We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Ultra-High Performance and Low-Cost Architecture of Discrete Wavelet Transforms

Mouhamad Chehaitly, Mohamed Tabaa, Fabrice Monteiro, Safa Saadaoui and Abbas Dandache

Abstract

This work targets the challenging issue to produce high throughput and low-cost configurable architecture of Discrete wavelet transforms (DWT). More specifically, it proposes a new hardware architecture of the first and second generation of DWT using a modified multi-resolution tree. This approach is based on serializations and interleaving of data between different stages. The designed architecture is massively parallelized and sharing hardware between low-pass and high-pass filters in the wavelet transformation algorithm. Consequently, to process data in high speed and decrease hardware usage. The different steps of the post/pre-synthesis configurable algorithm are detailed in this paper. A modulization in VHDL at RTL level and implementation of the designed architecture on FPGA technology in a NexysVideo board (Artix 7 FPGA) are done in this work, where the performance, the configurability and the generic of our architecture are highly enhanced. The implementation results indicate that our proposed architectures provide a very high-speed data processing with low needed resources. As an example, with the parameters depth order equal 2, filter order equal 2, order quantization equal 5 and a parallel degree P = 16, we reach a bit rate around 3160 Mega samples per second with low used of logic elements (≈400) and logic registers (≈700).

Keywords: Mallat binary tree algorithm, DWPT, IDWPT, lifting scheme wavelet, FIR filter, parallel-pipeline architecture, VHDL-RTL modeling, FPGA

1. Introduction

We notice in the last year a wide usage of wavelet transform theory in different domain like telecommunications, image and video processing, data compression, optical fiber, encryption and others. But these domains are evolved extremely which require a new wavelet transform architecture with low cost target technology that can provide a high-speed data processing and low power consummation. In parallel FPGA technology is massively blossomed to come very popular and to be a target technology of many applications, in particular of Discrete Wavelet Packet Transform (DWPT).

Although there are tons of research elsewhere, the talking of efficient hardware implementation of wavelet transform is still a complex mission and depend directly on the target application. Where in each application, there is a compromise between the different constraints: processing speed, implementation cost, and power consumption.

1.1 Related works

Since 1980, the crucial date of the born of "Wavelet Transform (WT)" with its founder J. Morlet, we found many works describe the hardware implementations of wavelet transforms. We note that the first work was done by Vishwanath Denk and Parhi [1], the authors propose an orthonormal DWT architecture combine a digit-serial processing technique with a lattice structure of quadratic mirror filter (QMF). After that, Vishwanath [2] and Motra [3], describe an efficient hardware implementation for DWT and Inverse DWT (IDWT). In 2001, Hatem et al. [4] worked in the reducing of the number of multipliers in the filters structure in a mixed parallel/sequential DWT architecture.

Wu and Hu [5] describe an implementation of DWPT/IDWPT in a strategy to minimize the number of multipliers and adders in symmetric filters using Embedded Instruction Codes (EIC). In other way to improve the data processing of DWT, Jing and Bin [6] implement the architecture on FPGA based on advanced distributed arithmetic (IDA), while Wu and Wang [7] used a multi-stage pipeline structure, although Palero et al. [8] work on the implementation of two-dimensional DWT architecture. Also, Hu and Jong [9] present two-dimensional DWT based on lifting scheme architecture that ensure a high throughput data processing.

Based on lifting scheme architecture, Fatemi and Bolouki [10] describe a pipeline and programmable DWPT architecture. Other important work, to optimize the hardware complexity of DWT based on coextensive distributive computation developed by Sowmya and Mathew [11]. Paya et al. [12] used a classical recursive pyramid algorithm (RPA) and polyphase decomposition to develop a new architecture for IDWPT based on the lifting scheme. Acharya [13] developed a systolic architecture for both DWPT/IDWPT with a fixed number of requirement pages. Farahani and Eshghi [14] described a new DWPT implementation based on a word-serial pipeline architecture and on parallel FIR filters banks. Sarah et al. [15] presented a convolution block suitable for DWT decomposition. Radhakrishnana and Themozhib [16] developed a new DWT architecture by using XOR-MUX adders and Truncations multipliers instead of the conventional adders and multipliers. Taha et al. [17] developed a parallel execution to perform Lifting Wavelet transform implementation with real time, while Shaaban Ibraheem et al. [18] presented a high throughput parallel DWT hardware architecture based on pipelined parallel processing of direct memory access (DMA).

Also, we have to notice that we found recently some orientation to software approach to compute DWPT/IDWPT on parallel processes to increase the data processing speed with optimization of the distributed computation. But the problem is still the required computing resources (concurrent network processors or processor cores) while the energy consumption is one of the critical criteria in most application domains, for that we do not include it in our bibliography.

1.2 Wavelet theory

In the previous work, we present a detailed review of the wavelet theory. Where we focus here on:

- the discrete wavelet packed transform which know as first generation of Discrete Wavelet Transform based on Mallat algorithm [19]

- the lifting scheme approach which know as first generation of Discrete Wavelet Transform based on

1.2.1 Review of DWPT and IDWPT

From the definition of wavelet theory, the DWPT and IDWPT of a signal $x[n]$ is set of approximation coefficients and detailed coefficients based on Mallat algorithm (or Mallat tree) and using FIR bank filter and inversely.

Based on Mallat the DWPT transform can be presented like decomposition, as shown in **Figure 1**.

Where the input signal is presented by the coefficients $D_0^0[k]$ in level zero with data sampling D_{in}. This amount of data (input signal) will be decomposed into two part:

i. High frequency signal presented by approximation coefficient $D_1^1[k]$ with half data sampling of original signal $(D_{in}/2)$ by using low-pass filter $h(n)$ and down sampling by a factor of two.

ii. Low frequency signal presented by detailed coefficient $D_1^0[k]$ with half data sampling of original signal $(D_{in}/2)$ by using high-pass filter $g(n)$ and down sampling by a factor of two.

Then the data path will be following the same processing in the next level with the same filters' characteristics. The depth in Mallat tree algorithm is equal to the number level and describe the needed filters equal to 2^{level} in each level. In general, the corresponding approximation and detailed coefficients in different levels in **Figure 1** are calculated as follows:

$$D_l^{2i}(k) = \sum_n h(n) D_{l-1}^i(2k - n) \tag{1}$$

$$D_l^{2i+1}(k) = \sum_n g(n) D_{l-1}^i(2k - n) \tag{2}$$

Where l presents the level and $i = 0, 1, \ldots, \left(2^{(l-1)} - 1\right)$.

