ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

FPGA Implementation of Image Compression Using SPIHT Algorithm

Mr.Vipin V[1], Miranda Mathews[2],

Assistant professor, Department of ECE, St. Joseph's College of Engineering & Technology, Palai, Kerala, India [1]

M.tech Student, Department of ECE, St. Joseph's College of Engineering & Technology, Palai, Kerala, India [2]

ABSTRACT: A VLSI architecture designed to perform real time image compression using SPIHT with arithmetic coder is described here. The main advantage of SPIHT is that it is fully progressive and provide higher PSNR. SPIHT without list is proposed in this paper and which uses breadth first search method. BFS method is suitable for VLSI implementation. In order to archive more performance a high speed arithmetic coder architecture designed with SPIHT. Matlab simulation is used for verifying the result.

Keywords: SPIHT, Arithmetic coder, PSNR, Breadth First Search

I. INTRODUCTION

In recent years, the demand of multimedia products have been growing fast, butconsume large amount of storage space. Therefore the theory of compression becomes more significant in information theory for reducing transmission bandwidth and memory. Compression is the process of encoding data using fewer bits. Image compression is one of the application of data compression. It reduces the amount of data required to represent an image. The objective of image compression is to reduce the redundancy of image and to store or transmit the image in efficient form. There are 3 types of redundancy, 1. Coding redundancy 2.Spatial redundancy, 3.Irrelevant information. In an image, neighboring pixels are correlated and contain redundant information. This type of redundancy is known as special redundancy. Pixel intensity can be predicted from neighboring pixels. This type of redundancy can be reduced by coding the difference between the neighboring pixels.

In this paper we are implementing an image compression technique in FPGA. The combination of DWT and SPIHT algorithm is used for image compression. Set Partitioning in Hierarchical Trees(SPIHT) is a wavelet based image compression method that offers good image quality, fast coding, and high PSNR. It is used for lossless image compression. For achieving higher performance, a high speedarchitecture of arithmetic coder is designed.

II. SPIHT IMAGE COMPRESSION

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

A. *SPIHT(Set Partitioning in Hierarchical Trees)*

SPIHT is a wavelet based image compression algorithm, proposed by Pearlman and Said in 1996. It offers variety of good characteristics

Good Image Quality

- High PSNR
- Fast coding and Decoding
- Used in lossless image compression
- A fully progressive bit stream

In SPIHT algorithm, the image first converted to wavelet coefficients . After that, SPIHT divides the wavelet transform into spatial orientation trees. It is shown in figure 1. Each node represent individual pixels. Every pixel is part of 2X2 block with its adjacent pixels. There are two types of SPIHT architecture.

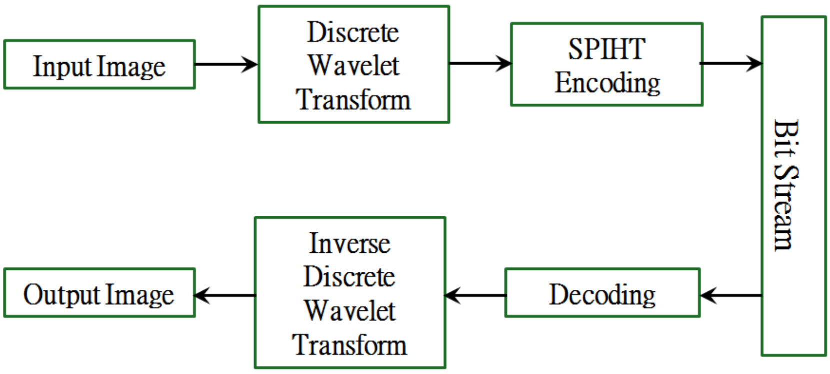

Input Image
Discrete Wavelet Transform
SPIHT Encoding
Bit Stream
Decoding
Inverse Discrete Wavelet Transform
Output Image

Fig. 1. SPIHT image compression block diagram

1) *SPIHT with list*
2) *SPIHT without list*

SPIHT with list needs dynamic operation and it is difficult for high speed application.SPIHT without list is used in real time application. This architecture uses fixed order processing through pixel nodes. It offers fast operation. In this paper we implement the SPIHT without list architecture with breadth first search order.

