International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:4, 2009

A Novel VLSI Architecture of Hybrid Image Compression Model based on Reversible Blockade Transform

C. Hemasundara Rao, *Student Member, IEEE,* M. Madhavi Latha, *Member, IEEE*

Abstract—Image compression can improve the performance of the digital systems by reducing time and cost in image storage and transmission without significant reduction of the image quality. Furthermore, the discrete cosine transform has emerged as the new state-of-the art standard for image compression. In this paper, a hybrid image compression technique based on reversible blockade transform coding is proposed. The technique, implemented over regions of interest (ROIs), is based on selection of the coefficients that belong to different transforms, depending on the coefficients is proposed. This method allows: (1) codification of multiple kernals at various degrees of interest, (2) arbitrary shaped spectrum,and (3) flexible adjustment of the compression quality of the image and the background. No standard modification for JPEG2000 decoder was required. The method was applied over different types of images. Results show a better performance for the selected regions, when image coding methods were employed for the whole set of images. We believe that this method is an excellent tool for future image compression research, mainly on images where image coding can be of interest, such as the medical imaging modalities and several multimedia applications. Finally VLSI implementation of proposed method is shown. It is also shown that the kernal of Hartley and Cosine transform gives the better performance than any other model.

Keywords—VLSI, Discrete Cosine Transform, JPEG, Hartley transform, Radon Transform

I. Introduction

COMPRESSION of an image data is important from the point of view of storage and transmission. To improve the efficiency of the image transmission through various networks and reduce the cost of the storage devices by increasing their capacity, image compression has become one of the most resolved techniques nowadays [1]. Image compression has two approaches. For the lossless compression, there is no partial reduction on image while performing the compression. The exact copy of the original image can be completely recovered. For the lossy compression, some irrelevant data will be thrown away during the compression. The recovered image is only an approximated version of the original image. The lossy compression technique used seems not to degrade the image quality. A non-linear filter for smoothing the resulting image would be suitable for image enhancement. In general, the overall compression ratio is acceptable it compresses to about 55-80% the size of the original image. A lossless compression technique could be performed additionally to increase the compression factor. There is no partial reduction on data while performing the compression. The exact copy of the original image can be completely recovered. For the lossy compression, some irrelevant data will be thrown away during the compression. The recovered image is only an approximated version of the original image [2].

This paper discussess the state of art technique hybrid blockade transform which can be implemented on text, image, and medical data. Section II gives overview of image compression. Section III discussess basics of image compression algorithms and hardware structures are explained in Section IV. Hybrid blockade transform is explained in section V and simulation resutls are shown in Section VI while synthesis results are shown in section VII. Finally conclusions are made in section VII.

C Hemasundara Rao is with Department of Electronics and Communications Engineering, Guru Nanak Engineering College, Hyderabad, Andhra Pradesh, India, e-mail: c_hemasundara@yahoo.co.in

M.Madhavi Latha is with Department of Electronics and Communications Engineering, Jawaharlal Nehru Technological University (JNTU), Hyderabad, India.

II. Overview of Image Compression

The general model of a still image compression framework can be described using a block diagram shown in Figure 1. Statistical analysis of a typical image indicates that there is a strong correlation among the neighbouring pixels. This causes redundancy of information in the image. The redundancy can be greatly removed by decorrelating the image with some sort of preprocessing in order to achieve compression. In general, still image compression techniques rely on two fundamental redundancy reduction principles-spatial redundancy reduction and statistical redundancy reduction. Spatial redundancy is the similarity of neighbouring pixels in an image and it is reduced by applying decorrelation techniques such as predictive coding, transform coding, subband coding, etc. The statistical redundancy reduction is popularly known as entropy encoding. The entropy encoding further reduces the redundancy in the decorrelated data by using variable-length coding techniques such as Huffman Coding, Arithmetic Coding, etc. These entropy encoding techniques allocate the bits in the codewords in such a manner that the more probably appearing symbols are represented with a smaller number of bits compared to the less probably appearing pixels, which aids in achieving compression. The decorrelation or preprocessing block in Figure 1 is the step for reducing the spatial redundancy of the image pixels due to strong correlation among the neighbouring pixels. In lossless coding mode, this decorrelated image is directly processed by the entropy encoder to encode the decorrelated

