

## **Design and Implementation of a Relaxed Haar Discrete Wavelet Transform Hardware module for Multimedia Compression**

James Ntaganda

*Electronic Engineering Department, Konkuk University, South Korea*

---

**Abstract:** Still Images and video clips are common formats for storing and sharing information in different fields. Information content in such multimedia formats is huge and requires compression. Compression is done to reduce redundant information content while maintaining good perceptual video or Image quality for Human Visual Systems (HVS) and involves signal transformation together with other operations. While MPEG-X and H.26X deploy DCT (Discrete Cosine Transform), Discrete Wavelet Transforms (DWTs) are common in different multimedia CODECs such as MJPEG2000, MJPEG-XR and MJPEG-LS. DWT like all other transformations requires intensive computation and consume more power. To mitigate the trade-off between computation duration, bulkiness of subsystems and power consumption, Software-Hardware co-design approach is used. In this approach, Hardware modules called hardware accelerators are reserved for intensive functions and are called by the main program as a subroutine, thus a good flexible design approach is required for such hardware circuit modules. In this paper, we revisit the fundamentals of Haar DWT. We exploit its flexibility and simplicity to design a relaxed Hardware module that can be used to process a 16x16 spatial micro block from an Image or a video frame without multiplication because multiplications takes longer and hence more power consumption. The Module is Implemented on FPGA

**Keywords**– Discrete Wavelet Transform, Multimedia Compression, Hardware Realization, FPGA

---

### **I. Introduction**

The demand for information exchange and storage in Image and video data formats is expected to grow in a proportional relation to teledensity and internet connectivity. However, Video binary information content is enormous to handle in overloaded data networks and computation involved is too intensive especially for portable devices. To reduce information content carried with in images, standardization groups recommend different specific methods and techniques in video compression for flexible compatibility across different platforms. Compression is done while keeping a good perceptual video quality for Human Visual Systems (HVS). High Efficiency Video Coding (HEVC) is a new Standard for video compression developed by the ISO and ITU-T [12]. Despite the success of MPEG-X and H.26X CODECs, Jason Spielfogel [4] points out that they continue to show drawbacks due to spatial-temporal compression involved in some specific scenes and applications. These drawbacks include [4]:

- (a) In scenes with high movement, temporal compression can create "artifacts" or remnants of previous Reference Frames, until the next Index frame (I-Frame) refreshes the scene.
- (b) If the entire scene is moving as with a PTZ camera, or if there is substantial movement within the scene of a fixed camera, then the Reference Frames can be at or near the same size as the I-Frame.
- (c) Because reference frames only measure the changes that have occurred since the Index frame, temporally coded video bitrates are not easily predictable in case of rapidly changing scenes.

Contrary, Motion JPEG (MJPEG) also known as Images per second coding or frame-per-frame compression techniques deliver a predictable file size making it easier to predict bandwidth. MJPEG2000 and latest MJPEG CODECs adopted DWT instead of Discrete Cosine Transform (DCT), thus blocking effect can be avoided [4]. Furthermore latest MJPEG features new functions such as progressive image transmission by quality or resolution, lossy and lossless compressions, region of interest (ROI) encoding, and good error resilience [9]. Due to these specific advantages, specific applications such as PTZ (Pan-Tilt-Zoom) cameras and IP based surveillance cameras still deploy MJPEG CODECs [4]. In their paper, Dragomir El Mezeni et al [5], describe in details JPEG-XR architecture presented in fig 1. Similarly, Christian Perra [6] suggested two architectures of MJPEG2000 and MPEG-XR, both of which deploy DWT at transformation stage. In this paper we only focus on developing a flexible hardware module for HDWT that can be integrated in Hardware-Software designs. The rest of the paper is organised as follows: Part I gives background and necessity for this work. In Parts II and III, we used Haar function to develop orthogonal Haar basis. In part IV we develop orthogonal basis which form the coefficients of a 16-element array after transformation. In Part V, we show the show image decomposition capability of HDWT. VI presents the design methodology. In part VII, by using

VHDL and synthesizing tools we simulate our design and present different views of the module designed. Place and routing is done on cyclone-IV FPGA module. In Part VIII, we present our Conclusive remarks.

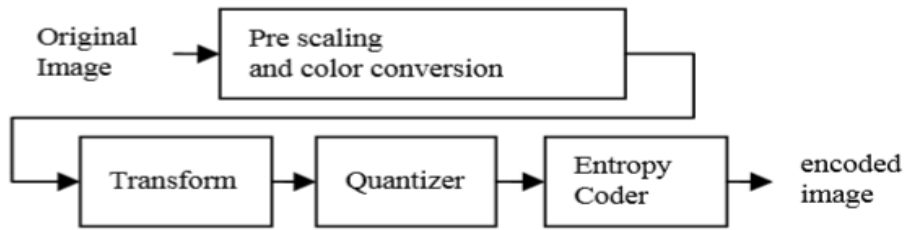


Figure1: JPEG XR coding flow, encoder side [6]

## II. Discrete Wavelet Transforms

Wavelets are mathematical tools for hierarchically decomposing functions [2]. They allow a function to be described in terms of a coarse overall shape, plus details that range from broad to narrow [2]. Regardless of whether the function of interest is an image, a curve, or a surface, Wavelets offer an elegant technique for representing the levels of detail present [2]. For any Image of NxN dimensions, each of its rows and columns can be represented as a linear combination of averaged and detailed coefficients (matrix entries) A. Jensen and A. la [3] presents thoroughly Ripples in Mathematics, Discrete Wavelet Transform and their hierarchical decompositions. The whole process is done by prediction and updating the new coefficients [3].