1) SPIHT with list: It uses 3 different lists to store the significant information about the wavelet coefficients. Three lists are1. List of insignificant set(LIS) 2. List of insignificant pixels(LIP) 3. List of Significant Pixels(LSP). LIP and LSP represent individual pixels and LIS represent a group of pixels. Initially LIS contain all the wavelet coefficients defined by spatial orientation tree. By travelling through each tree node, LIS are portioned and tested for significant state.The following function is used for testing the significant state.

$$S_n(\tau) = \begin{cases} 1, \max_{(i,j)\in\tau}\{|C_{i,j}|\} \geq 2^n \\ 0, otherwise \end{cases} \qquad (1)$$

The threshold,τ , for the first bit-plane is equal to 2_n, and

$$n = \log_2(\max(i,j)|C(i,j)|) \qquad (2)$$

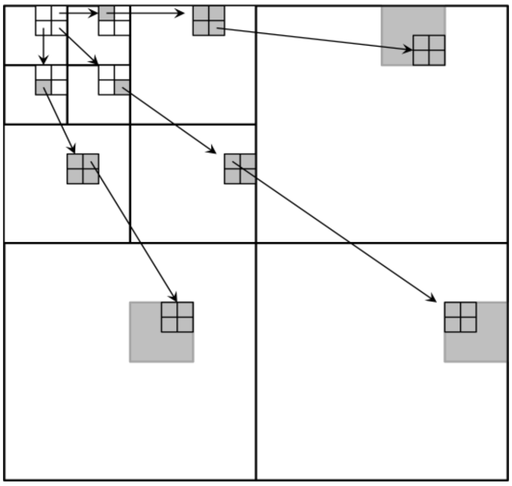
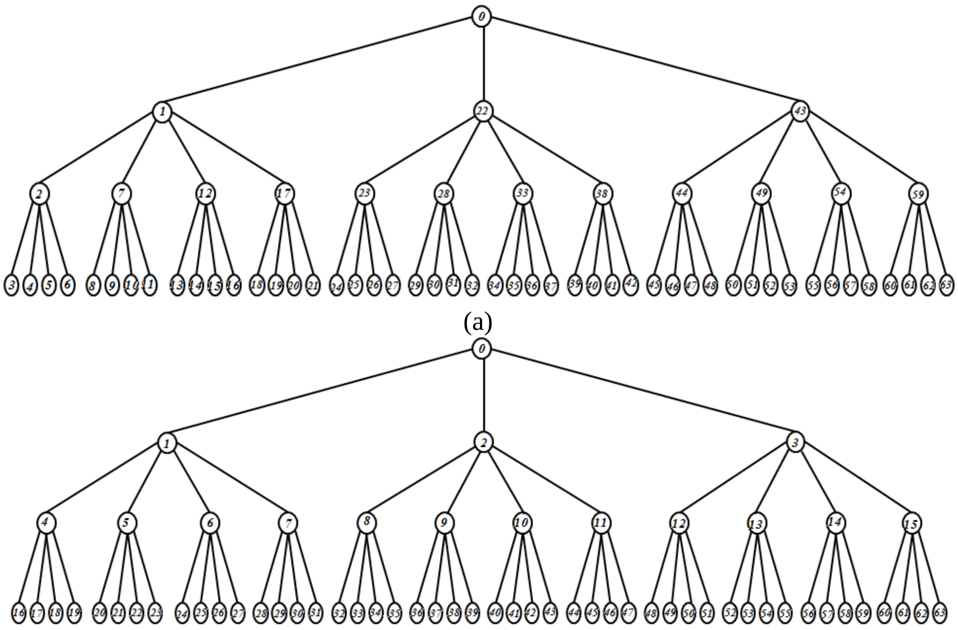


Fig 2. Spatial orientation trees

$C_{i,j}$ is the coefficient value for(i,j) in wavelet domain. If the magnitude of the nodes in the set are less than the predefined threshold, then the set is insignificant. If it is greater than the threshold then it moves to list of significant pixels.The insignificant sets remain in the LIS; the significant sets are partitioned into subsets, which are processed in the same manner and at the same resolution until each significant subset has exactly one coefficient. Finally, each pixel in the LSP is refined with one bit. The above mentioned procedure is then repeated for the subsequent resolution.

