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 1

Dynamic Reconfigurable on the Lifting Steps Wavelet Packet Processor with Frame-Based Psychoacoustic Optimized Time-Frequency Tiling for Real-Time Audio Applications

Alexey Petrovsky, Maxim Rodionov and
Alexander Petrovsky

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/51604

1. Introduction

The discrete wavelet packet transform (DWPT) as a generalization of the standard wavelet transform provides a more flexible choices for time–frequency (time-scale) representation of signals [1] in many applications, such as the design of cost-effective real-time multimedia systems and high quality audio transmission and storage. In parallel to the definition of the ISO/MPEG standards, several audio coding algorithms have been proposed that use the DWPT, in particular, adaptive wavelet packet transform, as the tool to decompose the signal [2],[3]. In practice, DWPT are often implemented using a tree-structured filter bank [2], [3],[4]. The DWPT is a set of transformations that admits any type of tree-structured filter bank, that provides a different time–frequency tiling map. Many architectures have been proposed for computing the discreate wavlet transform in the past. However, it is not the case for the DWPT. There are very few papers regarding the development of specific architectures for the DWPT. In [5] is designed a programmable DWPT processor using two-buffer memory system and a single multiplier–accumulator (MAC) to calculate different subbands. Method [6] exploits the in-place nature of the DWPT algorithm and uses a single processing element consisting of multipliers in parallel and adders for each low-pass and high-pass filters – wavelet butterflies (is the number of filter taps) to increase the throughput. In [7] is also proposed a folded pipelined architecture to speed up the throughput. It consists of MACs communicated by memory banks to compute each level of the total decomposition levels.

INTECH
open science | open minds

© 2013 Petrovsky et al.; licensee InTech. This is an open access article 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.

Applying the lifting scheme [8] for the construction of wavelets filter bank allows significantly reduce the number of arithmetic operations that are necessary to compute the transform. A folded parallel architecture for lifting-based DWPT was presented in [9]. It consists of a group of MACs operating in parallel on the data prestored in a memory bank. In [10] is proposed an architecture using a direct implementation of a lifting-based wavelet filter to perform one level of DWPT at a time. The main drawback of these existing architectures is that they all use memory to store the intermediate coefficients and involve intense memory access during the computation. A recursive pyramid algorithm based folded architecture for computing lifting-based multilevel DWPT is presented in [11]. However, the scheduling and control complexity is high, which also introduce large numbers of switches, multiplexers and control signals. The architecture is not regular and need to be modified for different number of level of DWPT computation. A folded architecture for lifting-based wavelet filters is proposed in [12] to compute the wavelet butterflies in different groups simultaneously at each decomposition level. According to the comparison results, the proposed architecture is more efficient than the previous proposed architectures in terms of memory access, hardware regularity and simplicity, and throughput. It is necessary to notice, that the architecture of the given processor is effective only for calculation of full tree DWPT. Here there is no technique of management for DWPT with best tree searching.

Algorithm transformation techniques have been employed in high-speed DSP system design is presented in [13]. All of the above mentioned techniques are applied during the processor design phase and their implementation is time invariant. Therefore, this class of signal processing techniques is referred as static techniques. Recently, dynamic techniques both of the circuit level and algorithmic level have been proposed [14]. These techniques are based on the principles that the input signal is usually non-stationary, and hence, it is better (from a coding perspective) to adapt the algorithm and architecture to the input signal. Such systems are referred to as reconfigurable signal processing systems [15],[16]. The key goal of these techniques is to improve the algorithm performance by exploiting variability in the data and channel.

Our approach is to design of dynamic algorithm transform (DAT) for design of application-specific reconfigurable lifting-based DWPT pipeline processor, in particular, for audio signal processing in real-time. The principle behind DAT techniques is to define parameter of input audio signals (subband entropy) and output encoded sequences (subband rate) for the given embedded processor architecture. Adaptive wavelet analysis for audio signal processing purposes is particularly interesting if the psychoacoustic information is considered in the DWPT decomposition scale. Due to the lack of selectivity of wavelet filter banks, the psychoacoustic information is computed in the wavelet domain.

