*2009 International Symposium on Computing, Communication, and Control (ISCCC 2009)*
*Proc .of CSIT vol.1 (2011) © (2011) IACSIT Press, Singapore*

# Efficient Address Generation Sub System for SPIHT based Memory-Efficient Image Compression Scheme

Anil V. Nandi [1], R.M.Banakar [2] and L.M.Patnaik [3]

[1] E&C department, B.V.B.College of Engg. & Tech., Vidyanagr, HUBLI-580 031, Karnataka, India.

[2] Professor, E&C department, B.V.B.College of Engg. & Tech., Vidyanagr, HUBLI-580 031, Karnataka, India.

[3] Vice Chancellor, Defense Institute of Advanced Technology, Girinagar, Pune, India.

**Abstract.** We present a hardware model of efficient address generation sub system for SPIHT in our unified model of image compression optimized for memory and throughput. The unified model utilizes Spatial Prediction (SP) and Set Partitioning In Hierarchical Trees (SPIHT). A single re-usable list is used in the proposed modified SPIHT algorithm, instead of three continuously growing linked lists existing in basic SPIHT algorithm. Augmented with exchange of significant and refinement passes, this leads to memory optimization at the algorithmic level while maintaining Peak Signal to Noise Ratio (PSNR) performance at lower bit-rates. The modified SPIHT algorithm yields reduction of memory requirements of the order of 65.16% on an average as compared to standard SPIHT. Effective 1D scheme for coefficient addressing is used in SPIHT modeling. The speed of accessing the coefficients is increased by 66.67% and throughput is increased by 61.11% using 1D addressing scheme as compared to 2D addressing scheme.

**Keywords:** Compression, Spatial prediction, SPIHT, Address Generation, PSNR.

## 1. Introduction

We have developed the transform domain compression method followed by tree coding approach to obtain the progressive transmission [1]. Discrete Wavelet Transform (DWT) [2] a popular transform method used in JPEG2000 [3] has the overheads like multiple routing passes, maintaining the tables, computationally complex blocks and large memory requirements. To reduce the computational complexity, the Spatial Prediction (SP) method is used to get the output similar to DWT [4]. The transformed output is further coded using the modified Set Partitioning In Hierarchical Tree (SPIHT) [5] coder to get the good PSNR performance depending on the available bandwidth [1].

The image processing hardware system consists of processor, memory and sub systems like instruction decoder, data / address bus and address generators for the data path across processor and memory. We have developed a hardware model of modified SPIHT processor system to obtain the progressive transmission of coefficients of a matrix obtained by spatial prediction unit. [6] [7]. In our unified model we have used two address generators, one for fetching the pixel elements from the source – Spatial Prediction Sub system (for processing by spatial prediction unit) and the other for SPIHT unit - SPIHT Sub system, for further processing [8] [9]. These hardware sub systems play a very important role in the overall performance of the hardware system. The overall architecture of image compression is developed based on the study of different existing architectures [10]. In this paper we have dealt with hardware model of SPIHT address generator sub system - SPIHT Sub system and its performance in terms of speed and throughput. Set Partitioning In Hierarchical Trees (SPIHT) is a very efficient coding method for compressing images decomposed by Discrete Wavelet Transform (DWT) like operations. As compared to Embedded Block Coding with Optimized Truncation (EBCOT) used in JPEG 2000, SPIHT has much simpler and straight forward coding

procedure and does not require lookup tables. However, SPIHT provides less compression efficiency. These features make SPIHT an appropriate algorithm for its implementation in hardware. The standard SPIHT uses 2D memory addressing scheme for scanning DWT coefficients. We have modified the addressing scheme to get 1D addresses for coefficients. We have designed and implemented both 1D and 2D address generator schemes and compared their performances [11]. Fixed memory data lists are used instead of dynamic allocation as in case of original SPIHT. This causes distortion to some extent but has the advantage of very high throughput and easier hardware implementation. Both 1D and 2D addressing based SPIHT are implemented and their performances are measured in terms of throughput, gate count, area and power consumption.