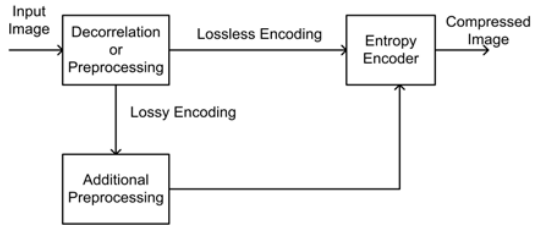

Fig. 1. Basic Image Compression Model

pixels using a variable-length coding technique. In the case of the lossy compression mode, the decorrelated image is subject to further preprocessing in order to mask or throw away irrelevant information depending on the nature of application of the image and its reconstructed quality requirements. This process of masking is popularly called quantization process. The decorrelated and quantized image pixels then go through the entropy encoding process to compactly represent them using variable-length codes to produce the compressed image.

III. Image Compression Algorithms

Image compression is the application of data compression on digital images. Infact, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. The best image quality at a given bit-rate (or compression rate) is the main goal of image compression. There are two forms of compression techniques used are lossless and lossy compression techniques.

A. Lossless Compression Techniques

Lossless compression [3] is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Run length encoding is a very simple method for compression of sequential data. It takes advantage of the fact that, in many data streams, consecutive single tokens are often identical. Run length encoding checks the stream for this fact and inserts a special token each time a chain of more than two equal input tokens are found. This special input advises the decoder to insert the following token n times into his output stream. Basic types of Run Length Encoding are discussed below.

1) Huffman Encoding: This algorithm [4], developed by D.A. Huffman, is based on the fact that in an input stream certain tokens occur more often than others. Based on this knowledge, the algorithm builds up a weighted binary tree according to their rate of occurrence. Each element of this tree is assigned a new code word, where at the length of the code word is determined by its position in the tree. Therefore, the token, which is most frequent and becomes the root of the tree, is assigned the shortest code. Each less common element is assigned a longer code word. The least frequent element is assigned a code word, which may have become twice as long as the input token. The compression ratio achieved by Huffman encoding uncorrelated data becomes something like 1:2. On slightly correlated data, as on images, the compression rate may become much higher, the absolute maximum being defined by the size of a single input token and the size of the shortest possible output token (max. compression = token size[bits]/2[bits]). While standard palletized images with a limit of 256 colors may be compressed by 1:4 if they use only one color, more typical images give results in the range of 1:1.2 to 1:2.5.

2) Entropy coding (Lempel/Ziv): The typical implementation of an entropy coder follows J. Ziv/A. Lempel's approach [5]. Nowadays, there is a wide range of so called modified Lempel/Ziv codings. These algorithms all have a common way of working. The coder and the decoder both build up an equivalent dictionary of meta-symbols, each of which represents a whole sequence of input tokens. If a sequence is repeated after a symbol was found for it, then only the symbol becomes part of the coded data and the sequence of tokens referenced by the symbol becomes part of the decoded data later. As the dictionary is build up based on the data, it is not necessary to put it into the coded data, as it is with the tables in a Huffman coder. This method becomes very efficient even on virtually random data. The average compression on text and program data is about 1:2; the ratio on image data comes up to 1:8 on the average Graphic Interchange Format (GIF) image. Here again, a high level of input noise degrades the efficiency significantly. Entropy coders are a little tricky to implement, as there are usually a few tables, all growing while the algorithm runs. LZ coding is subject to patents owned by IBM and Unisys (formerly known as Sperry).

3) Area Coding: Area coding is an enhanced form of run length coding, reflecting the two dimensional character of images. This is a significant advance over the other lossless methods. For coding an image it does not make too much sense to interpret it as a sequential stream, as it is in fact an array of sequences, building up a two dimensional object. Therefore, as the two dimensions are independent and of same importance, it is obvious that a coding scheme aware of this has some advantages. The algorithms for area coding try to find rectangular regions with the same characteristics. These regions are coded in a descriptive form as an element with two points and a certain structure. The whole input image has to be described in this form to allow lossless decoding afterwards. The possible performance of this coding method is limited mostly by the very high complexity of the task of finding largest areas with the same characteristics. Practical implementations use recursive algorithms for reducing the whole area to equal sized sub-rectangles until a rectangle does fulfill the criteria defined as having the same characteristic for every pixel. This type of coding can be highly effective but it bears the problem of a nonlinear method, which cannot be implemented in hardware. Therefore, the performance in terms of compression time is not competitive, although the compression ratio is.