## III. Haar Function, Its Scaling And Shifting Nature

Haar function has existed for decades and John E. Shore [11] in his concise paper presented the best concept on Haar function. In this paper we exploit the flexibility and simplicity of the scaling functions in equation (1) and (2), to generate orthogonal basis for Haar wavelets in a multi-resolution analysis (MRA) window.

$$\psi(x) = \begin{cases} 1, 0 \leq x \leq 1/2 \\ -1, 1/2 \leq x \leq 1 \\ 0, elsewhere \end{cases} \dots\dots\dots(1)$$

$$\varphi_{i,j} = \begin{cases} 1, 0 \leq x \leq 1 \\ 0, elsewhere \end{cases} \dots\dots\dots(2)$$

Using equations (1), (2) by scaling and shifting, MATLAB™ simulations, generates Haar wavelet MRA (Multi-resolution Analysis) graphical representation in figure 1. They represent orthogonal basis.

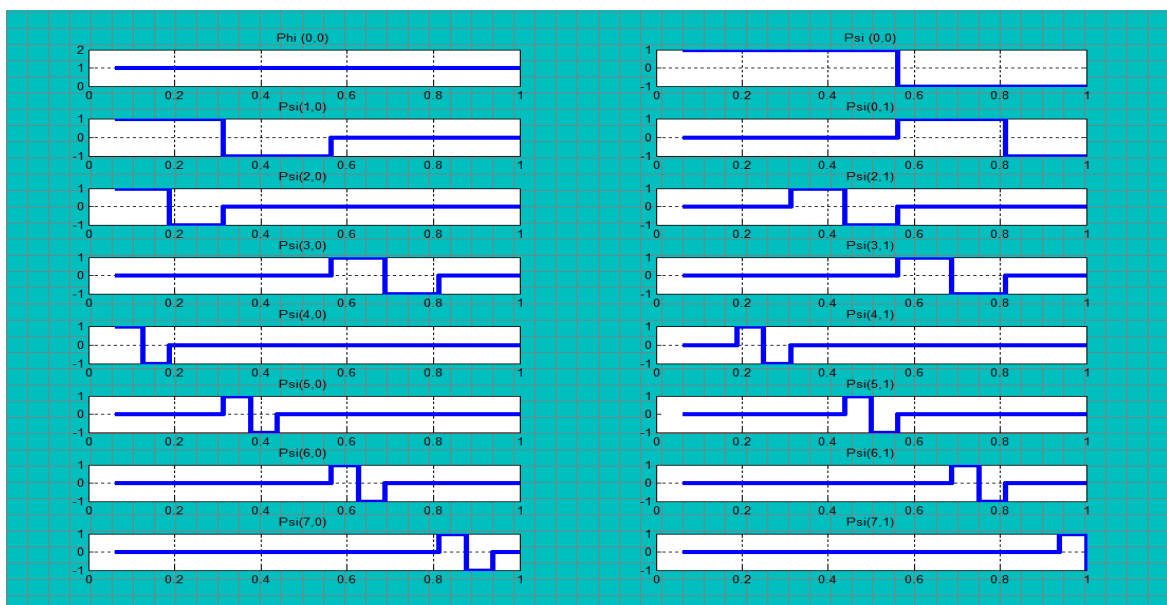


Figure 2: Simulation of Scaling and shifting of Haar function in an MRA window

### IV. Haar Wavelet Transform Function And Basis

Images can be represented in form of array vectors to store respective rows and columns of an NxM resolution. In situations where N=M, the whole image is stored in a square matrix representation. If we represent a row or a column of image is a vector  $I \in R^{16} = [S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_{10} S_{11} S_{12} S_{13} S_{14} S_{15} S_{16}]$ . We can represent  $I(x)$  in terms of Haar wavelet [13] coefficients by equation (3)

$$I(x) = \sum_{i,j \in Z} \langle I, \psi_{i,j} \rangle \psi_{i,j} \dots \dots \dots (3)$$

It then follows that, using equations (1) and (2),  $I \in R^{16}$  can be represented by equation (4)

$$I(x) = I_1\phi_{0,0} + I_2\psi_{0,0} + I_3\psi_{1,0} + I_4\psi_{1,1} + I_5\phi_{2,0} + I_6\psi_{2,1} + I_7\psi_{3,0} + I_8\psi_{3,1} \\ + I_9\psi_{4,0} + I_{10}\psi_{4,1} + I_{11}\psi_{5,0} + I_{12}\psi_{5,1} + I_{13}\psi_{6,0} + I_{14}\psi_{6,1} + I_{15}\psi_{7,0} + I_{16}\psi_{7,0} \dots \dots \dots (4)$$

Using the scaling functions and shifting operations, given in figure 2, we can obtain the wavelet components of  $I(x) \in R^{16}$ . These coefficients,  $I_1$  to  $I_{16}$  are calculated and presented in equations (5) to (20), extended from [13].

$$I_1 = \int_0^{1/16} s_1\phi_{0,0}(x)dx + \int_{1/16}^{2/16} s_2\phi_{0,0}(x)dx + \int_{2/16}^{3/16} s_3\phi_{0,0}(x)dx + \int_{3/16}^{4/16} s_4\phi_{0,0}(x)dx + \int_{4/16}^{5/16} s_5\phi_{0,0}(x)dx \\ + \int_{5/16}^{6/16} s_6\phi_{0,0}(x)dx + \int_{6/16}^{7/16} s_7\phi_{0,0}(x)dx + \int_{7/16}^{8/16} s_8\phi_{0,0}(x)dx + \int_{8/16}^{9/16} s_9\phi_{0,0}(x)dx + \int_{9/16}^{10/16} s_{10}\phi_{0,0}(x)dx \\ + \int_{10/16}^{11/16} s_{11}\phi_{0,0}(x)dx + \int_{11/16}^{12/16} s_{12}\phi_{0,0}(x)dx + \int_{12/16}^{13/16} s_{13}\phi_{0,0}(x)dx + \int_{13/16}^{14/16} s_{14}\phi_{0,0}(x)dx + \int_{14/16}^{15/16} s_{15}\phi_{0,0}(x)dx \\ + \int_{15/16}^{16/16} s_{16}\phi_{0,0}(x)dx = \frac{s_1+s_2+s_3+s_4+s_5+s_6+s_7+s_8+s_9+s_{10}+s_{11}+s_{12}+s_{13}+s_{14}+s_{15}+s_{16}}{16} \dots \dots \dots (5)$$