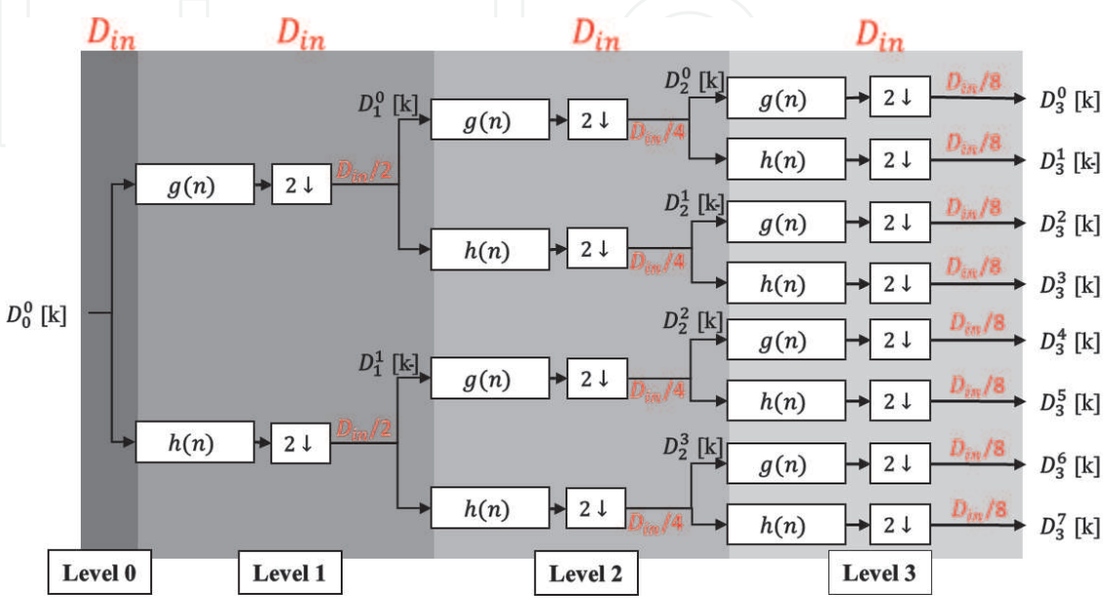


Figure 1.
DWPT three level transform based on Mallat algorithm.

As proposed Mallat in [19], the corresponding transfer functions of $h(n)$ and $g(n)$ are derived in the following equations:

$$H(Z) = H_0 + H_1 z^{-1} + H_2 z^{-2} + \ldots + H_{L-1} z^{-(L-1)} \quad (3)$$

$$G(Z) = G_0 + G_1 z^{-1} + G_2 z^{-2} + \ldots + G_{L-1} z^{-(L-1)} \quad (4)$$

where z^{-1} indicates the delay for one ***samplingperiod*** and L is the order of filters depends on the used mother wavelet.

In inverse way, the reconstruction of signal or IDWPT without loss of information is possible based on two important properties of wavelets: admissibility and regularity. Similar to decomposition way, the reconstruction operation is following an iterative method and the corresponding coefficients in different levels are calculated as follows:

$$D_l^i(k) = \sum_n \overline{h}(n) D_{l+1}^{2i}(2k - n) + \sum_n \overline{g}(n) D_{l+1}^{2i}(2k - n) \quad (5)$$

For example, the reconstruction of signal in three level based always on Mallat algorithm is presented in **Figure 2**.

Where $\overline{h}(n)$ and $\overline{g}(n)$ are the conjugated low-pass and high-pass of $h(n)$ and $g(n)$. Mallat used the quadratic mirror filter (QMF) of corresponding transfer functions $H, G, \overline{H}$ and $\overline{G}$ to ensure the perfect reconstruction of the original signal.

1.2.2 Review of lifting scheme discrete wavelet transform

Based on the wavelet theory, we can consider that the lifting wavelet theory is the second generation of DWPT. The strategy in this generation is to reduce the impact of the high pass and the low pass filters by replacing it into a sequence of smaller filters: update filters and predict filters. Therefore, the convolution

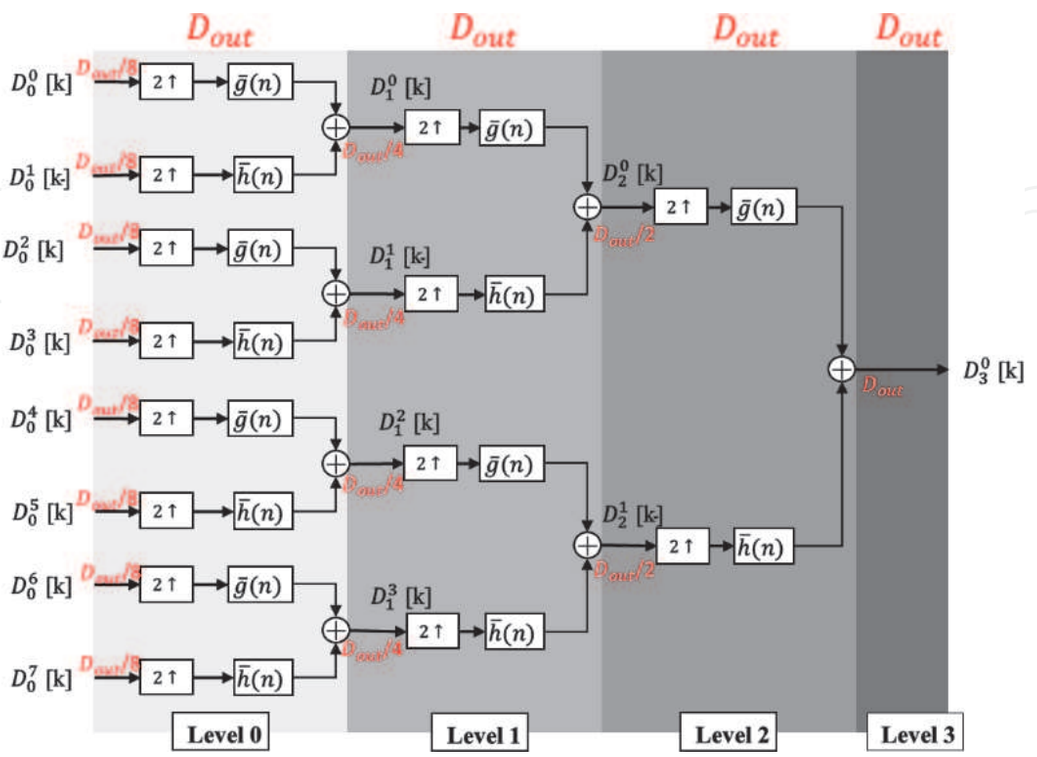


Figure 2.
IDWPT three level transform based on Mallat algorithm.

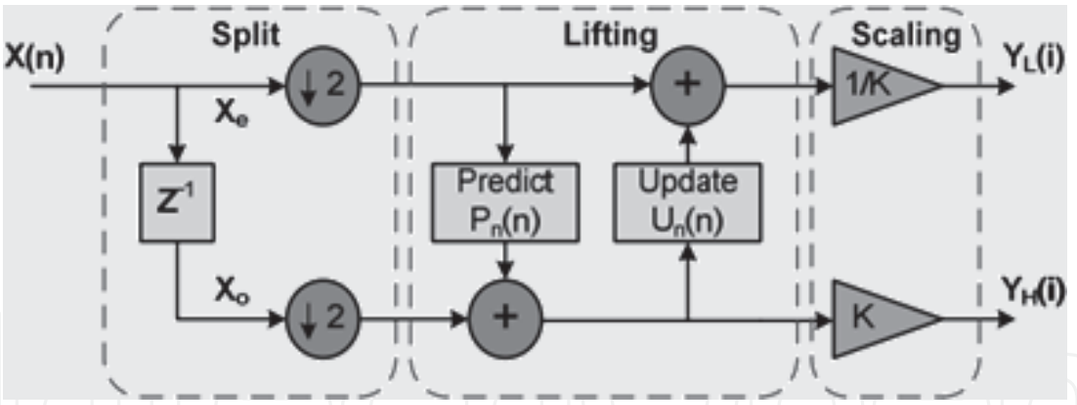


Figure 3.
Kernel of the lifting wavelet transform.

computations are reduced by comparison to the first generation which naturally reduce the design complexity by maintaining the same quality and speed.

