# *FPGA Implementations of Bireciprocal Lattice Wave Discrete Wavelet Filter Banks*

**Jassim M. Abdul-Jabbar and Sama N. M. Al-Faydi**
**Computer Engineering Department- University of Mosul**

## Abstract

In this paper, a special type of IIR filter banks; that is the bireciprocal lattice wave digital filter (BLWDF) bank, is presented to simulate scaling and wavelet functions of six-level wavelet transform. 1st order all-pass sections are utilized for the realization of such filter banks in wave lattice structures. The resulting structures are a bireciprocal lattice wave discrete wavelet filter banks (BLW-DWFBs). Implementation of these BLW-DWFBs are accomplished on Spartan-3E FPGA kit. Implementation complexity and operating frequency characteristics of such discrete wavelet 5th order filter bank is proved to be comparable to the corresponding characteristics of the lifting scheme implementation of Bio. 5/3 wavelet filter bank. On the other hand, such IIR filter banks possess superior band discriminations and perfect roll-off frequency characteristics when compared to their Bio. 5/3 wavelet FIR counterparts.

**Keywords**: All-Pass Sections, Bireciprocal Lattice Wave Filters (BLWDFs), Bireciprocal Lattice Wave Digital Discrete Wavelet Filter Banks (BLW-DWFBs), IIR Wavelet Filter Banks, Bio. 5/3 wavelet filter bank, Processing Element (PE), FPGA Implementations.

# البناء بإستخدام FPGA لأجراف مرشحات مويجية متقطعة من النوع الموجي المتشابك ثنائي التبادل

## الخلاصة

في هذا البحث، قُدّم نوع خاص لأجرافِ مرشِحاتٍ ذات الدفقة غير المحدودة IIR، وهي أجراف مرشحاتٍ رقمية من النوعِ الموجي المتشابكِ ثنائي التبادلِ (BLWDF) لمحاكاةِ الدوالِ المعيارية والمويجية (scaling and wavelet functions) للتحويلِ المويجي المتقطع بمستوياتٍ ستّة. وتم توظيف مقاطع إمرارٍ كليةٍ من المرتبة الأولى لتحقيقُ أجرافَ المرشحاتِ تلك بهياكل موجية متشابكة. أن الهياكل الناتجة تمثل أجراف مرشحات مويجية متقطعة من النوع الموجي المتشابك ثنائي التبادل (BLW-DWFBs). لقد تم بناء تلك الـ BLW-DWFBs باستخدام أداة لشريحة FPGA نوع Spartan-3E. إن تعقيد البناء لأجراف المرشحات المويجية المتقطعة تلك بالمرتبة 5 وتردد التشغيل قد أُثبِت أنّهما مقاربان لنظيريهما في بناء أجراف المرشحات المويجية نوع Bio. 5/3 . ومن جهة أخرى، فإنَّ إجراف المرشحات تلك هي من نوع IIR ومعروفة بأنها تَمتلكُ تمييزاً متفوّقاً وخصائصَ تدرجٍ ترددي متقنةٍ بالمقارنة مع نظرائها من نوعِ أجراف المرشحات المويجية نوع Bio. 5/3 ذات الدفقة المحدودة FIR .

**الكلمات الدالة:** مقاطع إمرارٍ كليةٍ، أجراف مرشحاتٍ رقمية من النوعِ الموجي المتشابكِ ثنائي التبادلِ (BLWDF)،أجراف مرشحات مويجية متقطعة من النوع الموجي المتشابك ثنائي التبادل (BLW-DWFBs)، أجراف مرشحات مويجية نوع IIR، أجراف مرشحات مويجية نوع Bio. 5/3، عنصر معالجة (PE)، البناء باستخدام صفوف البوابات القابلة للبرمجةَ حقلياً (FPGA).

## Introduction

The wavelet transform, generated by iteration of orthonormal two-band power-complementary recursive filter banks with perfect reconstruction is considered. Orthonormal filter banks can be realized using finite impulse response (FIR) or infinite impulse response (IIR) filters. It is known that IIR filter structures which are constructed using two real all-pass wave lattice sections can be implemented with low complexity structures that are robust to finite precision errors [1]. The main advantage of IIR filter banks is its good frequency selectivity. Linear phase and orthonormality are not mutually exclusive properties for IIR filter banks, as they are in FIR case.

In this paper, orthogonal wavelet transforms are examined using computationally efficient IIR perfect reconstruction quadrature-mirror filter (QMF) banks. The decomposition is done using low-pass branch iterated filter banks. Bireciprocal lattice wave digital filters (BLWDFs) are employed in 5th order to form some intermediate filter banks. The wave lattice components of these IIR filter banks are real all-pass wave adaptors of the 1st order. Thus, new bireciprocal lattice wave discrete wavelet filter banks (BLW-DWFBs) are introduced.

