Hardware Implementation of Multilevel Two Dimensional Haar Wavelet Transform Using FPGA

Khamenees Khalaf Hasan\textsuperscript{1,2*}, Yaroub Ghazi Jasim\textsuperscript{1,2}, Mohammed Rasheed Salih\textsuperscript{1,2}

\textsuperscript{1}Salahaddin Province, Tikrit, Iraq, P.O. Box (45)
\textsuperscript{2}Department of Electrical Engineering, Tikrit University, Salahaddin, Iraq

\*Corresponding author: kaljomaily@tu.edu.iq

Abstract. Focusing on the intensive computations involved in the discrete wavelet transform (DWT), the design of efficient hardware architectures for a fast computation of the transform has become imperative, especially for real-time applications. To constrain the complexities of the design, a basic linear algebra approach is used to denote the signal flow graph of forward DWT (FDWT) and inverse DWT (IDWT) architectures. Based on this context, the DWT was selected along with the theorized Haar function being the mother wavelet, as the main analytical method for this study. The proposed FDWT and its IDWT hardware architecture filter generated similar results compared to the MATLAB model for the seven levels of DWT decomposition. Simulations were performed using grayscale images of different sizes to validate the proposed design and attain speed performance appropriate for a number of real-time applications. The proposed FDWT filter produced 700 slices of hardware logic and register element area, which comprises less than 2% of the Altera DE2 development board Cyclone II Field Programmable Gate Array (FPGA) hardware area.

1. Introduction
Since the discovery of DWT by Mallat, several architectures for DWT processor have been proposed [1]. The DWT operation determines the volume of computations in successive resolution levels. In addition to providing, a significant reduction in memory uses and computational complexity, the wavelet coefficients calculated by Haar wavelet are identical to those computed by convolution filter-bank. Consequently, the number of multiplications and additions compared to the filter-bank approach are reduced, resulting in a more efficient use of power and chip area. Its modular structure is well suitable for hardware implementation [2].

A Haar wavelet is the simplest type of wavelet [3]. The Haar wavelet function is one of the original types of wavelet which is still in use because of its theoretical simplicity, precise reversible rapid computation and memory efficiency, given it can be calculated in place without applying a temporary array [4]. Haar wavelets are derivatives of a mathematical operation referred to as the Haar transform (HT). This operation explains the wavelet decomposition of each signal into two components: average and difference [5]. The averaging and differencing decomposition is a very valuable method although the calculations can rapidly become somewhat cumbersome for large matrix sizes [6]. The HWT can be implemented via two approaches, i.e. matrixes multiply method and linear algebra equations. The linear algebra method offers an output similar to the matrixes multiply method, but obtains fewer resources. The calculations inherent in this transformation are compatible with energy efficient hardware implementation as it entails a reversible sequence of additions, subtractions and divisions [7].
This paper embarks on designing a fast hardware compatible and resource-efficient architecture for the computation of 1-D and 2-D DWTs. This paper provides a detailed overview of an efficient HAAR wavelet transform HWT architecture for the FDWT in addition to IDWT. This paper is organized as follows. Section two describes the related works. Section three explains the design of FPGA wavelet architecture. Section four discuss the performance results and the conclusions are made in section five.

2. Material and Experimental Procedures
The architecture proposed in this study is a direct RC architecture, comprising a forward DWT (FDWT) and its inverse (IDWT), typified by variable levels of transformation were discussed in this subsection, the. Each one of these two modules is designed as hierarchical scheme that applies 1D-DWT processing module two times to represent 2-D DWT processing [8], [9]. The module can be repeatedly applied on the same image for multilevel processing [10], [11]. The small size of the hardware is maintained by means of an iterative single computing 1D-DWT module for higher level computations and reversing to external memory between levels. A single 1D-DWT filter is incorporated into the RC architecture implementations to lower hardware costs.