B. Lossy Compression Techniques

Lossy compression techniques are one where compressing data and then decompressing it retrieves data that may well be different from the original, but is close enough to be useful in some way. Lossy compression is most commonly used to compress still images, especially in applications such as streaming media and internet telephony. Some of the lossy compression techniques are discussed below.

1) Vector Quantization: A vector quantizer [6] can be defined mathematically as a transform operator T from a K-dimensional Euclidean space R^K to a finite subset X in R^K made up of N vectors. This subset X becomes the vector codebook, or, more generally, the codebook. Clearly, the choice of the set of vectors is of major importance. The level of distortion due to the transformation T is generally computed as the most significant error (MSE) between the 'real' vector x in R^K and the corresponding vector x' = $T(x)$ in X. This error should be such as to minimize the Euclidean distance d. An optimum scalar quantizer was proposed by Lloyd and Max. Later on, Linde, Buzo and Gray resumed and generalized this method, extending it to the case of a vector quantizer. The algorithm that they proposed is derived from the KNN clusterisation method, and is performed by iterating the following basic operations:

Subdivide the training set into N groups (called 'partitions' or 'Voronoi regions'), which are associated with the N codebook letters, according to a minimum distance criterion; The centroids of the Verona regions become the updated codebook vectors; Compute the average distortion: if the percent reduction in the distortion (as compared with the previous step) is below a certain threshold, then STOP.

Once the codebook has been designed, the coding process simply consists in the application of the T operator to the vectors of the original image. In practice, each group of n pixels will be coded as an address in the vector codebook, that is, as a number from 1 to N. The LBG algorithm for the design of a vector codebook always reaches a local minimum for the distortion function, but often this solution is not the optimal one. A careful analysis of the LBG algorithm's behavior allows one to detect two critical points: the choice of the starting codebook and the uniformity of the Verona regions' dimensions. For this reason some algorithms have been designed that give better performances. With respect to the initialization of the LBG algorithm, for instance, one can observe that a random choice of the starting codebook requires a large number of iterations before reaching an acceptable amount of distortion. Moreover, if the starting point leads to a local minimum solution, the relative stopping criterion prevents further optimization steps.

IV. From Algorithms to VLSI Architectures

The envisaged mass application of the discussed video communication services calls for coding equipment of low manufacturing cost and small size. Manufacturing cost are dominated by the number of integrated circuits (chips), chip packaging and silicon area per chip. The number of chips can be kept small by large area silicon on the other hand. Because of the defect density, production of very large area chips is not economic. Thus, high complexity systems require in general several chips for implementation.

The cost for chip packaging is related to the number of pads. For this reason the system partitioning into chips has to consider the interconnects. By application of advanced semiconductor technologies high complexity systems can be implemented with a moderate number of chips. It is the goal to achieve a VLSI implementation with the smallest silicon area for a specified source rate. The source rate of digital video signals has a large range according to the kind of application[7]. It is obvious that the hardware structures for low source rates such as the Quarter Common Intermediate Format(QCIF) format with 9 Mb/s are different from those for very high source rates such as HDTV with 1.9 Gb/s. The required silicon area for VLSI implementation of algorithms is related to the required resources such as logic gates, memory, and the interconnect among the modules. The amount of logic depends on the concurrency of operations. A figure of merit for the required concurrency can be derived by, the number of concurrency of operations

$$N_{con,op} = R_S.N_{op,pel}.T_{op} \quad (1)$$

with R_S, as source rate in pixels per time unit, $N_{op,pel}$ number of operations per pixel and T_op as average time for performing an operation. The number of operations per pixel is an average value derived by counting all operations required for performing the coding scheme. The video coding schemes are periodically defined over a basic interval. In case of hybrid coding schemes as applied in MPEG this interval is a macro block. Almost all tasks of the coding scheme are defined on a macro block of 16 $\times$ 16 luminance pixels and the associated chrominance pixels. For this reason counting have to be performed over one macro block. The number of operations depends on the specific algorithms. For example block matching based on exhaustive search requires much more operations than search strategies. The number of operations of a complete coding scheme (encoder and decoder) is in the order of 200 when applying block matching with 2D log search and a fast DCT algorithm. With at present available technologies, T_{op} is in the order of 20 ns. From this follows that the number of concurrent operations ranges from 5 to 1000, depending on the video format.