Many examples on the design of orhtonormal wavelet transform implemented with IIR filter pairs were considered in 2005 by S. Damjanović and L. Milić [1]. Also in 2005, the frequency transformations of such wavelet IIR filter banks were also presented by S. Damjanović, *et al.* [2]. Low complexity half-band IIR filters were presented and realized in 2006 by L. Milić, *et al.* [3] using two path polyphase structures utilizing all-pass filters as components. Such realization was accomplished without taking phase linearity into considerations.

Linear phase realizations are introduced in this paper for the proposed BLW-DWFBs. Recently, many attempts have been reported concerning hardware implementations of such two-band wavelet IIR filter banks [4-6]. The proposed 5th order BLW-DWFBs are implemented on Spartan-3E FPGA kit. It is shown that, such implementations are comparable to the lifting scheme implementations of the corresponding Bio.5/3 FIR wavelet structures, besides possessing superior band discriminations and perfect roll-off frequency characteristics.

In addition to this introductory section, this paper is divided into seven major sections. In section 2, the design and realization of the bireciprocal lattice wave digital filters are reviewed. Section 3 presents the IIR Wavelet Filter Banks. In section 4, Implementations of BLWDF bank structures as a wavelet transform are accomplished. Section 5 presents two implementation techniques of the applied filter, namely; pipelining and bit-serial techniques. Section 6 is devoted to the implementation results and finally, section 7 summarizes some conclusions.

## Bireciprocal Lattice Wave Digital Filters; Design and Realization

A bireciprocal Lattice Wave Digital Filter (BLWDF), as shown in Fig. 1, is a special form of a LWDF [7] that reduces the number of adaptors under certain conditions and can also be used as a base for decimation and interpolation filters. In BLWDF transfer functions, numerators are polynomials in $z^{-2}$ [8]. This reduces the implementation complexity of BLWDFs. It also increases the throughput, compared to the same order LWDFs implementations [9]. So, BLWDFs conducted in this paper, represent the most efficient family of IIR filters and are therefore of great interest.

The all-pass functions $A_0(z)$ and $A_1(z)$ can be expressed respectively, as [10]

$$A_0(z) = \prod_{i=2,4,\dots}^{(N+1)/2} \frac{\alpha_i + z^{-1}}{1 + \alpha_i z^{-1}} \dots\dots\dots (1)$$

and

$$A_1(z) = \prod_{i=3,5,\dots}^{(N+1)/2} \frac{\alpha_i + z^{-1}}{1 + \alpha_i z^{-1}} \dots\dots\dots (2)$$

The overall transfer functions of *BLWDF* can then be written for the low-pass side as

$$H_o(z) = \frac{1}{2}[A_o(z) + z^{-1}A_1(z)] \dots\dots(3)$$

and for the high-pass side as

$$H_1(z) = \frac{1}{2}[A_o(z) + z^{-1}A_1(z)] \dots(4)$$

## IIR Wavelet Filter Banks

A very efficient way for representing the QMF bank can be obtained by using polyphase structure[11]. QMF banks, composed of two all-pass filters, are known to be one of the best circuits for building up a multi-channel IIR filter banks. They can completely eliminate the aliasing error and amplitude distortion [12]. Fig. 2 shows the two channel all-pass filter based IIR QMF banks [13].

## Implementation of BLWDF Bank Structure as a Wavelet Transform

In this section, the proposed structures for implementing the 5th order *BLWDF* as wavelet transform are presented.

### *One-level analysis FBs using 5th order BLWDF*

An efficient hardware implementation of the BLWDF bank structure of discrete wavelet transform can be accomplished using number of Processing Elements (PEs). Each processing element in Fig.(3) is a generic term that refers to a hardware element that executes a stream of instructions.

In Fig. (3), the *Control Unit* sends the clock signal to the processing elements, *PE-A* and *PE-B* to declare the work at each rising edge clock cycle. The internal construction of *Spilt Stage* in Fig. 3 is shown in Fig.(4). It splits the samples that are read from *Block RAM* into even and odd samples, where even samples are fed to the *PE-A* and odd samples to the *PE-B*.

*PE-A* represents a processing element that contains the 1st order all-pass filter $A_1(z)$ and *PE-B* contains the 1st order all-pass filter $A_0(z)$. The transfer function of these two all-pass filters can be represented as in (2) and (1), respectively. These transfer functions can be realized in discrete-time domain as difference equations.

For an all-pass filter $A_1(z)$, the corresponding difference equation is

$$y_2[n] = \beta\, x_{even}[n] + x_{even}[n-1]$$

$$-\beta\, y_2[n-1] \ldots\ldots (5)$$

and for an all-pass filter $A_0(z)$, the corresponding difference equation is

$$y_1[n] = \alpha\, x_{odd}[n] + x_{odd}[n-1]$$
$$-\alpha\, y_1[n-1] \ldots.\ldots (6)$$

where $x_{odd}$ represents the odd samples of the input signal, $x_{even}$ represents the even samples of the input signal, $\alpha$ represents the all-pass filter $A_0(z)$ coefficient and equal to $\alpha_2$ = 0.1875 and $\beta$ represents the all-pass filter $A_1(z)$ coefficient and equal to $\alpha_3$ = 0.6875. A full derivations and calculations of these coefficients are given in Ref. [14].