$$I_2 = \int_0^{1/16} s_1\psi_{0,0}(x)dx + \int_{1/16}^{2/16} s_2\psi_{0,0}(x)dx + \int_{2/16}^{3/16} s_3\psi_{0,0}(x)dx + \int_{3/16}^{4/16} s_4\psi_{0,0}(x)dx + \int_{4/16}^{5/16} s_5\psi_{0,0}(x)dx \\ + \int_{5/16}^{6/16} s_6\psi_{0,0}(x)dx + \int_{6/16}^{7/16} s_7\psi_{0,0}(x)dx + \int_{7/16}^{8/16} s_8\psi_{0,0}(x)dx + \int_{8/16}^{9/16} s_9\psi_{0,0}(x)dx + \int_{9/16}^{10/16} s_{10}\psi_{0,0}(x)dx \\ + \int_{10/16}^{11/16} s_{11}\psi_{0,0}(x)dx + \int_{11/16}^{12/16} s_{12}\psi_{0,0}(x)dx + \int_{12/16}^{13/16} s_{13}\psi_{0,0}(x)dx + \int_{13/16}^{14/16} s_{14}\psi_{0,0}(x)dx + \int_{14/16}^{15/16} s_{15}\psi_{0,0}(x)dx \\ + \int_{15/16}^{16/16} s_{16}\psi_{0,0}(x)dx = \frac{s_1+s_2+s_3+s_4+s_5+s_6+s_7+s_8-s_9-s_{10}-s_{11}-s_{12}-s_{13}-s_{14}-s_{15}-s_{16}}{16} \dots \dots \dots (6)$$

$$I_3 = \int_0^{1/16} s_1\psi_{1,0}(x)dx + \int_{1/16}^{2/16} s_2\psi_{1,0}(x)dx + \int_{2/16}^{3/16} s_3\psi_{1,0}(x)dx + \int_{3/16}^{4/16} s_4\psi_{1,0}(x)dx + \int_{4/16}^{5/16} s_5\psi_{1,0}(x)dx \\ + \int_{5/16}^{6/16} s_6\psi_{1,0}(x)dx + \int_{6/16}^{7/16} s_7\psi_{1,0}(x)dx + \int_{7/16}^{8/16} s_8\psi_{1,0}(x)dx + \int_{8/16}^{9/16} s_9\psi_{1,0}(x)dx + \int_{9/16}^{10/16} s_{10}\psi_{1,0}(x)dx \\ + \int_{10/16}^{11/16} s_{11}\psi_{1,0}(x)dx + \int_{11/16}^{12/16} s_{12}\psi_{1,0}(x)dx + \int_{12/16}^{13/16} s_{13}\psi_{1,0}(x)dx + \int_{13/16}^{14/16} s_{14}\psi_{1,0}(x)dx + \int_{14/16}^{15/16} s_{15}\psi_{1,0}(x)dx \\ + \int_{15/16}^{16/16} s_{16}\psi_{1,0}(x)dx = \frac{s_1+s_2+s_3+s_4-s_5-s_6-s_7-s_8}{16} \dots \dots \dots (7)$$

$$\begin{aligned}
 I_4 = & \int_0^{1/16} s_1 \psi_{1,1}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{1,1}(x)(x) dx + \int_{2/16}^{3/16} s_3 \psi_{1,1}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{1,1}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{1,1}(x) dx \\
 & \int_{5/16}^{6/16} s_6 \psi_{1,1}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{1,1}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{1,1}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{1,1}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{1,1}(x) dx + \\
 & \int_{10/16}^{11/16} s_{11} \psi_{1,1}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{1,1}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{1,1}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{1,1}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{1,1}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{1,1}(x) dx = \frac{s_9 + s_{10} + s_{11} + s_{12} - s_{13} - s_{14} - s_{15} - s_{16}}{16} \dots \dots \dots (8)
 \end{aligned}$$

$$\begin{aligned}
 I_5 = & \int_0^{1/16} s_1 \psi_{2,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{2,0}(x)(x) dx + \int_{2/16}^{3/16} s_3 \psi_{2,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{2,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{2,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{2,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{2,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{2,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{2,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{2,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{2,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{2,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{2,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{2,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{2,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{2,0}(x) dx = \frac{s_1 + s_2 - s_3 - s_4}{16} \dots \dots \dots (9)
 \end{aligned}$$

$$\begin{aligned}
 I_6 = & \int_0^{1/16} s_1 \psi_{2,1}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{2,1}(x)(x) dx + \int_{2/16}^{3/16} s_3 \psi_{2,1}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{2,1}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{2,1}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{2,1}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{2,1}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{2,1}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{2,1}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{2,1}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{2,1}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{2,1}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{2,1}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{2,1}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{2,1}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{2,1}(x) dx = \frac{s_5 + s_6 - s_7 - s_8}{16} \dots \dots \dots (10)
 \end{aligned}$$

$$\begin{aligned}
 I_7 = & \int_0^{1/16} s_1 \psi_{3,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{3,0}(x)(x) dx + \int_{2/16}^{3/16} s_3 \psi_{3,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{3,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{3,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{3,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{3,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{3,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{3,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{3,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{3,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{3,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{3,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{3,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{3,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{3,0}(x) dx = \frac{s_9 + s_{10} - s_{11} - s_{12}}{16} \dots \dots \dots (11)
 \end{aligned}$$

$$\begin{aligned}
 I_8 = & \int_0^{1/16} s_1 \psi_{3,1}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{3,1}(x)(x) dx + \int_{2/16}^{3/16} s_3 \psi_{3,1}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{3,1}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{3,1}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{3,1}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{3,1}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{3,1}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{3,1}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{3,1}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{3,1}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{3,1}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{3,1}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{3,1}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{3,1}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{3,1}(x) dx = \frac{s_{13} - s_{14} - s_{15} - s_{16}}{16} \dots \dots \dots (12)
 \end{aligned}$$

