IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 11, 2017, accepted November 19, 2017, date of publication November 22, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2776293

A Framework for Hardware Efficient Reusable IP Core for Grayscale Image CODEC

ANIRBAN SENGUPTA[1], (Senior Member, IEEE), DIPANJAN ROY[1], (Student Member, IEEE), SARAJU P. MOHANTY[2], (Senior Member, IEEE), AND PETER CORCORAN[3], (Fellow, IEEE)

[1]Department of Computer Science and Engineering, IIT Indore, Indore 453552, India
[2]Department of Computer Science and Engineering, University of North Texas, Denton, TX 76203 USA
[3]Department of Engineering and Informatics, NUI Galway, Galway, Ireland

Corresponding author: Dipanjan Roy (phd1501201007@iiti.ac.in)

This work was supported by the Council of Scientific and Industrial Research under Grant 22/730/17/EMR-II.

ABSTRACT This paper proposes two major novelties. First, we provide a mathematical framework for hardware resource efficient IP core-based image compression and decompression (CODEC). The framework includes CODEC functions that are capable of determining the pixel intensities of a compressed gray scale image using significantly lesser hardware resources. Digital pixel values of the original image are fed as an input to the functions of proposed IP framework and compressed digital pixel values of compressed image generated. Similarly, digital pixel values of the compressed image are fed into other functions of the proposed framework for image decompression. Second, the second novelty is using the derived IP functions to propose designs of reusable IP cores for complete Haar wavelet transformation (HWT)-based lossy image CODEC. Testing of images from various data sets (NASA, medical applications, and so on) in terms of hardware resources, image quality, and compression efficiency have indicated that the proposed IP core framework was successful in achieving hardware efficient CODEC compared with JPEG and conventional HWT CODECs.

INDEX TERMS IP core, CODEC, hardware efficient, pixel intensity.

I. INTRODUCTION

With the advancement of multi-media technologies and digital systems like digital camera, smart phone, scanner, tablets etc. high-resolution images can be captured easily [15], [19], [20]. Due to the better quality, these high-resolution images occupy large storage space, take high transmission time and large bandwidth to upload/download an image [13], [14]. An efficient Intellectual Property (IP) block/ reusable core [11], [12] for image compression and decompression can compute (generate) the compressed image as well as reconstruct it through a single step computation each while preserving the compression efficiency and quality parameters of a captured image. Image compressions are of two types: a) lossless, where no data loss is occurred; b) lossy, where less relevant data are discarded. Cameras in medical imaging [6], satellite imaging, forensic imaging use lossless image compression [7], [8], while camera in smart phones, tables, digicam, scanner etc uses lossy image compression.

In the year 2000 Joint Photographic Experts Group (JPEG) proposed Discrete Wavelet Transformation (DWT) based image compression technique [1]–[3]. Haar Wavelet Transformation (HWT) based image compression is one of the efficient forms of DWT [9] based image compression technique [10]. HWT decomposes each signal into two components, one is called average (approximation) or trend and the other is known as difference (detail) or fluctuation.

Wavelet based image compression for volumetric medical imaging is discussed in [4]. Both lossy and lossless image compression is performed through directional wavelet transforms, block-based intra-band prediction and arbitrary decomposition structures. A new algorithm is proposed in [5] to select a threshold value through statistical analysis. The proposed algorithm is capable to maximize the compression ratio while minimizing the redundancy. Further reduction of image details is also achieved through Huffman encoding. None of these aforementioned approaches propose any functions for dedicated HWT-based image compressing and decompressing hardware or presents the design flow of an IP core for image compression and decompression.

Rest of the paper is organized as follows: Section II highlights the novelties of proposed approach, Section III

TABLE 1. Reduction of designer effort obtained when applying proposed IP based method in HWT hardware image compression.

Proposed IP core based hardware HWT based image CODEC		HWT matrix multiplication hardware based image CODEC		Proposed IP core based hardware HWT based image CODEC		HWT matrix multiplication hardware based image CODEC	
Resources required for IP core 1 based Image compression	Resources required for IP core 2 based Image decompression	Resources required for Image compression	Resources required for Image decompression	Speed of computation of IP core 1 for image compression	Speed of computation of IP core 2 for image decompression	Speed of computation for image compression	Speed of computation for image decompression
4 ASU, 4 Mul	4 ASU, 2 Adder, 2 Mul	512 Mul, 511 Adder	512 Mul, 511 Adder	4 pixels per execution	4 pixels per execution	1 pixel per execution	1 pixel per execution

TABLE 2. Reduction of designer effort obtained when applying proposed IP based method in JPEG hardware image compression

Proposed IP core based hardware HWT based image CODEC		JPEG/DCT matrix multiplication hardware based image CODEC		Proposed IP core based hardware HWT based image CODEC		JPEG/DCT matrix multiplication hardware based image CODEC	
Resources required for IP core 1 based Image compression	Resources required for IP core 2 based Image decompression	Resources required for Image compression	Resources required for Image decompression	Speed of computation of IP core 1 for image compression	Speed of computation of IP core 2 for image decompression	Speed of computation for image compression	Speed of computation for image decompression
4 ASU, 4 Mul	4 ASU, 2 Adder, 2 Mul	8 Mul, 7 Adder	8 Mul, 7 Adder	4 pixels per execution	4 pixels per execution	1 pixel per execution	1 pixel per execution

introduces proposed novel framework for IP block based HWT lossless image compression; Section IV introduces proposed novel IP core based design process for HWT-based image compressor and decompressor. Section V presents the analysis and results while Section VI presents the conclusion.

II. NOVEL CONTRIBUTIONS OF THIS PAPER

a) Proposes multiple functions for IP block based HWT image compression and image reconstruction. The IP functions are capable to directly determine the pixel intensities of a compressed gray scale image and can be used as a 'back box' in image processing tools as library where uncompressed digital pixel values of original image is fed as input to the IP block (representing a set of functions) and compressed digital pixel values of compressed image is generated. Similarly, digital pixel values of compressed image are fed into another IP black box (representing a set of functions) for image reconstruction.

b) Using the derived IP functions for both compression/decompression, to propose the system design of a dedicated reusable soft IP cores for complete HWT based image compression and image reconstruction. Both designs have been successfully tested on Intel Cyclone FPGA. The IP core designs can be used directly as a macro-block (CODEC) in SoCs or standalone ASICs. The reduction of designer effort obtained when applying proposed IP core based compression method compared to normal hardware based HWT and DCT/JPEG based compression is shown in Table I, Table II Further reduction in designer effort when applying proposed IP block based compression compared to normal software based HWT and DCT/JPEG based compression is shown in Fig. 1. The block diagram representation of proposed IP block/core based HWT image compression and decompression is shown in Fig.2.

III. PROPOSED FRAMEWORK FOR IP BASED HWT LOSSY IMAGE COMPRESSION

A. PROBLEM FORMULATION

For a gray scale input image of size N × N design an IP core for HWT-based image compression and decompression.

B. PROCESSING INPUT IMAGE

In the proposed approach, an N × N gray scale digital image with 8-bit depth is considered as input. An N × N matrix is generated by calculating the pixel intensity of each coordinate of the input image. Fig. 3 shows a generic 512 × 512 input image in the form of a matrix (A). The subscript and superscript of each element indicate the row number and column number of the element respectively. For example, the pixel intensity of 3rd row and 510th column of matrix 'A' is represented by element m_3^{510}. In a gray image, the pixel values lay between 0 and 255, where 255 indicates pure white and 0 indicates pure black.

C. BACKGROUND ON HAAR WAVELET TRANSFORMATION

In the process of Haar wavelet based transformation of input data two types of coefficient are generated: a) scaling coefficient and b) wavelet coefficient. Scaling coefficient represents the sum of two consecutive data samples and divided by two while wavelet coefficient represents the difference of two consecutive data samples and divided by two. Thus scaling coefficients represent the high-frequency signals known as coarse details of the data and wavelet coefficients represent the low-frequency signals known as finer details of the data.