Both *PE-A* and *PE-B* receive the inputs and perform the filtering operation as in (5) and (6), respectively. Then, the output manipulated samples $y_1[n]$ and $y_2[n]$, are fed to the *Lattice Part* to perform operations that are shown in Fig. (5) and generate the approximation and detail coefficients outputs $c[n]$ and $d[n]$, respectively.

The first output samples of $c[n]$ and $d[n]$ need two clock cycles to be derived out from the structure of Fig.(3), while the rest of the samples will appear after that in a throughput of 2 samples per clock cycle.

As soon as the first coefficient is generated, the *Control Unit* in Fig. 3 enables the data_ready signal which indicates that the data is ready to be written into the *RAM*. The *Control Unit* then enables the write_enable signal and sends the write address to the *RAM* so that the wavelet coefficients are written in the appropriate location in the *RAM*. Thus, the *Control Unit* has to perform an appropriate scheduling in order to store the wavelet coefficients in the memory.

This structure will continue its operation for all input samples. When saving *Ń* samples in *Block RAM*, the number of output samples of $c[n]$ is *Ń/2* and equals those of $d[n]$. So, the size of *APP RAM* for saving $c[n]$ will be *Ń/2* and the size of *DET RAM* for saving $d[n]$ is also *Ń/2*.

***One-level synthesis FBs using 5th order BLWDF***

The two parallel all-pass filters used in this section are $A_0(z^{-1})$ and $A_1(z^{-1})$ and are shown in Fig. (6) by the processing elements *PEr-B* and *PEr-A*, respectively. These filters can be implemented as in $A_0(z)$ and $A_1(z)$ all-pass filters, respectively, by reversing the inputs and outputs of these filters using *LIFO* register [15] as shown in Fig. (7).

The *LIFO Reg* in Fig. (7), stores the incoming samples and then the last input sample will be sent first to the filter and so on until reaching the first stored input sample. This operation will take a long time for waiting the *LIFO Reg* to fill. So to solve this problem, the *LIFO Reg* must be removed and substituted by another equivalent way. The efficient equivalent way that is used instead of *LIFO Reg1* is to read the approximation and detail coefficients from *APP RAM* and *DET RAM* respectively from down to up. This means that the reverse operation is realized before *Lattice Part* without encountering any delays as with *LIFO Reg1*. Also *LIFO Reg2* can be removed and substituted by saving the outputs of *PEr-A* and *PEr-B* in Fig. 6 in *Block RAM* from down to up.

The resulting all-pass filters are the same as those all-pass filters in analysis side. The $A_0(z)$ all-pass filter has the difference equation

$$y_{1r}[n] = \alpha\, d_l[n] + d_l[n-1] \\ -\alpha\, y_{1r}[n-1] \;\ldots\ldots (7)$$

and the $A_1(z)$ all-pass filter has the difference equation

$$y_{2r}[n] = \beta\, c_l[n] + c_l[n-1] \\ -\beta\, y_{2r}[n-1] \;\ldots\ldots (8)$$

with

$$c_l[n] = 0.5\big[c[n] + d[n]\big] \ldots\ldots\ldots\ldots (9)$$

and

$$d_l[n] = 0.5\big[c[n] - d[n]\big] \ldots\ldots\;\ldots (10)$$

At the beginning, when the *Control Unit* in Fig. 6 sends the clock signal, the *Lattice Part* starts to read the two inputs $c[n]$ and $d[n]$ that come from the *APP RAM* and *DET RAM*, respectively. This reading starts from bottom to top at each clock and a lattice operation similar to that shown in Fig.(5) b is performed. *PEr-A* and *PEr-B* make the primary processing represented by (7) and (8) on $c_l[n]$ and $d_l[n]$ inputs. Two clock cycles are needed to complete the filter's processing for the first input samples of $c_l[n]$ and $d_l[n]$ and get the output while the rest of the output samples will get out consequently in a throughput of 2 samples per clock cycle.

When the outputs from *PEr-A* and *PEr-B* are derived out, the *Control Unit* enables the data_ready signal which indicates that the data is ready to be written into the *Block RAM*. The *Control Unit* then enables the write_enable signal and sends the write address to the *Block RAM* so that the reconstructed signal is written in the *Block RAM*. This operation will continue for all input samples *Ń*, such that *Ń/2* samples for $c[n]$ and *Ń/2* samples for $d[n]$ are achieved. The reconstructed signal will be also *Ń* samples and that will be saved in *Block RAM*.

## Implementation Techniques

The two most common implementing techniques in *DSP* systems as considered in this paper are pipelining and bit-serial techniques. The following subsections describe such implementations of the proposed *BLWDF* banks for wavelet transform.

### *Pipeline implementation*

The beauty of a pipelined design is that the new data can start processing before the prior data finished. The efficiency of the pipelined design speed faces the penalty of having a large design area as compared with the bit-serial implementation [8].