$$\begin{aligned}
 I_9 = & \int_0^{1/16} s_1 \psi_{4,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{4,0}(x) dx + \int_{2/16}^{3/16} s_3 \psi_{4,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{4,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{4,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{4,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{4,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{4,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{4,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{4,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{4,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{4,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{4,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{4,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{4,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{4,0}(x) dx = \frac{s_1 - s_2}{16} \dots \dots \dots (13)
 \end{aligned}$$

$$\begin{aligned}
 S_{10} = & \int_0^{1/16} s_1 \psi_{4,1}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{4,1}(x) dx + \int_{2/16}^{3/16} s_3 \psi_{4,1}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{4,1}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{4,1}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{4,1}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{4,1}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{4,1}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{4,1}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{4,1}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{4,1}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{4,1}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{4,1}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{4,1}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{4,1}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{4,1}(x) dx = \frac{s_3 - s_4}{16} \dots \dots \dots (14)
 \end{aligned}$$

$$\begin{aligned}
 I_{11} = & \int_0^{1/16} s_1 \psi_{5,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{5,0}(x) dx + \int_{2/16}^{3/16} s_3 \psi_{5,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{5,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{5,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{5,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{5,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{5,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{5,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{5,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{5,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{5,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{5,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{5,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{5,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{5,0}(x) dx = \frac{s_5 - s_6}{16} \dots \dots \dots (15)
 \end{aligned}$$

$$\begin{aligned}
 I_{12} = & \int_0^{1/16} s_1 \psi_{4,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{4,0}(x) dx + \int_{2/16}^{3/16} s_3 \psi_{4,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{4,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{4,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{4,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{4,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{4,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{4,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{4,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{4,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{4,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{4,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{4,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{4,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{4,0}(x) dx = \frac{s_7 - s_8}{16} \dots \dots \dots (16)
 \end{aligned}$$

$$\begin{aligned}
 I_{13} = & \int_0^{1/16} s_1 \psi_{4,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{4,0}(x) dx + \int_{2/16}^{3/16} s_3 \psi_{4,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{4,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{4,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{4,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{4,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{4,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{4,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{4,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{4,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{4,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{4,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{4,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{4,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{4,0}(x) dx = \frac{s_9 - s_{10}}{16} \dots \dots \dots (17)
 \end{aligned}$$

$$\begin{aligned}
 I_{14} = & \int_0^{1/16} s_1 \psi_{4,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{4,0}(x) dx + \int_{2/16}^{3/16} s_3 \psi_{4,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{4,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{4,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{4,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{4,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{4,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{4,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{4,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{4,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{4,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{4,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{4,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{4,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{4,0}(x) dx = \frac{s_{11} - s_{12}}{16} \dots \dots \dots (18)
 \end{aligned}$$

$$\begin{aligned}
 I_{15} = & \int_0^{1/16} s_1 \psi_{4,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{4,0}(x) dx + \int_{2/16}^{3/16} s_3 \psi_{4,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{4,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{4,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{4,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{4,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{4,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{4,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{4,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{4,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{4,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{4,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{4,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{4,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{4,0}(x) dx = \frac{s_{13} - s_{14}}{16} \dots \dots \dots (19)
 \end{aligned}$$

$$\begin{aligned}
 I_{16} = & \int_0^{1/16} s_1 \psi_{4,0}(x) dx + \int_{1/16}^{2/16} s_2 \psi_{4,0}(x) dx + \int_{2/16}^{3/16} s_3 \psi_{4,0}(x) dx + \int_{3/16}^{4/16} s_4 \psi_{4,0}(x) dx + \int_{4/16}^{5/16} s_5 \psi_{4,0}(x) dx \\
 & + \int_{5/16}^{6/16} s_6 \psi_{4,0}(x) dx + \int_{6/16}^{7/16} s_7 \psi_{4,0}(x) dx + \int_{7/16}^{8/16} s_8 \psi_{4,0}(x) dx + \int_{8/16}^{9/16} s_9 \psi_{4,0}(x) dx + \int_{9/16}^{10/16} s_{10} \psi_{4,0}(x) dx \\
 & + \int_{10/16}^{11/16} s_{11} \psi_{4,0}(x) dx + \int_{11/16}^{12/16} s_{12} \psi_{4,0}(x) dx + \int_{12/16}^{13/16} s_{13} \psi_{4,0}(x) dx + \int_{13/16}^{14/16} s_{14} \psi_{4,0}(x) dx + \int_{14/16}^{15/16} s_{15} \psi_{4,0}(x) dx \\
 & + \int_{15/16}^{16/16} s_{16} \psi_{4,0}(x) dx = \frac{s_{15} - s_{16}}{16} \dots \dots \dots (20)
 \end{aligned}$$

### V. Image Decomposition BY Haar Dwt

One of the powerful features of wavelet transformation is that image can be decomposed and compacted into energy levels. Using MATLAB™ software simulation, we present Image decomposition in both vertical hierarchies in figure 3 and in square energy compaction levels in figure 4. Horizontal, vertical and diagonal details can be seen along decomposition. Details coefficients help in image synthesis. With this energy compaction, it is easier for quantization process that comes next to transformation in most of Multimedia CODECS

