

## FPGA Implementation of JPEG2000 Image Compression using Modified DA based DWT and Lifting DWT-IDWT Technique

G.Bheemeswara Rao\*, T. Surya Kavitha\*\*, U. Yedukondalu\*\*\*

(Department of Electronics and Communication Engineering, Aditya Engineering College, AP, India)

### ABSTRACT

Today's electronic equipment comes with user friendly interfaces such as keypads and graphical displays. As images convey more information to a user, it is many of the equipment today have image displays and interfaces. Hence most of the signal processing technologies today has dedicated hardware that act as co-processors to compress and decompress images. In this project, a 2D image compression using modified DA based DWT IDWT is discussed and proposed a new technique called lifting DWT and is implemented on Spartan 3e FPGA EDK. This uses less silicon area compared with previous technique for the compression of 64X64 sized image. The silicon area used in this new approach for the compression of 64X64 image is approximately same as that of the silicon area used by the previous technique for the compression of 32X32 sized image.

**Keywords:** Discrete wavelet Transformation, Distributed Arithmetic, Embedded Development Kit, FPGA, Lifting technique.

### I. INTRODUCTION

JPEG2000 is a compression standard for still images intended to overcome the shortcomings of the existing JPEG standard, makes use of the wavelet and sub-band technologies. It supports the features like lossless and lossy compression, protective image security, Region of interest coding, Robust to bit errors [1]. The JPEG 2000 standard works on image tiles. The source image is partitioned into rectangular non-overlapping blocks in a process called tiling. These tiles are compressed independently as though they were entirely independent images. All operations, including component mixing, wavelet transform, quantization, and entropy coding, are performed independently on each different tile. The nominal tile dimensions are powers of two, except for those on the boundaries of the image. Tiling is done to reduce memory requirements, and since each tile is reconstructed independently, they can be used to decode specific parts of the image, rather than the whole image. Compression is accomplished through the use of the encoder, which is presented in Figure-1

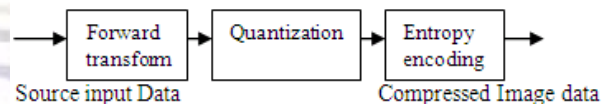


Figure1: Block diagram of JPEG2000.

Before the forward discrete wavelet transform, or DWT, is applied to each tile, all image tiles are DC level shifted by subtracting the same quantity, such as the component depth, from each sample. DC level shifting involves moving the image tile to a desired bit plane, and is also used for region of interest coding, which is explained later. This process is pictured in Figure-2.

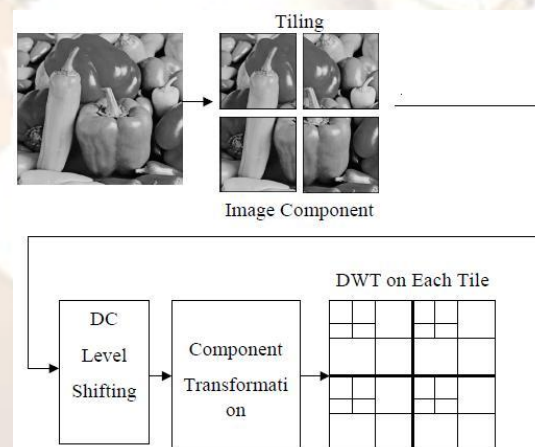
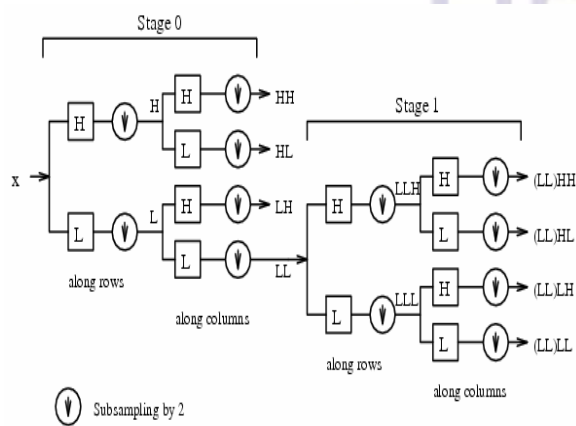


Figure 2: Forward Transformation.