### Six-level analysis structure using pipeline implementation

The block diagram of the proposed analysis side scalable structure for six-level *DWT* is shown in Fig. (8). The main units in the structure are the Processing Elements (PEs). Each PE consists of one-level analysis filter bank of the same 5th order *BLWDF* type shown in Fig. (3). Therefore, the *PE* structure depends on the chosen filter, the filter order and the selected structure of the filter banks.

In Fig.(8), when the *Control Unit* sends the rising edge clock signal, the first two samples of input signal of size *Ń* are first sent from *Block RAM* to the first processing element *PE1*. *PE1* receives the input, performs the filtering operation, and generates the first level of approximation and detailed coefficients. The approximation coefficients or the scaling coefficients, which are the outputs of the low-pass filter, are stored in the *Temporal Buffer1* in order to divide the *PE1* approximation output samples into even and odd samples. Even and odd samples are fed simultaneously to the next *PE* at the same clock for further processing. The detail coefficients or the wavelet coefficients of *PE1* which are the outputs of the high-pass filter are stored in *DET RAM1*. After that, *PE2* processes the approximation coefficients which are stored in *Temporal Buffer1* as they are generated to produce the second level of coefficients. Similarly, as soon as the second level coefficients are available, *PE3* starts processing them from *Temporal Buffer2* and the third level coefficients are produced and so on until the last output stage of *PE6*. Both approximation and detailed coefficients are then stored in the *APP RAM6* and other six *DET RAMs* (from *DET RAM1* to *DET RAM6*), respectively.

The *Control Unit* in Fig. 8 provides communication between different *PEs* as well as loads wavelet coefficients into different *DET RAMs* as they are being generated by the *PEs*. This *Control Unit* also provides clock signals to all the other units for synchronization purposes. Thus, the *Control Unit* has to perform an appropriate schedule, in order to store the wavelet coefficients from all levels in the memory.

**<u>Six-level synthesis structure using pipeline implementation</u>**

The wavelet filters constructed from $5^{th}$ order *BLWDF* for one-level synthesis side structure (shown in Fig. 6) can also be used for six-level synthesis side filter banks structure of *DWT* as shown in Fig. 9, where the main processing element that refers to one-level synthesis side filter bank is *PEr*. The *PEr* structure depends on the chosen filter.

In the structure of Fig. 9, the input samples represent the output of the six-level analysis side structure which are the approximation or scaling coefficients and wavelet coefficients saved previously in *APP RAM6* and in Six *DET RAMs*, respectively. The output of this structure represents the reconstructed signal from these coefficients after passing through several processings by using wavelet filters. The wavelet synthesis levels in Fig. 9 are represented by *PEri*. Manipulations can start in parallel whenever the approximation and detailed coefficients are available in these *RAMs*. Initially, *PEr1* reads at every rising edge clock from down to up one sample from the *APP RAM6* and another from *DET RAM6*. Then *Lattice Operation* is implemented for filtering operation. The output is saved in the *Temporal Buffer1*, Simultaneously, *PEr2* processes the samples that are read from up to down in *Temporal Buffer1*and from down to up in *DET RAM5*. The output of this level is saved in *Temporal Buffer2*. Other processing elements work as *PEr2* until the last level where the output is saved in *Block RAM* from down to up in order to build the constructed signal. All operations are controlled by the *Control Unit*.

***Bit-serial implementation***

A major advantage of bit-serial technique is that it is an area efficient implementation. This implementation becomes the most suitable to be used when the area is considered an important factor for the designer.

***<u>Six-level analysis structure using bit-serial implementation</u>***

In this structure, there is only one main processing element (*PE*) used instead of six processing elements to perform the whole processing. This *PE* as in Fig.10 represents a one-level analysis filter bank of the type shown in Fig. (3). The *Control Unit* manages the operations of *PE* by sending a clear signal whenever each level is completed to clear the variables and signals of the pervious operation on the *PE* in order to prepare it for another level operation.

Initially in Fig.(10), the *PE* receives the input samples from *Block RAM*. The rising edge clock signal from *Control Unit* is also received and the filtering operation is performed, then the output coefficients are saved for level one in *Temporal APP RAM* and *DET RAM1*. As soon as level one is completed, the *Control Unit* sends a clear signal to reset *PE*. Then, level two will start with the new input signal that is fed from *Temporal APP RAM* and the output coefficients for this level will be saved in *Temporal APP RAM* besides *DET RAM2*. This operation will continue for every level until level six. The output coefficients of this level will be saved as final outputs in *Temporal APP RAM* and *DET RAM6*.

***<u>Six-level synthesis structure using bit-serial implementation</u>***

The synthesis filter banks structure as shown in Fig. 11 consists of main processing element *PEr* that represents a one-level synthesis filter bank of the type shown in Fig. 6 to perform both lattice and filtering operation with two multiplexers (*MUX*) for multiplexing between *RAMs* with a *Control Unit* to manage the whole operation.