By definition, the lifting wavelet transform is dividing to three steps: Split, Lifting, and Scaling, as shown in **Figure 3**.

In the split steps, the input signal ***X(n)*** will be divided into two sub sequences odd and even. The obtained sub-signal will be modified in lifting steps, by using alternating prediction and updating filters. And finally, a scaling operation is used to obtain an approximated and detailed signal.

1.3 Contributions and work organization

In this work, our goal is to develop a high performance, low cost implementation and configurable new hardware architecture of discrete wavelet transform based on Mallat algorithm [19]: first generation (based on Discrete Packet Wavelet Transform - DWPT) and second generation (lifting scheme Discrete Wavelet Transform) by exploitation of this suitable FPGAs environment. In order to provide the low hardware cost and the high processing speed by design, we develop a new generic parallel-pipeline architecture avoiding the complexity of the traditional architectures with the massif need for hardware resource by: i) intelligent sharing of hardware computing resources (multipliers and adders) among the different filters and stages, ii) design a linear architecture to limited impact of filter and wavelet order. To improve the high performance (data processing speed and hardware cost) of our proposal, we will perform different simulation function of selected wavelet family, transformation depth, filtering order and coefficient quantization. In VHDL at the RTL level, we modeled our architectures and we synthesized it using Altera Quartus Prime Lite, targeting an Intel/Altera Cyclone-V FPGA.

This work is organized as follows: in Section 2, we introduce our linear non-parallel and P-parallel architecture of first generation for both the DWPT and IDWPT along with simulation results. In Section 3, our linear non-parallel architecture for second generation based on lifting scheme is described. Finally, conclusion is given in Section 4.

2. Hardware implementation of first generation

2.1 DWPT

As shown in **Figure 1**, we notice that in a given stage ***k***, each filter proceeds the same amount of data and half data rate by comparison of filter in the adjacent level

$k-1$. The number of needed filters (low-pass and high-pass filters) in a given level k is 2^k. Furthermore, the amount of procced data in each level is the same.

So, the tree architecture of Mallat have a big regularity of the behavior of filters in different levels. Which leads us to develop an ultra high speed data processing with low hardware consumption (this constraint is critical in modern application that need high throughput with low power consumption). To achieve that we think to develop an evolving architecture by retransform the exponential tree to linear one, as shown in **Figure 4**.

A high throughput rate with lower hardware resources are provided in this architecture by linearization of classic Mallat tree and parallelization the used transposed FIR filter. To achieve our goal by minimizing the hardware consumption, we proposed a shared computational resource (multipliers and adders) between the low-pass and high pass filters as shown in **Figure 5**.

In this structure, we propose a modified transposed FIR filter corresponding to H/G blocks in **Figure 4**, this model is look like the serial FIR filter in the theory of FEC coding. The H/G blocks can process in parallel P inputs sampling (signals) and consequently P outputs sampling (signals) in each clock cycle and consequently the P-parallel DWPT (**Figure 4**) are able to transform P sampling in each clock cycle.

Furthermore, this architecture is suitable for all wavelet family where we need just to change the coefficients of high-pass and low-pass for each family. Where the data handling (filter coefficients or signal sampling) of the low-pass and high-pass filter between different stages is dedicated to specific block in our architecture; we called it "buffers block". The main role of "buffers block" is to interleaving data from stage to the next stage and to manage data between low-pass and high-pass filter in the same stage. Their structure is detailed in **Figure 6**.

To procced the same amount of data in the original Mallat binary tree b (of course multiplied by the degree of parallelize P), the buffer blocks should be working with this mechanism:

a. The parameter k describe the stage, change from 1 to max depth of wavelet transform. The parameter P present the degree of parallelism and must respect the dyadic rule, $P = 2^k,\ x\ \mathbb{N}^+$.

b. The structure of buffer block is based on the concept of manipulation of data speed transfer in register level, where we built up inside P sub-blocks, each block has two registers/buffers level speed: "Fast Buffer" and "Slow Buffer". On each clock cycle, the Fast Buffer take data from the output of previous stage and achieve P-shift. While, Slow Buffer sub-blocks are powered to take data from the Fast Buffer registers of the same stage and then achieve P-shift on two-clock cycles.

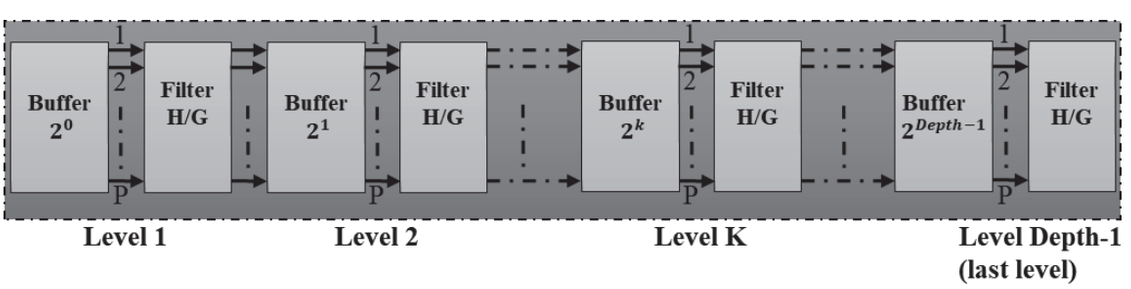


Figure 4.
Datapath diagram of linear and P-parallel proposed DWPT architecture.

