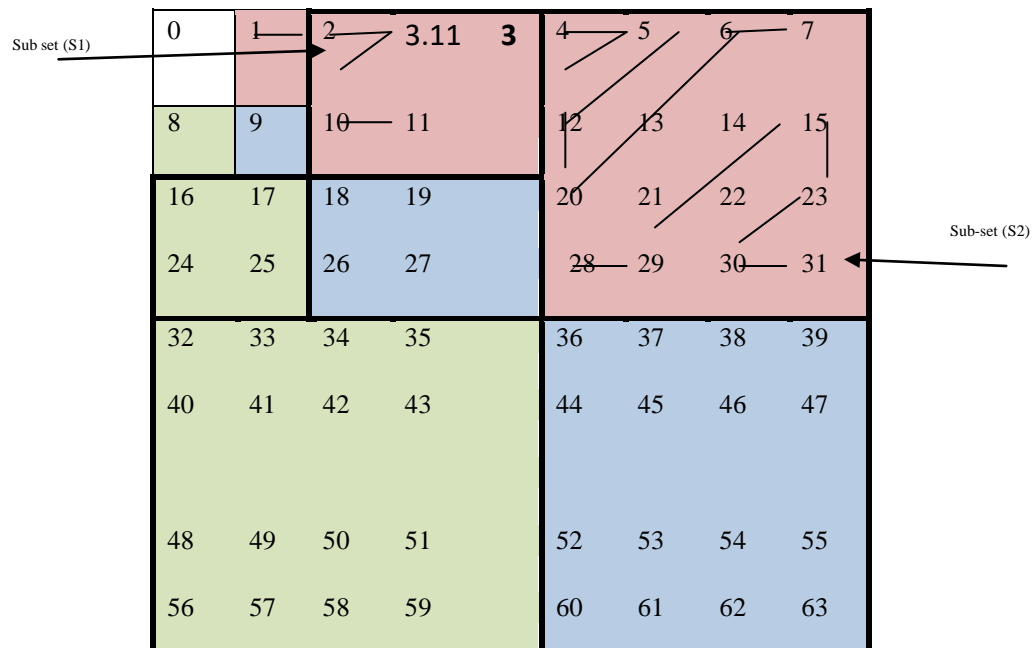


Figure 3. Three trees in 8x8 block with zigzag scan

The zigzag scanning order is performed on each tree then each value is compared to threshold in parallel. The parallelization of the computation process of each tree reduces the computation time. The number of levels where trees are compared to threshold depends on the required output data rate. The less number of levels is the less data rate. Parallel processing has been adopted in implementing the SPIHT coding, where each tree is processed individually. Figure 4 shows a flowchart of LM-SPIHT parallel processing. The output of SPIHT in previous step is encoded using runlength encoding. Runlength is useful when applied on the SPIHT output bitstream. Since the SPIHT output are represented by three values only (1: significant list, 10: significant pixel, and 0: insignificant pixel) this makes the output contains a long successive runs of 10 or 0. Flowchart of Runlength encoding is as in Figure 5.

This step reduces the number of required bits to be transmitted to the receiver. Hence reducing the data rate of the overall system to makes it more compatible to work in the 5G mobile networks. Table 1 shows the LM-SPIHT algorithm step by step. The output of SPIHT in previous step is encoded using runlength encoding. Runlength is useful when applied on the SPIHT output bitstream. Since the SPIHT output are represented by three values only (1: significant list, 10: significant pixel, and 0: insignificant pixel) this makes the output contains a long successive runs of 10 or 0. Flowchart of Runlength encoding is as in Figure 5. This step reduces the number of required bits to be transmitted to the receiver. Hence reducing the data rate of the overall system to makes it more compatible to work in the 5G mobile networks. Table 1 shows the LM-SPIHT algorithm step by step.