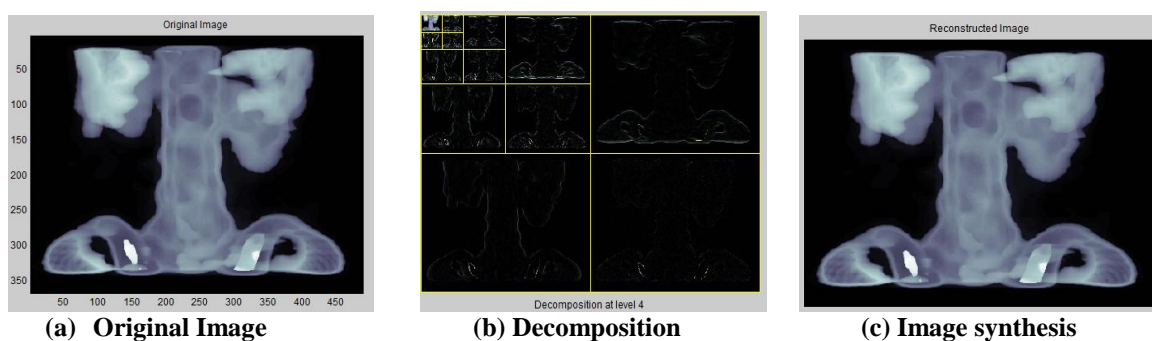
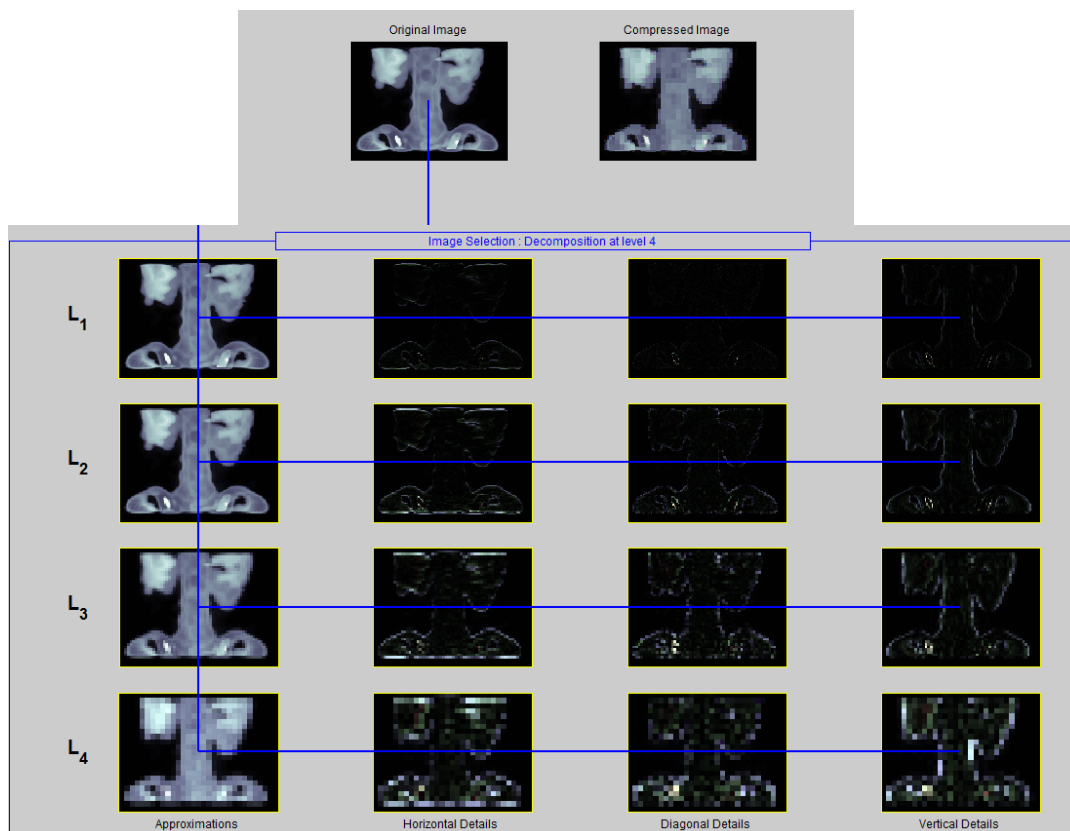