The required interconnects between the operative part and the memory depends highly on the access rate. Considering one large external memory for storing video data and intermediate results the number of parallel bus lines for connecting the operative part and the memory becomes approximately

$$N_{bus} = R_S.N_{acc,pel}.T_{acc} \quad (2)$$

where $N_{acc,pel}$ specifies the mean number of accesses per pel and T_{acc} the memory access time. Because of 1- and 2-operand operations $N_{acc,pel}$ larger than $N_{op,pel}$. For simplicity let us assume that $N_{acc,pel}$. T_{acc} is in the same order than $N_{op,pel}$ T_{op}. From this follows as a figure of merit the determined concurrency. Thus the number of bus lines becomes very large. Taking into consideration that the access rate is mainly

influenced by multiple accesses of image source data and intermediate results, the access rate to an external memory can be essentially reduced by assigning a local memory to the operative part. The size of the local memory depends on the specific access structure of the coding algorithm. Because of the periodicity over the macro block the local memory is in the order of a macro block size.

A. Function Specific Architecture

Considering available technologies and the required computational requirements for video encoders and decoders, the use of dedicated or function specific implementations, is often mandatory. For high volume consumer products, the silicon area optimization brought by dedicated architectures, compared to programmable architectures, leads to lower production cost. In this section, we will develop two examples of function specific architectures: the DCT computation and the motion estimation. In each case, we will describe the algorithms, review different architectures and compare their performances in terms of speed and silicon area. Finally, a panorama of previously reported function specific chip-sets will be established.

1) Discrete Cosine Transform (DCT): DCT helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the Discrete Fourier Transform (DFT): it transforms a signal or image from the spatial domain to the frequency domain. With an input image, A, the coefficients for the output image B, are as shown in Figure 2:

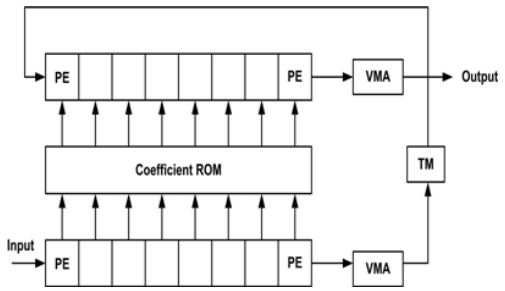

Fig. 2. Separated DCT implementation

The input image is $N2$ pixels wide by $N1$ pixels high; $A(i, j)$ is the intensity of the pixel in row 'i' and column 'j'; $B(k1, k2)$ is the DCT coefficient in row $k1$ and column $k2$ of the DCT matrix. All DCT multiplications are real. This lowers the number of required multiplications, as compared to the DFT. The DCT input is an 8 $\times$ 8 array of integers. This is called as process element (PE) as shown in Figure 2. These PEs are given to Vector Merging Adder (VMA). The PEs which are obtained from coefficient ROM, again processed by VMA and later is given to Transposition Memory (TM) from where the final output is taken. This array contains each pixel's gray scale level; 8 bit pixels have levels from 0 to 255. The output array of DCT coefficients contains integers; these can range from -1024 to 1023. For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT.

2) Block Matching: Two-dimensional suffix tries seem quite restrictive because they only provide information about square matches. However, there are many problems in deriving a data structure that can represent rectangles of arbitrary shape. There is no unique shape identifying a position in the array. Moreover, a particular shape may be incapable of uniquely specifying a position if it is not capable of intersecting both the right and bottom boundaries of the array.