2D-Haar wavelet transformation is comprised of: forward transformation of data and inverse transformation of data. Forward transformation is a two-step process i.e. level 1 forward transformation on input data and level 2 forward transformation on level 1 transformed data. Similarly, inverse

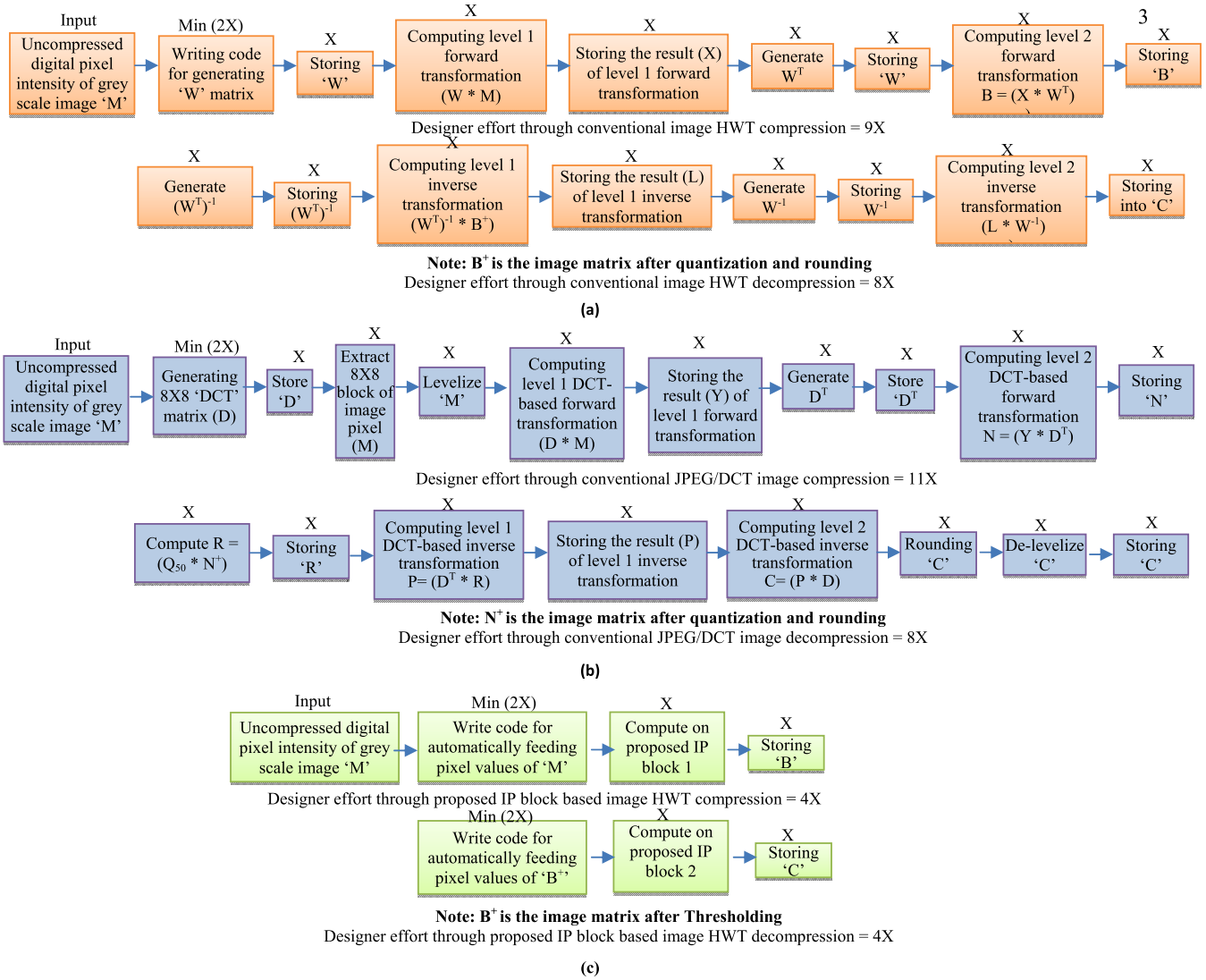

FIGURE 1. Reduction of designer effort obtained when applying proposed IP based method in Software Image Compression. (Reduction of 9× and 13× compared to conventional HWT image compression and JPEG/DCT image compression respectively). (a) Total designer effort for conventional lossy image HWT CODEC = min (17×) + 1× (Thresholding) = min (18×). (b) Total designer effort for lossy JPEG image CODEC = min (19×) + 2× (Quantization) + 1× (Rounding) = min (22×). (c) Total designer effort for lossy image CODEC through proposed approach IP based HWT CODEC= min (8×) + 1× (Thresholding) = min (9×).

transformation is also a two-step process i.e. level 1 inverse transformation on compressed data and level 2 inverse transformation on level 1 decompressed data.

D. PROPOSED IP CORE DESIGN FOR HWT-BASED IMAGE COMPRESSION

In the proposed IP core design for HWT-based image compression, we have introduced two functions to perform pixel intensity computation for level 1 forward transformation. The proposed functions transform the input image column-wise i.e. compute scaling coefficient by adding two vertically consecutive pixel intensities and divided by two for 1 to $N/2^{th}$ row, and compute wavelet coefficient by subtracting two vertically consecutive pixel intensities and divided by two for $(N/2 + 1)^{th}$ to N^{th} row. Thus divides the input image horizontally into two halves. The upper half represents the coarse details containing scaling coefficients (high frequencies) and the lower half represents the finer details containing wavelet coefficients (low frequencies). The proposed functions to perform pixel intensity computation for level 1 forward transformation are:

$$X_n^p = \left(\frac{m_{2n}^p + m_{2n-1}^p}{2}\right) \tag{1}$$

$$X_{\frac{N}{2}+i}^p = \left(\frac{m_{\frac{N}{2}+i-j+1}^p - m_{\frac{N}{2}+i-j}^p}{2}\right) \tag{2}$$

Where, N is the dimension of the square input image; n is the variable ranging from 1 to N/2, increases in every row; i is the variable ranging from 1 to N/2, increases in every row;

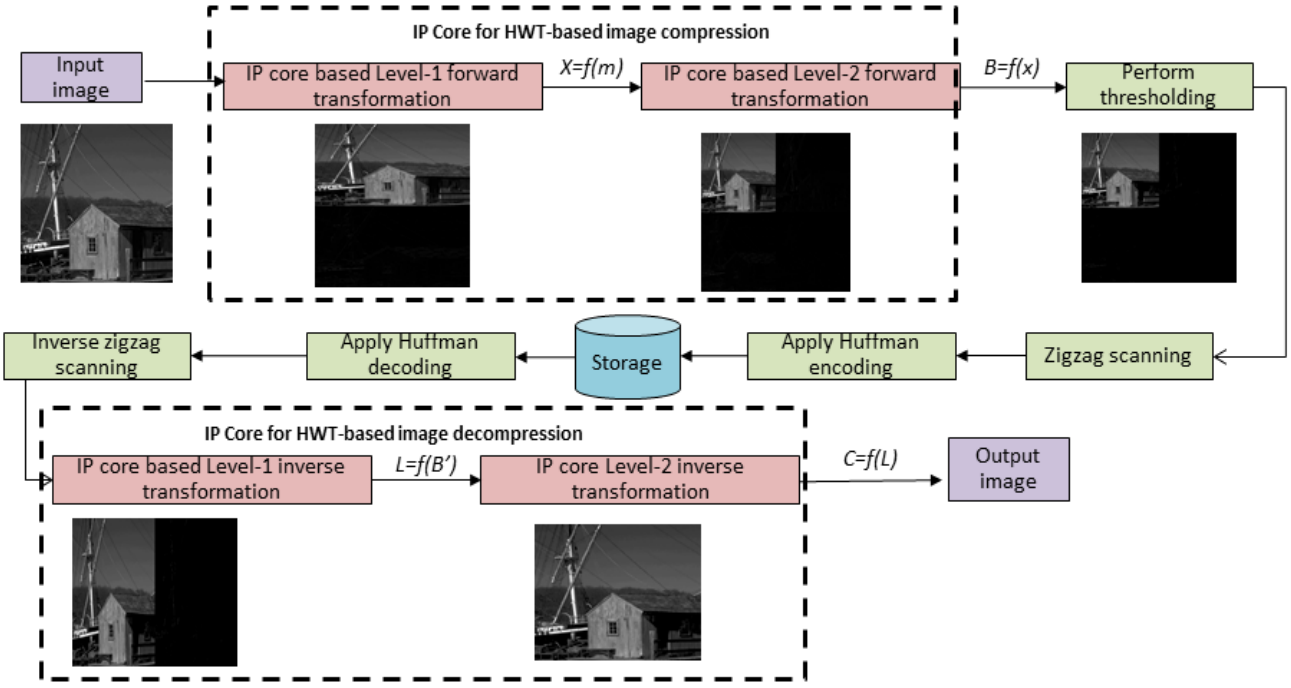

FIGURE 2. Framework for IP based image compression and decompression through dedicated hardware.

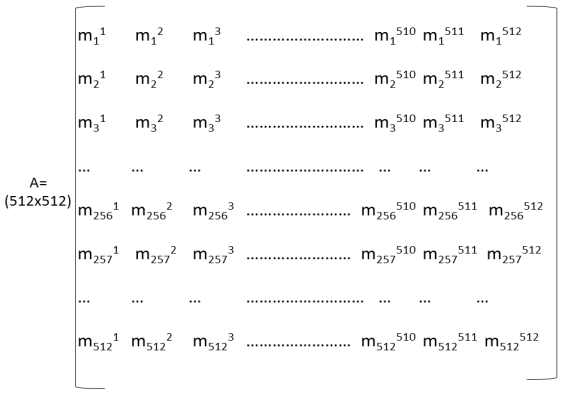

FIGURE 3. Generic pixel value of fa 512 × 512 image.