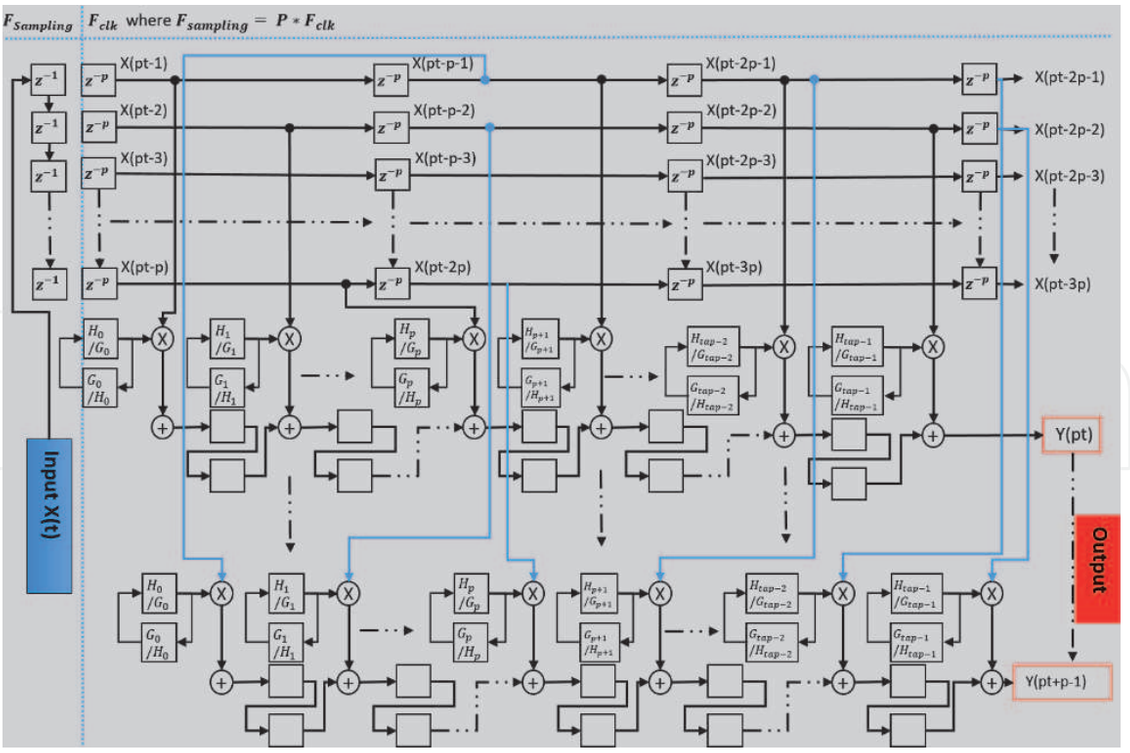


Figure 5.
P-parallel transposed FIR filter structure.

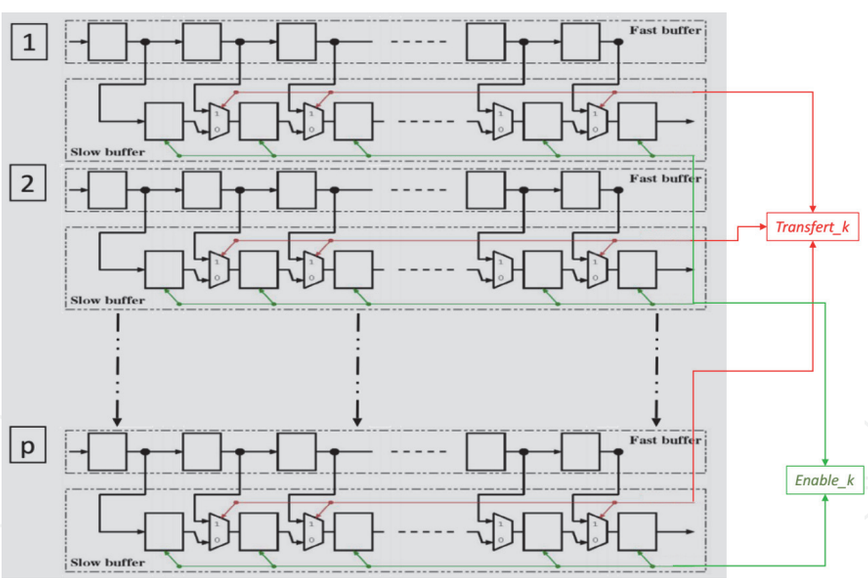


Figure 6.
General view of buffer block structure (in stage ***k****) of parallel DWPT architecture.*

c. The size Buffer blocks (number of fast registers and slow registers) depends on two parameters: the stage presented by ***k*** and the parallel degree presented by ***P*** .

d. To manage the data path between "Slow Buffer" and "Fast Buffer", we specified two control signal: "$enable_k$" and "$transfer_k$", where the $enable_k$ signal (in green) is dedicated to control the shift rate between the different registers in Slow Buffers sub-blocks and "$transfer_k$" signal (in red) is to

manage the data transfer from the fast buffer sub-block to the slow buffer sub-block. Technically these two control signals give the permission to transfer all data from the registers of the Fast Buffer to the Slow Buffer registers simultaneously in each 2^k clock cycle (in a given stage k).

The operation in the "**d**" stage combine the synchronization of data from stage k to stage $k-1$ and down sampling by factor **2** without using an extra memories or DSP block. Where the playing in the time between buffers, give us the possibility to procced only half data from the Fast Buffer to the Slow Buffer on every 2^k cycle. Furthermore, the slow speed of the Slow Buffer ensures the twice (to respect the concept proposed by Mallat algorithm) presented of leaving Slow Buffer data to the next stage.

To centralize the architecture, we developed a control block or control unit to manage all control signals in different stage, as shown in **Figure 7**.

2.2 IDWPT

As a reverse way of P-parallel DWPT transform, this section is dedicated to present our proposed model of P-parallel IDWPT.

As we mention in the section of P-parallel DWPT transform, the reconstruction process has also a big regularity, where in **Figure 2**, we notice that each filter proceeds the same amount of data and half data rate by comparison to filter in the adjacent level. The number of needed filters (low-pass and high-pass filters) in a given level K is 2^k. Furthermore, the amount of proceed data in each level is the same. This leads us to develop an ultra high speed data processing with low cost resources consummation.

We introduce the concept of linearize and serialize in our pipeline and P-parallel architecture to eliminate the impact of exponential evolution of the number of used filters. So, as shown in **Figure 8**, we develop a novel architecture.

In this architecture, in each stage we implement only one modified filter instead of using $P * 2^k/2$ low pass filters and $P * 2^k/2$ high pass filters. It is important to mention that the number modified transposed FIR filter bank increased linearly as a function of depth order which it was exponential in the classic architecture.

To achieve our goal by minimizing the hardware consumption, we develop a Blocks Filter $\overline{H}$ / $\overline{G}$ which is a modified reconstruction P-parallel FIR filters by shared computational resource (multipliers and adders) between the low-pass and high pass filters and with a similar structure of that present in **Figure 4**. The only

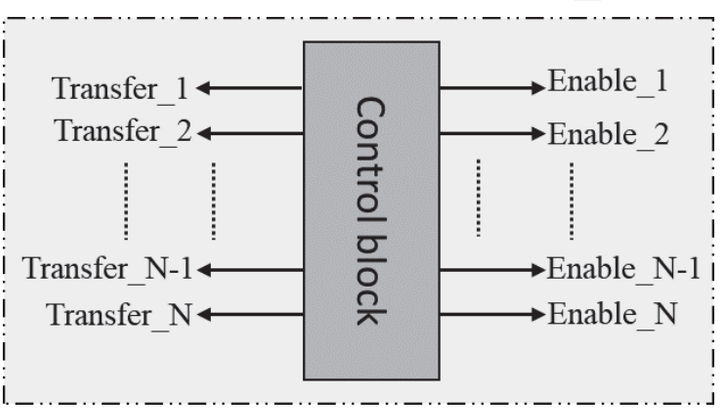


Figure 7.
Control block.