In addition to the parameter N, which specifies the number of characters in the N = n $\times$ n input array (all of our results hold both for square and non-square input arrays, but we assume the square case for simplicity), we let m denote the maximum horizontal or vertical dimension of a match (so the largest possible match has M = m $\times$ m characters). For most practical applications m may be taken to be a relatively small constant, but for theoretical purposes we take m to be a parameter that could be anywhere in the range 1 to n. The idea is to employ a data structure that is a factor of m larger, but still allows fast matching. For a fixed constant k, the construction of Giancarlo and Grossi [8] can be adapted to produce a (k, *) suffix trie that represents all rectangular sub-arrays of height k, visited in column-major order (similarly, a (., k) suffix trie can be defined for all sub-arrays of width k visited in row-major order). However, for practical values of m, it may be faster to employ the simple strategy (that can be viewed as a generalization of the simple strategy discussed at the start of Fiala and Greene [31]) that runs in O(mN) time.

3) Programmable Architecture: In contrast to function oriented approaches with limited functionality, programmable architectures enable the processing of different tasks under software control. The particular advantage of Programmable architectures is the increased flexibility. Changes of architectural requirements, e.g., due to changes of algorithms or an extension of the aimed application field, can be handled by software changes. Thus a generally cost-intensive redesign of the hardware can be avoided. Moreover, since programmable architectures cover a wider range of applications, they can be used for low-volume applications, where the design of function specific VLSI chips is not an economical solution. On the other hand, programmable architectures require a higher expense for design and manufacturing, since additional hardware for program control is required. Moreover, programmable architectures require software development for the envisaged application. Although several vendors provide development tools, including high level language compilers, the expense for software development may not be neglected and has to be considered for deciding which type of architecture, function oriented or programmable, has to be used for a specific application. Image processing, especially video coding applications; often require a real-time processing of the image data. To achieve this goal, parallelization strategies have to be employed. A variety of programmable and dedicated VLSI architectures for video coding applications have been presented in the past [9-12]. for real-time video compression, are presented. These examples clarify the wide range of architectural alternatives for the VLSI implementation of processors for video coding applications.

V. Reversible Blockade Transform

A 2D reversible blockade transform and its inverse have an implementation as a sequence of lifting steps arranged for reduced computational complexity (i.e., reducing a number of non-trivial operations)[13]. This transform pair has energy compaction properties similar to the discrete cosine transform (DCT), Discrete Hartley Transform (DHT) and is also lossless and scale-free. As compared to a separable DCT transform implemented as 1D DCT transforms applied separably to rows and columns of a 2D data block, as shown in fig.3, the transforms operations are re-arranged into a cascade of elementary transforms, including the 2×2 Hadamard transform, and 2×2 transforms incorporating lifting rotations. These elementary transforms have implementations as a sequence of lifting operations[14]. We propose reversible coding methods that

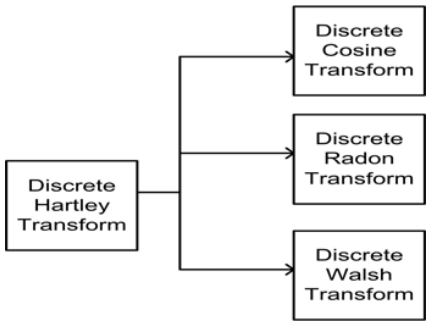

Fig. 3. Reversible Blockade Transform

correspond to the discrete Walsh transform (DWT), Discrete Cosine Transform (DCT), Discrete Radon Transform (DRT), and Discrete Hartley Transform (DHT)[15]. Furthermore, we propose a method that uses the difference of the n-th order, a method of which the number of levels of the transform coefficient is the same as that of the input signal, and a reversible overlap transform coding method. Simulation shows that the compression efficiency of the proposed method is almost the same as that of predictive coding.

VI. Simulation Results

Simulation results have been carried out for Blockade Transform. The idea behind Blockade tranform is to develop another transform from other two transform techniques and it depends on the application. All the experiments are done using Verilog language and Matlab. Table 1 shows the compression ratios of different transforms and it is observed from Table 1 that DHT gives better compression than any other transform which are considered in this work.

The image is first converted into binary format and it is given to blockade transform block. The output of the block is also binary and using MATLAB the image is re-constructed. The noise levels must be considered in hybrid transform and alter must be used to reduce the noise levels. For all Hybrid Transforms, experiments have been conducted and compression ratios have been obtained and tabulated in Table 1. The images which are considered here are benchmarks and taken from [16].