The structure of Fig. 11 will run when the first rising edge clock signal derives out from *Control Unit*. At level one, the *PEr* receives the input signal from *Temporal APP RAM* and *DET RAM6*. Reading from both *RAMs* will start from down to up. Then lattice and filtering operation are performed and outputs of level one are saved in *Temporal Block RAM*. When level one operation is completed, the *Control Unit* sends a clear signal to reset *PEr* and prepare it for level two operation that will start with the feeding of input signals from *Temporal Block RAM* and *DET RAM5*. Reading *DET RAM5* starts from bottom to up. The level two outputs will be saved in the same *Temporal Block RAM* and this operation will continue for every level until the last level, where the input will be fed into *PEr* from *Temporal Block RAM* and from *DET RAM1*. The output represents the reconstructed signal and saved also in the same *Temporal Block RAM*. The saving will also start from down to up.

**Implementation Results and comparisons**

The two techniques in previous sections have been implemented using one of the available Xilinx FPGA devices; Spartan-3E kit. This device has a capacity of (4656) logic slices and can operate at a maximum clock speed of 50 MHz. Therefore, performance is usually measured with respect to two evaluation metrics; the throughput (sample rate) which is given in terms of the clock speed, and device utilization, which is given in terms number of FPGA logic slices used by the implementation. Table 1 explains the hardware resources required for one-level wavelet transform using $5^{th}$ order BLWDF. Table (2) shows the Hardware resources required for pipeline implementation for six-level wavelet transform using $5^{th}$ order BLWDF. Also, Table 3 explains the hardware resources required for bit-serial implementation for six-level wavelet transform using $5^{th}$ order BLWDF. Finally, Table 4 shows the frequency values that generated by these implementation techniques.

From Tables (1) and (4), it can be seen that when implementing this structure of one-level wavelet transform using $5^{th}$ order *BLWDF* on FPGA Spartan-3E kit, the percentage occupied area is 3% and the operating frequency is 72.369 $MH_z$ . These results are of the same orders given in Ref. [16] for the lifting scheme implementations of Bio. 5/3 wavelet filter bank, where the percentage occupied area is approximately 3% and the operating frequency is 71.295 $MH_z$. On the other hand, half-band filters in wavelet transform using $5^{th}$ order *BLWDF* provide excellent pass/stop-band discrimination as compared to that of Bio. 5/3 wavelet filters (see Fig. 12).

From Tables 2 - 4, the percentage occupied areas for pipeline and bit-serial implementations for six-level wavelet transform using $5^{th}$ order *BLWDF* are 27% and 24%, respectively, while the maximum operating frequencies (in analysis sides) are respectively, 72.369 $MH_z$ and 58.824 $MH_z$. These results highlight again that bit-serial implementation is the most suitable to be used when the area is an important factor, while pipeline implementation is the most suitable to be used when the operating frequency is a constraint.

## Conclusions

A new bireciprocal lattice wave discrete wavelet filter banks (BLW-DWFBs) have been introduced and FPGA implemented. Lifting scheme Bio. 5/3 wavelet filter bank implementation complexities have been approximately maintained by IIR half-band filter bank that provide more excellent pass/stop-band discrimination characteristics. Bireciprocal lattice wave digital filters (BLWDFs) have been employed in $5^{th}$ order to form such intermediate IIR filter banks using low-complexity and robust implementation structures. Thus, new bireciprocal lattice wave discrete wavelet filter banks (BLW-DWFBs) have been achieved with perfect characteristics.

## References

1. S. Damjanović and L. Milić, "Examples of Orthonormal Wavelet Transform Implemented with IIR Filter Pairs", The 2005 International 114 Workshop on Spectral Methods and Multirate Signal Processing– SMMSP2005, Riga, Latvia, pp. 19 – 27, June, 2005.
2. S. Damjanović, L. Milić and T. Saramaki, "Frequency Transformations in Two-Band Wavelet IIR Filter Banks", EUROCON 2005; The International Conference on Computer as a Tool, Vol. 1, pp. 88 – 91, 2005.
3. L. Milić, S. Damjanović and M. Nikolic, "Frequency Transformations of IIR Filters With Bank Applications", IEEE Asia

Pacific Conference on Circuits and Systems, Singapore, December 2006.

