Design and Implementation of Lifting Wavelet Transform Using Field Programmable Gate Arrays

T B Taha¹, R Ngadiran² and P L Ehkan²
¹IT Department, Faculty of Sciences, Tishk International University, Erbil, Iraq
²School of Computer and Communication Engineering, University Malaysia Perlis (UniMAP), Perlis, Malaysia

phd.taha@gmail.com

Abstract. Lifting Wavelet transform (LWT) has an extensive usage in different image processing applications as image compression and information hiding. LWT is considered a good solution for hardware designs as it relies only on integer calculations while applying the wavelet transform. In this paper, an FPGA design and implementation of LWT is presented, the implementation is achieved using VHDL coding without importing off-shelf components which make the proposed design applicable to a wide range of devices. The design is based on parallel execution to perform LWT implementation with real time response. The design utilized 421 logic registers of DE2 Cyclone II (EP2C35F672C6) FPGA device with 151.91MHz frequency.

1. Introduction
Lifting Wavelet Transform (LWT), was first introduced by Sweldens is considered to be the second generation of wavelets. The main difference that distinguishes LWT from classic wavelet construction is that Fourier transform is not used in the signal transform calculations [1]. In compare to conventional DWT structure, LWT has better computational efficiency in terms of using a lower number of mathematical operations, which results in a smaller implementation area, less power consumption, and lower design complexity. Due to its “in place calculation” where the produced coefficients replaces the old image pixels without extra memory utilization, and its integer-to-integer transformation abilities, LWT can be easily implemented by hardware [2].

The implementation of LWT in hardware is necessary for different applications that need the transformation for implementing digital circuits in real time manner, as image compression and perceptual mapping. The hardware implementation of digital circuits is required because it accelerates the execution speed from hours to several minutes or seconds [3]. However, in order to achieve high speed and real time execution, hardware design needs to be concentrate since the resources as logical components and frequency are limited. Generally to apply digital circuits on hardware, different technologies had been used as application specific integrated circuits (ASIC) and field programmable gate arrays (FPGAs). Cost reduction played a major role in the widespread use of FPGAs since there was a reduction of 100% in the cost of building blocks from 1990 to 2003 [4]. At the same time, the continuous increase in the cost of application specific integrated circuits ASICs compared to advances in semiconductor manufacturing made the FPGA a better alternative for embedded systems, as the cost of making incremental changes to FPGA designs is negligible when compared to the large expenses involved in the re-manufacturing of ASICs [5][6]. FPGA is also considered as one a successful
technology for developing systems whose require real time operation, as it has parallel execution nature where several processes can be executed concurrently which produce instance response [7].

Accordingly, the implementation of wavelet transform using FPGA devices will lead to excel the execution of different wavelet-based image processing applications.

A three-level DWT was implemented by Karmani, Djemal & Tourki [8] for a watermarking scheme for images and videos. A pipelined 2D-scan-based architecture was used. The DWT was also used by Darji, Lad, Merchant & Chandorkar [9] by proposing a pipelined blind FPGA-based watermark approach using a quantization method. A greyscale image was used as the host image and a binary image was used as the watermark. In this study, not all the steps of the embedding process were implemented using FPGA, since the DWT was performed using Matlab.

DWT which was used in mentioned studies and several other studies as [10-11], has higher computational complexity than LWT, hence its FPGA implementation used to be resource and time consuming. The LWT was utilized instead, as the model by [12], where the analysis the computational features of 2D DWT algorithm and propose a parallel architecture called Fragment-based Interleaving Dual Pipelines (FIDP). Another lifting based implementation is achieved by [2], in which the 5/3 lifting-based wavelet transform is modeled and simulated using Matlab.

From literature, it’s found that there is a need to design wavelet transform using simple calculations, i.e LWT, and the implementation of LWT needs to be designed and simulated codes without importing off-shelf components, to make the design applicable to wide range of applications and devices. In this paper, an optimized LWT implementation is presented using FPGA devices, and all the implementation is achieved using only VHDL codes without relying on Matlab and other external functions or applications.