## 1.1. Address generation unit for SPIHT

Hardware implementation of SPIHT has the following difficulties namely addressing and frequent transactions among coefficient lists. The addressing method of the wavelet coefficients is of 2D type. Since the search for the descendants of a given coefficient has to be performed very frequently, it is necessary to design a dedicated circuit to compute the addresses, which consumes many clock cycles. Also, the transactions among three lists, List of Insignificant Set (LIS), List of Insignificant Pixel (LIP) and List of Significant Pixel (LSP) are required frequently. Linked lists can be effectively used in computing machines using software for addition and deletion of the entries in the lists. Implementing linked lists in hardware is a difficult task. To overcome the above difficulties, we have proposed a modified SPIHT suitable for hardware implementation. Using a suitable technique, the 2D coefficients can be arranged in a 1D coefficients' memory array, which makes operation of searching the descendants very simple. Instead of dynamic linking lists, fixed-size arrays are used for three lists. Transactions among the lists do not require storage of coordinates, but only the flags are changed. Checking for the contents of the lists becomes convenient. The throughput will be improved and the coding procedure is simplified. Fixed-size lists lead to decrease in average PSNR by about 0.19 dB.

## 1.2. Original SPIHT and modification

The structure of decomposed coefficients using our spatial prediction method is shown in Figure 1. SPIHT algorithm uses this structure for achieving further compression. In such structure, the elements are called 'wavelet coefficients', when DWT is used. In this structure, 'a' is a root node (does not have any descendents), 'b', 'c', and 'd' are parent nodes which have descendents. The arrows indicate the descendants of a particular element. For example, element 'b' has descendents b1, b2, b3 and b4. In turn, b1, b2, b3 and b4 have descendents [b11, b12, b13, b14], [b21, b22, b23, b24], [b31, b32, b33, b34] and [b41, b42, b43, b44] respectively. Searching for descendants of a specific parent element requires the movement of a pointer to a specific set of locations.

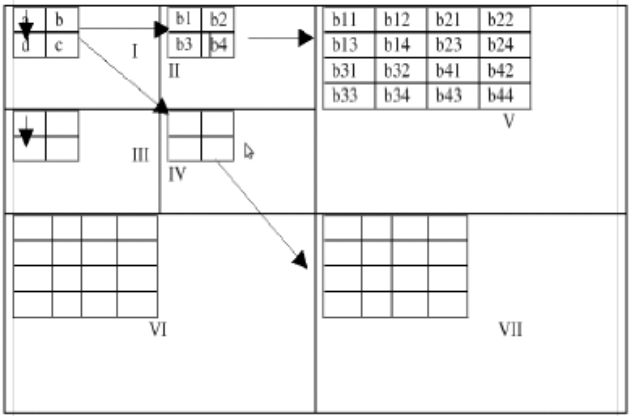


Fig.1. Data structure used in SPIHT

SPIHT uses the coding of elements in the structure in a progressive manner which lie in a particular bit plane. As per the algorithm, the elements will be accessed in terms of sets of (2 x 2) in a progressive manner. So, we need the information about parent-child relationships of elements. We have used a novel method to calculate the addresses of descendant elements of a parent element node. In this method, the search for

descendants of a specific parent node becomes easy, since the addresses of descendents will be stored in consecutive memory addresses as compared to complex addressing present in the original structure. This makes the memory-read operation more efficient and reduces the switching frequency of the address bus. The address generation circuit has a shifter and an increment-by-one operation. Furthermore, the above relation does not change when the image dimension is changed. This address generation scheme is done in order to simplify the hardware design without much loss in coding efficiency. All the definitions in the original SPIHT algorithm are identical except the 2D addresses being replaced by 1D addresses. We have used 8 bit symbols to represent 2D addresses. In original SPIHT, there are three lists to be constructed, called LIS, LIP and LSP. When an element in LIS changes its type, transactions among the lists are necessary. Insertion and deletion operations of the lists are necessary. Dynamic linking lists have to be employed to construct the three lists. It is easy to construct dynamic linking lists with high level programming languages. However, design of a dedicated circuit for this purpose is not easy task. These operations reduce the throughput also. If N is the total number of coefficients, based on the proposed addressing method, addresses of the coefficients which have descendants are from 0 to (N/4)-1. Tables with N/4 entries are used for the lists. They are LIS(i). LIP(i), and LIS(i) for i = 0,..., (N/4)-1. LIS(i)=0 is the case for node-i that has no descendant. Similar definitions are used for LSP(i) and LIP(i) that can be found in the algorithm. Encoding can be done by scanning the tables. Since linking lists are used in the original SPIHT, the nodes that become significant earlier will be placed near the heads of the lists. These nodes are more important than the nodes that become significant later. Therefore, the bits for the earlier nodes will also be output first. The proposed algorithm does not have such scheme. Hence, the original SPIHT tends to have better PSNR performance than the proposed algorithm. The PSNR performance of the proposed algorithm is quite acceptable particularly at lower bit-rates.