Unlike to MATLAB, VHDL codes does not support many built in functions such as the convolution. Consequently, while implementing the algorithm in VHDL; linear equations of FDWT and IDWT are used. Like all wavelet transforms, the HWT decomposes a discrete signal into two subsignals of half its length. One subsignal is a running average or trend; the other subsignal is a running difference or fluctuation (Walker, 2008). The FDWT module which consists of adder and right shifter is used to obtain the low-pass average and high-pass difference components. The first current and next pixel samples are given to the adder then the output of the adder is right shifted by one to give (division by two) of the average wavelet components. The difference component is calculated by subtracting the current and next pixel values with a shift to right operation (Stromme and McGregor, 1997; Al Muhit et al., 2004):

Data converter reads the data from external memory array and restores it back in to an external memory array module using Matlab programs that convert the input grayscale format image into a hexadecimal format file prior to saving it in the memory. These data are afterwards utilized as input into the memory module, thereby producing wavelet coefficients text data which is in hexadecimal (ASCII text) form by dwt entity. The hexadecimal format text file is transformed again using Matlab programs into grayscale format image files which are employed at the output stages in order to examine the content of the memory files as shown in Figure 1. Simulation was implemented for various image sizes and transformation levels for each size. The test images in the grayscale format which are initially saved in the memory module are deprived of its image header information to produce a raw text file. Memory read and writes processes are use text file command.
3. Results and Discussion

Following the design completion of every VHDL module, each one is required to be assembled independently for precision analysis and debugging. The VHDL modeling for HWT has been analyzed, synthesized, and simulated using the same Altera Development board DE2, CycloneII FPGA family and chipset EP2C35F672C6N which has been used previously. The Quartus II software tool (V 9.1 SP2 web edition) has built in tools for performing these operations. The VHDL model for HWT is tested for justifying the performance of FDWT and IDWT.

From the first 8 selected DWT sequence of samples of Lena image, it is supposed that the memory stores couple of coefficients, low coefficient L0=A1 at address 0 and high coefficient H0=A2 at address 1 and so on, as illustrated in Figure 2. The pair of coefficients L0=A1, H0=A2 is formed through IDWT filtering of one level of decomposition HWT of the 256×256 Lena test image version. Given that, input coefficients are presently stored in the external memory module, L0 and H0 can be stored in their addresses 0 and 1.

Figure 1. FDWT system block diagram

Figure 2. Waveform indicating DWT results of memory module

Similarly, all pairs of the resultant wavelet coefficients L(i) and H(i), can be saved at addresses, 2i and 2i+1, in that order as shown in Figure 3.
The intermediate $256 \times 256$ wavelet HEX coefficients data are stored in the external memory and then the dump signal is activated which signifies the end of the transformation process. The 2-D-DWT module generate ready signal to the outside environment indicating the end of first level of decomposition process after 989565 clock cycles required for coding of $256 \times 256$ pixels through FDWT filtering as shown in Figure 4 as well as later in the Table 1, illustrating the number of clock cycle usage for 2-D DWT coding process of HWT.

Although the significant increase in the number of cycles used for the 2-D DWT coding process results in higher image size versions, as Table 1 shows, the clock cycle usage is also increased with the rise in the levels of decomposition and it ranges from 64323 clock cycles at $(N=64, L=1)$, to 5264303 clock cycles at $(N=512, L=7)$, in the 2-D DWT coding process of HWT as illustrated in Table 1.
| Level | Clock cycles | 64×64 pixels | 128×128 pixels | 256×256 pixels | 512×512 pixels |
|-------|--------------|--------------|---------------|---------------|---------------|
| Level 1 | 64323 | 249725 | 989565 | 3943805 |
| Level 2 | 80128 | 312448 | 1237888 | 4931968 |
| Level 3 | 84291 | 328649 | 1300611 | 5180291 |
| Level 4 | 85414 | 332812 | 1316614 | 5243014 |
| Level 5 | 85737 | 333935 | 1320777 | 5259017 |
| Level 6 | 85840 | 334258 | 1321900 | 5263180 |
| Level 7 | - | 334361 | 1322223 | 5264303 |
| **Frequency** | **131.18 MHz** | **129.25 MHz** | **130.89 MHz** | **111.35 MHz** |

After performing synthesis and other verification processes, a Register Transfer Level (RTL) simulation of FDWT Module has been achieved. The RTL or Technology Map helps to check the design visually. The functionality of the individual hardware components comprising the 2-D -DWT chipset was tested using ModelSim-Altera 6.5b. Test benches are often used to provide an automated testing environment wrapper around the design. The VHDL behavioral report can be created by means of synthesis, which implies it can be translated into physically attainable circuits, using Computer Aided Design tools. A simulation flow summary report is generated after performing this compilation. The performance of this proposed arithmetic unit has shown significant improvements in the total number of computations and given less equivalent gate counts as shown in in Figure 5. The proposed FDWT filter produced 700 slices (LES), which comprises less than 2 % of the hardware area.