The paper is organized as follow, in section two a brief description of LWT equations is presented. Section three describes the methodology of LWT implementation on 2D image. The implementation results are shown in section 4. The paper is concluded in section 5.

2. Lifting Wavelet Transform
The filter bank method is used to apply the Discrete Wavelet Transform (DWT) on digital signals is depend on high pass filter and low pass filers to create the detail and approximation bands respectively as shown in figure 1.

![Figure 1. Wavelet transform using filters.](image)

The equivalent processes in LWT is the usage of three steps, Splitting, Prediction and Update. In the Split phase, the signal is divided into odd and even samples. To apply the prediction, which is equivalent to the high pass filter (h) in the filter banks, equation (1) is used, which is the difference between the odd element \(X_{2i+1}\) and the average of its two even neighbors \(X_{2i}, X_{2i+2}\). The output will then be the detail band (Di).
\[ D_i = X_{2i+1} - \frac{(X_{2i} + X_{2i+2})}{2} \] (1)

For the update, which is equivalent to the low pass filter \( \hat{g} \) in the filter banks, equation (2) is used. In each update application, the value of the even sample is added to the summation of the previously calculated detail coefficient and the recently created one, and the result is divided by four. The output will be the approximation band \( S_i \).

\[ S_i = X_{2i} + \frac{(D_i + D_{i-1})}{4} \] (2)

For two-dimensional signals as digital images, the LWT is applied on rows first followed by columns application. The implementation of the equation above on FPGA device is described in the next section.

3. FPGA Implementations

In this section, a detailed description of the design flow of LWT implementation is presented. As shown in Figure 2, input signal \( X \) is fed to the LWT and the produced output is approximation band \( S_n \) and detail band \( D_n \). The implementation will be described in details for one dimensional signal in the following sub-section followed by two-dimensional implementation.

![Figure 2. Block diagram of LWT system](image)

A one-dimensional signal of 16 elements is considered as a test input to the design. Values of test signal input a \( X= 6, 5, 1, 9, 5, 11, 4, 3, 5, 0, 6, 4, 9, 6, 5, 7 \) that are taken from memory initialization file (mif) that is loaded to M4K RAM which consist of 8 locations each of 16 bits. Each pixel is represented by 8 bits, since each grey scale level pixel has value between 0 to 255. Test signal is shown in figure 3

![Figure 3. Simulation results of one dimensional LWT application](image)

According to Equations (1) and (2), to obtain one output of each of D (Detail) and S (Approximation) band, three elements from the input are required. Practically, three new elements are required in the first clock followed by two new elements and one from previous reading in subsequent clocks. For instance,
the first data inputs according to X signal in the above example must be, 6,5,1 followed by 1,9,5, 5,11,4, and so on.

Three values will be carried by three signals, and implemented using three registers, Reg1sig, Reg2sig, and Reg3sig as shown in figure 4, which depicts the RTL (Register Transfer Level) of the first part of design that responsible of reading the pixels in the appropriate sequence and deliver them to the LWT logic. Registers' values will be fed to LWT procedure for calculating D and S coefficients. LWT, as integer-based transformation, is one clock process because all the LWT-related calculations are performed by variables within a process. The RTL of LWT implementation of LWT is shown in Figure 5.

![Figure 4. RTL for one level LWT](image)

*Figure 4. RTL for one level LWT*

*(Part1: input registers initialization)*

![Figure 5. RTL for one level LWT](image)

*Figure 5. RTL for one level LWT*

*Figure 5. RTL for one level LWT*

*(Part2: LWT implementation on input data)*

In two dimensional signals, further LWT implementation is applied again on each of detailed band D, and approximation band S individually as the same way it was applied on original sequence. S and D are further divided into S1, D1 and S2, D2, respectively. This operation is applied on digital images by considering them as two-dimensional arrays. In the process of applying LWT on an image of size (n
× n), each row is considered as one-dimensional signal, the result is detail and approximation bands with width n/2 and length n. Another LWT operation is applied on each produced band by column wise to obtain four n/2 × n/2 bands as shown in figure 6.