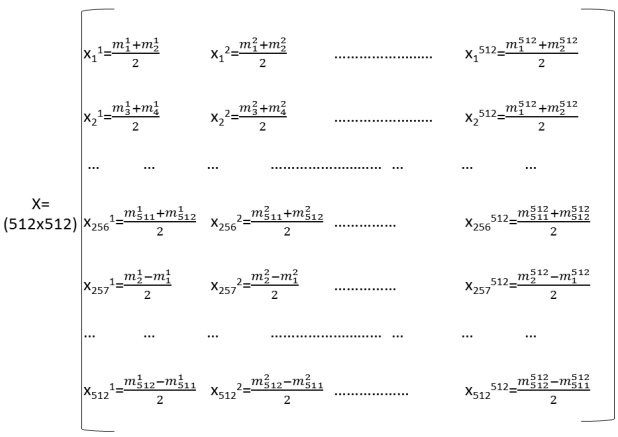

FIGURE 4. Forward HWT-based level 1 transformed image matrix.

p is the variable ranging from 1 to N, increases in every column; j is the variable ranging from N/2 to 1, decreases as 'i' increases. Eqn.1 is used for calculating 1 to N/2th row and Eqn.2 is used for calculating (N/2+1)th to Nth row of the input image. The corresponding level 1 forward transformation image in the form of matrix 'X' is shown in Fig.4 where calculation of pixel intensity is performed based on the aforementioned equations.

We have also introduced two functions to perform pixel intensity computation for level 2 forward transformations on the level 1 transformed image. The proposed functions transform the level 1 transformed image row-wise i.e. compute scaling coefficient by adding two horizontally consecutive pixel intensities and divided by two for 1 to N/2th column, and compute wavelet coefficient by subtracting two horizontally consecutive pixel intensities and divided by two for (N/2 + 1)th to Nth column. Thus divides each half of level 1 transformed image into vertically two halves and finally divides the input image into four quarters. The upper-left quarter contains the scaling coefficients of scaling coefficient (high-high frequencies), the upper-right quarter contains the scaling coefficients of wavelet coefficient (high-low frequencies), the lower-left quarter contains the wavelet coefficients of scaling coefficient (low-high frequencies) and the lower-right quarter contains the wavelet coefficients of wavelet coefficient (low-low frequencies). The proposed functions

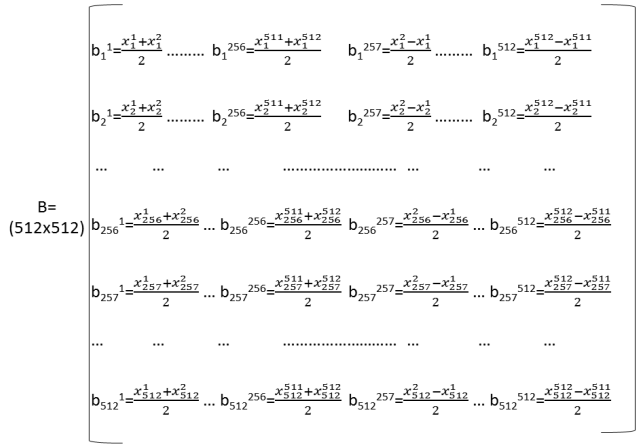

FIGURE 5. Forward HWT-based level 2 transformed image matrix.

to perform pixel intensity computation for level 2 forward transformations are:

$$B = \left(\frac{x_n^q \pm x_n^p}{2}\right) \tag{3}$$

$$B = \left(\frac{x_{\frac{N}{2}+i}^q \pm x_{\frac{N}{2}}^p}{2}\right) \tag{4}$$

Where, N is the dimension of the square level 1 transformed image; n is the variable ranging from 1 to N, increases in every row; i is the variable ranging from 1 to N/2, increases in every row; p is the odd variable ranging from 1 to N, increases in every column; q is the even variable ranging from 2 to N, increases in every column. The 1st and 2nd part of Eqn.3 is used for calculating upper-left and upper-right quarter of the level 1 transformed image respectively. The 1st and 2nd part of Eqn.4 is used for calculating lower-left and lower-right quarter of the level 1 transformed image respectively. The corresponding level 2 forward transformation image in the form of matrix 'B' is shown in Fig.5 where calculation of pixel intensity is performed based on the aforementioned equations. The HWT-based compressed image matrix (B) can be generated in one step computation from the input image matrix (A) through the following proposed functions:

$$B = \left(\left(\frac{m_{2n}^q + m_{2n-1}^q}{2}\right) + \left(\frac{m_{2n}^p + m_{2n-1}^p}{2}\right)/2\right) \tag{5}$$

$$B = \left(\left(\frac{m_{2n}^q + m_{2n-1}^q}{2}\right) - \left(\frac{m_{2n}^p + m_{2n-1}^p}{2}\right)/2\right) \tag{6}$$

$$B = \left(\left(\frac{m_{\frac{N}{2}+i-j+1}^q - m_{\frac{N}{2}+i-j}^q}{2}\right) + \left(\frac{m_{\frac{N}{2}+i-j+1}^p - m_{\frac{N}{2}+i-j}^p}{2}\right)/2\right) \tag{7}$$

$$B = \left(\left(\frac{m_{\frac{N}{2}+i-j+1}^q - m_{\frac{N}{2}+i-j}^q}{2}\right) - \left(\frac{m_{\frac{N}{2}+i-j+1}^p - m_{\frac{N}{2}+i-j}^p}{2}\right)/2\right) \tag{8}$$

Where, variable N, n, i, p, q are defined earlier; j is the variable ranging from N/2 to 1, decreases as 'i' increases. Eqn. 5 and 6 is calculated from Eqn. 3 and Eqn. 7and 8 is calculated from Eqn. 4 with the help of Eqn. 1 and 2.

To generate the compressed image matrix 'B', **Eqn. 5** is used for calculating intensity of pixels corresponding to both row and column index 1 to N/2; **Eqn. 6** is used for calculating intensity of pixels corresponding to row index 1 to N/2 and column index > than N/2 to N; **Eqn. 7** is used for calculating intensity of pixels corresponding to row index > than N/2 to N and column index 1 to N/2; **Eqn. 8** is used for calculating intensity of pixels corresponding to both row and column index > than N/2 to N. Those aforementioned functions can be used as a dedicated macro block in any image processing toolbox software like Matlab, OpenCV directly to perform HWT-based image compression.

E. COMPRESSION AND DECOMPRESSION OF IMAGE DATA

Thresholding is performed on the compressed image pixel (matrix) 'B' using function 'f', where 'f' is defined as follows:

$$B' = f(B, T) = \begin{cases} B, & b_n^p > T \\ 0, & b_n^p \leq T \end{cases} \tag{9}$$

Where, B' are the compressed image pixels after applying threshold based on the aforementioned function and 'T' is the hard threshold value. The compressed image matrix B' is then converted from a 2D matrix to a 1D array through zigzag scanning. Thereafter the 1D array is encoded through Huffman encoding in order to generate the bit stream data of the compressed image to store it in a storage device. To decompress the image from the stored data, the stored bit stream data is decoded through Huffman decoding and then the compressed image matrix B' is reconstructed through inverse zigzag scanning.

F. PROPOSED IP CORE DESIGN FOR HWT-BASED IMAGE DECOMPRESSION (IMAGE RECONSTRUCTION)

HWT-based image decompression can be achieved in two different ways. One way is, combining upper-left with lower left quarter together as well as upper-right with lower-right quarter together. This divides the level 1 decompressed image vertically into two halves (left half and right half), subsequently then combining left half and right half together thus generating the final decompressed image. Second way is, combining upper-left with upper-right quarter together and lower-left with lower-right quarter together. This divides the level 1 decompressed image horizontally into two halves (upper half and lower half), subsequently then combining upper half and lower half together thus generating the final decompressed image. In the proposed IP core design for HWT-based image decompression, we have introduced two functions for each of the cases to perform pixel computation for level 1 inverse transformation and two functions for each of the cases to

$$L=_{(512\times512)}\begin{bmatrix} l_1^1=(b'^1_1-b'^1_{257})\, l_1^2=(b'^2_1-b'^2_{257}) \ldots\ldots & l_1^{512}=(b'^{512}_1-b'^{512}_{257}) \\ l_2^1=(b'^1_1+b'^1_{257})\, l_2^2=(b'^2_1+b'^2_{257}) \ldots\ldots & l_2^{512}=(b'^{512}_1+b'^{512}_{257}) \\ \ldots\quad\ldots\quad\ldots\quad\ldots\ldots\ldots & \ldots\quad\ldots \\ l_{256}^1=(b'^1_{128}+b'^1_{384})\, l_{256}^2=(b'^2_{128}+b'^2_{384}) \ldots\ldots & l_{256}^{512}=(b'^{512}_{128}+b'^{512}_{384}) \\ l_{257}^1=(b'^1_{129}-b'^1_{385})\, l_{257}^2=(b'^2_{129}-b'^2_{385}) \ldots\ldots & l_{257}^{512}=(b'^{512}_{129}-b'^{512}_{385}) \\ \ldots\quad\ldots\quad\ldots\quad\ldots\ldots\ldots & \ldots\quad\ldots \\ l_{511}^1=(b'^1_{256}-b'^1_{512})\, l_{511}^2=(b'^2_{256}-b'^2_{512}) \ldots\ldots & l_{511}^{512}=(b'^{512}_{256}-b'^{512}_{512}) \\ l_{512}^1=(b'^1_{256}+b'^1_{512})\, l_{512}^2=(b'^2_{256}+b'^2_{512}) \ldots\ldots & l_{512}^{512}=(b'^{512}_{256}+b'^{512}_{512}) \end{bmatrix}$$