Figure 4: Spine decomposition at four levels

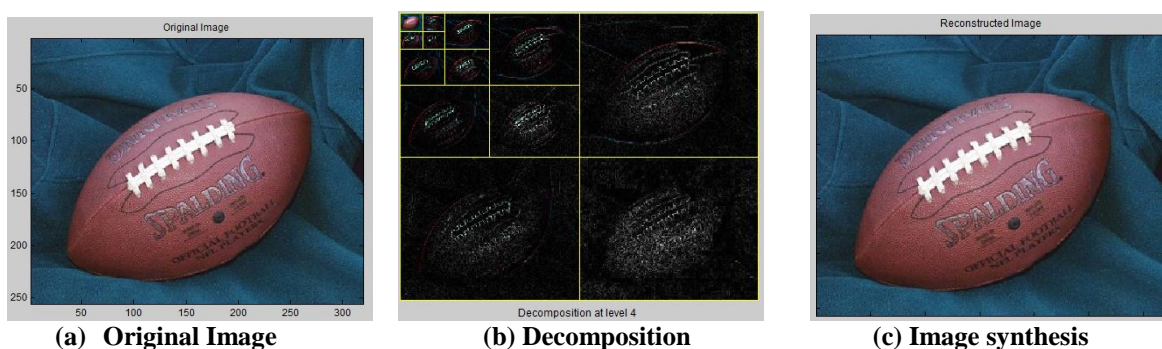


Figure 5: Foot ball decomposition at four levels

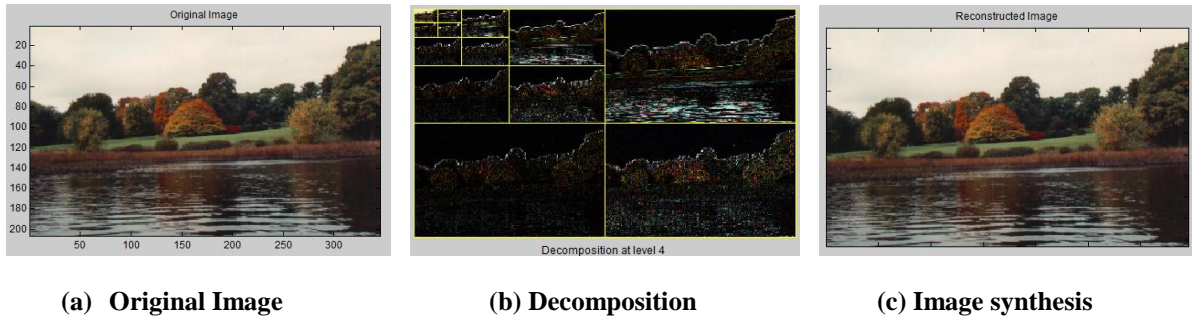


Figure 6: Autumn decomposition at four levels

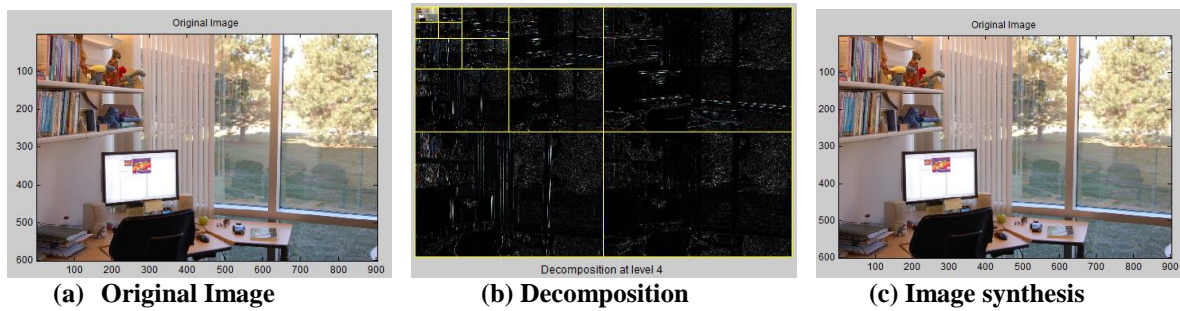


Figure 7: Office decomposition at four levels

### VI. Hardware Design For Relaxed Haar Discrete Wavelet Module

Different approaches to DWT hardware implementation have been published. The main approach used is prediction and updating [1] but the memory required during respective intermediate stages is immense. Different papers also have shown lifting and techniques which involves Filters. Coefficients involved in this approach makes precision problem. In this paper we revisit the conventional DCT, FFT and DST basic processing element [10] of two inputs and two outputs in figure 8. The upper output involves addition only while the lower involves subtraction. This simple structure becomes handy in the whole design process. By using equations (5) to (20) and by re-arrange the inputs starting with even elements( $S_0, S_2, S_4, \dots$ ) of a 16-element array and then odd elements( $S_1, S_3, S_5, \dots$ ), in section IV, an elegant, simple and flexible structure for hardware implementation is presented in figure 9. The dotted paths in figure 9 indicate signal routes to output but no operation involved. Instead of multiplication matrices, we use bit shifting operations in hardware description languages (HDLs).

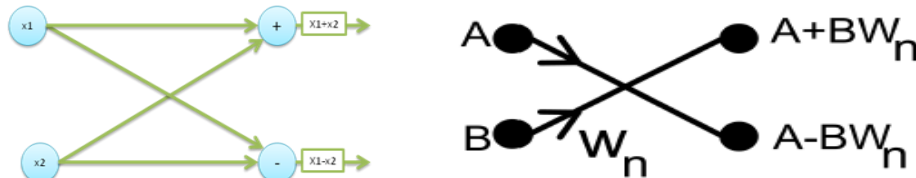


Figure 8 : Basic processing unit [10]



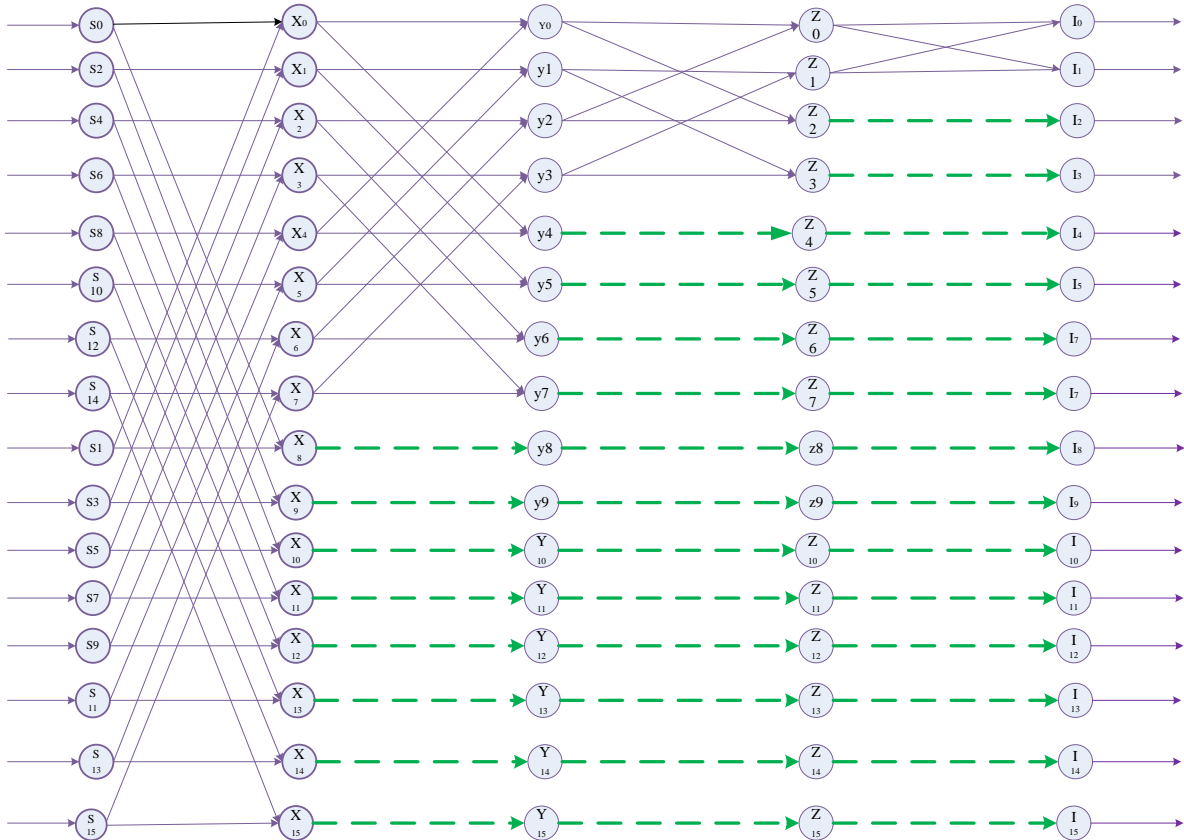


Figure 9: Butterfly structure for HDWT designed from equations (5) to (20)