2)SPIHT with Breadth First Search :

Coding order of a coefficient tree based on SPIHT with list having different order for different image. It is a set partitioning strategy for making SPIHT efficient. But for hardware implementation we need a fixed path, that is an approximate order. So we define an order for processing tree coefficient for speedup processing time and reduce memory size. There are two searching order. One way is Depth First Search(DFS) and other one is Breadth First Search(BFS) as shown in figure 3.



(a)

(b)

Fig. 3. a. The travel order of Depth First Search, b. The travel order of Breadth First Search

From earlier results SPIHT with BFS is more efficient than SPIHT with DFS. The PSNR(Peak Signal to noise Ratio) performance of BFS is slightly higher than the DFS.BFS can code more significant information than the DFS. Also the order of BFS is closer to that of SPIHT with lists. From these characteristics BFS is the good way for hardware implementation. There are two kinds of symbols involved in coding part. The first kind is significance map symbols(MAP) for sign bit and position bit of significant coefficients. The other is successive approximation quantization symbols (SAQ) for magnitude bit of significant coefficients[].The output order of SPIHT is MAP first and then the SAQsymbols. Pixel’s significant state at the ***p***th bit- plane is defined by the following formula:

$$\begin{cases} Sig^{p}(i,j) = Sig^{p+1}(i,j) + Mag^{p}(i,j) \\ Sig^{msb}(i,j) = 0 \end{cases} \qquad (3)$$

$Sig_p(i,j)$ is the significant state of pixel at(i,j) in ***p***th bit plane, *$Mag_p(i,j)$* is the corresponding magnitude. From this formula we can obtain the relationship between significance and magnitude of each coefficient. The significant state of coefficient at *p_{th}* bit plane is obtained from logic OR operation.

III. System Architecture And Implementation

Figure 4 illustrates the simple architecture of SPIHT with BFS method. The input to the system is raw image pixels and are provided into transform module first. i.e., Discrete Wavelet Transform(DWT).After that wavelet coefficients are arranged in a tree structure through Tree Generation Module, which reads each node from wavelet memory by the BFS way. The state information also generated by the same module.

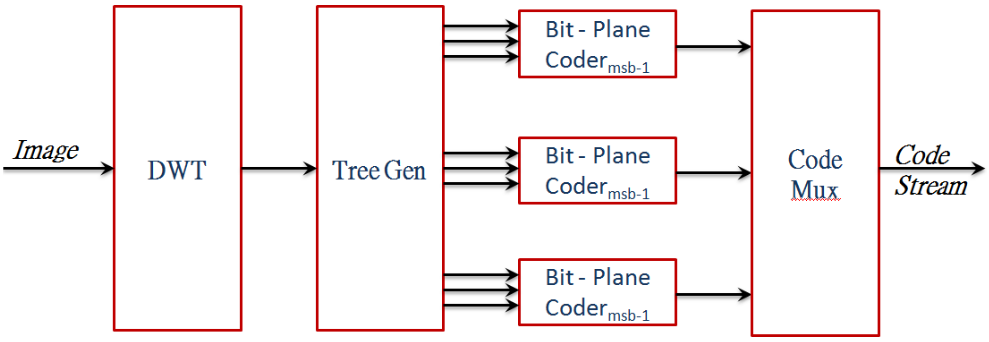


Fig. 4.Architecture of SPIHT BFS

A. *Arithmetic Coder*

Arithmetic coding is a form of entropy encoding. It is used in lossless compression. It differs from other entropy encoding, such as Huffman encoding. Because in Huffman coding each symbol coded., but in arithmetic coding the entire message coded into a single number, a fraction of n(0.0 ≤ n ≤ 1.0).