### II. ABOUT DWT ARCHITECTURE

Image consists of pixels that are arranged in two dimensional matrix, each pixel represents the digital equivalent of image intensity. In spatial domain adjacent pixel values are highly correlated and hence redundant. In order to compress images, these redundancies existing among pixels needs to be eliminated. DWT processor transforms the spatial domain pixels into frequency domain information that are represented in multiple sub-bands, representing different time scale and frequency points. Human visual system is very much sensitive to low frequency and hence, the decomposed data available in the lower sub-band region and is selected and transmitted, information in the higher sub-bands regions are rejected depending upon required information content. In order to extract the

low frequency and high frequency sub-bands DWT architecture shown in figure below is used. As shown in the figure, input image consisting rows and columns are transformed using high pass and low pass filters. The filter coefficients are predefined and depend upon the wavelets selected. First stage computes the DWT output along the rows, the second stage computes the DWT along the column achieving first level decomposition. Low frequency sub-bands from the first level decomposition is passed through the second level and third level of filters to obtain multiple level decomposition as shown in Figure 3 below.



The decomposition of the image using 2-level DWT is shown in figure-4 [8].

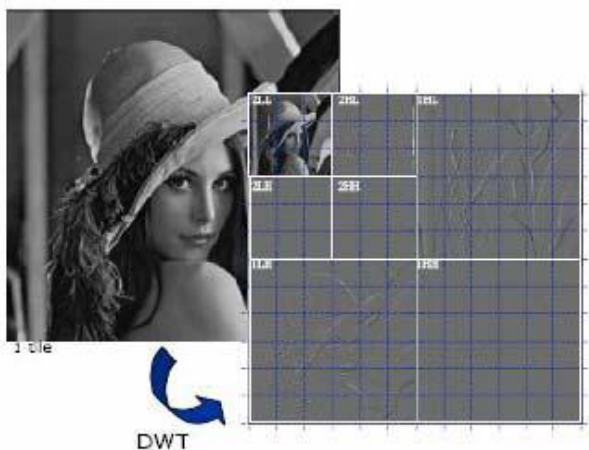


Figure 4: Second level decomposition of an Image.

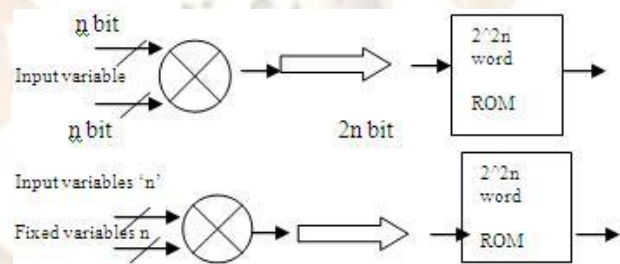
### III. DISTRIBUTIVE ARITHMETIC BASED 2D DWT/IDWT ARCHITECTURE

The computational modules of DWT consist of filters,  $H(t)$  and  $G(t)$  can be defined with the transfer function of these filters as,

$$G(z) = g_0 + g_1 z^{-1} + \dots + g_{(l-1)} z^{-(l-1)}$$

$$H(z) = h_0 + h_1 z^{-1} + \dots + h_{(l-1)} z^{-(l-1)}$$

In this section, we first outline how to perform multiplication by using memory based architecture. Following this, we briefly explain architecture for DWT filter bank. Using this we show complete design for block based DWT. In computing DWT, multipliers are the fundamental computing elements. Since these multipliers consume significant area, the number of multipliers and adders that can be employed on a chip is limited. The memory based approach provides an efficient way to replace multipliers by small ROM tables such that the DWT filter can attain high computing speeds with a small silicon area. Traditionally, multiplication is performed using logic elements such as adders, registers etc. However, multiplication of two n-bit input variables can be performed by a ROM table of size of  $2^{2n}$  entries. Each entry stores the pre-computed result of a multiplication. The speed of the ROM table lookup is faster than that of hardware multiplication if the look-up table is stored in the on chip memory. In general  $2^{2n}$ -word ROM table is too large to be practical. In DWT, one of the input variables in the multiplier can be fixed. Therefore, a multiplier can be realized by  $2^n$  entries of ROM as shown in figure 5.