4. C. Jing and H. Y. Bin, "Efficient Wavelet Transform on FPGA Using Advanced Distributed Arithmetic", IEEE Eighth International Conference on Electronic Measurement and Instruments (ICEMI), 2007.
5. I. Dalal, " An Efficient FPGA Implementation of the Discrete Wavelet Transform", M.Sc. Thesis, Electrical and Computer Engineering, North Carolina State University, 2008.
6. M. Nagabushanam and S. Ramachandran, " Design and Implementation of Parallel and Pipelined Distributive Arithmetic Based Discrete Wavelet Transform IP Core", European Journal of Scientific Research, ISSN 1450-216X Vol. 35, No.3, pp.378-392, 2009.
7. H. Johansson and L. Wanhammar, "Wave Digital Filter Structures for High-Speed Narrow-Band and Wide-Band Filtering", IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 46, No. 6, June 1999.
8. J. M. Abdul-Jabbar, " An Analytical Design Procedure For Bireciprocal Lattice Wave Digital Filters with Approximate Linear Phase", Alrafidain Engineering Journal, Collage of Engineering, University of Mosul, Mosul, Iraq, Vol. 17, No. 6, PP. 42-52 , Dec. 2009.
9. H. Ohlsson, "Studies on Implementation of Digital Filters with High Throughput and Low Power Consumption", M.Sc. Thesis, Electrical Engineering, Linköping University, Sweden, June 2003.
10. L. Milić and T. Saramaki, "Three Classes of IIR Complementary Filter Pairs with an Adjustable Crossover Frequency", Proceedings of the 2003 International Symposium on Circuits and Systems, Vol. 4, pp. IV-145 - IV-148, May 2003.
11. E. Galijasevic and J. Kliewer, "on The Design of Near-Perfect-Reconstruction IIR QMF Banks Using FIR Phase-Compensation Filters", University of Kiel, 2nd International Symposium on Image and Signal Processing and Analysis , Pula, Croatia, pages 530-534, June 2001.
12. X. Zhang and H. Iwakura, "Design of QMF Banks Using Allpass Filters", IEE Electronics Letters, Vol.31, No.3, pp.172-174, 1995.
13. S. Mitra, "Digital Signal processing", McGraw-Hill, New York, NY, 10020, ISBN 0-471-48422-9, 2001.
14. J. M. Abdul-Jabbar and S. N. M. Al-Faydi, "Design and Realization of Bireciprocal Lattice Wave Discrete Wavelet Filter Banks", to be published in Alrafidain Engineering Journal, Collage of Engineering, University of Mosul, Mosul, Iraq.
15. S. Damjanović, L. Milić,"A Family of IIR Two-Band Orhonormal QMF Filter Banks", Serbian Journal of Electrical Engineering, Vol. 1, No. 3, pp. 45-56, September 2004.

16. Z. Abed Ali, "Two-Dimensional Discrete Wavelet Transform Architecture for JPEG2000 filters", M. Sc. Thesis in Computer Engineering, Collage of Engineering, University of Mosul, Mosul, Iraq, 2007.