## 2. Hardware modelling

The hardware implementation of modified SPIHT is necessary to achieve higher speed, throughput, low power and area. These are very difficult to achieve in software based approach implemented on a general purpose processor. After spatial prediction based decomposition, the coefficients are obtained. The address generator generates the addresses from where the coefficients are picked up and placed in the coefficients memory. The coefficients are now stored in the sequential manner as discussed previously. The addresses generated by the address generator are used by the scanner block to read the coefficients from matrix (Refer Figure 2) and put them into coefficient lists' memory (LIP, LIS and LSP) of SPIHT encoder. These are then compared with the threshold value and the corresponding code is generated from code generator as per the SPIHT algorithm steps. The SPIHT encoder is shown in Figure 3.

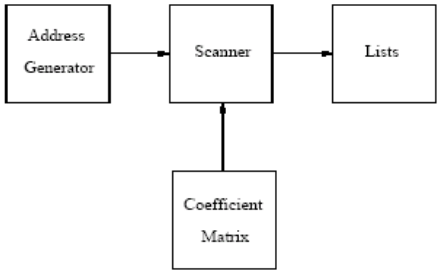


Fig.2. SPIHT memory

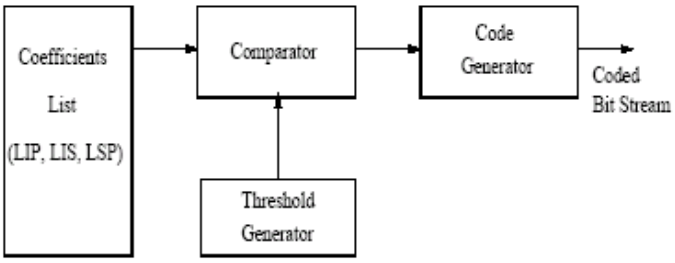


Fig.3. SPIHT encoder

### 2.1. Address Generator and scanner

The hierarchical tree structure obtained after image decomposition has a particular pattern for accessing the coefficients. The relation between the row and the column values of a parent coordinate and its four descendants are as follows (Refer Figure 1). Suppose the parent be 'b' and its row and column address be X[0] and Y[0] respectively. Then the row and column addresses for its descendants b1, b2, b3, b4 are as given below.

b1: X[1]= 2X[0]; Y[1]= 2Y[0] b2: X[2]= X[1]; Y[2]= Y[1]+1
b3: X[3]= X[1]+1; Y[3]= Y[2]-1 b4: X[4]= X[3]; Y[4]= Y[3]+1

With this pattern, the tree grows for each of the nodes and it stops at the last element of the matrix. The address generator has been designed to implement the above mentioned pattern in hardware. In hardware, the above mentioned mathematical formulae are implemented as follows.

Multiplication by two is equivalent to rotating the binary value of the given element by one bit to the left. Addition and subtraction by one is implemented using an incrementor and a decrementor respectively. Whenever a value remains same as in the previous iteration, a register is used to fetch and store it. The four descendants of each parent node are produced in four stages one by one in one cycle. In the next cycle, the row and the column address of the next parent node is fetched and the process is repeated. The values of 2D addresses are initialized as follows.

X[1]= 0 ; Y[1]= 0 X[2]= 0 ; Y[2]= 1 X[3]= 1 ; Y[3]= 0 X[4]= 1 ; Y[4]= 1

The above four values are the first four nodes among which (X[1], Y[1]) i.e. (0, 0) does not have any descendants. So the next node i.e. (0, 1) has to be accessed. At this stage, the index value is 4, since the last produced node (1, 1) i.e. (X[4], Y[4]) has index of 4. To access (0,1) i.e. (X[2], Y[2]), we use the formula,

Index of the parent=(Index of the last produced node/4)+1

In this case, next index = (4/4) + 1 = 2, which is the desired one.