Let L be the set of symbols, Every symbol i ∈ L, and Probability

$$P_i \in [0,1], if \sum_{i \in L} P_i = 1 \qquad (4)$$

Codeword for symbol sequence $X_M = \{X_1,X_2, ... X_M\}$ and there are bits of number.

$$\sigma(X^M) = q(X^M) + \frac{p(X^M)}{2} \qquad (5)$$

Where $p(X^M)$ is the probability of sequence X^M and $q(X^M)$ is the cumulative frequency of X^M. Arithmetic coder is designed based on three v-size registers: low, high and range. Cumulative count of symbol i is represented by *cum_freq[i]*,
i.e.,

$$cum\_freq[i] = \sum\nolimits_{j \le i} P_j \qquad (6)$$

The interval for the symbol i is [cum_freq[i-1], cum_freq[i]].
If the current probability interval is [low,high), then the update can be done by the following formula:

$$\begin{cases} range = high - low + 1 \\ high = low + range \times \dfrac{cum\_freq[i-1]}{cum\_freq[0]} \\ low = low + range \times \dfrac{cum\_freq[i]}{cum\_freq[0]} \end{cases} \qquad (7)$$

The precision registers grows with M increments. The normalization procedure explained below.

1) If high< HALF : “0” bit written into output . where HALF= *0.5 X 2*^{v}
2) If low ≥ HALF : “1” bit written into output
3) otherwise, output is not defined. In this condition *bit_to_follows* counter is increased. Then if condition 1 is satisfied then a “0” followed by *bit_to_follows* ones written into output. If condition 2 satisfied then a “1” bit and *bit_to_follows*

zeros are written into output. After that to avoid underflowing the registers low and high are scaled. Arithmetic Coder has shorter length coding. If the input symbol's probability is high, then shrink of coding interval will be slow. On other words , if there are some rare symbols then the speed of shrink will be fast.

Detailed architecture of SPIHT encoding with arithmetic coder shown in figure. The original image fed to line based lifting wavelet engine. The wavelet coefficients from line based lifting wavelet engine are written into wavelet coefficient buffers. The processor dispatcher receives these coefficients from wavelet coefficient buffer in breadth first search way. These coefficients allocates to one of the arithmetic coders from eight processors array through internal bus. The parallel architecture provide a wide precision and compression ratio range. The output from arithmetic coder sent to the corresponding code FIFO by code FIFo dispatcher through the internal bus. The Read FIFO and Truncate module generate the final code, which read each code FIFO from top to bottom and truncate the code according to the bit rate requirement. The *power management part* reduce the power consumption in the system. It will stop the clock input for unused bit- planes of each arithmetic coder based on maximal bit-plane register file. The configuration and control part responsible for the settings of parameters like image resolution, wavetype, decomposition and target bit rate.

The Arithmetic coder consist of mainly three parts, *Tree Con* , *bit plane context FIFOs array* and the *coding core*. The Tree construction part generate spatial orientation tree and it visit the wavelet coefficient by the breadth first search order. The bit plane context array consists of twelve FIFOs, which store the context values of twelve bit plane. By using this high speed arithmetic coder we can improve the PSNR about 0.5dB.