FIGURE 6. Inverse HWT-based level 1 transformed image matrix.

$$C=_{(512\times512)}\begin{bmatrix} c_1^1=(l_1^1-l_1^{257}) & c_1^2=(l_1^1+l_1^{257}) \ldots\ldots & c_1^{511}=(l_1^{256}-l_{257}^{512}) & c_1^{512}=(l_1^{256}+l_{257}^{512}) \\ c_2^1=(l_2^1-l_2^{257}) & c_2^2=(l_2^1+l_2^{257}) \ldots\ldots & c_2^{511}=(l_2^{256}-l_2^{512}) & c_1^{512}=(l_2^{256}+l_2^{512}) \\ \ldots\quad & \ldots\quad & \ldots\quad & \ldots \\ c_{256}^1=(l_{256}^1-l_{256}^{257}) & \ldots\ldots & c_{256}^{511}=(l_{256}^{256}-l_{256}^{512}) & c_{256}^{512}=(l_{256}^{256}+l_{256}^{512}) \\ c_{257}^1=(l_{257}^1-l_{257}^{257}) & \ldots\ldots & c_{257}^{511}=(l_{257}^{256}-l_{257}^{512}) & c_{257}^{512}=(l_{257}^{256}+l_{257}^{512}) \\ \ldots\quad & \ldots\quad & \ldots\quad & \ldots \\ c_{511}^1=(l_{511}^1-l_{511}^{257}) & \ldots\ldots & c_{511}^{511}=(l_{511}^{256}-l_{511}^{512}) & c_{511}^{512}=(l_{511}^{256}+l_{511}^{512}) \\ c_{512}^1=(l_{512}^1-l_{512}^{257}) & \ldots\ldots & c_{512}^{511}=(l_{512}^{256}-l_{512}^{512}) & c_{512}^{512}=(l_{512}^{256}+l_{512}^{512}) \end{bmatrix}$$

FIGURE 7. Inverse HWT-based level 2 transformed image matrix.

perform pixel computation for level 2 inverse transformations. For the first case, the proposed functions for level 1 inverse transformation transforms the compressed image column-wise i.e. subtract i^{th} and $(N/2 + i)^{th}$ row's pixel intensity for odd rows, and add i^{th} and $(N/2 + i)^{th}$ row's pixel intensity for even rows. Thus level 1 HWT-based inverse transformation represents the level 1 decompressed image into two halves. The left half represents the coarse details containing scaling coefficients (high frequencies) and the right half represents the finer details containing wavelet coefficients (low frequencies). The proposed functions for level 2 inverse transformation transforms the level 1 transformed image row-wise i.e. subtract i^{th} and $(N/2+i)^{th}$ column's pixel intensity for odd columns, and add i^{th} and $(N/2+i)^{th}$ column's pixel intensity for even columns. Thus level 2 HWT-based inverse transformations construct the complete decompressed image.

For the next case, the proposed functions for level 1 inverse transformation transforms the compressed image row-wise i.e. subtract i^{th} and $(N/2 + i)^{th}$ column's pixel intensity for odd columns, and add i^{th} and $(N/2 + i)^{th}$ column's pixel intensity for even columns. Thus level 1 HWT-based inverse transformation represents the decompressed image into two

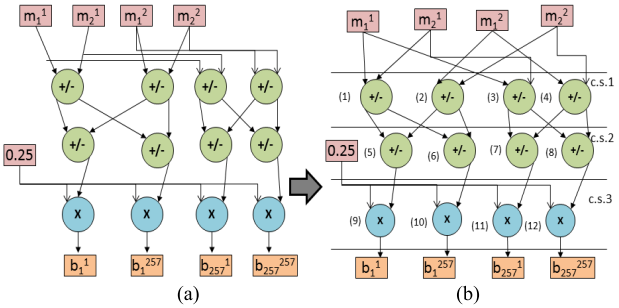

FIGURE 8. (a) DFG of HWT-based image compression, (b) Scheduled DFG based on 4 adder-subtractor unit and 4 multiplier.

halves. The upper half represents the coarse details containing scaling coefficients (high frequencies) and the lower half represents the finer details containing wavelet coefficients (low frequencies). The proposed functions for level 2 inverse transformation transforms the level 1 transformed image column-wise i.e. subtract i^{th} and $(N/2 + i)^{th}$ row's pixel intensity for odd rows, and add i^{th} and $(N/2+i)^{th}$ row's pixel intensity for even rows. Thus level 2 HWT-based inverse transformations construct the complete decompressed image. The proposed functions to perform first type of pixel intensity computation for level 1 inverse transformation are:

$$L = \left(b'^z_x - b'^z_{\frac{N}{2}+x}\right) \tag{10}$$

$$L = \left(b'^z_x + b'^z_{\frac{N}{2}+x}\right) \tag{11}$$

Where, N is the dimension of the square compressed image matrix; z is the variable ranging from 1 to N, increases in every column; x is the variable ranging from 1 to N/2, increases in every alternate row. Eqn.10 is used for calculating the pixel value of odd rows and Eqn.11 is used for calculating the pixel value of even rows of the compressed image. The corresponding level 1 decompressed image in the form of matrix 'L' is shown in Fig.6 where calculation of each pixel is performed based on the aforementioned equations. The proposed functions to perform first type of pixel intensity computation for level 2 inverse transformations are:

$$C = \left(l^z_x - l^{\frac{N}{2}+z}_x\right) \tag{12}$$

$$C = \left(l^z_x + l^{\frac{N}{2}+z}_x\right) \tag{13}$$

Where, N is the dimension of the square level 1 decompressed image matrix; z is the variable ranging from 1 to N/2, increases in every alternate column; x is the variable ranging from 1 to N, increases in every row. Eqn.12 is used for calculating the pixel value of odd columns and Eqn.13 is used for calculating the pixel value of even columns of the level 1 decompressed image. The corresponding level 2 decompressed image in the form of matrix 'C' is shown in Fig.7 where calculation of each pixel is performed based on the aforementioned equations. The proposed functions