Table 1.LM-SPIHT Step by Step Process

step 1	Convert input image into pixel values using MATLAB
step 2	Store pixels in text file.
step 3	Divide image into 8x8 blocks.
step 4	Perform wavelet transform on each block.
step 5	If 'LL' quarter of wavelet output then
step 6	Compute threshold
step 7	Divide block into three trees
step 8	If at least one value in subset (S1) \geq threshold then
step 9	If value in S1 \geq threshold then out=10
step 10	If value in S1 < threshold then out=0
step 11	Else
step 12	Skip subset (S1)
step 13	If at least one value in subset (S2) \geq threshold then
step 14	If value in S2 \geq threshold then out=10
step 15	If value in S2 < threshold then out=0
step 16	Else
step 17	Skip subset (S2)
step 18	Compute refinement for significant values
step 19	Go to step 8 using threshold=threshold/2
step 20	if n=0 or exceed data rate then runlength encoding for out code

4. HARDWARE IMPLEMENTATION

The system was implemented using netFPGA-CML-IG Kintex-7 board in the stand alone form of connection as shown in Figure 6. Xilinx ISE Design Suite 14.7 is used to write the VHDL code which has to be then compiled with zero errors. A test bench code is written in VHDL.



Figure 6. NetFPGA board in standalone mode of connection

The implementation starts by wavelet transforming the input image. The image is uploaded from the text file into ROM component otherwise it will consume a lot, if not all of the FPGA device. For simplicity, the ROM could be preloaded from a file on disk containing the original input image. In the hardware synthesis part, the RTL view of the wavelet top level design is depicted in Figure 7.

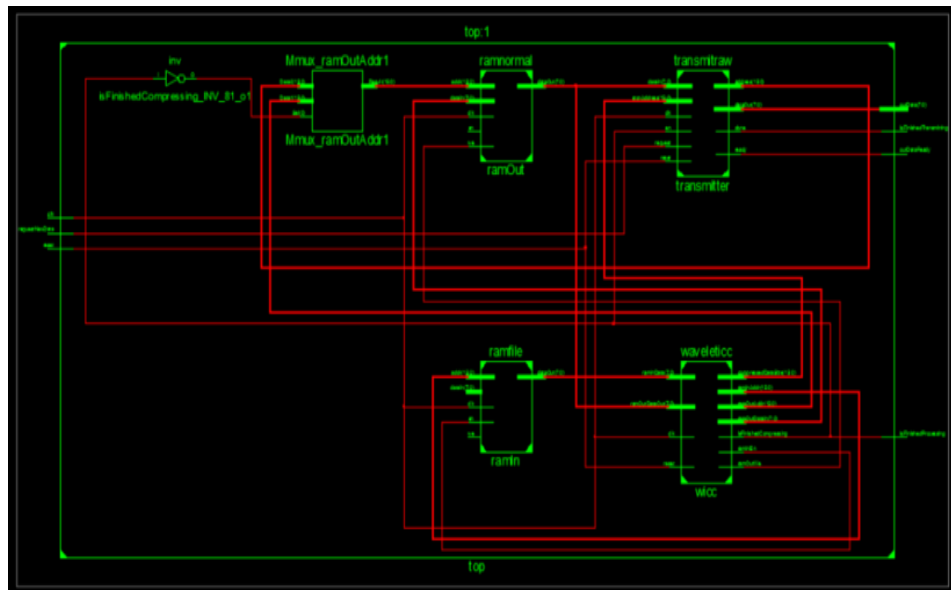


Figure 7. RTL view of the wavelet top model

To implement one level 2D-DWT, the input bits are passed to shift registers with 2-bits to perform FIFO operation. The output bits are then passed to the low pass filter and high pass filter in parallel. The operation continues until all rows in the image completion. Then the output of each filter is again passed to 2-bit shift register to perform the FIFO operation. Also the same operation is done for bits in each column. The Register Transfer Logic view of the wavelet circuit is shown in Figure 8. Next step is implementing the low and high pass filters. The low-pass filter can be implemented as an average between the current value of a signal, and the previous value as (6) :