To access the look-up table, we have the same number of registers as filter coefficients. The input  $x[n]$  will enter into the serial shift register which has to access the look-up table. When the next input comes into first register, the old value will be pushed into the next register. In the same way, when next values come into registers, the old values will go off from the registers. Now, to get the address from the input values, we consider the bit positions and get the values of inputs by that bit position. For example, if we want to get the first address, we have to consider the LSBs of all serial registers. By this address we will get the first position value. In the same manner, we have to get all bit position addresses and get the corresponding values from the look-up table. While adding, we have to shift the values by the bit position value and give them to adder. Finally, we have the result, which is the convolution of the filter coefficients and the inputs. The accessing of look-up table is as shown in the figure 6. The same architecture will be used for the both high-pass and low-pass filters. If the input is 8-bit length, then we require 8 clock cycles to get the convolved value. The same

implementation of LUTs based Distributed Arithmetic is shown in figure 7.

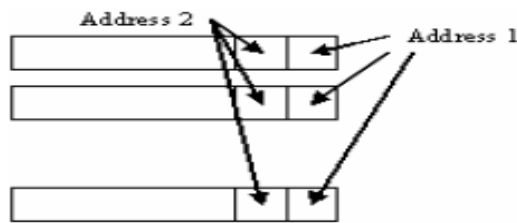


Figure 6: Accessing the ROM Tables.

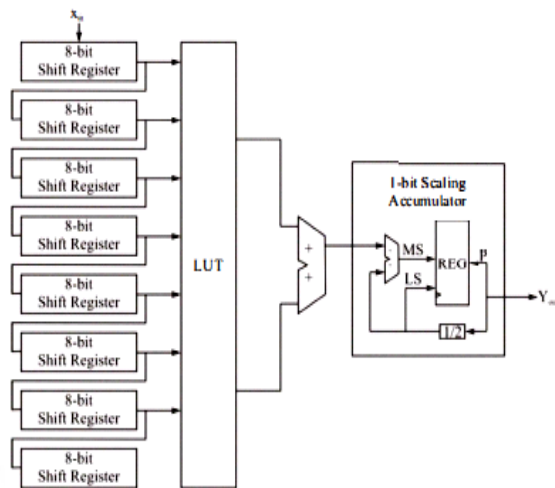


Figure 7: Distributive Arithmetic.

#### IV. MODIFIED DA BASED DWT ARCHITECTURE

To speed up the process we can go for the parallel implementation of the Distributive Arithmetic (DA). The structure is as shown in the Figure.8.

In parallel implementation, we divide the input data into even samples and the odd samples based on their position. Even we can split the filter coefficients into even and odd samples. So, the even samples convolve with the even and odd filter coefficients and at the same time the odd samples also convolve with the same coefficients.

So, by the same time we are getting the result for both even and odd samples of input. In order to further increase the speed and reduce the area, the LUT can be further split into four stages, and can be accessed by the input values for data read.

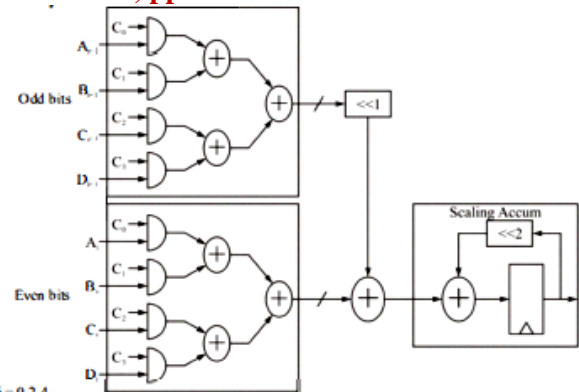


Figure 8: Parallel implementation of DA based DWT.