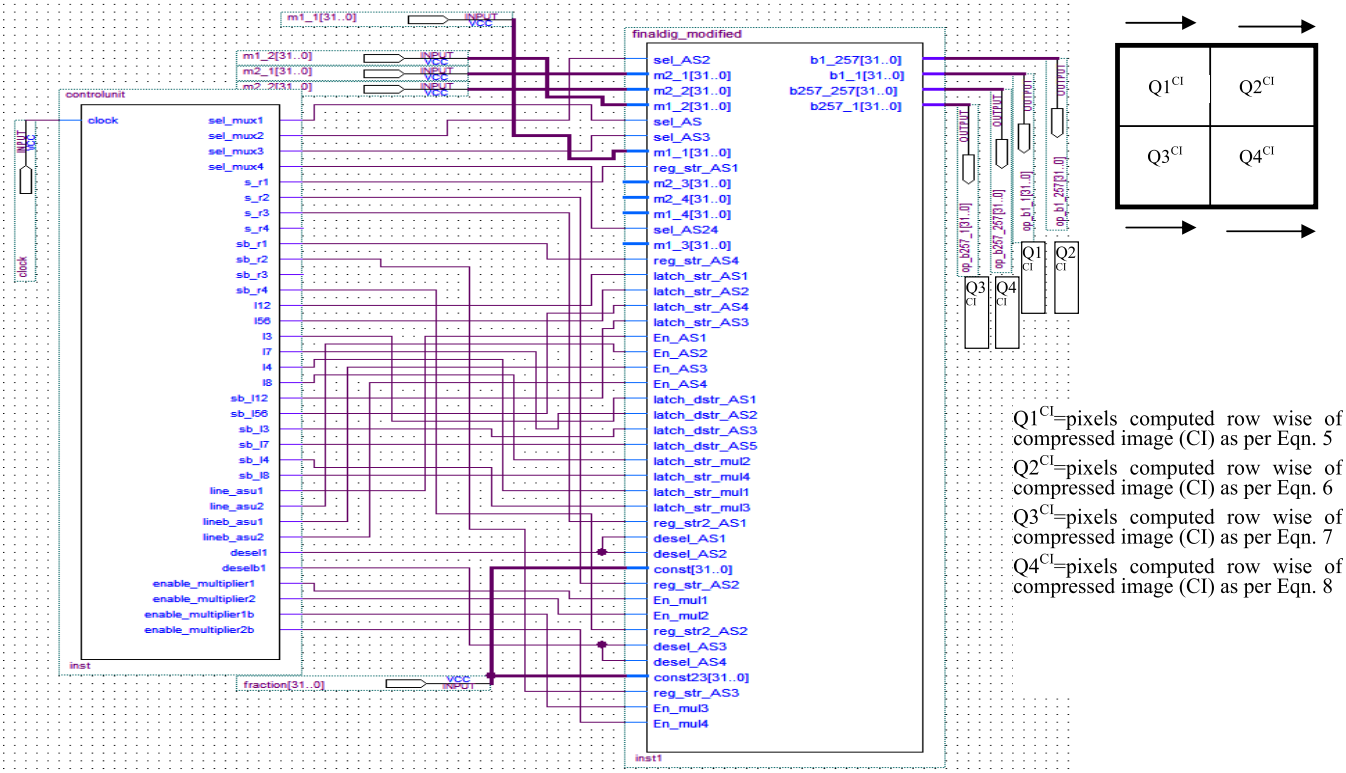

FIGURE 9. **IP core for HWT-based image compressor system.** ***The four output of the IP core are computed for four quarters and increases row-wise in each quarter of compressed image (CI).***

to perform next type of pixel intensity computation for level 1 inverse transformation are:

$$L = \left(b'^{z}_{x} - b'^{\frac{N}{2}+z}_{x}\right) \tag{14}$$

$$L = \left(b'^{z}_{x} + b'^{\frac{N}{2}+z}_{x}\right) \tag{15}$$

Where, N is the dimension of the square compressed image matrix; z is the variable ranging from 1 to N, increases in every column; x is the variable ranging from 1 to N/2, increases in every alternate row. Eqn.14 is used for calculating the pixel value of odd rows and Eqn.15 is used for calculating the pixel value of even rows of the compressed image. The corresponding level 1 decompressed image in the form of matrix 'L' is shown in Fig.6 where calculation of each pixel is performed based on the aforementioned equations.

The proposed functions to perform next type of pixel intensity computation for level 2 inverse transformations are:

$$C = \left(l^{z}_{x} - l^{z}_{\frac{N}{2}+x}\right) \tag{16}$$

$$C = \left(l^{z}_{x} + l^{z}_{\frac{N}{2}+x}\right) \tag{17}$$

Where, N is the dimension of the square level 1 decompressed image matrix; z is the variable ranging from 1 to N, increases in every column; x is the variable ranging from 1 to N/2, increases in every alternate row. Eqn.16 and Eqn. 17 are used for calculating the pixel value of odd rows and even rows of the level 1 decompressed image.

The HWT final decompressed image pixel intensities (of matrix C) can be generated in one step computation from compressed image matrix (B') through following proposed functions:

$$\mathrm{C} = \left(b'^{g}_{i} - b'^{g}_{\frac{N}{2}+i}\right) - \left(b'^{\frac{N}{2}+g}_{i} - b'^{\frac{N}{2}+g}_{\frac{N}{2}+i}\right) \tag{18}$$

$$\mathrm{C} = \left(b'^{g}_{i} - b'^{g}_{\frac{N}{2}+i}\right) + \left(b'^{\frac{N}{2}+g}_{i} - b'^{\frac{N}{2}+g}_{\frac{N}{2}+i}\right) \tag{19}$$

$$\mathrm{C} = \left(b'^{g}_{i} + b'^{g}_{\frac{N}{2}+i}\right) - \left(b'^{\frac{N}{2}+g}_{i} + b'^{\frac{N}{2}+g}_{\frac{N}{2}+i}\right) \tag{20}$$

$$\mathrm{C} = \left(b'^{g}_{i} + b'^{g}_{\frac{N}{2}+i}\right) + \left(b'^{\frac{N}{2}+g}_{i} + b'^{\frac{N}{2}+g}_{\frac{N}{2}+i}\right) \tag{21}$$

Where, N is the dimension of the square input image matrix; g is the variable ranging from 1 to N/2, increases in every alternate column; i is the variable ranging from 1 to N/2, increases in every alternate row. To reconstruct the each pixel value of matrix 'C' Eqn.18 is used for odd row and odd column elements, Eqn.19 is used for odd row and even column elements, Eqn.20 is used for even row and odd column elements, Eqn.21 is used for even row and even column elements of matrix B'.

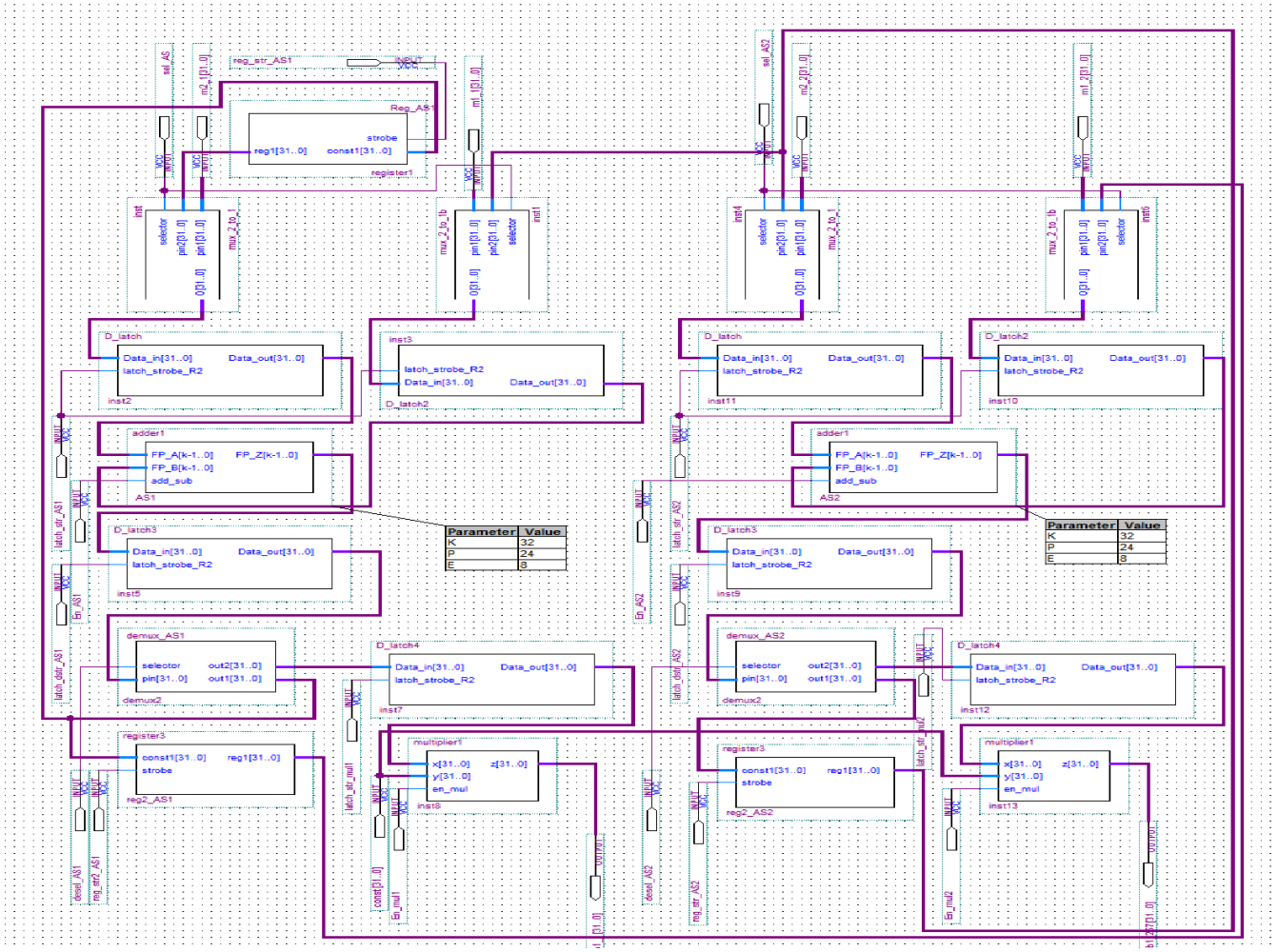

FIGURE 10. Schematic system of datapath processor of HWT-based image compressor (computing two pixels in parallel).