$$y_L = \frac{x[n] + x[n-1]}{2} \quad (6)$$

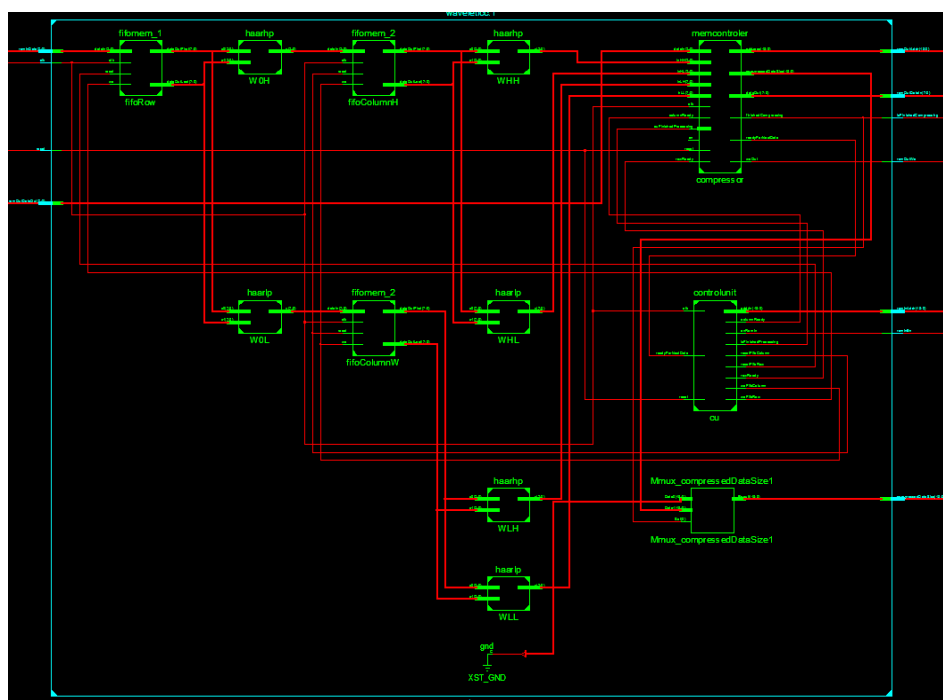


Figure 8. RTL view of the wavelet filters

Intuitively, the high-pass filter is calculated by determining the distance between the average and one of the signal values (or, in other words, the difference between the current and the previous signal value, divided by 2): as (7):

$$y_H = \frac{x[n] - x[n-1]}{2} \quad (7)$$

The data cannot be directly stored to the output RAM, because the compression yet needs to be performed. For this purpose, a memory controller device is created which takes the decomposition values and controls the address and data buses of the output RAM to store the correct values. This operation is performed for the red, green, and blue layers of the image in parallel. Figure 9 depicts the simulation results of wavelet transform. A sample of the SPIHT encoder code written in Xilinx ISE is shown in Figure 10.