The division by four is performed by using a shift register, which shifts right by two bit positions. So the index is actually a pointer to a pointer, which points to the row, and the column addresses which in turn points to the actual pixel value. The row and column addresses are stored in a memory array one each for row and column address. Once the row and column addresses are produced we need to search and access the actual pixel values. The pixel values are scanned from the matrix and stored in a one dimensional memory array in a row major fashion. For a small (2 x 2) matrix, the arrangement is shown in Figure 4. The elements present in (2 x 2) matrix are scanned as per the addresses produced in our scheme and are placed in memory sequentially. The same pattern applies to matrices of any size. This is performed by a scanner program. Therefore the row and column addresses produced should be converted to their equivalent one-dimensional address. For a (N x N) matrix and row major scanning, 1D address is given by

1-dimensional address = Row address x N + column address.

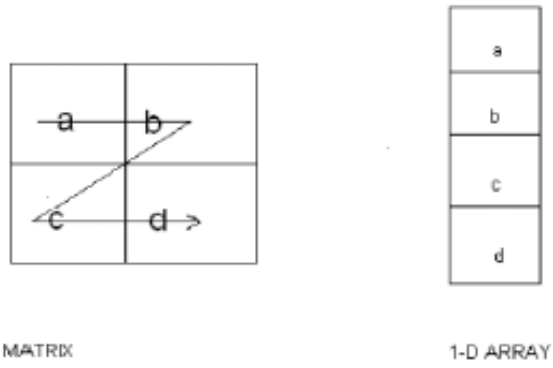


Fig.4. a) Raster scan order b) Conversion to 1D array

It is to be noted here that the addressing starts from 0 and not 1 for both the 2D and 1D arrays. Thus the random access of coefficients is converted into sequential access. Maximum magnitude calculator is a part of address generator. Maximum magnitude coefficient is computed by searching all the coefficients of the image matrix using sequential addressing for the coefficients. This process of finding the maximum value of coefficient slows down the system performance. By using Depth First Search order algorithm [12] performance can be improved. Maximum magnitude calculation phase calculates the maximum value of coefficients and rearranges the following information for SPIHT block. a) Maximum magnitude of four child trees b) Current maximum magnitude. The address generator generates the fixed order addresses and picks the coefficients data from source.

Initialization of Input Data: The SPIHT encoder performs the next operation. It applies the SPIHT algorithm on the pixels and performs the encoding. The input data to this encoder should be categorized into three lists - List of Insignificant Pixels (LIP), List of Insignificant Sets (LIS) and List of Significant Pixels

(LSP). Initially the LSP should contain no pixels. The LIP should contain all the four root nodes and LIS should contain all the descendants in the proper sequence. The corresponding pixels are accessed from the matrix and placed in these lists using the procedure described earlier. Once the lists are initialized they are fed as input to the encoder module.

### 2.2. SPIHT block

We have implemented SPIHT block, which does coding based on data from lists and magnitude calculator. The coefficient blocks are read from higher level to lower level. The fixed-point numerical representation is used for each decomposition level. When the block receives data, it generates the code corresponding to information in each block that contributes to each bit-plane. The SPIHT block generates bit stream, which is sent to the FIFO for each bit-plane. The data received in FIFO vary in size depending on the coefficient values in each bit-plane. Maximum size of FIFO is kept at 400 bits after verification by software program [13] for a class of (256 x 256) pixel images. The FIFO arranges the block of data into regular size of 16 bit words for memory access.

### 2.3. Platform for Implementation

Hardware is modeled using VHDL under Xilinx 7.1i with modelsim simulator. XST is used to synthesize VHDL code and generate the net list. The design is placed and routed on Vertex-4 LX FPGA reconfigurable system with XC4VLX15 device consisting of 13,824 logic cells, Block RAM of 864 bits and 1536 CLBs.

## 3. Results and discussions

The test image used for address generation scheme is 'Barbara' (Grey, 256 x 256 pixels). The hardware model is synthesized using Vertex-4 (XC4VLX15) model FPGA. After synthesis, the FPGA resources are tabulated as shown in Table I. The maximum clock frequency and throughput corresponding to 1D and 2D addressing schemes are also tabulated. The 1D memory scheme can operate at faster clock frequency, about 1.6667 times more as compared to 2D scheme and the throughput is increased by about 1.6111 times as compared to 2D scheme. The FPGA resource utilization remains almost same in both the schemes.

| Memory | Area in % (FPGA) | | | Max.Clock Frequency | Throughput |
|---|---|---|---|---|---|
| | FF | LUT | RAM | | |
| **2D Memory Scheme** | **6** | **7** | **5** | **27 MHz (930 Clock cycles)** | **18 MPixels/Sec** |
| **1D Memory Scheme** | **6** | **7** | **4** | **45 MHz (783 Clock cycles)** | **29 MPixels/Sec** |