IV. DESIGN OF IP CORE FOR HWT-BASED IMAGE COMPRESSOR AND DECOMPRESSOR

A. IP CORE DESIGN FOR HWT-BASED IMAGE COMPRESSION

To design an IP core for HWT-based image compression a Data Flow Graph (DFG) for HWT-based image compression is designed based on the Eqn. 5 – Eqn.8 shown in Fig. 8 (a). The DFG is then scheduled based on designer specified resources e.g. four 32-bit adder-subtractor unit (ASU) and four 32-bit multipliers, shown in Fig. 8(b) (*Note: four 32-bit ASU 0 and four 32-bit Mul are chosen as device support for I/O pins beyond this are not available with Cyclone II FPGA. Further, 32 –bit resources have been selected because each resource operates on floating point value during computation which requires 32 bit IEEE single precision format in normalized scientific notation).* The green nodes represent ASU and the blue nodes represent multiplier in the graph. The multiplexing scheme is performed on the scheduled DFG based on the resource constraints. Multiplexing scheme is the process of representing each resource with corresponding inputs and outputs. For example in Fig 8(b), operation 1 and operation 5 are computed through same functional unit i.e. ASU1. Similarly, opn 2 & opn 6; opn 3 & opn 7; opn 4 & opn 8 are computed through ASU2, ASU3, and ASU4 respectively. The datapath processor of the complete IP core is designed based on the multiplexing scheme of each system resource. Multiplexer and demultiplexer can be integrated easily into the datapath based on the multiplexing scheme. The control unit controls different components of the datapath and makes synchronization between them. It is responsible for activating and deactivating signals like selector, deselector, latch strobe, enabler etc. in the datapath processor so that the components like multiplexer, demultiplexer, latch, functional unit (adder, multiplier etc.), register etc. response at the right time. Our proposed IP core design for HWT-based image compression system is shown in Fig.9 where the left block is the controller and the right block represents the datapath processor of the complete IP core design of HWT-based image compressor system. Each component of datapath processor and the control unit of HWT-based image compres-

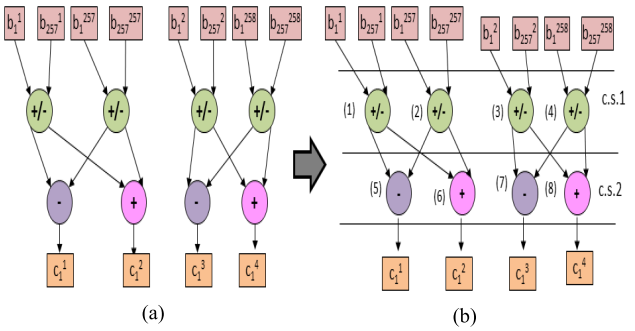

FIGURE 11. (a) DFG of HWT-based image decompression, (b) Scheduled DFG based on 4 adder-subtractor unit, 2 adder and 2 subtractor.

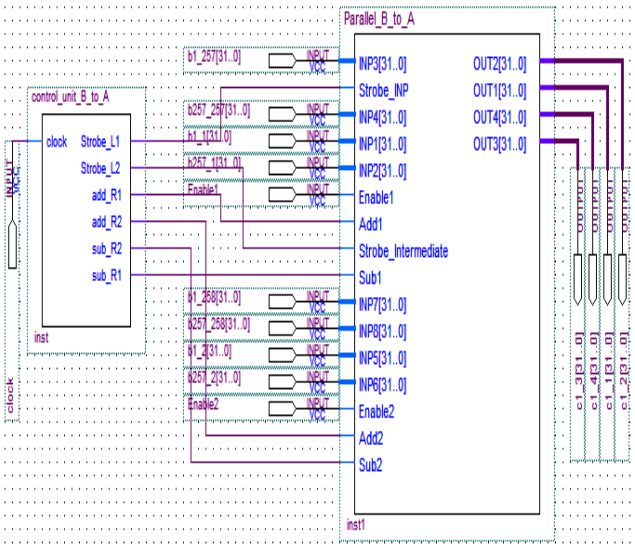

FIGURE 12. System of IP core for HWT-based image decompressor. ***(Note: The four output of the IP core are computed row-wise of decompressed image).***

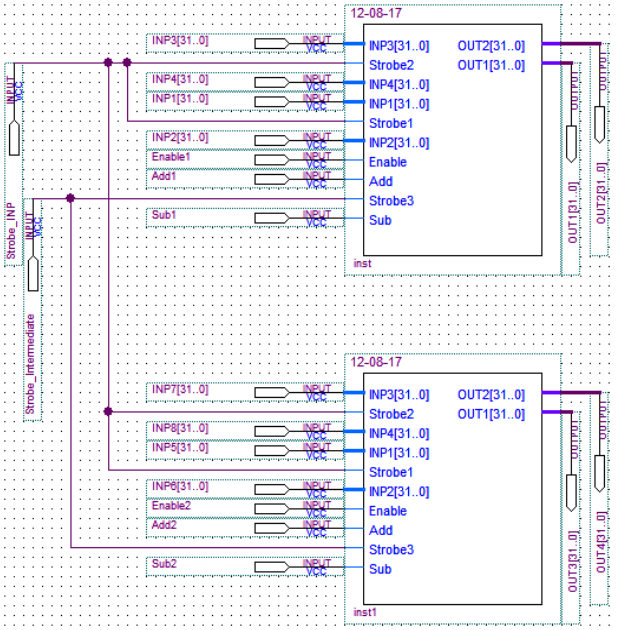

FIGURE 13. Internal System of datapath processor for HWT-based image decompressor.

TABLE 3. Comparison of device utilization for proposed IP cores, normal HWT and JPEG CODEC.

IP Core	Total logic elements	Total register	Total pins
Proposed Forward HWT	4360	583	433
Proposed Inverse HWT	5329	391	384
Normal Forward HWT	> 33216	>34593	-
Normal Inverse HWT	>33216	>34593	-
Standard JPEG CODEC	12121	1826	322

sor are designed using VHDL as the hardware description language. The IP core is capable to accept the 8-bit digital pixel value *(Note: we have considered 8-bit as it provides the maximum grayscale shades)* of a grayscale input image, perform level 1 and level 2 forward transformations and generate the digital pixel value of the corresponding compressed image. For example, as shown in Fig. 8, m_1^1, m_2^1, m_1^2, m_2^2 is taken as inputs which represents the pixel intensities of input image as shown in Fig. 3 and generates the outputs as b_1^1, b_1^{257}, b_{257}^1, b_{257}^{257} which represents the pixel intensities of compressed image as shown in Fig. 5. Similarly, the remaining pixel intensities of the compressed image are determined. The internal schematic block diagram representation of datapath processor is shown in Fig. 10. This novel IP core of HWT-based image compressor system can be used as a black box reusable core in a camera SoC where a designer does not need to know internal process of HWT image compression.

B. IP CORE DESIGN FOR HWT-BASED IMAGE DECOMPRESSION

To design an IP core for HWT-based image decompression a DFG is designed again based on the Eqn. 18 – Eqn.21 shown in Fig. 11(a). The DFG is then scheduled based on four 32-bit ASU, two 32-bit adders and two 32-bit substractors, shown in Fig. 11(b). (*Note: four 32-bit ASU, two 32-bit adders and two 32-bit subtractors are chosen because device support for I/O pins beyond this is not available with Cyclone II FPGA. Further, 32 –bit resources have been selected because each resource operates on floating point value during computation which requires 32 bit IEEE single precision format in normalized scientific notation).*The green nodes represent ASU, the purple node represents subtractor and the pink nodes represent adder in the graph. In our proposed hardware design for HWT-based image decompression shown in Fig.12 where, the left block is the control unit and the right block represents the datapath processor of the complete hardware design of HWT-based image decompressor system. Each component of datapath and the control unit of HWT-based image decompressor are designed using VHDL as the hardware description language. The hardware is capable to accept the pixel value of a compressed image, perform level 1 and level 2 inverse transformations and generate the pixel value of the corresponding decompressed image. For example, as shown in Fig. 11, b_1^1, b_1^{257}, b_{257}^1, b_{257}^{257} is taken as inputs which represents the pixel

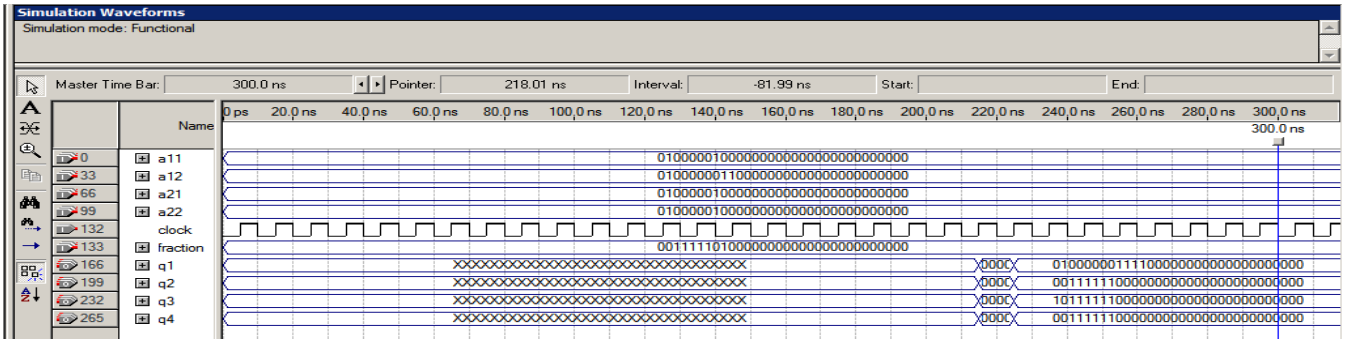

FIGURE 14. Simulation of IP core 1: 2D Image compressor (for test image shown in Fig.2). *Note:* $q1 = b_1^1$, $q2 = b_1^{257}$, $q3 = b_{257}^1$, $q4 = b_{257}^{257}$ *of compressed image.*