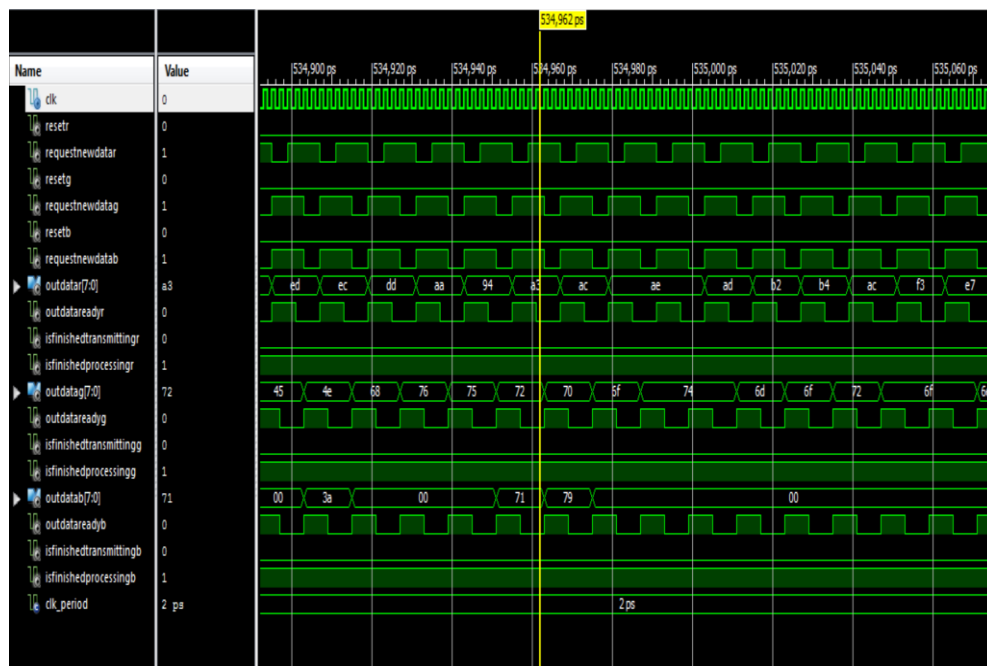


Figure 9. Simulation results of wavelet transform

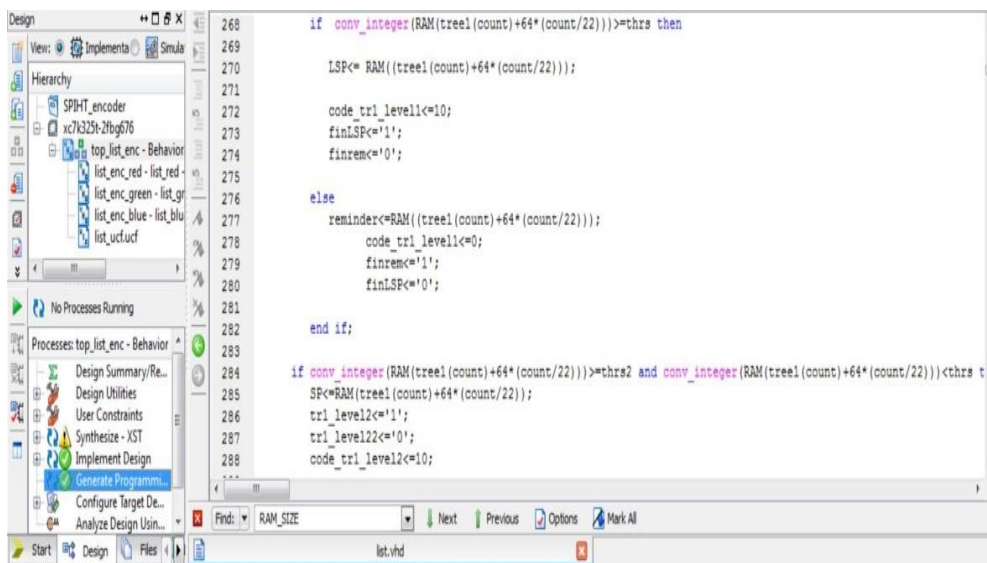


Figure 10. Sample code of SPIHT encoder

The output of wavelet transform is next SPIHT encoded (note only LL is being encoded). The RTL configuration is in Figure 11. The same operation is done for red, green, and blue in parallel. The simulation results of SPIHT encoding shown in Figure 12. A sample of the SPIHT decoder code in ISE is shown in Figure 13. The VHDL code is downloaded to netFPGA device using JTAG connector and by using the SVF (Serial Vector Format) mode, multiple code programs are simultaneously downloaded in the sequence shown in Figure 14.