TABLE I
Compression Ratios of Different Transforms

Transform	Compression Ratio (%)
DCT	72
Randon Transform	65
Walsh Transform	69
Hartley	79

We have tested various image formats and data formats like text, image and ECG. It is observed from Table 2 that for smaller images blockade transform (DRT+DHT) gives better performance and for higher size images the blockade transform (DWT+DHT) gives better resutls. We have also tested for medical data like ECG but the blockade transform may not give better resutls.

TABLE II
Hybrid Blockade Transform

Sno	Data Memory	Type of Image	Transform	CR (%)
1	50KB	Text	DCT+DHT	58
2	268KB	Image	DRT+DHT	69
3	841KB	Image	DRT+DHT	79
4	2MB	Image	DWT+DHT	82
5	3MB	Image	DWT+DHT	84
6	20KB	ECG	DCT+DHT	42

VII. Synthesis Results

The Blockade Transform architecture as shown in Figure 3 for implementing 2-dimensional hybrid blockade transform could be synthesized and implemented using the Cadence tools tools for 90 nm technology node. The library cells have been designed to synthesize both single kernel transforms as well as Blockade Hybrid Transform. In the following sections, synthesis of both single kernel and hybrid transforms have been discussed. For each single kernel transform and blockade transform power and area are mentioned and comparisons are shown in Table 3 and 4.

A. Area

Cadence Tool provides a predictable path from RTL to placed gates. It is a unified environment for general logic and high-performance datapath synthesis, DFT analysis and insertion, physical synthesis, power optimization and static timing analysis. The Physical Netlist (placed gates) generated by Cadence Simulator provides a clean handoff between the RTL designer and the layout engineer, eliminating time-consuming layout-to-synthesis iterations for timing closure. Blast Create is a complete solution for logic designers that includes fast, full-featured, high capacity synthesis, testability analysis, power optimization and detailed placement capabilities as well as incremental or full static timing analysis. The combination of gainbased synthesis with Cadence Fixed Timing methodology enables fast implementation and analysis without sacrificing accuracy, and ensures delivery of a high-quality Physical Netlist for predictable timing closure.

TABLE III
AREA OF EACH TRANSFORM

Sno	Transform	Area(mm2)
1	DWT	721
2	DCT	1124
3	DHT	924
4	DRT	636
5	Blockade (DWT+DHT)	1342

B. Power

Cadence Spectre provides a comprehensive RTL-to-GDSII solution for power optimization and management. Optimal power management is enabled throughout the implementation flow with power-aware synthesis, physical optimization and routing, enabling designers to minimize power and ensure uniform power distribution. Cadence Spectre is fully integrated with RTL-to-GDSII implementation flow to enable continuous power, timing and area tradeoffs throughout the design flow. By addressing power very early in the flow, Cadence spectre ensures good Quality of Results (QOR) and minimizes iterations for faster turnaround time. It provides dynamic power savings by using clock gating techniques that also include support for test logic. It enables dynamic power reduction by recognizing special directives in the RTL to insert power gating cells and retention. Using Spectre, experiments are conducted and results are tabulated in Table 4.

TABLE IV
POWER DISSIPATED BY EACH TRANSFORM

Sno	Transform	Power
1	DWT	42.89
2	DCT	92.12
3	DHT	58.44
4	DRT	32.91
5	Blockade (DWT+DHT)	124.12

C. Layouts of Various Transforms

Using standard cell library of Cadence Tools, the Discrete Walsh Transform (DWT) has been synthesized. The layout of DWT is shown in Figure 4. All the transforms have been considered for 8×8 only. As discussed before, the 2-dimensional 8×8 image is divided into 256 sub-matrices. Each matrix is treated as 8-bit row matrix. These bits are given to DST in parallel for all 256 row matrix. The output is taken and verified. For this work, only standard cell is used hence only built-in NAND and NOR gates are taken into account. It is observed from the layout that the DWT has more congestion at Input/Output (I/O) pads than at the center of the chip which reflects that the fact it has more power consumption compared to DCT which is shown in Table. 4.