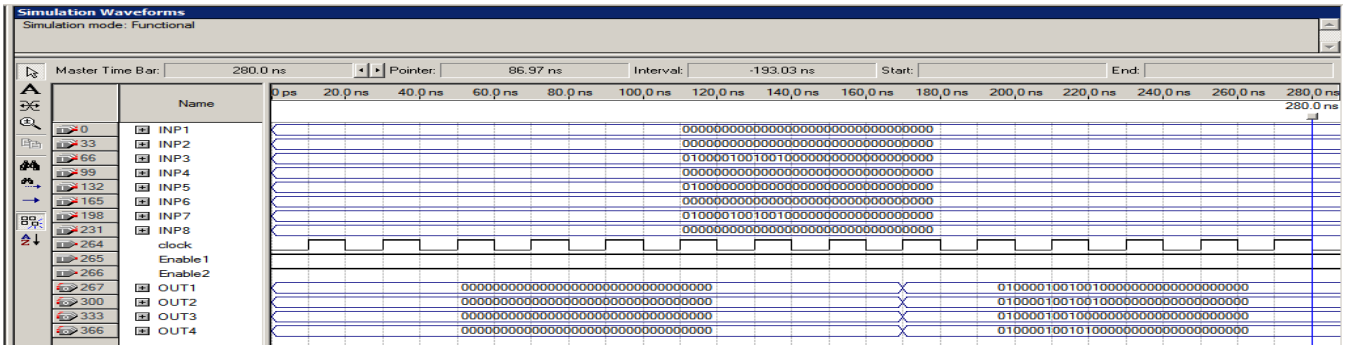

FIGURE 15. Simulation of IP core 2: 2D Image decompressor (for test image shown in Fig.2). *Note:* $OUT1 = c_1^1$, $OUT1 = c_1^2$, $OUT1 = c_1^3$, $OUT1 = c_1^4$ *of decompressed image.*

TABLE 4. For IP based HWT compressed image for T = 25.

Images	Original size (bits)	Compressed size (bits)	Compression efficiency (%)	MSE	PSNR
Image 1	2097152	653383	68.84	86.2	28.78
Image 2	2097152	540771	74.21	174	25.72
Image 3	3276800	812689	75.20	102	28.03
Image 4	3276800	1001555	69.44	148	26.44
Image 5	3276800	989390	69.81	61.4	30.25
Image 6	2097152	459841	78.07	239	24.34

TABLE 5. For JPEG/DCT compressed image.

Images	Original size (bits)	Compressed size (bits)	Compression efficiency (%)	MSE	PSNR
Image 1	2097152	710608	66.12	35.81	32.59
Image 2	2097152	743664	64.54	38.93	32.77
Image 3	3276800	1133296	65.41	49.9	31.15
Image 4	3276800	1733456	47.1	77.3	29.25
Image 5	3276800	1304352	60.19	35.41	32.64
Image 6	2097152	942320	55.07	96.17	28.3

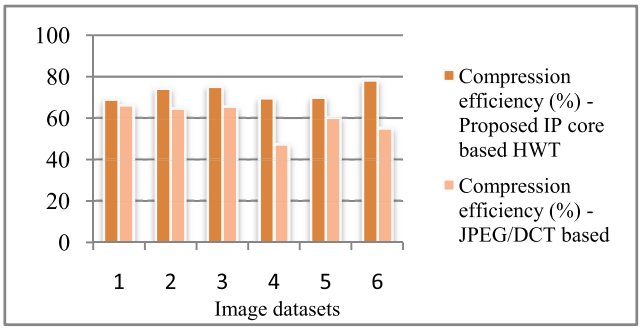

FIGURE 16. Comparison of compression efficiency (%) for proposed vs. standard JPEG/DCT based compression.

intensity of compressed image as shown in Fig. 5 and generates the outputs as c_1^1, c_1^2 or c_2^1, c_2^2 (depending on the enabler of ASU) which represents the pixel intensity of decompressed image as shown in Fig. 7. The complete datapath processor shown is Fig. 12 is implemented in two sub-block diagram as shown in Fig. 13 and the detailed schematic representation of each block of Fig. 13 is shown in Fig. 14. This novel hardware of HWT-based image decompressor can be used as a black box for a user who has no knowledge about the internal process of HWT-based image decompression.

Both IP cores are ready to simulate in a synthesis tool and can be emulated in any FPGA (Field Programmable Gate Array) device. The hardware design of both the IPs is implemented in Altera Quartus II 7.2. The simulation result and the device utilization summary are discussed in the next section.

V. RESULTS AND ANALYSIS

CT images [16], NASA images [17] and standard 512 × 512 gray scale test images [18] are used as image dataset to verify and compare the proposed framework and IP core designs for HWT-based image compression & decompression with normal HWT-based IP core and JPEG/DCT IP core. As mentioned earlier all the aforementioned IP cores

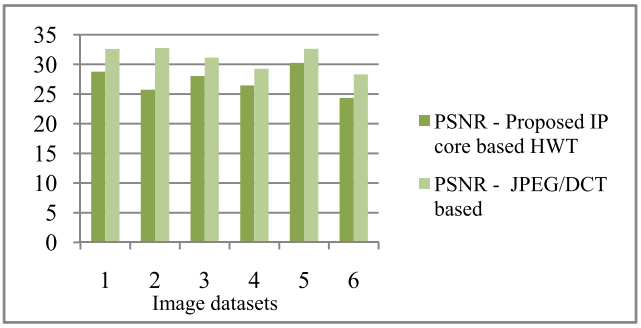

FIGURE 17. Comparison of PSNR for proposed vs. standard JPEG/DCT based compression.

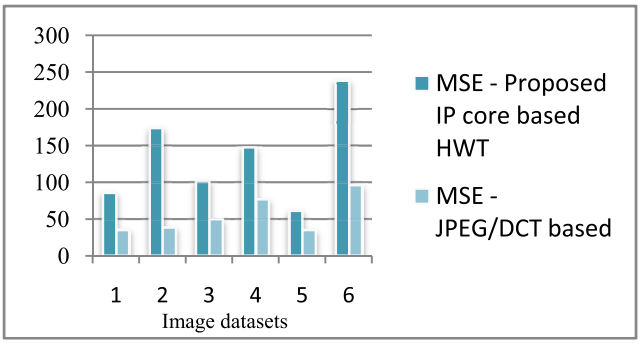

FIGURE 18. Comparison of MSE for proposed vs. standard JPEG/DCT based compression.

are implemented in Altera Cyclone II family, device no. EP2C35F672C6. Table III reports the comparison between proposed HWT-based CODEC, normal HWT-based IP core and standard JPEG/DCT IP core in terms of total used logic elements, registers, and I/O pin. As shown in Table III our proposed HWT-based IP core uses less resources whereas the normal HWT-based IP core can not be implemented in our used device due to lack of I/O pins. The simulation result for IP core 1 and 2 is shown in Fig. 14 and 15 respectively which indicates that the designed IP core/IP block framework was successful in compression and decompression of test image (shown in Fig.2). Similarly, successful results were obtained for all tested images selected from 3 datasets [16]–[18]. Total six images are selected from 3 datasets [16]–[18] to report the compression efficiency for different threshold (T) values. Additionally, the Mean Square Error (MSE) [1], [2] and the Peak Signal to Noise Ratio (PSNR) [1], [2] of the compressed images is also reported. Table IV reports the comparison between the original image and the proposed IP based compressed image for hard threshold $T = 25$ in terms of storage size in bits. Further, it also reports the compression efficiency percentage, MSE and PSNR of all the test images. Table V reports the same quality parameters of the standard JPEG/DCT based compressed image. It can be observed that the compressed image generated through proposed HWT-based IP core achieves higher compression efficiency compared to standard JPEG/DCT method. The graphical comparison of compression efficiency, PSNR and MSE is also shown in Fig. 16, 17 and 18 respectively.

VI. CONCLUSION AND FUTURE WORK

In this paper, a novel IP design based HWT image compression and decompression including its mathematical framework is proposed. The models can be used as a dedicated macro block in the library of an image processing toolbox to perform end to end HWT-based image compression. Further, the designed hardware can be used as an IP core in digital camera systems to perform image compression and decompression. Our future works aims to develop IP based video CODEC through mathematical functions, followed by subsequent validation in commercial synthesis tool.