2. Flexible tree structured signal expansion based on DWPT

DWPT algorithm is a generalization of the discrete wavelet transform that can be represented as a filter bank with a tree structure [3] (see figure 1). Within a given node number n of the tree

http://dx.doi.org/10.5772/51604

at any level l ($n = 0 \,..\, .2^{l-1}, l \in Z$) input $x_{l,\,n,k}$, (k– signal samples) is separated by low-frequency (LF) $x_{l+1,2\cdot n,k}$ and high frequency (HF) $x_{l+1,2\cdot n+1,k}$ components using a pair of wavelet filters $h(z)$ and $g(z)$ with finite impulse response (FIR), after which each subband signal down-sampling by factor of two. Function block that implements this separation of the input signal is called a dual-channel filter bank analysis.

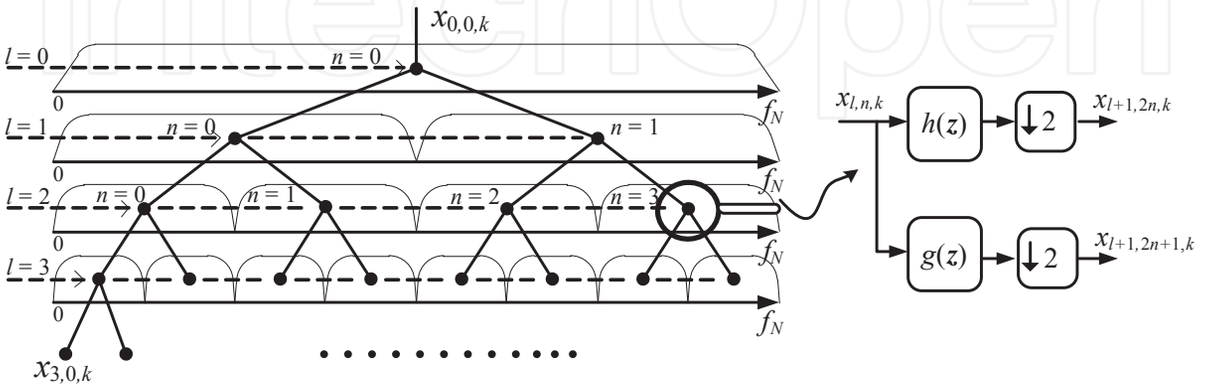


Figure 1. DWPT tree structure (left) and dual-channel filter bank – wavlet baterfily (right)

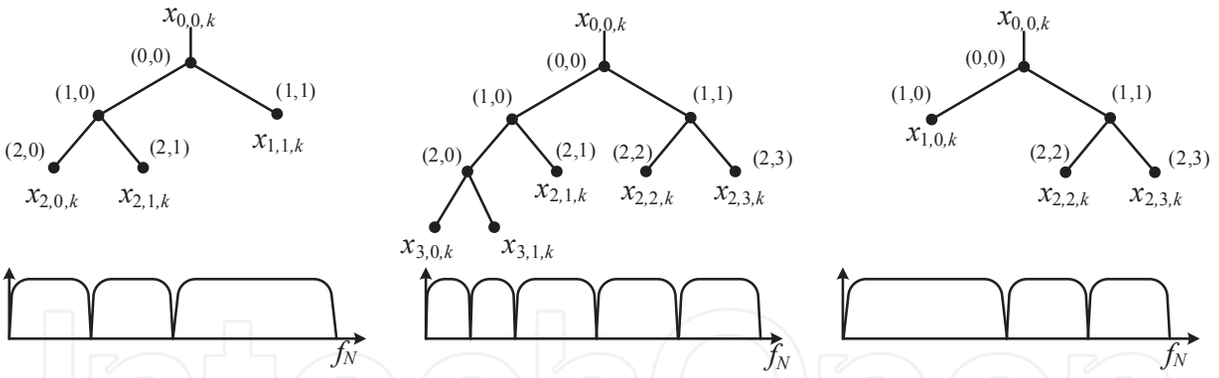


Figure 2. DWPT tree structure examples and corresponding magnitude response of then filter bank

Thus, a specific node (l, n) corresponds to the frequency range ($n \cdot 2^{-l}$, $(n+1) \cdot 2^{-l}$), normalized to the Nyquist frequency (f_N). At each level of decomposition frequency resolution increases twice, but twice the resolving power decreases over time. DWPT is a complete decomposition of the signal in the low and high frequencies. Variation of the resolution in frequency and time domains allows for a more detailed decomposition, for example, in the lower frequencies, which leads to an increase in the frequency resolution and reduced over time. This feature has adapted DWPT. The advantage is the ability to DWPT sufficiently flexible selection of tree decomposition (see figure 2), based on the nature of the signal. The choice of tree structure can

be performed based on pre-known features of the signal, and executed dynamically, "arranged" for the current frame processing [2].