There are five open source standard cell libraries [17], the vsclib, wsclib, vxlib, vgalib and rgalib. They have been drawn with the Graal software from Alliance, part of an extensive open source software suite for designing integrated circuits with a standard cell design methodology [18]. The libraries have been characterized in a generic 90nm technology. The spice model comes from the University of California,

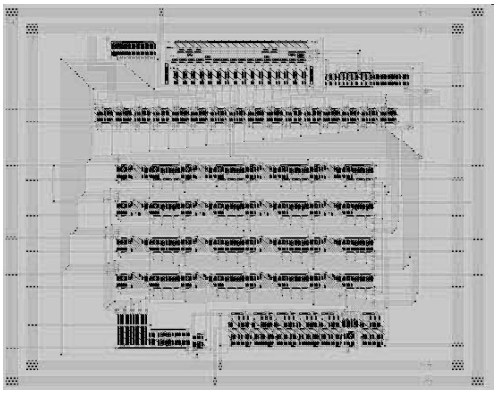

Fig. 4. Layout of Discrete Hartley Transform

Berkeley, and the layout rules are slightly oversized 0.13um ones which should make the layout compatiblewith most foundry rules. These cells are considered for Cadence tools to synthesize DHT+DCT. The floor-planning of combined transform is shown in Figure 5. Considering the standard

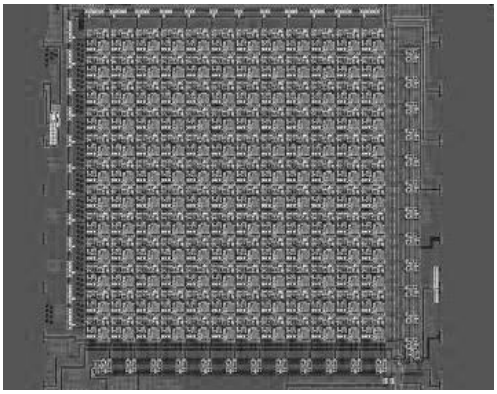

Fig. 5. Layout of Discrete Radon Transform

cells again for synthesizing Blockade transform DHT+DWT, the 2-dimensional 8×8 image is considered and is divided into 256 sub-matrices. Each matrix is treated as 8-bit row matrix. These bits are given to DHT+DWT in parallel for all 256 row matrix as shown in fig 6. The output is taken and verified. It is observed from the layout that the DHT+DWT has more congestion at the middle of the chip as well as at Input/Output (I/O) pads which reflects that the fact it has more power consumption compared to all other blockade transform techniques as well single kernel transforms and this is shown in Table 4.

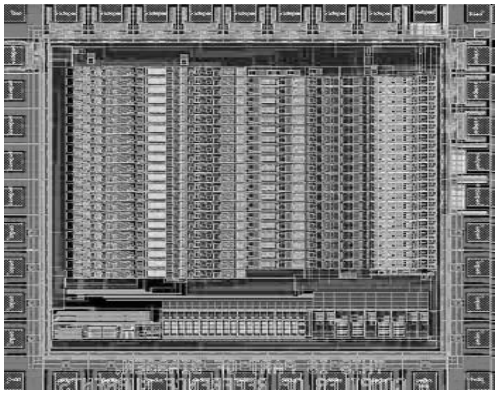

Fig. 6. Layout of Blockade Transform

VIII. Conclusions

A new reversible blockade transform for image compression and data compression is developed. Hartley transform has been considered as basis for all hybrid transforms and transformation coefficients have been generated from Hartley transform kernel. Simulations have been conducted for all hybrid transforms and it is proved that DHT+DCT is better compression ratio when compared to all other transform techniques. Experiments have been conducted by considering various standard library cells, and the layouts of single kernel and hybrid transforms have been drawn. Using synthesis tools power and area of single kernel and hybrid transforms have been noted down and it is proved that DHT+DCT is giving better results compared to other techniques.

References

[1] Abbas Razavi, Rutie Adar, Issac Shenberg, "VLSI Implementation of an Image Compression Algorithm with a New Rate Control Capability" *IEEE International Conference on Multimedia*, Aug 1992, CA, USA, pp. V.669- V.672

[2] Jie Guo; Cheng-ke Wu; Yun-song Li; Ke-yan Wang; Song, J."Memory-efficient architecture including DWT and EC for JPEG2000" *9th International Conference on Solid-State and Integrated-Circuit Technology,* 2008. 20-23 Oct. 2008. pp. 2192-2195.