REFERENCES

[1] S. Benchikh and M. Corinthios, "A hybrid image compression technique based on DWT and DCT transforms," in *Proc. Int. Conf. Adv. Infocom Technol.*, Wuhan, China, 2011, pp. 1–8.
[2] A. M. G. Hnesh and H. Demirel, "DWT-DCT-SVD based hybrid lossy image compression technique," in *Proc. Int. Image Process., Appl. Syst. (IPAS)*, Hammamet, Tunisia, Nov. 2016, pp. 1–5.
[3] *Information Technology—JPEG 2000 Image Coding System: Core Coding System*, document ISO/IEC 15444-1 | ITU-T Rec. T.800, 2002.
[4] T. Bruylants, A. Munteanu, and P. Schelkens, "Wavelet based volumetric medical image compression," *Signal Process., Image Commun.*, vol. 31, pp. 112–133, Feb. 2015.
[5] A. A. Nashat and N. M. H. Hassan, "Image compression based upon wavelet transform and a statistical threshold," in *Proc. Int. Conf. Optoelectron. Image Process.*, Warsaw, Poland, Jun. 2016, pp. 20–24.
[6] A. Bilgin and M. W. Marcellin, "Applications of reversible integer wavelet transforms to lossless compression of medical image volumes," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Aug. 1998, p. 411.
[7] S. Li, H. Yin, X. Fang, and H. Lu, "Lossless image compression algorithm and hardware architecture for bandwidth reduction of external memory," *IET Image Process.*, vol. 11, no. 6, pp. 379–388, 2017.
[8] G. Scarmana and K. McDougall, "Exploring the application of some common raster scanning paths on lossless compression of elevation images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Milan, Italy, Jul. 2015, pp. 4514–4517.
[9] R. K. Lama, S. Shin, M. Kang, G.-R. Kwon, and M.-R. Choi, "Interpolation using wavelet transform and discrete cosine transform for high resolution display," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2016, pp. 184–186.
[10] C. Yu and S.-J. Chen, "Design of an efficient VLSI architecture for 2-D discrete wavelet transforms," *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 135–140, Feb. 1999.
[11] A. Sengupta, "Evolution of the IP design process in the semiconductor/EDA industry [hardware matters]," *IEEE Consum. Electron. Mag.*, vol. 5, no. 2, pp. 123–126, Apr. 2016.
[12] A. Sengupta, "Cognizance on intellectual property: A high-level perspective [hardware matters]," *IEEE Consum. Electron. Mag.*, vol. 5, no. 3, pp. 126–128, Jul. 2016.
[13] P. M. Corcoran, P. Bigioi, and E. Steinberg, "Wireless transfer of images from a digital camera to the Internet via a standard GSM mobile phone," in *Proc. Int. Conf. Consum. Electron. (ICCE)*, Los Angeles, CA, USA, Jun. 2001, pp. 274–275.
[14] P. M. Corcoran, P. Bigioi, and E. Steinberg, "Wireless transfer of images from a digital camera to the Internet via a standard GSM mobile phone," *IEEE Trans. Consum. Electron.*, vol. 47, no. 3, pp. 542–547, Aug. 2001.
[15] I. Andorko, P. Corcoran, and P. Bigioi, "A dual image processing pipeline camera with CE applications," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2011, pp. 737–738.
[16] *Public Lung Database to Address Drug Response*. Accessed: 2017. [Online]. Available: http://www.via.cornell.edu/databases/crpf.html
[17] *NASA Image and Video Library*. Accessed: 2017. [Online]. Available: https://images.nasa.gov/#/

[18] *Dataset of Standard 512×512 Grayscale Test Images*. Accessed: Aug. 18, 2017. [Online]. Available: http://decsai.ugr.es/cvg/CG/base.htm
[19] E. Kougianos, S. P. Mohanty, G. Coelho, U. Albalawi, and P. Sundaravadivel, "Design of a high-performance system for secure image communication in the Internet of Things," *IEEE Access J.*, vol. 4, pp. 1222–1242, 2016.
[20] S. P. Mohanty, "A secure digital camera architecture for integrated real-time digital rights management," *J. Syst. Archit.*, vol. 55, nos. 10–12, pp. 468–480, Dec. 2009.

ANIRBAN SENGUPTA (M'09–SM'17) is an Assistant Professor in computer science and engineering at IIT Indore. He has been an active researcher with strong publications in the emerging areas of hardware security, IP core protection, privacy and digital rights management for electronics devices, and forensic engineering for security. He has 133 publications and patents, with bulk of them in IEEE and IET periodicals. His patents have been cited in various industry patents, such as IBM Corporation, Siemens, Qualcomm, and STC University of Mexico. Several of his IEEE publications have appeared in Top 50 Most Popular Articles from the IEEE Periodicals. He has supervised around 15 candidates, including several Ph.D.s, many of whom are/were placed in academia and industry. He was a recipient of the Sir Visvesvaraya Faculty Research Fellow by the Ministry of Electronics and IT and the Best Research Paper Award 2017 by IIT Indore. He was awarded the prestigious IEEE Distinguished Lecturer (Renowned DL of IEEE CE Society) by IEEE Consumer Electronics Society in 2017, as well as the Outstanding Associate Editor Award from the IEEE Computer Society Technical Committee on VLSI (TCVLSI) Letter Editorial Board of the IEEE Computer Society in 2017. He was awarded the highest rating of Excellent by the expert committee of Department of Science and Technology (DST) based on the performance (output) in an externally funded project in 2017. His research achievements have received wide media coverage as IET International News, U.K., in December 2017 issue of IET Member News periodical (Vol.9, Issue 46, pp.5). He has been inducted into the Executive Committee of the IEEE Computer Society Technical Committee on VLSI in 2017. He has successfully commissioned special issues on hardware security, privacy, IP protection in the IEEE TVLSI, IET CDT, the IEEE Access, and the IEEE CEM, which have received wonderful response from scientific community. Furthermore, due to high quality special issue in the IEEE Access in 2016, he has also been conferred an award from the IEEE Access for his outstanding contribution. His research ideas have been awarded competitive funding by various prestigious agencies, such as DST, the Council of Scientific and Industrial Research, and the Department of Electronics and IT. He is currently an Editor-in-Chief of the IEEE VLSI Circuits and Systems Letter of the IEEE Computer Society Technical Committee on VLSI. He also currently serves in several editorial positions as a Senior Editor, an Associate Editor, an Editor, and a Guest Editor of several IEEE transactions/journals, IET, and Elsevier journals, including the IEEE Transactions on Aerospace and Electronic Systems, the IEEE Transactions on Very Large Scale Integration (VLSI) Systems, the IEEE Access, *IET Journal on Computer and Digital Techniques*, *Elsevier Microelectronics Journal*, *IEEE Consumer Electronics Magazine*, and the IEEE VLSI Circuits and Systems Letter. He also serves as a Guest Editor of the IEEE Transactions on VLSI Systems, *IET Computers and Digital Techniques*, and the IEEE Access. He is the Technical Program Chair of the 36th IEEE International Conference on Consumer Electronics 2018 in Las Vegas, the 15th IEEE International Conference on Information Technology 2016, the 3rd IEEE International Symposium on Nanoelectronic and Information Systems 2017, and the 2019 IEEE International Symposium on VLSI in Florida.

DIPANJAN ROY (S'16) was a Software Development Engineer at the Amazon Development Center, Bengaluru. He is currently a Research Scholar in computer science and engineering at IIT Indore.

SARAJU P. MOHANTY (SM'08) is currently a Professor with the University of North Texas. He has authored 230 research articles, three books, and invented four U.S. patents. His research is in smart electronic systems. His Google Scholar h-index is 28 and i10-index is 82. He serves as the Chair of the Technical Committee on VLSI, the IEEE Computer Society. He has been recognized as an IEEE Distinguished Lecturer by the Consumer Electronics Society (CESoc) in 2017. He was conferred the Glorious India Award in 2017 for his exemplary contributions to the discipline. He received the Society for Technical Communication 2017 Award of Merit for his outstanding contributions to the *IEEE Consumer Electronics Magazine.* He was conferred the 2016–2017 UNT Toulouse Scholars Award for sustained excellent scholarship and teaching achievements. He was the recipient of the 2016 PROSE Award for the Best Textbook in Physical Sciences and Mathematics from the Association of American Publishers for the book titled *Nanoelectronic Mixed-Signal System Design* (McGraw-Hill, 2015). He is the Editor in Chief of the *IEEE Consumer Electronics Magazine*.

PETER CORCORAN (F'10) is currently the Personal Chair in electronic engineering at the College of Engineering and Informatics, NUI Galway. He has co-authored 300 technical publications and co-invented 300 granted U.S. patents. He is the Founding Editor of the *IEEE Consumer Electronics Magazine*.