#### V. PROPOSED LIFTING TECHNIQUE FOR DWT

The lifting scheme has been developed as a flexible tool suitable for constructing the second generation wavelet [6]. It is composed of three basic operation stages: splitting, predicting, and updating. Fig.9, 10 shows the lifting scheme of the wavelet filter computing one dimension signal:

- Split step: where the signal  $x(n)$  is split into even  $x(2n)$  and odd  $x(2n+1)$  points, because the maximum correlation between adjacent pixels can be utilized for the next predict step.
- Predict step: The even samples are multiplied by the predict factor and then the results are added to the odd samples to generate the detailed coefficients( $d_j$ ) ie., high pass coefficients:  

$$HP(2n-1) = x(2n-1) - (x(2n) + x(2n+1))/2$$
- Update step: the detailed coefficients computed by the predict step are multiplied by the update factors and then the results are added to the even samples to get the coarse coefficients( $s_j$ ) ie., low pass coefficients:  

$$LP = x(2n) + (HP(2n-1) + HP(2n+1) + 2)/4$$

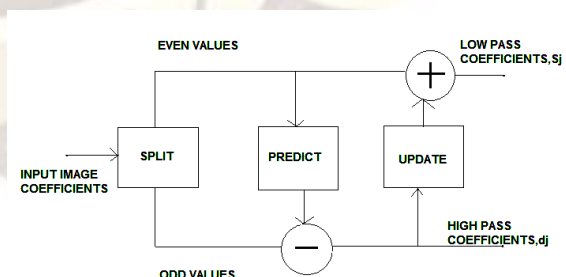


Fig 9: Block diagram of forward lifting scheme

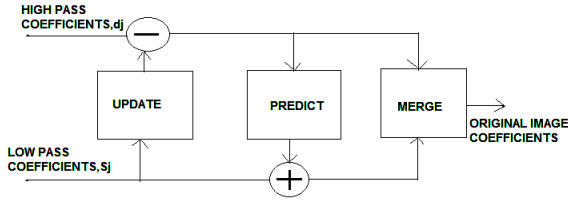


Fig 10: Block diagram of inverse lifting scheme

The proposed VLSI architecture shown in Fig.11 performs 2-D DWT with line-based method [2][6], which consists of five key modules: data choose module, the row DWT module, the column DWT module, DWT control unit and external RAM. An  $N/4$  external RAM is used to store the LL band output coefficients to carry out the multi-level decomposition, where N represents the width and the height of the input image. The DWT control unit controls the time sequence of the whole system.

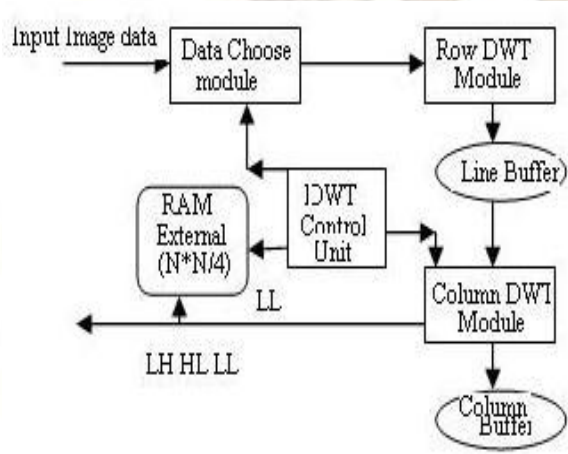
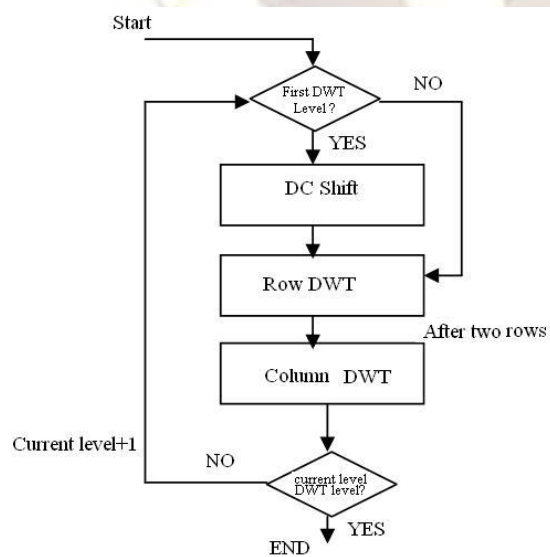


Fig 11: Proposed 2D DWT Architecture.