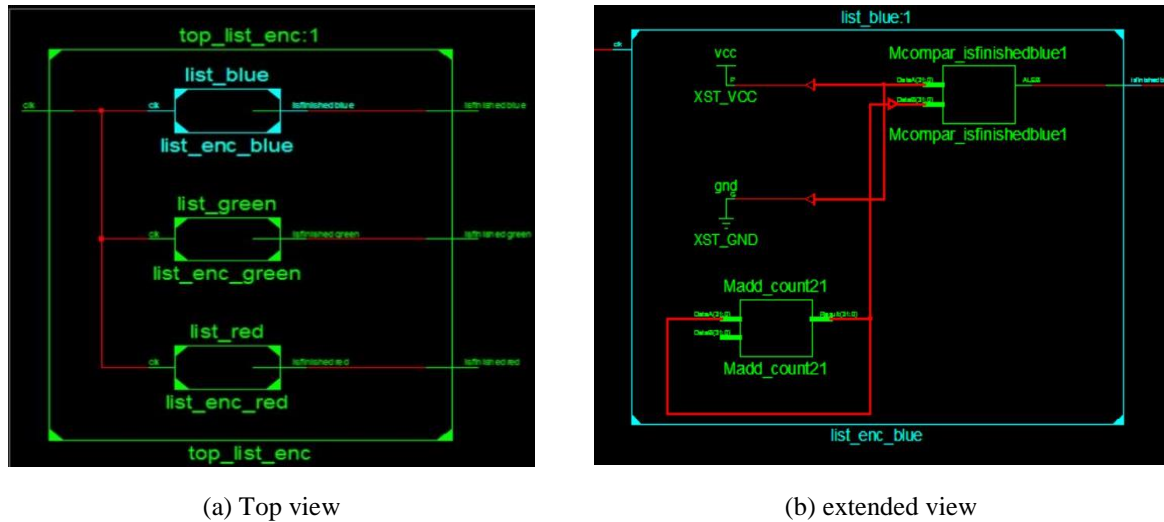


Figure 11. RTL view of the SPIHT encoder

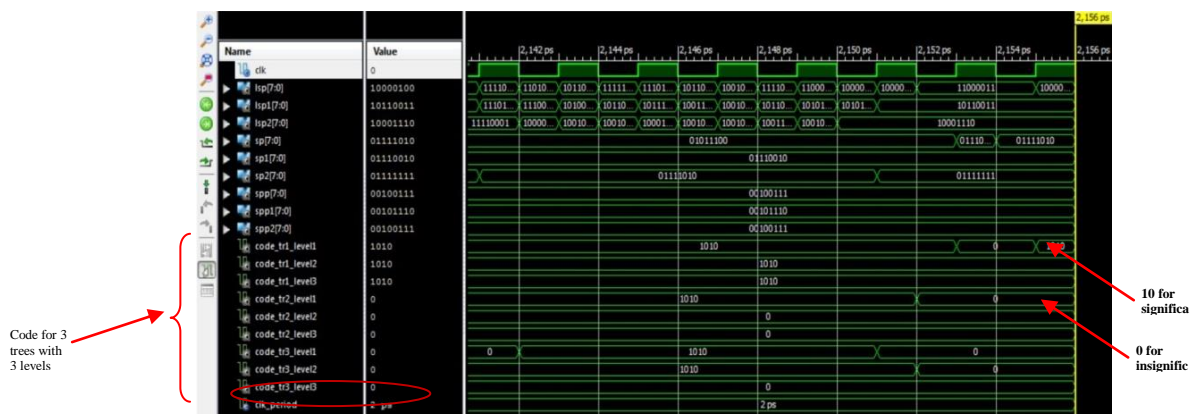


Figure 12. Simulation result of SPIHT encoder

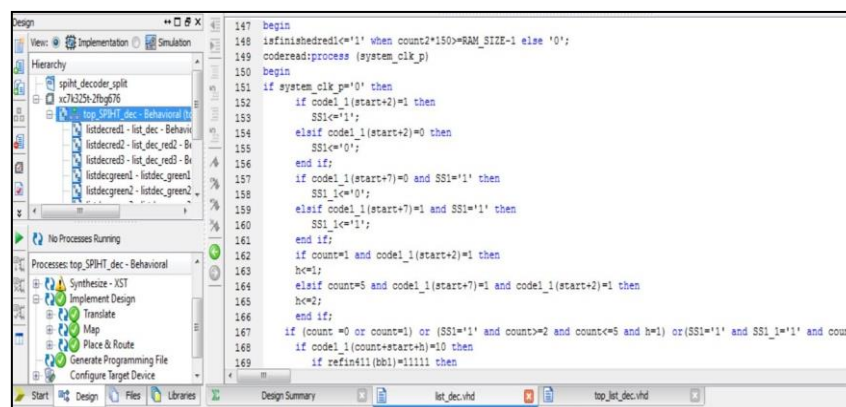


Figure 13. Sample code of SPIHT encoder

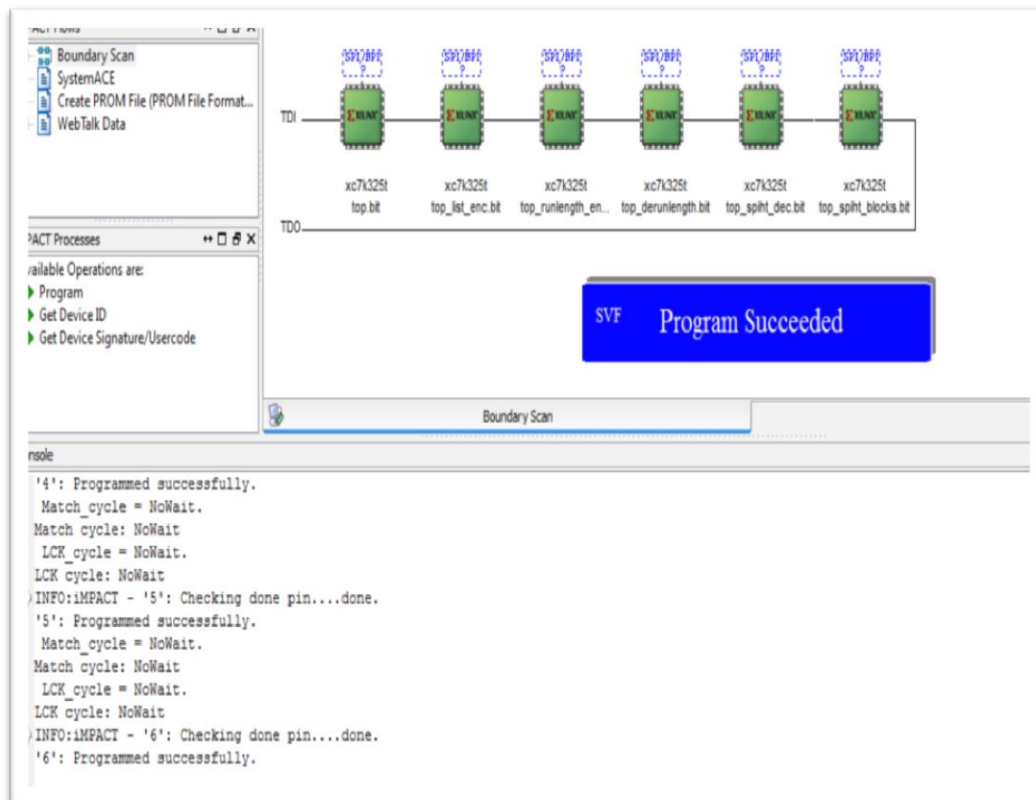


Figure 14. Xilinx iMPACT software

5. RESULTS AND DISCUSSION

The input image for DWT, LM-SPIHT and the restored image from DWT and LM-SPIHT for Compression are presented in MATLAB software. Figure 15 shows the encoding and decoding output for wavelet transform with LM-SPIHT.

As it is noticed the difference between original and reconstructed image is undetectable by human eyes. The difference is only noticeable through calculation or graphical representation. Since the PSNR Performance for the Lena image using Wavelet transform and LM-SPIHT technique is 51.4 dB and MSE is 46.56 dB which is greater than 40 dB, then the two images are virtually indistinguishable. The compression ratio in Lena image is 40% shows that the image has been compressed. As shown in Table 2.