![Simulation flow summary report](image_url)

**Figure 5.** A simulation flow summary report for four FDWT versions using Lena test image; a- 256×256 pixels version, b- 512× 512 pixels version,
Figure 6 shows the synthesis report with synopsis of main hardware device utilization in terms of registers and total logic elements LEs by number of LUT inputs and the Model-sim simulation result.

After synthesis and place-and-route are completed, the Power Analyzer is used for full information on the design power estimates. Figure 7 shows the RTL schematic of the proposed 2-D DWT architecture with different blocks and their interconnections. The total number of logic elements LEs used determines hardware size.
4. Conclusions
The objective of this study has been to develop a scheme for the design of hardware resource-efficient and high-speed architectures for the computation of the 1-D and 2-D DWT. Efficient low-power consumption architecture for multilevel decomposition FDWT and IDWT are outlined. A distinct 1D-DWT filter is integrated into the proposed linear algebra HWT based coding by means of multilevel RC architecture implementation to lessen hardware use. The RC is performed by applying the forward 1D DWT in both horizontal and vertical directions of the image, for a chosen amount of levels. Multiple levels of computations can be implemented in this architecture, with intermediate results between levels saved in the external memory. The proposed architectures have been examined using diverse images, different levels of decompositions and clock frequencies. The results showed that the amount of occupied LEs is relatively less than LEs in the target device.

5. References

[1] Mallat, S. G. 1989. A theory for multiresolution signal decomposition: the wavelet representation. IEEE transactions on pattern analysis and machine intelligence, 11, (7), 674-693.

[2] Al-Azawi, S. 2018. Low-Power, Low-Area Multi-level 2-D Discrete Wavelet Transform Architecture. Circuits, Systems, and Signal Processing, 37, (1), 444-458.

[3] Walker, J. S. 2008. A primer on wavelets and their scientific applications, CRC press.

[4] Talukder, K. H. & Harada, K. 2010. Haar wavelet based approach for image compression and quality assessment of compressed image, International Journal of Applied Mathematics, 36, (1), pp.1-9.

[5] Phang, C. & Phang, P. 2008. Modified fast and exact algorithm for fast haar transform, International Journal of Computer Science and Engineering, 2, pp. 55-58.

[6] Al Muhit, A., Islam, M. S. & Othman, M. 2004. VLSI implementation of discrete wavelet transform (DWT) for image compression. Proceedings of the 2nd International Conference on Autonomous Robots and Agents, ICARA, Palmerston North, New Zealand, pp. 13-15.

[7] Stromme, O. & McGregor, D. R. 1997. Study of wavelet decompositions for image/video compression by software codecs, Sixth International Conference on Image Processing and Its Applications, 1, pp. 61-63.

[8] Hasan, K. K., Ngah, U. K., & Salleh, M. F. M. 2013. Multilevel decomposition Discrete Wavelet Transform for hardware image compression architectures applications. In Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on (pp. 315-320). IEEE.

[9] Hasan, K. K., Ngah, U. K., & Salleh, M. F. M. 2013. Low complexity image compression architecture based on lifting wavelet transform and embedded hierarchical structures. In Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on (pp. 305-309). IEEE.

[10] Hasan, K., Ngah, U. K., & Salleh, M. 2014. Hierarchical hardware architecture of discrete wavelet transform for image compression. Int. J. Comput. Sci. Inf. Technol. Res, 2, (4), 232-241.

[11] Hasan, K. K., & Abdul–Amir, S. O. 2009. DCT/DPCM Hybrid Coding For Interlaced Image Compression. Tikrit Journal of Engineering Sciences, 16, (1), 121-132.

Acknowledgments
The authors would like to gratefully acknowledge Tikrit University, Engineering college for their support to work on this paper.