**VII. Design SYNTHESIS, Placing AND Routing**

Using Hardware description language (HDL), structure presented in figure, we used ModelSim® and Quartus software to realize the design presented. Figure 10 shows ModelSim® wire connectivity and binary simulations involve using different test vectors. Similarly figure 11 shows RTL view with the help of Quartus® software after place, routing as and synthesis. It should be noted that the input and output have equal bit length (8 bits). This is because during shifting process (dividing by 16) we shifted rightwards by 4bit position. The output became 8bits instead of 12 bits after four successful stages that involved addition and subtraction.

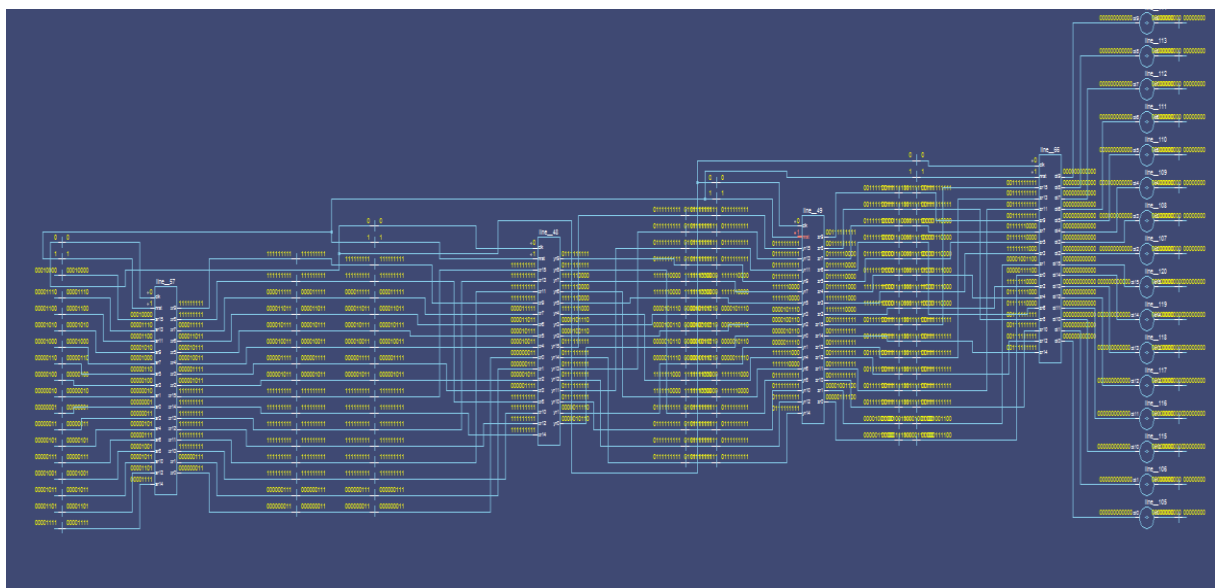


Figure 10: Dataflow tracing using ModelSim®

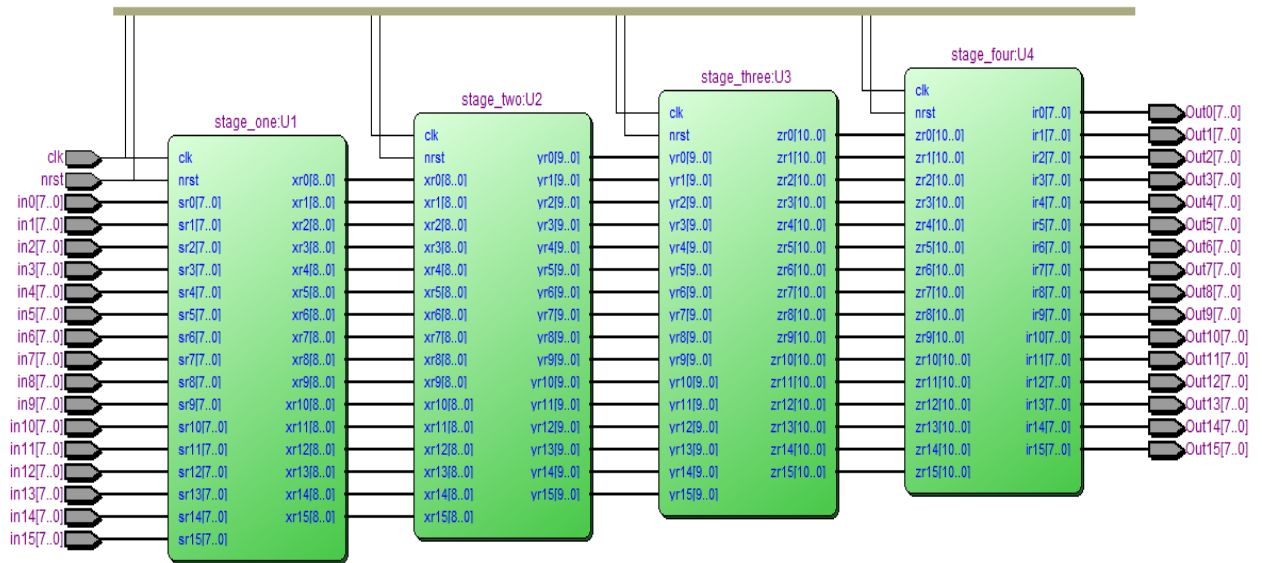


Figure 11: Register Transfer Level (RTL) view using Quartus®

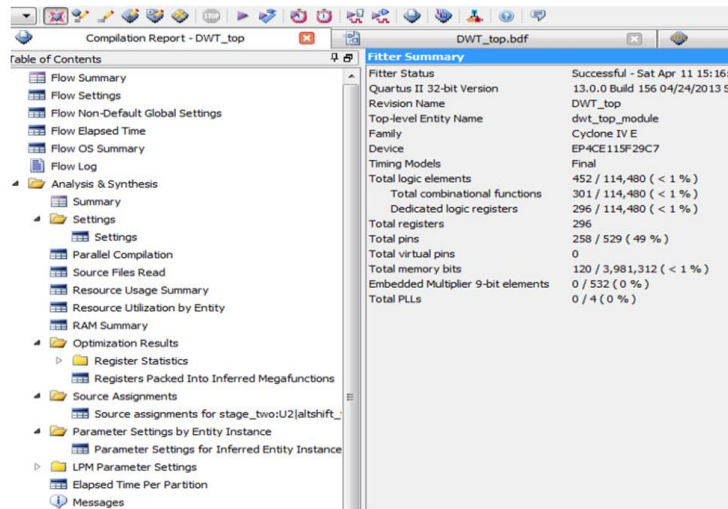


Figure 12: Fitter summary with no multiplication involved

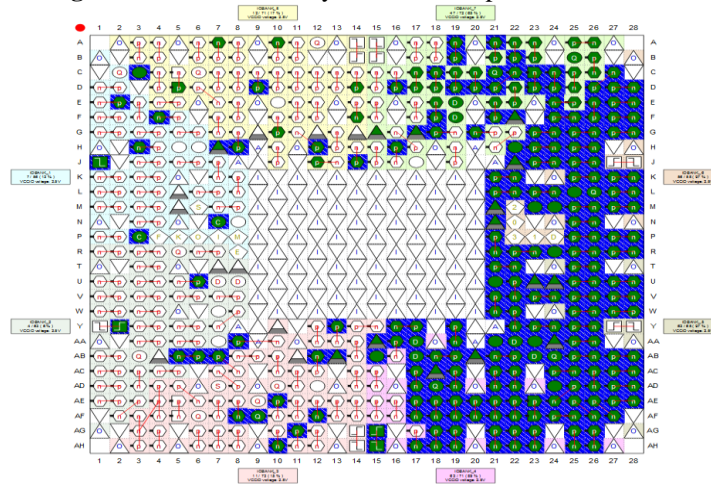


Figure 13: Cyclone IV E-EPCE115F29C7 wired bond after place and routing to the FPGA

### VIII. Conclusion

In this paper, we explored the basic principles of Haar Function and exploited the simplicity of Haar wavelet transformation to construct a butterfly circuit structure with zero multiplication operation. With VHDL, we simulated, placed and routed our design to FPGA. We implemented the design on Cyclone IV EPCE115F29C7 module. Using this approach, CODECs using DWT at their signal transformation stage, can integrate such module into their Hardware-Software system with ease.

### References

- [1]. Chih-Hsien Hsia, et al, Memory-efficient architecture of 2-D lifting-based discrete wavelet transform, Journal of the Chinese Institute of Engineers, Published online: 28 Jun 2011.
- [2]. Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics: A primer, part 1. *IEEE Computer Graphics and Applications*, 15(3):76–84, May 1995.