Figure 15. Output results

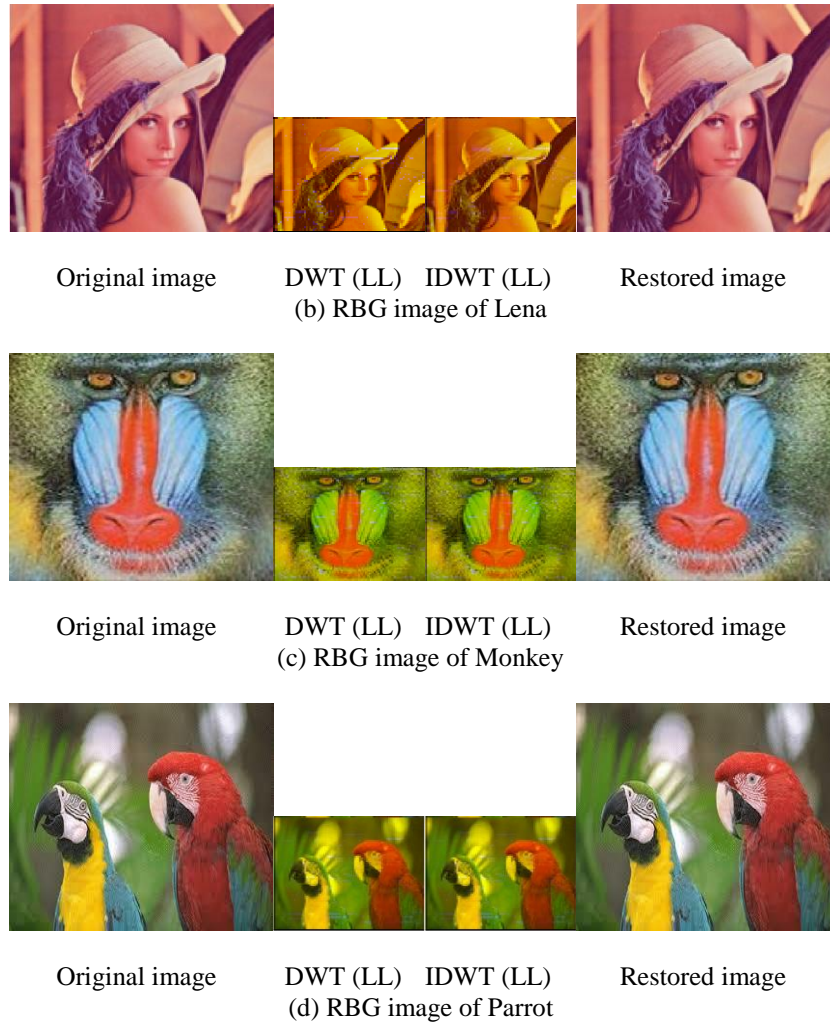


Figure 15. Output results

Table 2. CR, MSE, and PSNR in dB

image	CR	MSE	PSNR
Lena	41.16	0.4656	51.45
Monkey	42.54	0.4523	51.57
Parrot	40.64	0.5421	50.79

MATLAB simulation for Lena image with a size of $176 \times 144 \times 3$ took 23.44ms. For the same image, the hardware implementation time using FPGA was calculated using the results obtained through timing simulation. The total time taken by the simulations was 1605.166 ns, thus, the total number of clock cycles can be calculated as below.

1 clock cycle in wavelet program=10 ns.

Total number of clock cycles for wavelet Encoder= $3000000000\text{ps}/10000\text{ps}=300000$

1 clock cycle in LM-SPIHT=2 ps.

Total number of clock cycles for LM-SPIHT Encoder= $2155\text{ps}/2\text{ps}=1077$

1 clock cycle in Runlength=2 ps.

Total number of clock cycles for Runlength Encoder= $3798\text{ps}/2\text{ps}=1899$

The Xilinx LM-SPIHT Encoder has clock cycle=5.298 ns.

The expected total time is taken by the wavelet transform and LM-SPIHT Encoder for an image size of $176 \times 144 \times 3$ is calculated as below. T_{MATLAB} and T_{FPGA} are simulation time for MATLAB implementation and VHDL implementation.

$T_{\text{MATLAB}}=23.44$ ms

$T_{\text{FPGA}}=\text{Total Number of Clock Cycles} * \text{Clock Period}=(1077+300000+1899)* 5.298 \text{ ns}=0.001605166848 \text{ ms}$

FPGA implementation of new LM-SPIHT colored image compression with reduced... (Yasmine M.)

$$\text{Speed-up} = T_{\text{MATLAB}}/T_{\text{FPGA}} = 14604$$

From the calculation it is being seen that the hardware implementation speeded up the processing of Lena image 14604 times faster than that of the results obtained from software (MATLAB) simulations, thereby making it highly promising for real-time and memory limited mobile communication. LM-SPIHT was compared with several models in literature by mean of speedup factor, FPGA processing time, and PSNR as in Table 3.

Table 3. Compression with Other Models

Model	Speedup factor	FPGA Processing time (ms)	PSNR (dB)
LM-SPIHT	14604	0.001605166848	51.45
In[13]	7129	0.01534629	41.53
In [7]	31.4	20.75	43

6. CONCLUSION

This paper focuses mainly on compressing images using wavelet transform along with the proposed LM-SPIHT; Adding another level of compression using lossless Runlength encoding. Wavelet transform along with LM-SPIHT technique enhance the quality of reconstructed image by mean of PSNR, MSE and so on. Haar wavelet transform makes itself a standard technique for its high efficiency. SPIHT provides variety of important characteristics like good image quality, high PSNR etc., along with HWT technique makes the proposed work a very efficient design. The purpose of this paper is to improve image compression algorithm that is fast, efficient and low memory compatible for real-time implementations. The design was successfully implemented, tested and validated on Xilinx netFPGA-CML-IG Kintex-7 utilizing 1,797 logic cells and achieved a maximum frequency of 260.3MHz. The hardware implementation of the proposed algorithm speeded up the processing of Lena image 14604 times faster than that of the results obtained from MATLAB simulations. Thus, proven the FPGA realization of the design is highly promising for real-time and memory limited mobile communication.

ACKNOWLEDGEMENTS

The authors would like to thank Al-Nahrain University, college of Information Engineering for providing the necessary hardware equipment required for hardware implementation.