Table I Results for 2D (Original SPIHT) and 1D (modified SPIHT) Memory accesses

## 4. Conclusions

In this work, we have presented an efficient hardware for image codec based on modified SPIHT. Efficient addressing method for accessing error matrix elements (coefficients) is used in the implementation. The hardware can be realized at low cost. The distortion measure will be the bottleneck for such technique at lower code rates, which will be addressed in system architecture. The hardware design is verified over Xilinx Vertex device. The clock frequency (speed) and throughput are increased to a greater extent with 1D addressing scheme as compared to that of 2D addressing scheme. The speed is increased by 66.67% and throughput is increased by 61.11%.

## 5. References

[1] Anil V. Nandi, R.M.Banakar, and L.M.Patnaik, "Memory-Efficient Spatial Prediction based Image Compression Scheme," Image and Vision Computing, Elsevier, Vol.25, No.6, June 2007, pp. 899-906.

[2] V. R. Algazi, and R. R. Estes, ”Analysis-based coding of image transform and subband coefficients,” *Proceedings SPIE Applications of Digital Image Processing XVIII,* Vol.2564, August 1995, pp. 11-21.

[3] D.Taubman, ”High performance scalable image compression with EBCOT,” *IEEE Transactions on Image processing,* Vol.9, July 2000, pp. 1158-1170.

[4] Chin-Hwa Kuo, Tzu-Chuan Chou, and Tay-Shen Wang, ”An efficient spatial prediction-based image compression scheme,” *IEEE Transactions on Circuits and Systems for Video Technology* Vol.12, No.10, Oct. 2002,pp.850-856.

[5] A.Said, and W.A.Pearlman, ”A new, fast, and efficient image codec based on set partitioning in Hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.6, No.3, June 1996, pp. 243-250.

[6] Anil V. Nandi, R.M.Banakar, ”Parallel implementation of SPIHT algorithm for compression of images”, P*roc. of Int.Conf. Industrial & Information Systems ICIIS2007,*Univ.of Peradeniya, Sri Lanka, Aug 8-11,2007, pp.329-333.

[7] Anil V. Nandi, R.M.Banakar, ”Throughput efficient Parallel implementation of SPIHT Algorithm”, *21*st *International Conference on VLSI Design and Embedded Systems,* Hyderabad, India, January 4-8, 2008, pp. 482-486. 1063-9667/08 $25.00, 2008 IEEE, DOI 10.1109/VLSI 2008.48, IEEE Computer Society.

[8] Anil V. Nandi, R.M.Banakar, ”High-Speed, High-Throughput Pipelined and Parallel Architecture for SPIHT Algorithm,”*Proc.12th IEEE VLSI Design & Test Symp. VDAT 2008,* Bangalore,India,July 23-26,2008, pp.434-438.

[9] Anil V. Nandi, R.M.Banakar, ”Efficient Architecture for SPIHT Algorithm for Compression of Images”, *Transaction on Electronics and Communications, International Symposium on Electronics and Telecommunications ISETC 2008,* Timisoara, Romania, pp. 150-154. Tom 53(67), Fascicola 1-2, 2008.

[10] Anil V. Nandi, R.M.Banakar, ”VLSI Architectures for Image Processing”, *Asia pacific Conf. on Parallel and Distributed Computing Technologies,* Vellore Institute of Technology, Vellore, India, Dec 13-15,2004,pp.483-488.

[11] Anil V. Nandi, R.M.Banakar, ”Efficient Memory sorting SPIHT Image Coder/Decoder,” *3*rd *International Conference on Computers and Devices for Communication CODEC-06,* Institute of Radio Physics and Electronics, University of Calcutta, India, December 18-20, 2006, pp. 166-169.

[12] T.Cormen, C.Leiserson and R.Rivest, ”Introduction to Algorithms,” 2/e, MIT Press, Cambridge, Massachusetts, September 2001. ISBN: 0262032937.

[13] Priyanka G, Shalini Neeli, Trupti R, Kavya P, Anil V. Nandi, and R.M.Banakar, ”ImageCodec using Wavelet/Set Partitioning In Hierarchical Tree approach,” *Proceedings of National Conference on Signal Processing and Communications-2006 NCSPC 2006,* at JNNCE, Shimoga, Karnataka, INDIA, pp. 81-89, 2006.