[3] Pellegri, P.; Novati, G.; Schettini, R., "Multispectral loss-less compression using approximation methods" in Proceedings of IEEE International Conference on Image Processing (ICIP), Volume 2, pp. 638-641, September 2005.

[4] Karp R, "Minimum-redundancy coding for the discrete noiseless channel", in Proceedings of IEEE Transactions on Information Theory, Volume 7, Issue 1, pp. 27-38, January 1961.

[5] Ziv, J., and A. Lempel, "A Universal Algorithm for Sequential Data Compression," in Proeedings of IEEE Transactions on Information Theory", vol. 23, no. 3, pp. 337-343, October, 1977.

[6] Allott, D., Clarke, R.J., "Shape adaptive activity controlled multistage gain shape vector quantisation of images", in Proceedings of IET Electronics Letters, Volume 21, Issue 9, April 25, 1985.

[7] Yibin Yang; Boroczky, L.; "A new enhancement method for digital video applications" in Proceedings of IEEE Transactions on Consumer Electronics, Volume 48, Issue 3, pp. 435-443, August 2002.

[8] Giancarlo, R. and Grossi, R. "On the construction of classes of suffix trees for square matrices: algorithms and applications"In Proceedings of ICALP. 1995.

[9] T. Fukushima, "A survey of image processing LSIs in Japan," IEEE 10th Int. Conf on Patt. Recog., Atlantic City, NJ, pp. 394-401, June 1990.

[10] K. Gaedke, H. Jeschke, and P. Pirsch, "A VLSI-based MIMD architecture of a multiprocessor system for real-time video processing applications," J. VLSI Signal Proc., vol. 5, pp. 159-169, Apr. 1993.

[11] W. Gehrke, R. Hoffer, and P. Pirsch, "A hierarchical multiprocessor architecture based on heterogeneous processors for video coding applications," Proc. ICASSP '94, vol. 2, IEEE Press 1994.

[12] J. Goto et al., "250-MHz BiCMOS super-high-speed video signal processor (S-VSP) ULSI," IEEE J. Solid-state Circ., vol. 26, no. 12, pp. 1876-1884, 1991.

[13] Komatsu, K.; Sezaki, K. "Reversible discrete cosine transform" *International Conference on Acoustics, Speech and Signal Processing, 1998*, Volume 3, 12-15 May 1998 pp.1769 - 1772.

[14] Lei Wang; Jiaji Wu; Licheng Jiao; Li Zhang; Guangming Shi, "Lossy to lossless image compression based on reversible integer DCT" *15th International Conference on Image Processing,* 2008, 12-15 Oct. 2008 pp.1037 - 1040.

[15] Soo-Chang Pei; Jian-Jiun Ding, "Reversible Integer Color Transform" *IEEE Transactions on Image Processing,* Volume 16, Issue 6, June 2007, pp.1686 - 1691.

[16] A. G. Weber, "The USC-SIPI Image Database. Version 5," USCSIPI Rep #315, Oct 1997 http://sipi.usc.edu/service/database/Database.htm.

[17] http://www.vlsitechnology.org/

[18] Ziping Hu; Verma, P.; Sluss, J. "Routing in Degree-Constrained FSO Mesh Networks" *International Conference Future Generation Communication and Networking,* 2008. Volume 1, 13-15 Dec. 2008 pp.208 - 215

C. Hemasundara Rao Mr. C. Hemasundara Rao graduated in B.Tech. from MU in 1992, post graduate in M.Tech. from JNTU in 1999 from JNTU and presently pursuing Ph.D from JNTU.

He has been actively guiding students in the area of VLSI and Image Processing. He has been published more than 4 National / International Conferences. Currently, he has been working as professor in Electronics and Communication Engineering Dept., Gurunank Engineering College, Hyderabad, Andhra Pradesh. India.

Dr. M. Madhavi Latha She is graduated B.Tech from Nagarjuna University in 1986, Post graduation in M.Tech from JNTU in 1993 and PhD from JNTU in 2002.

She has been actively involved in research and guiding students in the area of signal and image processing, VLSI (mixed signal design) and hardware implementation of speech CODECs. She has published more than 25 papers in National/International Coferences and 3 Journal papers. Currently, she has been working as professor in ECE, JNTU college of Engineering, Hyderabad, Andhra Pradesh, India.