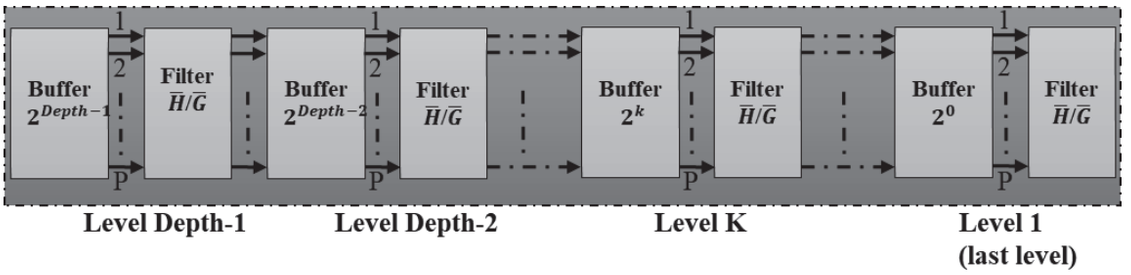


Figure 8.
Data path diagram of linear and P-parallel proposed IDWPT architecture.

difference is the coefficients filters values. Consequently, the P-parallel FIR filter is able to filter ***P*** sampling in each clock cycle.

The data manage and interleaving between filters from the first to the end stages is dedicated to the buffer block. Their structure is detailed in **Figure 9**.

To ensure the data management between reconstruction high-pass and low-pass filter, we play on the timing of buffer register: slow buffer and fast buffer. To procced the same amount of data in the original Mallat binary tree b (of course multiplied by the degree of parallelize ***P***), the buffer blocks should be working with this same mechanism as that used in the previous section.

The "fast buffer" achieve P-shift on each clock cycle while the "slow buffer" achieve P-shift on two-clock cycles. To manage the data follow path between "Slow Buffer" and "Fast Buffer", we specified two control signals: "$\boldsymbol{enable_k}$" and "$\boldsymbol{transfer_k}$". The "$\boldsymbol{enable_k}$" signal (in green) is dedicated to control the shift rate between the different registers in Slow Buffers sub-blocks and the "$\boldsymbol{transfer_k}$" signal (in red) is to manage the data transfer from the fast buffer sub-block to the

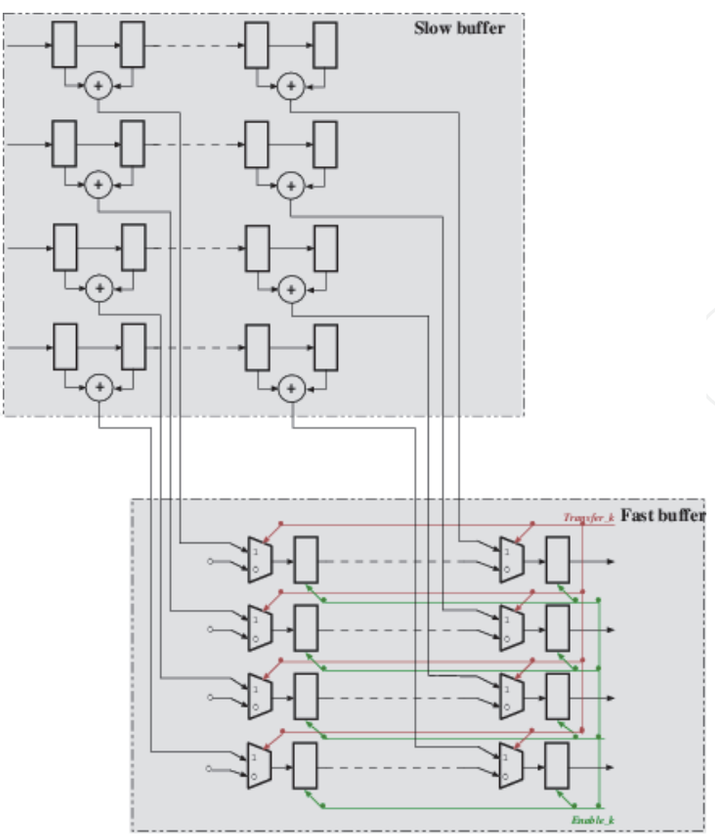


Figure 9.
General view of buffer block structure (in stage ***k*** *and degree of parallelization* $\boldsymbol{P = 4}$*) of parallel IDWPT architecture.*

slow buffer sub-block. Also, we used a control block unit to manage the control signals. The structure of control block is similar to that present in **Figure 7**.

2.3 Implementation results

Following our strategy, we develop a new pipeline and P-parallel architectures for DWPT and IDWPT. These architectures are full reconfigurable at synthesis. The reconfigurable parameters are the wavelet scale or the depth of DWPT and IDWPT, the filter coefficient and data quantization, the order of modified $\boldsymbol{H}/\boldsymbol{G}$ and $\overline{\boldsymbol{H}}/\overline{\boldsymbol{G}}$ filters, and the degree of parallelism.

Also, these architectures are partially reconfigurable after synthesis function the value of filters coefficients (that mean implicitly the order of filters). This feature, we give the possibility to work with different wavelet family without re-synthesis the FPGA carte where we load dynamically after synthesis the filter coefficients of the corresponding wavelet.

Our aim in this part is to study the performance of these architecture to record the impact of different parameters on:

- The speed of data process that mean the clock frequency (given in MHz) of implemented architecture. Where from the degree of parallelism and clock frequency, we can obtain the data sampling rate of our DWPT and IDWPT architectures.

- The hardware consumption, which represented the logic registers $\boldsymbol{l_r}$ and the logic elements $\boldsymbol{l_e}$.

In the following procedure of the implementation of our new architectures of DWPT and IDWPT transforms on the same FPGA split, we respect these constraints:

- These architectures (pipeline and P-parallel DWPT and IDWPT) are designed and modeled in VHDL at the RTL level.

- Theoretically, we do not have a limitation of parallelism degree but we should take into consideration the exist technology (hardware side) and the value must respect the dyadic rule, i. e. $\boldsymbol{P} = \boldsymbol{2^x}$, $\boldsymbol{x}$ $\mathbb{N}^+$.

- We used Altera Quartus software premium lite edition to synthesis our architectures and Intel/Altera Cyclone-V FPGA as a target technology with a speed grade of −7. For the real implementation, we used an FPGA board from Xilinx product called NexysVideo development board based on Artix-7 FPGA as a target technology.

2.3.1 Real implementation setup

To evaluate the proposed solution, a real implementation setup is depicted in **Figure 10**, where we used the UART connector to send and receive data from PC to NexysVideo board and inversely. Initial verification has been realized by sending the coefficients of Low-pass and High-pass filter after synthesis. Additional verification has been realized when received the reconstructed data.

The different simulations results are shown in **Tables 1–3**.