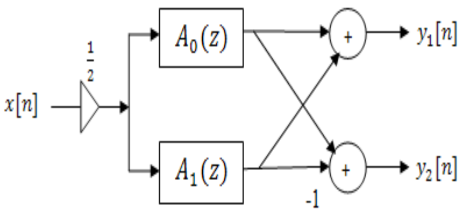


**Fig. 1 BLWDF block diagram**.

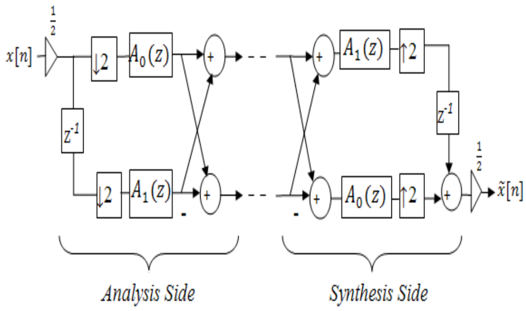


**Fig. 2 Polyphase realization of the IIR wavelet filter bank.**

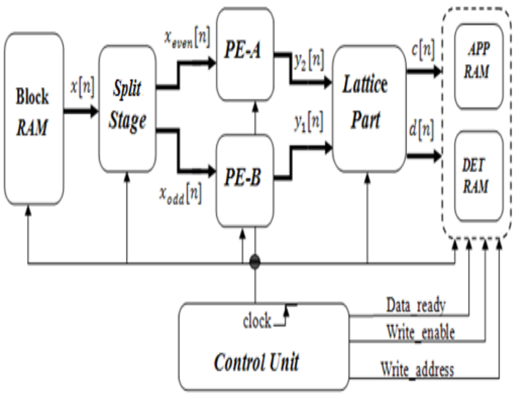


**Fig. 3 Structure of one-level wavelet transform analysis side using 5th order BLWDF.**

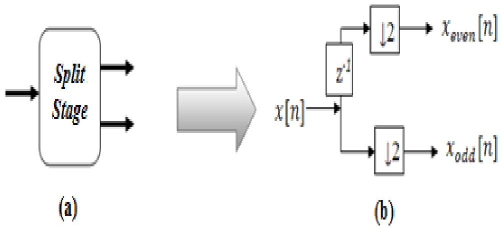


**Fig. 4 Spilt stage processing element**
**(a) Block diagram,**
**(b) Internal construction.**

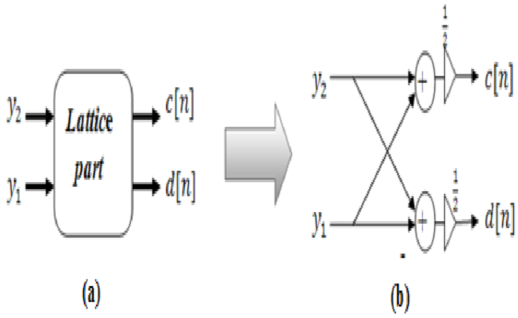


**Fig. 5 Lattice part operation**
**(a) Block diagram,**
**(b) Internal construction.**

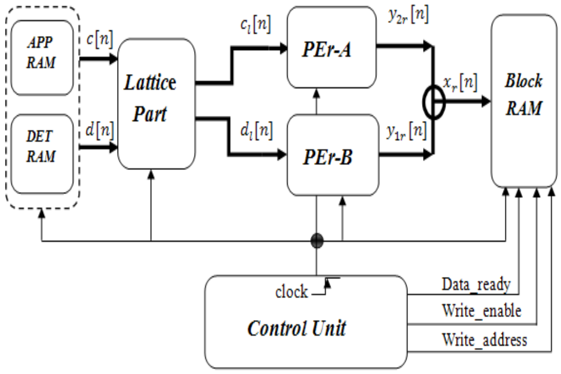


**Fig. 6 Structure of one-level wavelet transform synthesis side using 5th order BLWDF**.

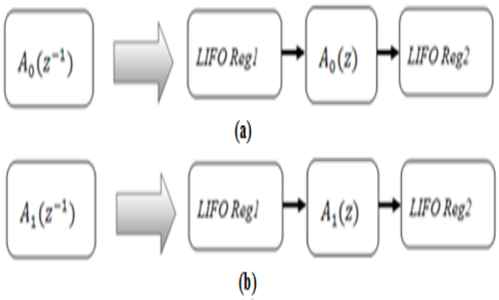


**Fig. 7 (a) Implementation of the transfer function $A_0(z^{-1})$.**
**(b) Implementation of the transfer function $A_1(z^{-1})$.**