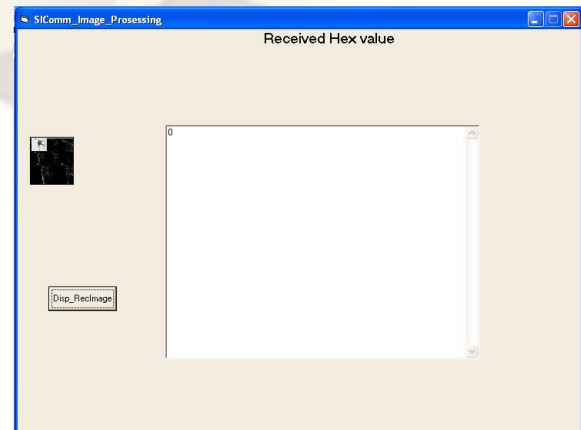
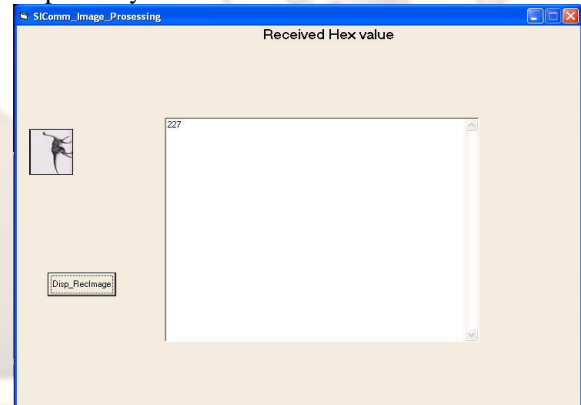


## VI. THE HARD WARE AND SOFTWARE IMPLEMENTATION

The entire design process is performed using Xilinx platform studio8.1, as our design was implemented using EDK (Embedded Development Kit) that helped the hardware designer to easily build, connect and configure embedded processor-based system. It includes Software Development Kit (SDK) for MicroBlaze and PowerPC – Including GNU C/C++ compiler and debugger; Xilinx Microprocessor Debug (XMD) target server; Data2MEM utility for bitstream loading and updating. The hard ware specification of our design was specified in the Microprocessor Hardware Specification file(MHS) using high level system C language. And the design was compiled using the compiled software routines that are available as an Executable and Linkable Format (ELF) file. The ELF file is the binary ones and zeros that are run on the processor hardware. These ones and zeros are transferred to and from the kit using RS 232 and JTAG cables. The image which is to be compressed is converted in to text file using MAT lab and the compressed and reconstructed images were displayed using Visual Basic Comport visual interface for 64X64 predesigned display on PC.

### a. SIMULATION RESULTS

The Original, Transformed and Inverted images are displayed in the consecutive fig 13, 14,15 respectively.



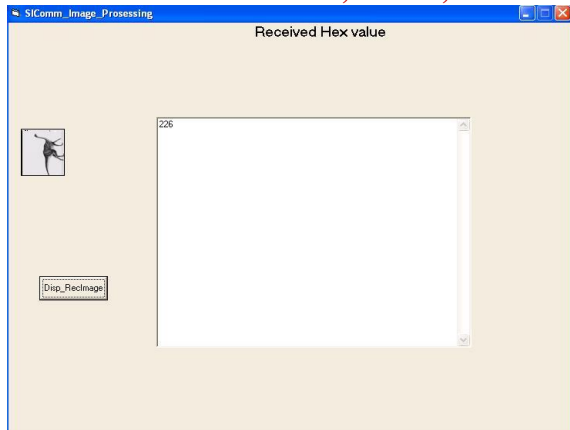


Fig 11, 12, 13: 2 level transformed and inverted images.

### b. SYNTHESIS RESULTS

The synthesis results are given below in fig 14 and the results are very satisfactory which reduces the silicon area used compared to the previous technique[9] used and those comparisons are given in the table 1.

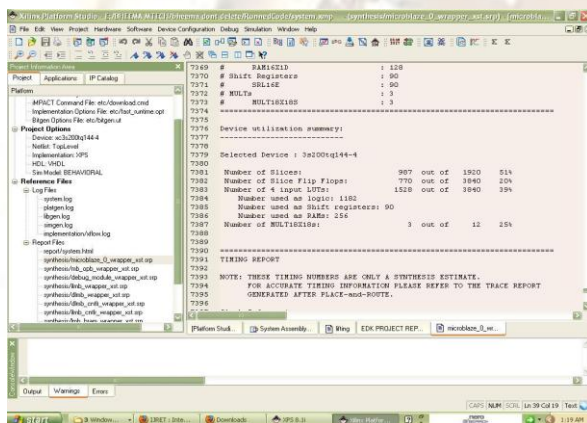


Fig 14: Synthesis result.