![Figure 6. LWT bands.](image)

In order speed up the calculations, concurrent transpose processes were initiated for both bands at the same time, another parallel computation was achieved by creating S1, D1 from S concurrently with creating S2, D2 from D. Figure 7 shows two dimensional LWT process where operations on the same column were processed simultaneously.

![Figure 7. Parallel architecture of LWT decomposition.](image)

4. Implementations Results
A 16×16 image was used for applying LWT. Image pixels was read row by row as shown in figure 8. After S and D band had been produced, the next LWT operation is applied column by column, hence, a transpose on S and D was needed. VHDL simulation results for the four bands are shown in figure 9.

![Figure 8. Reading image pixels](image)

![Figure 9. Produced LWT coefficients for different bands using VHDL](image)
The LWT was implemented on Altera DE2 Cyclone II (EP2C35F672C6) FPGA device. The LWT implementation utilized 538 logic elements at maximum frequency of 151.91MHz. A total of 421 logic register was used for the 16×16 image, it had better logic utilization than the model proposed by Bao-feng & Yong [12] where 466 logic registers were utilized for 2×2 image, in addition, the implementation is done using VHDL without relying on off-shelf components; hence, the implementation results can be considered in the practical implementations and in different platforms.

5. Conclusions
In this paper, an implementation of LWT is achieved on FPGA using VHDL. The algorithm is based on parallel processing for higher execution speed. The implementation consumed 421 logic register from DE2 Cyclone II (EP2C35F672C6) FPGA device with maximum frequency 151.91MHz for 16 × 16 sample image. The utilization and the execution speed denoted to the ability of perform the transform using the proposed implementation efficiently and in real time. Different LWT-based operations can be implemented based on the proposed methods, as compression, watermarking and perceptual mapping.

References
[1] Sweldens, W. Lifting scheme: a new philosophy in biorthogonal wavelet constructions. In Wavelet applications in signal and image processing III International Society for Optics and Photonics 2569 68-80.
[2] Gholipour, M. Design and implementation of lifting based integer wavelet transform for image compression applications International Conference on Digital Information and Communication Technology and Its Applications pp. 161-172.
[3] Wang, X., Qin, Q., & Cheng, Y. Design and implementation of digital image watermark based on FPGA. Recent Advances in Computer Science and Information Engineering pp. 223-229.
[4] Rodriguez-Andina, J. J., Moure, M. J., & Valdes, M. D. Features, design tools, and application domains of FPGAs IEEE Transactions on Industrial Electronics 54 1810-1823.
[5] M. Thompson, "FPGAs Accelerate Time to Market for Industrial Designs," EE Times (July 2, 2004). Accessed on 9-May-2019 from http://www.us.design-reuse.com/articles/8190/fpgas-accelerate-time-to-market-for-industrial-designs.html.
[6] National Instruments. (2012). Introduction to FPGA Technology: Top 5 Benefits. Accessed on 9- May, 2019, from http://www.ni.com/white-paper/6984/en/#toc2.
[7] Sulaiman, N., Obaid, Z. A., Marhaban, M. H., & Hamidon, M. N. Design and implementation of FPGA-based systems-a review Australian Journal of Basic and Applied Sciences 3 3575-3596.
[8] Karmani, S., Djemal, R., & Tourki, R. Computer Standards & Interfaces Efficient hardware architecture of 2D-scan-based wavelet watermarking for image and video. Computer Standards & Interfaces, 31 801–811.
[9] Darji, A. D., Lad, T. C., Merchant, S. N., & Chandorkar A. N. Watermarking hardware based on wavelet coefficients quantization method. Circuits Systems, and Signal Processing 32 2559–2579.
[10] Joshi, A. M., Darji, A., & Mishra, V. Design and implementation of real-time image watermarking 2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC) pp. 1-5.
[11] Karthigaikumar, P., Anumol, & Baskaran, K. FPGA implementation of high speed low area DWT based invisible image watermarking algorithm. Procedia Engineering 30 266–273.
[12] Li, B. F., & Dou, Y. Fidp: A novel architecture for lifting-based 2D DWT in JPEG2000 International Conference on Multimedia Modeling pp. 373-382.