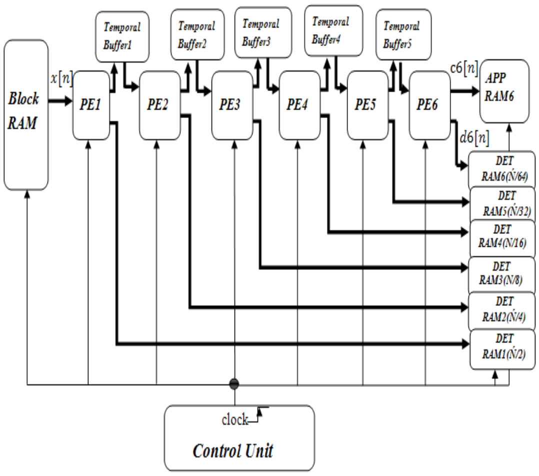


**Fig. 8 Structure of six-level wavelet transform analysis side using pipeline technique.**

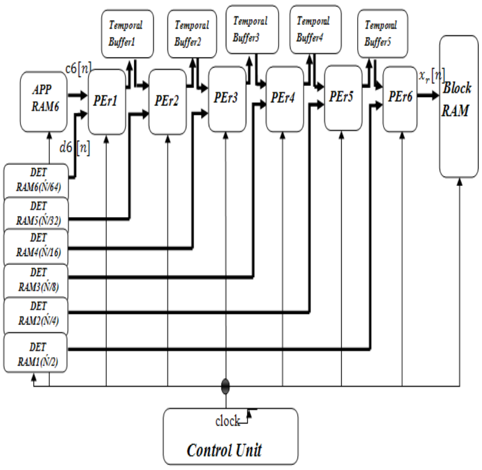


**Fig. 9 Structure of six-level wavelet transform synthesis side using pipeline technique.**

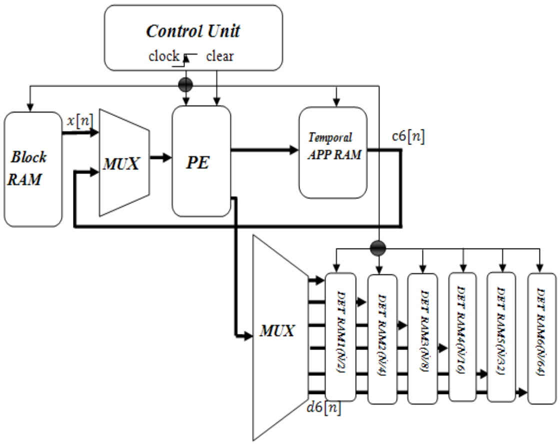


**Fig. 10 Structure of six-level wavelet transform analysis side using bit-serial technique.**

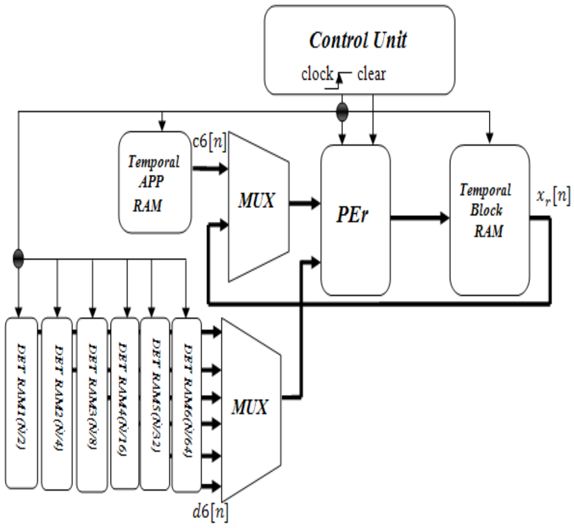


**Fig. (11) Structure of six-level wavelet transforms synthesis side using bit-serial technique.**

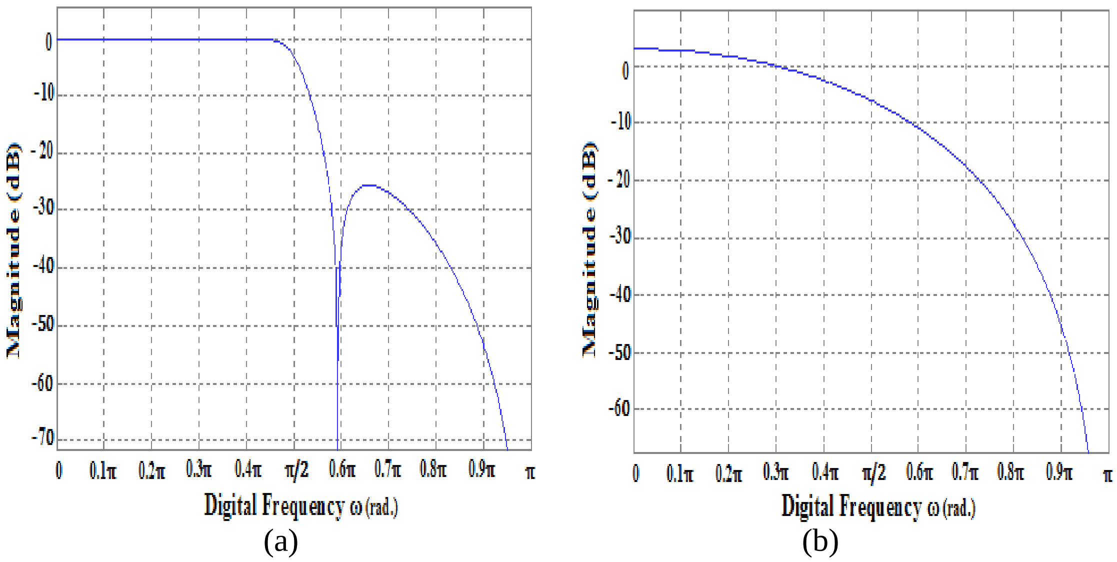


**Fig. (12) Magnitude responses for (a) 5th order BLWDF, (b) Bio. 5/3 wavelet filter.**

**Table (1) Hardware resources required for one-level wavelet transform using 5th order BLWDF.**

| Side | resource | used | available | Utilization ratio |
|---|---|---|---|---|
| Analysis Side Structure | Slices | 180 | 4656 | 3% |
| | Slice flip flops | 172 | 9312 | 1% |
| | 4-input LUTs | 218 | 9312 | 2% |
| Synthesis Side Structure | Slices | 261 | 4656 | 5% |
| | Slice flip flops | 227 | 9312 | 2% |
| | 4-input LUTs | 339 | 9312 | 3% |

**Table (2) Hardware resources required for pipeline implementation for six-level wavelet transform using 5th order BLWDF.**

| Side | resource | used | available | Utilization ratio |
|---|---|---|---|---|
| Analysis Side Structure | Slices | 1277 | 4656 | 27% |
| | Slice flip flops | 1120 | 9312 | 12% |
| | 4-input LUTs | 1556 | 9312 | 16% |
| Synthesis Side Structure | Slices | 1808 | 4656 | 38% |
| | Slice flip flops | 1565 | 9312 | 16% |
| | 4-input LUTs | 2249 | 9312 | 24% |

**Table (3) Hardware resources required for bit-serial implementation for six- level wavelet transform using 5th order BLWDF.**

| Side | resource | used | available | Utilization ratio |
|---|---|---|---|---|
| Analysis Side Structure | Slices | 1162 | 4656 | 24% |
| | Slice flip flops | 1166 | 9312 | 12% |
| | 4-input LUTs | 1399 | 9312 | 15% |
| Synthesis Side Structure | Slices | 1565 | 4656 | 33% |
| | Slice flip flops | 1562 | 9312 | 16% |
| | 4-input LUTs | 1967 | 9312 | 21% |

**Table (4) Periods and frequencies for different implementation structures.**

| Level and Technique | Analysis/ Synthesis | Minimum Period (n sec) | Maximum Frequency (MHz) |
|---|---|---|---|
| One-level | Analysis Side | 13.818 | 72.369 |
| | Synthesis Side | 16.350 | 61.162 |
| Six-level Pipeline Technique | Analysis Side | 13.818 | 72.369 |
| | Synthesis Side | 17.418 | 57.411 |
| Six-level Bit-serial | Analysis Side | 17 | 58.824 |
| | Synthesis Side | 22.068 | 45.314 |