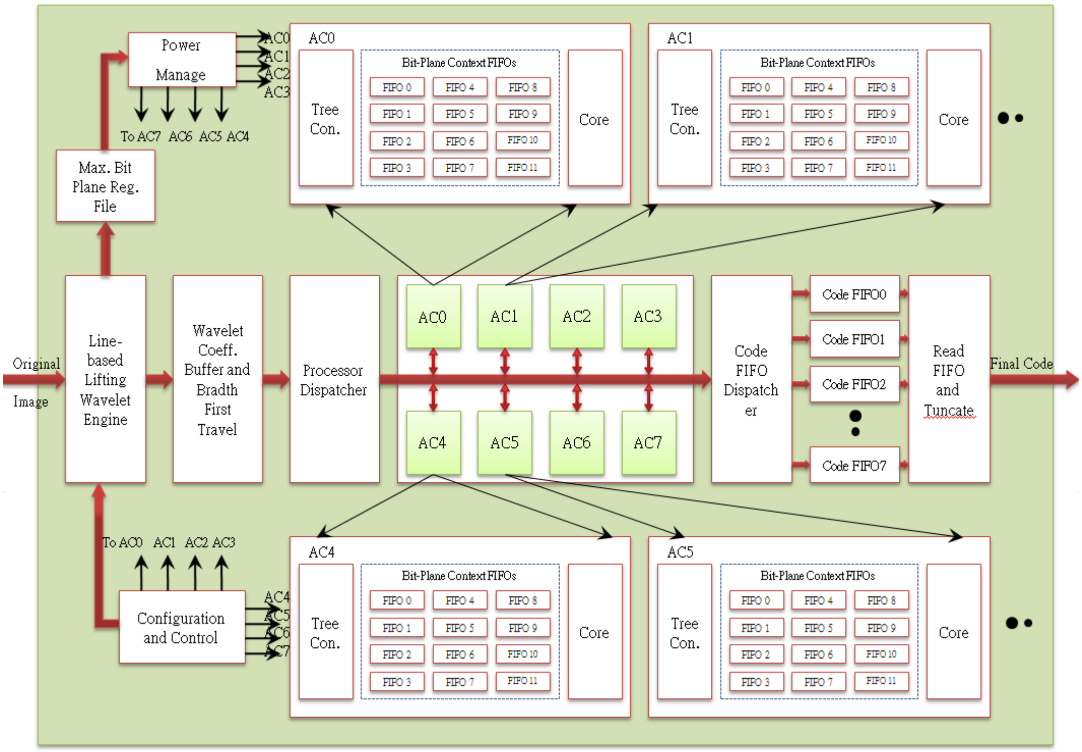


Fig. 5. Architecture of SPIHT encoding.

B. FPGA Implementation

FPGAs are an array of logic gates, which can be programmed to perform variety of tasks. FPGA offers a low cost , high speed and flexible solution over ASICs. A single FPGA can be used for many tasks, it is fabricated in higher volume, which reduce fabrication costs. Also we can reprogram, and which allows design modification and bug fixes without the need to construct a new hardware system. Reprogramming takes only few milliseconds.

Spartan 3E FPGA is used in our project. The main advantages are high speed connectivity, high performance DSP solutions and low cost. The software used for FPGA implementation is Xilinx ISE 10.1. There are softcore microprocessor(MicroBlaze) and hard core embedded microprocessors (PowerPC). MicroBlaze is a virtual microprocessor that includes the following properties.

- Thirty- two 32-bit general purpose registers
- 32- bit address line
- Single issue pipeline
- Separate 32-bit instruction and data bus.

IV. RESULT

SPIHT image compression is simulated in matlab. A 512 x 512 image is considered as the input image. After applying one level wavelet transformation, the input image is divided into four frequency sub-bands. Wavelet transformed coefficients are compressed by applying SPIHT algorithm, we got a result PSNR = 56.5583 and MSE = 0.1436.Matlab R2010a is used for simulation.

TABLE I
MATLAB RESULT

DWT Time	0.021 seconds
Encoding Time	2.696 seconds
Decoding Time	1.100 seconds
PSNR	56.4647
MSE	0.1166