Based on the results in **Tables 1–3**, we observed that when we increase the quantization order from 5 to 16 this increases linearly the logic and element registers

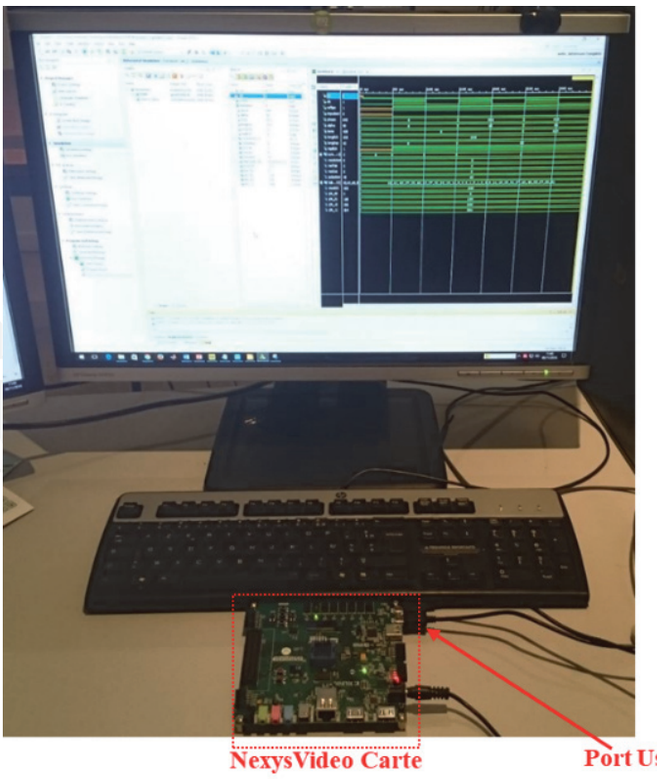


Figure 10.
Lab implementation setup.

Design parameters (Depth, Filter order, and Quantification)	Clock frequency (MHz)	Resources usage ($l_e l_r$)		
(2, 2, 5)	203.8	*205*	(471,296)	**(109, 186)**
(3, 2, 5)	200.21	*201.82*	(756,510)	**(166,312)**
(4, 2, 5)	197.37	*196.16*	(1204,899)	**(244,505)**
(2, 4, 5)	200.87	*152.88*	(879,456)	**(265,286)**
(3, 4, 5)	185.05	*152.58*	(1299,719)	**(379,442)**
(4, 4, 5)	193.71	*153.37*	(1941,1171)	**(483,665)**
(2, 16, 5)	189.2	*144.03*	(3299,1416)	**(1447,886)**
(3, 16, 5)	192.3	*137.44*	(4794,1924)	**(1983,1222)**
(4, 16, 5)	185.08	*136.24*	(6397,2614)	**(2457,1625)**
(2, 2, 16)	122.62	*132.36*	(2571,905)	**(578,582)**
(3, 2, 16)	119.79	*135.34*	(4216,1599)	**(833,972)**
(4, 2, 16)	123.14	*133.69*	(5850,2853)	**(1102,1572)**
(2, 4, 16)	120.56	*104.57*	(5038,1324)	**(1594,902)**
(3, 4, 16)	118.57	*102.77*	(7521,2260)	**(2174,1388)**
(4, 4, 16)	115.33	*100.61*	(10,374,3636)	**(2772,2084)**
(2, 16, 16)	114.16	*94.14*	(4902,4402)	**(7719,2822)**
(3, 16, 16)	126.16	*92.08*	(6805,5729)	**(10,557,3884)**
(4, 16, 16)	124.23	*90.49*	(9107,7752)	**(13,469,5156)**

Table 1.
Implementation results of pipeline and P = 4 parallel DWPT (italic) and IDWPT (bold) architectures.

Design parameters (Depth, Filter order, and Quantification)	Clock frequency (MHz)		Resources usage ($l_e l_r$)	
(2, 2, 5)	217.31	*207.04*	(1109,504)	**(109, 186)**
(3, 2, 5)	212.45	*195.77*	(1699,935)	**(166,312)**
(4, 2, 5)	213.4	*198.73*	(2754,1531)	**(244,505)**
(2, 4, 5)	217.9	*147.15*	(2120,897)	**(265,286)**
(3, 4, 5)	202.6	*148.39*	(3050,1197)	**(379,442)**
(4, 4, 5)	206.59	*147.65*	(4603,2023)	**(483,665)**
(2, 16, 5)	201.14	*136.44*	(7689,2447)	**(1447,886)**
(3, 16, 5)	202.16	*133.05*	(12,176,3166)	**(1983,1222)**
(4, 16, 5)	196.82	*131.56*	(14,956,4571)	**(2457,1625)**
(2, 2, 16)	95.77	*128.75*	(6079,1696)	**(578,582)**
(3, 2, 16)	97.8	*123.08*	(9279,2735)	**(833,972)**
(4, 2, 16)	98.82	*128.04*	(13,489,5011)	**(1102,1572)**
(2, 4, 16)	97.04	*99.98*	(12,032,2582)	**(1594,902)**
(3, 4, 16)	94.2	*98.87*	(17,549,3965)	**(2174,1388)**
(4, 4, 16)	88.01	*98.95*	(24,311,6363)	**(2772,2084)**
(2, 16, 16)	99.15	*90.16*	(11,263,7856)	**(7719,2822)**
(3, 16, 16)	102.89	*86.02*	(14,750,11,451)	**(10,557,3884)**
(4, 16, 16)	100.24	*86.1*	(21,314,13,091)	**(13,469,5156)**

Table 2.
Implementation results of pipeline and P = 8 parallel DWPT (italic) and IDWPT (bold) architectures.

and decreases logarithmically the clock frequency from around 200 MHz to around 100 MHz. As expected, the impact of depth and order of filters is too weak on the clock frequency and increases linearly the logic and element registers while it was exponential with Mallat binary tree. It is important to notice that the small latency in our architectures give us the possibility to process data in ultra high speed (in the gate of Giga-samples/clock cycle) without requiring any extra memory or DSP blocks.

It is important to notice that the incrementation of the functional frequency is directly proportional to the parallel degree. When we exceed the order of parallelism to 32, the needed resources overcome the capacity of NexysVideo board. To vanquish this problem, we suggest two possible solutions:

i. Under the strategy of minimizing the used hardware of Discrete Wavelet Transform, we look forward to the lifting scheme wavelet transform as a second DWPT transform generation. Section 3 is dictated to describe in details this suggested proposal.

ii. Another possible solution is to upgrade this work with a new FPGA family like Ultra scale. This new FPGA architecture present a high-performance environment which deliver the optimal balance between the required system performance (with 783 k to 5541 k Logic cells) and the smallest power envelope. But it remains a very expensive solution.

Design parameters (Depth, Filter order, and Quantification)	**Clock frequency (MHz)**		**Resources usage ($l_e l_r$)**	
(2, 2, 5)	210.36	*197.47*	(3668, 652)	**(389,668)**
(3, 2, 5)	209.23	*195.35*	(6019, 960)	**(573,1096)**
(4, 2, 5)	209.02	*194.97*	(8655, 1243)	**(742,1576)**
(2, 4, 5)	181.83	*147.54*	(5991, 1689)	**(1091,1008)**
(3, 4, 5)	178.58	*142.31*	(8380, 2363)	**(1410,1526)**
(4, 4, 5)	178.37	*141.98*	(11,181, 2601)	**(1552,2036)**
(2, 16, 5)	169.1	*127.6*	(30,012, 4881)	**(5894,3048)**
(3, 16, 5)	167.28	*124.88*	(37,172, 6575)	**(7300,4106)**
(4, 16, 5)	167.43	*125.09*	(38,374, 7680)	**(7536,4796)**
(2, 2, 16)	106.07	*125.25*	(11,116, 3395)	**(2183,2120)**
(3, 2, 16)	105.11	*123.02*	(17,679, 4330)	**(2704,3472)**
(4, 2, 16)	104.98	*122.71*	(25,389, 4830)	**(3016,4986)**
(2, 4, 16)	91.2	*92.6*	(31,336, 5137)	**(6154,3208)**
(3, 4, 16)	90.55	*91.29*	(39,361, 7764)	**(7730,4848)**
(4, 4, 16)	91.85	*93.93*	(42,687, 10,342)	**(8383,6458)**
(2, 16, 16)	86.43	*83.17*	(26,408, 15,572)	**(30,619,9736)**
(3, 16, 16)	86.57	*83.44*	(33,859, 20,959)	**(39,257,13,104)**
(4, 16,16)	85.9	*82.16*	(36,348, 24,456)	**(42,143,15,290)**

Table 3.
Implementation results of pipeline and P = 16 parallel DWPT (italic) and IDWPT (bold) architectures.