- [3]. A. Jensen and A. la Cour-Harbo, *Ripples in Mathematics, the Discrete Wavelet Transform* Springer-Verlag 2001.
- [4]. Jason Spielfogel, *Why we like MJPEG compression*, <http://www.securityinfowatch.com/article/10561410/why-we-like-mjpeg-compression> [Accessed on 4<sup>th</sup> may, 2015].
- [5]. Dragomir El Mezeni et al , JPEG-XR encoder implementation on a heterogeneous multiprocessor system, 5th European Conference on Circuits and Systems for Communications (ECCSC'10), November 23–25, 2010, Belgrade, Serbia
- [6]. Cristian Perra , Re-encoding JPEG images for smart phone applications, 21st Telecommunications forum TELFOR 2013, IEE, 2013
- [7]. Koichi Hattori, Hiroshi Tsutsui et al, A High-Throughput Pipelined Architecture for JPEG XR Encoding, IEEE Conference Publications, IEEE Conference Publications, 2009. .DOI: [10.1109/ESTMED.2009.5336818](https://doi.org/10.1109/ESTMED.2009.5336818)
- [8]. Harish Yagain, Srinivas Donapati, Addressing the Interoperability Issues While Using JPEG-XR , 2011 International Symposium on Electronic System Design
- [9]. C. A. Christopoulos T. Ebrahimi and A. N. Skodras, JPEG2000: The New Still Picture Compression Standard, Media Lab, Ericsson Research, Ericsson Radio Systems AB, S-16480 Stockholm, Sweden
- [10]. Saini, S. ; Mahajan, A. ; Mandalika, Implementation of Low Power FFT Structure using a Method Based on Conditionally Coded Blocks, IEEE Conference Publications 2010,
- [11]. JOHN E. SHORE, On the Application of Haar Functions , Concise Papers , IEEE TRANSACTIONS ON COMMUNICATIONS, MARCH 1973
- [12]. Iain Richardson, The new standard for video coding released by ISO MPEG and ITU-T VCEG <http://www.vcodex.com/h265.html> [accessed may 6<sup>th</sup> 2015]
- [13]. Zunera Idrees, Eliza Hashemiaghjekandi, Image Compression by Using Haar Wavelet Transform and Singular Value Decomposition, School of computer Science, Physics and mathematics Linnaeus University.