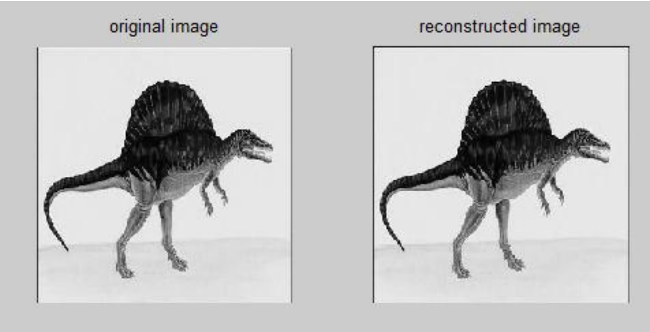


Fig 7. Original and reconstructed image in Matlab

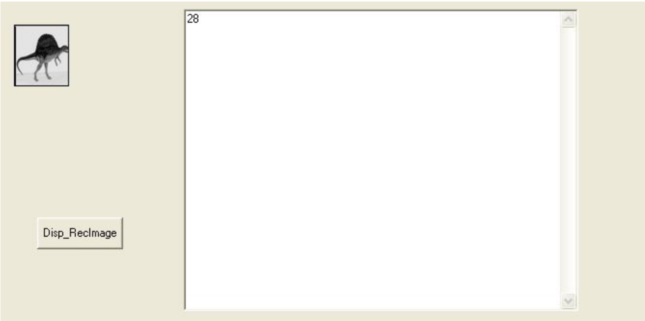

28

Fig.8. Implementation in FPGA.Input images

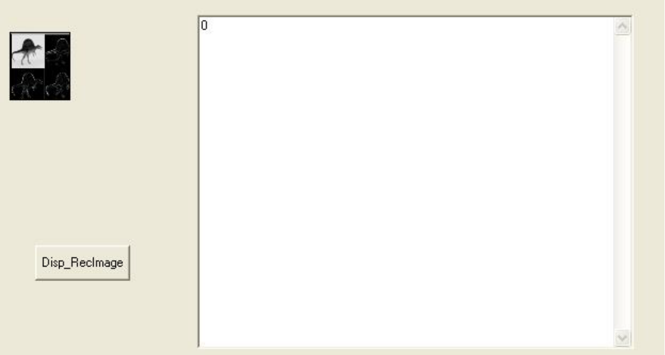


Fig. 9.Compressed image

For FPGA implementationa8 bits/pixel and size is 64 x 64 image is used. Results are shown in below figure8 and figure 9.From the device utilization summery, utilization of 4 input LUTs is 77%. This is shown in Table II.

TABLE III
DEVICE UTILIZATION SUMMARY

Elements in FPGA Amount	**Amount Utilized**	**Amount Available**	**% Utilization**
Number of Slices	1850	1920	96%
Number of Slice Flip Flop	2315	3840	60%
Number of 4 input LUTs	2961	3840	77.10%
Number of GCLK	3	8	37.5%
Number of BRAMs	4	12	33%
Number of MULTI 18x18s	3	12	25%
Number of DCMS	1	4	25%

V. CONCLUSION

An efficient VLSI implementation of image compression is presented here. SPIHT with arithmetic coder improves the peak signal to noise ratio. Another highlight of this system is SPIHT without list. The breadth first search is a fixed order method, which reduce the time requirement. Arithmetic coding makes itself a standard technique for its high efficiency. SPIHT with arithmetic coding architecture provide a high throughput memory efficient design.

REFERENCES

[1] Kai Liu, Evgeniy Belyaev, and Jie Guo, "VLSI Architecture of Arithmetic Coder Used in SPIHT", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 20, Issue 4 ,pp. 697-710 April 2012.

[2] Kai Liu & Jie Lei & YunSong Li,"A Novel VLSI Architecture of SPIHT Using Breadth First Search for Real-Time Applications", Journal of Signal Processing Systems (2012) 68:113-8 125 DOI 10.1007/s11265-011-0581-2

[3] W.-B. Huang, A. W. Y. Su, and Y.-H. Kuo,"VLSI implementation of a modified efficient SPIHT encoder", IEICE Transaaction Fundamentals, E89-A, volume 12, pp. 3613-3622, December 2006

[4] A. Said and W. A. Pearlman, "A new ,fast and efficient image codec based on set partitioning in hierarchical trees", IEEE Transaction on Circuits Systems for Video Technology, volume 6, no. 3, pp. 243-249, March 1996

[5] Y. Wiseman,"A pipeline chip for quasi arithmetic coding", IEICE Transaction Fundamentals, volume E84-A, issue 4, pages 1034-1041, April 2001.

[6] T. W. Fry and S. A. Hauck,"SPIHT image compression on FPGAs", IEEE Transaction Circuits Systems for Video Technology, volume 15, no. 9, pp. 1138-1147, September2005.