3. Hardware implementation of second generation

Actually, we find that some new applications especially in modern wireless communication require high throughput but at the same time a low energy consumption. For this reason, we look for the second discrete wavelet generation or lifting scheme wavelet transform. Because the lifting wavelet theory is by nature require less multiplier/adder blocks and consequently low energy.

So, our aim in this section is to conserve the ultra high speed data process and also reduce the hardware conception by introducing the linearization concept in the classic lifting scheme DWPT and IDWPT tree as shown in **Figures 11** and **12**.

A new pipeline and linear lifting scheme DWPT and IDWPT architectures are presented in **Figures 11** and **12**. These new architectures ensure the data speed proceed like the classic lifting scheme transform but with less hardware not affected by the wavelet depth.

The $\boldsymbol{P}/\boldsymbol{U}$ Filter Blocks and $\overline{\boldsymbol{P}}/\overline{\boldsymbol{U}}$ Filter Blocks in linear lifting scheme of DWPT and IDWPT architectures, receptively, are the modified predicted and updated filter. In **Figure 13**, we present the structure of the modified $\boldsymbol{P}/\boldsymbol{U}$ Filter Blocks (the same for $\overline{\boldsymbol{P}}/\overline{\boldsymbol{U}}$ Filter Blocks, we just change the coefficients values) which can process the same amount of data (same functionality) on given stage in the classic lighting scheme tree. This Filter Blocks can process two samplings in one clock cycle.

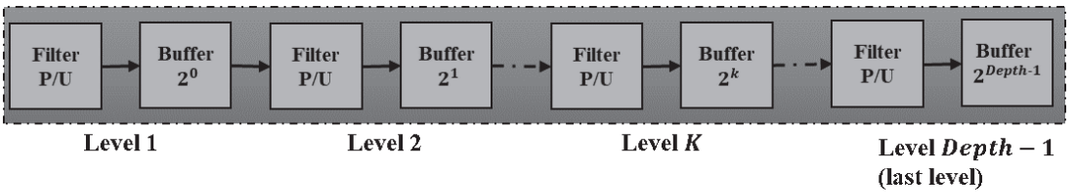


Figure 11.
Data path diagram of linear proposed lifting scheme DWPT architecture.

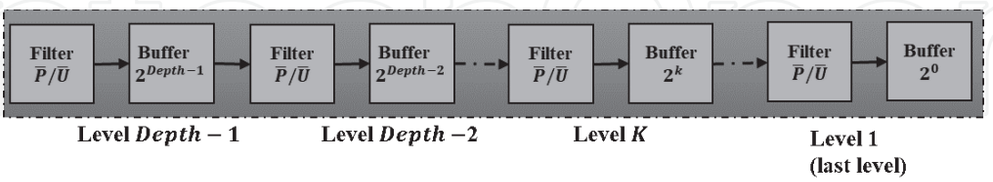


Figure 12.
Data path diagram of linear proposed lifting scheme IDWPT architecture.

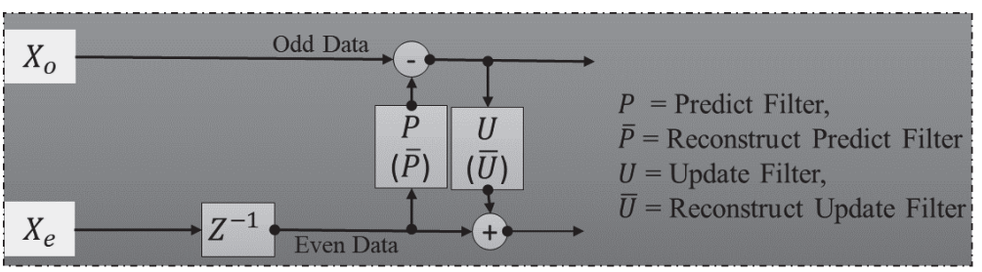


Figure 13.
Structure of the modified predict/update filters and their conjugate in stage ***k****.*

4. Comparison

To evaluate the performance of our architecture, a comparison section is important to prove the potential of our work and to lead us to a new innovated architecture.

In **Table 4**, we present a comparison between our proposed architectures and other achieved architectures of Discrete Wavelet Transform in literature. Without doubt, this table presents the potential of our linear pipeline and parallel architecture where on one hand it ensures a high frequency data processing and on the other hand a full reconfigurable structure using less hardware. Additionally, without missing an important feature, we implemented our architecture without using a memory or DSP blocks which gives us a privilege to more optimization of the used hardware in the next FPGA generation.

5. Conclusion

In this work, we propose ultra-high throughput with low hardware consumption of first generation and second generation of discrete wavelet packet transform. Where title of example, from **Table 3**, with a quantization order = 5, depth order = 2, filter order = 2 and degree of parallelism $\boldsymbol{P}$ = 16, we obtain a clock frequency = 210.36 MHz, theoretically can proceed 3365.52 Mega samplings in one clock cycle with low hardware used $\boldsymbol{l_e}$ = 3668 and $\boldsymbol{l_r}$ = 652.

	Sung et al. [20]	Marino et al. [21]	Mohanty et al. [22]	Madis-hetty et al. [23]	Wang et al. [24]	Wu et al. [5]	Meihua et al. [25]	Proposed parallel architecture
Wavelet	Daub-4	Quadri-filter	Daub-4	Daub-6	Lifting- Db4	Quadri-filter	Quadri-filter	Arbitrary
DWPT/ IDWPT	Both	DWPT	DWPT	DWPT	Both	Both	DWPT	Both
Logic cell	N/A	N/A	426	1040	N/A	30,192 (Logic element)	1835 (Logic element)	3668 (Logic element)
Technology	Xilinx XC2V4000	N/A	CMOS 90 nm	Xilinx Virtex 6	CMOS 180 nm	0.35 μm	Altera EP20K200E	Altera Cyclone IV & V
Max. Freq.(Mhz)/ Bitrate	N/A	N/A	Feq.: 20	Feq.: 306.15	Feq.: 20	Feq.: 100	Feq.: 29	Bitrate:718.13 Feq.: 197.47
Memory	Yes	N/A	Yes	Yes	Yes	N/A	Yes	No
Quantization (bits)	N/A	N/A	N/A	N/A		32	N/A	Test with 5 to 16 (up to the limit of manufacturing technology)
DSP	N/A	N/A	N/A			N/A	N/A	No
Depth	3	N/A	N/A	4	3	3 (up to 6)	3	2, 3 and 4 (up to the limit of manufacturing technology)
Parallel degree	No	No	No	No	No	No	No	4,8 and 16(up to the limit of manufacturing technology)

Table 4.
Comparison of proposed architectures with other works.