Parameters	Existing	Proposed
Device used	Xc2vp30ff896-7	3s200tq144-4
Image size	32x32	64x64
No. of slices Used	832	987
No. of slice flipflops	634	770
No. of 4 input LUTs	1186	1528
No. of MULT 18X18s	2	3

Table 1: Comparison of device utilization.

From the above comparison we can say that even we were used lower end FPGA for the design implementation it uses less silicon area i.e., approximately same silicon area for the compression of image size double the size of the image compressed using DA based Technique.

### VII. CONCLUSION

In this paper we proposed a technique for software-implementation of a lifting with the goal of getting a customizable lifting DWT-core which can be used as a module in implementing a bigger system irrespective of one's choice of implementation platform. Here we have written the core processor Microblaze is designed and implemented using XILINX ISE 8.1 Design suite and Xilinx Platform Studio (XPS), the algorithm is written in system C Language and tested in Embedded Development Kit (EDK) by interfacing a test circuit with the PC using the RS232 cable. The test results are seen to be satisfactory. The area taken and the speed of the algorithm are also evaluated.

### ACKNOWLEDGEMENT

We wish to acknowledge Aditya Engineering College for providing resources to carry out this work successfully.

### REFERENCES

- [1] David S. Taubman, Michael W. Marcellin - JPEG 2000 – Image compression, fundamentals, standards and practice", Kluwer academicpublishers, Second printing - 2002.
- [2] G. Knowles, "VLSI Architecture for the Discrete Wavelet Transform," Electronics Letters, vol.126, pp. 1184-1185,1990.
- [3] M, Vishwanath, R. M. Owens, and M. 1. Irwin, "VLSI Architectures for the Discrete Wavelet Transform," IEEE Trans. Circuits And Systems II, vol. 42, no. 5, pp. 305-316, May. 1995.
- [4] AS. Lewis and G. Knowles, "VLSI Architectures for 2-D Daubechies Wavelet Transform without Multipliers". Electron Letter, vol.27, pp. 171-173, Jan 1991.
- [5] K.K. Parhi and T. Nishitani "VLSI Architecture for Discrete Wavelet Transform", IEEE Trans. VLSI Systems, vol. 1, pp. 191-202, June 1993.
- [6] P.Y Chen, "VLSI implementation for one-dimensional multilevel liftingbased wavelet transform", IEEE Trans. on Computers, vol. 53, pp.386-398, 2004.
- [7] C. Chakrabarti and M. Vishwanath, "Architectures for Wavelet Transforms: A Survey", Journal of VLSI Signal Processing, Kulwer, vol.10, pp. 225-236,1995.
- [8] David S. Tabman and Michael W. Marcelliu, "JPEG 2000 – Image Compression, Fundamentals, Standards an
- [9] FPGA implementation of modified DA based DWT by Cyril nagaraj, Naga Bhushanam.



**\*G.Bheemeswara Rao** received his B.Tech Degree in 2009 from JNTK University,AP,India. He is currently working toward Post Graduation degree in the Department of Electronics and Communication Engineering, Aditya Engineering College,AP,India. His Research Interests are VLSI Implementation of Digital Systems, Image Processing, Communication Systems.



**\*\*T.Surya Kavita** received her post Graduation Degree from JNTUK University,AP,India. Currently she is working as Associate Professor in the Department of Electronics and Communication Engineering in Aditya Engineering College,AP,India with as Experience of 10 Years in teaching .She got a professional membership in IETE, ISTE. Her Research interests are Image and SignalProcessing, Communications and Digital System Design Implementation.



**\*\*\*U. Yedukondalu** received his Post Graduation Degree in the Stream of Electronics and Currently he is working towards the Ph.D in the Department of Electronics and Communication Engineerign . He is the Head of the Electronics Department in Aditya Engineering College,AP,India with an Experience of 13Years in teaching .His Research Interests include Analog and Digital Integrated circuits and VLSI Architecture implementation of the Digital Circuits, Image Processing .