REFERENCES

- [1] Farzad Khalvati, "Computational Redundancy in Image Processing". Ph.D. thesis in Electrical and Computer Engineering, Canada, 2008.
- [2] T.Vijayakumar, S.Ramachandran, "FPGA Implementation of 2D-DWT and SPIHT Architecture for Lossless Medical Image Compression", *International Journal of Scientific & Engineering Research*, Volume 4, Issue 8, August-2013
- [3] Jörg Ritter, G'orschwin Fey, Paul Molitor, "SPIHT implemented in a XC4000 device", the 45th Midwest Symposium on Circuits and Systems, *IEEE*, vol 1, page I-239, 2002.
- [4] Ch.Samson, VUK Sastry, "An Improved Run Length Encoding Scheme for Image Compression", *International Journal Of Engineering and Computer Science*, Volume 6 Issue 3, Page No. 20716-20720, March 2017.
- [5] A Hazarathaiah, and B Prabhakara Rao, "Medical Image Compression using Lifting based New Wavelet Transforms", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 4, No. 5, October 2014, pp. 741~750 ISSN: 2088-8708
- [6] Thomas W. Fry, and Scott Hauck, "SPIHT Image Compression on FPGAs", *IEEE Transactions on Circuits and Systems for Video Technology*, Volume: 15, Issue: 9, pp. 1138 – 1147, Sept. 2005.
- [7] Thomas W. Fry, and Scott Hauck, "SPIHT Image Compression on FPGAs", *IEEE Transactions on Circuits and Systems for Video Technology*, Volume: 15, Issue: 9, pp. 1138 – 1147, Sept. 2005.
- [8] Renjini L, and Jyothi R L, "Wavelet Based Image Analysis: A Comprehensive Survey", *International Journal of Computer Trends and Technology (IJCTT)*, Volume 21, No. 3, Mar 2015.
- [9] Ismahane Benyahia, et.al., "Evaluation of the Medical Image Compression using Wavelet Packet Transform and SPIHT Coding", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol.8, No.4, August 2018 , pp. 2139~2147, ISSN: 2088-8708, DOI: 10.11591/ijece.v8i4.pp2139-2147.
- [10] Dipalee Gupta, and Siddhartha Choubey, "Discrete Wavelet Transform for Image Processing", *International Journal of Emerging Technology and Advanced Engineering*, Volume 4, Issue 3, March 2015.
- [11] Tobias Blaser, Stephan Senn, et. al, "Wavelet-based Compression using the SPIHT Algorithm", Semester Thesis, Chip Design Project, 1st April 2006.

- [12] M. B. I. Reaz, M. Akter, et.al, "Design and Implementation of A Modified SPIHT Algorithm For Image Compression", *IEEE Second International Workshop on Computer Science and Engineering*, 28-30 Oct. 2009.
- [13] Li Y., Song J., Wu C., Liu K., Lei J., Wang K., "FPGA Design of Listless SPIHT for Onboard Image Compression" In: Huang B. (eds), "Satellite Data Compression". Springer, New York, NY, 2012.
- [14] Yashaswini P R1, and Mr. Ravi Kiran, "Design and Implementation of Strip based Modified SPIHT algorithm for Compression and Decompression using Verilog", *International Research Journal of Engineering and Technology (IRJET)*, Volume: 03 Issue: 07 , July-2016.
- [15] Varsha Bansal, Pratishtha Gupta, Suhail Tomar, "The Implementation of Run Length Encoding for RGB Image Compression", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* Volume 3 Issue 12, December 2014.

BIOGRAPHIES OF AUTHORS



YasmineMazinTabra received her B.Sc. Degree and M.Sc. Degree in Information Engineering from Al-Nahrain University, Baghdad, Iraq in 2005 and 2009. Since 2009, she has been working as a Lecturer in Information and Communication Department/Information Engineering College/Al-Nahrain University. Currently, she is studying Ph.D. in Information and Communication Engineering. Her fields of interest are Mobile technology, Massive MIMO, Applications of Signal Processing in Direction Finding and Image Processing, and Multimedia Communication.



Dr. Bayan Mahdi Sabbar received his B.Sc. in Electrical Engineering, in 1980, M.Sc. in Digital Communications Systems, in 1983, and Ph.D in (Electrical Engineering)/high resolution array Signal Processing, in 1987. Currently, he is a head of Information and Communication Department, Al-Nahrain University, Baghdad, Iraq. His Fields of Interest are Adaptive systems, High resolution algorithms and their applications, Applications of Signal Processing in Direction Finding and Image Processing, and Identifications Systems based on Signal Processing.