Based on the results in **Tables 1–3**, these architectures ensure high operating frequency which is low affected of wavelet depth and filters order because in our structures we maintained a short critical path of effective data path. Furthermore, these architectures are pipelined and P-parallel, modeled in VHDL at the RTL level, generic and fully reconfigurable in pre-synthesis function of the quantization of the filter coefficients and data sampling, the depth of wavelet transform, the order of the filters, and the degree of parallelism.

Last, but not least, our developed architectures are reconfigurable post-synthesis, which is not the case for most of the previous work as shown in the comparison in **Table 4**. Where the values of filters coefficients can be load at run-time which provides a great flexibility in experimental usage in contrary to all previous works.

This work is still in progress where we are making many simulations/verifications in different contexts to verify if the simulation results will agree or not with the implementation results. As perspectives, we work on new version of FIR filter and in parallel another work to create an IP core (Intellectual Property core) FIR to be used with different FPGA boards and in different applications. A natural way of this work is to develop a different parallel version of hardware implementation in FPGA of lifting scheme wavelet transform.

Author details

Mouhamad Chehaitly[1*], Mohamed Tabaa[2], Fabrice Monteiro[3], Safa Saadaoui[2] and Abbas Dandache[3]

1 University of Montpellier, LIRMM Lab, France

2 EMSI Casablanca, LPRI Lab, Morocco

3 University of Lorraine, LGIPM Lab, France

*Address all correspondence to: che.liban.tly@hotmail.com

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

References

[1] Denk T and Parhi K. Architectures for lattice structure based orthonormal discrete wavelet transforms. IEEE International Conference on Application Specific Array Processors, pp. 259–270, August 1994.

[2] Vishwanath M and Owens R. A common architecture for the DWT and IDWT. IEEE International Conference on Application Specific Systems, Architectures and Processors (ASAP), pp. 193–198.8, August 1996.

[3] Motra A, Bora P and Chakrabarti I. An efficient hardware implementation of DWT and IDWT. IEEE Conference on Convergent Technologies for the Asia-Pacific Region (TENCON), vol. 1, pp. 95–99, October 2003.

[4] Hatem H, El-Matbouly M, Hamdy N and Shehata K-A. VLSI architecture of QMF for DWT integrated system. Circuits and Systems, MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium on (Volume: 2), pp. 560–563, 2001. doi:10.1109/MWSCAS.2001.986253

[5] Wu B-F and Hu Y-Q. An efficient VLSI implementation of the discrete wavelet transform using embedded instruction codes for symmetric filters. IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 9, pp. 936–943, September 2003.

[6] Jing C and Bin H-Y. Efficient wavelet transform on FPGA using advanced distributed arithmetic. 8th IEEE International Conference on Electronic Measurement and Instruments (ICEMI'2007), pp. 2–512–2-515, Aug. 2007.

[7] Wu Z and Wang W. Pipelined architecture for FPGA implementation of lifting-based DWT. 2011 International Conference on Electric Information and Control Engineering, pp. 1535–1538, April 2011.

[8] Palero R, Gironés R and Cortes A. A novel FPGA architecture of a 2-D wavelet transform. Journal of VLSI signal processing systems for signal, image and video technology, vol. 42, no. 3, pp. 273–284, 2006.

[9] Hu Y and Jong C. A memory-efficient high-throughput architecture for lifting-based multi-level 2-D DWT. IEEE Transactions on Signal Processing, vol. 61, no. 20, pp. 4975–4987, Oct. 2013.

[10] Fatemi O and Bolouki S. A Pipeline, Efficient and Programmable Architecture for the 1-D Discrete Wavelet Transform using Lifting Scheme. The Second Conference On Machine Vision, Image Processing & Applications (MVIP 2003), Tehran 2003.

[11] Sowmya K-B and Mathew J. Discrete Wavelet Transform Based on Coextensive Distributive Computation on FPGA. Materials Today: Proceedings, Second International Conference on Large Area Flexible Microelectronics (ILAFM 2016): Wearable Electronics, December 20th-22nd, 2016.

[12] Paya G, Peiro M, Ballester F and Herrero V. A new inverse architecture discrete wavelet packet transform architecture. IEEE, Signal Processing and Its Applications, 7803–7946, 443–446 vol. 2, 2003.

[13] Acharya T. A Systolic Architecture for Discrete Wavelet Transforms. IEEE, Digital Signal Processing Proceedings, 13th International Conference on Volume 2, 571–574 vol. 2, 1997.

[14] Farahani M and Eshghi M. Architecture of a Wavelet Packet Transform Using Parallel Filters.

TENCON 2006 - IEEE Region 10 Conference, 1–4244–0548-3, 1–4, 2006.

[15] Farghaly S-H and Ismail S-M. Floating-Point FIR-Based Convolution Suitable for Discrete Wavelet Transform Implementation on FPGA, 2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2019, pp. 158–161, DOI: 10.1109/NILES.2019.8909290.

[16] Radhakrishnan P and Themozhi G. FPGA implementation of XOR-MUX full adder based DWT for signal processing applications. Microprocessors and Microsystems, Volume 73, 2020, 102961, ISSN 0141–9331, doi:10.1016/j.micpro.2019.102961.

[17] Taha T-B, Ngadiran R and Ehkan P-L. Design and Implementation of Lifting Wavelet Transform Using Field Programmable Gate Arrays. MS&E 767.1 (2020): 012041.

[18] Mohammed Shaaban I et al. High-throughput parallel DWT hardware architecture implemented on an FPGA-based platform. Journal of Real-Time Image Processing 16.6 (2019): 2043–2057.

[19] Mallat S. A wavelet tour of signal processing. Academic Press, 1999.

[20] Sung T-Y et al. Low-power multiplierless 2-D DWT and IDWT architectures using 4-tap Daubechies filters. In Proc. Seventh Int. Conf. PDCAT, pp. 185–190, 2006.

[21] Marino F. Two fast architectures for the direct 2-D discrete wavelet transform. IEEE Trans. Signal Process, vol. 49, no. 6, pp. 1248–1259, 2001.

[22] Mohanty B-K and Meher P-K. Memory-efficient high-speed convolution-based generic structure for multilevel 2-D DWT. IEEE Trans. Circuits Syst. Video Technol., vol. 23, pp. 353–363, 2013.

[23] Madishetty S, Madanayake A, Cintra R and Dimitrov V. Precise VLSI Architecture for AI Based 1-D/2-D Daub-6 Wavelet Filter Banks with Low Adder-Count. IEEE Transactions on circuits and systems-I: regular paper, Vol. 61, No. 7, 1984–1993, July 2014.

[24] Wang C et al. Near-threshold energy-and area-efficient reconfigurable DWPT/DWT processor for healthcare - monitoring Applications. IEEE Transactions on Circuits and Systems II: Express Briefs, 62(1), 70–74, 2015.

[25] Meihua X et al. Architecture research and VLSI implementation for discretewavelet packet transform. High Density Microsystem Design and Packaging and Component Failure Analysis, 2006. HDP'06. Conference on. IEEE, 2006.