3. Dynamic transformation of DWPT decomposition

We present adaptive DWPT tree derived via DAT's. The principle behind DAT is to define parameter of input signals (subband entropy) and output sequences (subband rate) for the given embedded processor architecture. In other hands, DAT techniques is to construct a minimum cost subband decomposition of DWPT by maximizing the minimum masking threshold (which is limited by the perceptual entropy (PE)) in every subband for the given embedded processor architecture and temporal resolution. Achieving this purpose, we suppose that the tree structure of DWPT decomposition is adapted, as closely as possible, to the critical bands ($CB\text{ - }WPD:(l,\ n)\in E_{CB}$) as shown in [14]. For the DWPT tree structure E_i the information density H belong to tree E_i is estimated as

$$H_{E_i}=\sum_{\forall(l,n)\in E_i}\sum_{k} w_{E_i}(k)\cdot ln\left(w_{E_i}(k)\right), \tag{1}$$

where

$$w_{E_i}(k)=\frac{|x_{l,n,k}|}{\sum_{\forall(l,n)\in E_i}|x_{l,n,k}|}, \tag{2}$$

here $x_{l,n,k}$ are wavelet coefficients, l is a decomposition level, n is the node number of decomposition level, k is the index of the current wavelet coefficient of the node $(l,\ n)$. H_{E_i} is estimates based on the wavelet coefficients of terminated nodes (nodes is a grey area in a figure 3).

The growing decision for DWPT tree based on the given H is being taken in terms of allowing the further decomposition of the WP tree can be expressed as:

$$H_{E_i}<H_{E_{i-1}}. \tag{3}$$

If (3) is true we continue the subband splitting process in DWPT tree, otherwise the suboptimal decomposition for the given frame of signal is founded.

The subband splitting process is managed based on the estimated values of PE in parent and child nodes of current DWPT tree structure. PE estimation is described in [17],[18],[19] and expressed as

$$PE_{l,n}=\sum_{k=0}^{K_{l,n}-1} log_2\left(2\left[n\ \text{int}\left(SMR_{l,n,k}\right)\right]+1\right), \tag{4}$$

http://dx.doi.org/10.5772/51604

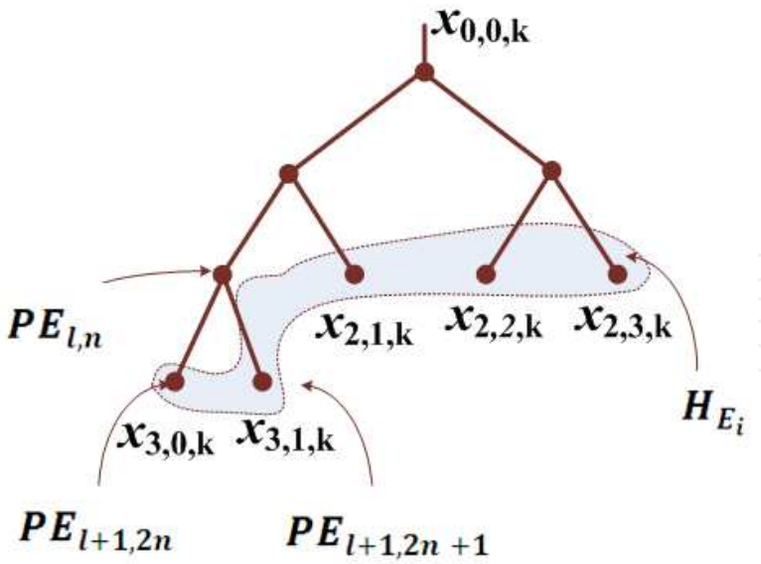


Figure 3. DWPT tree growing process

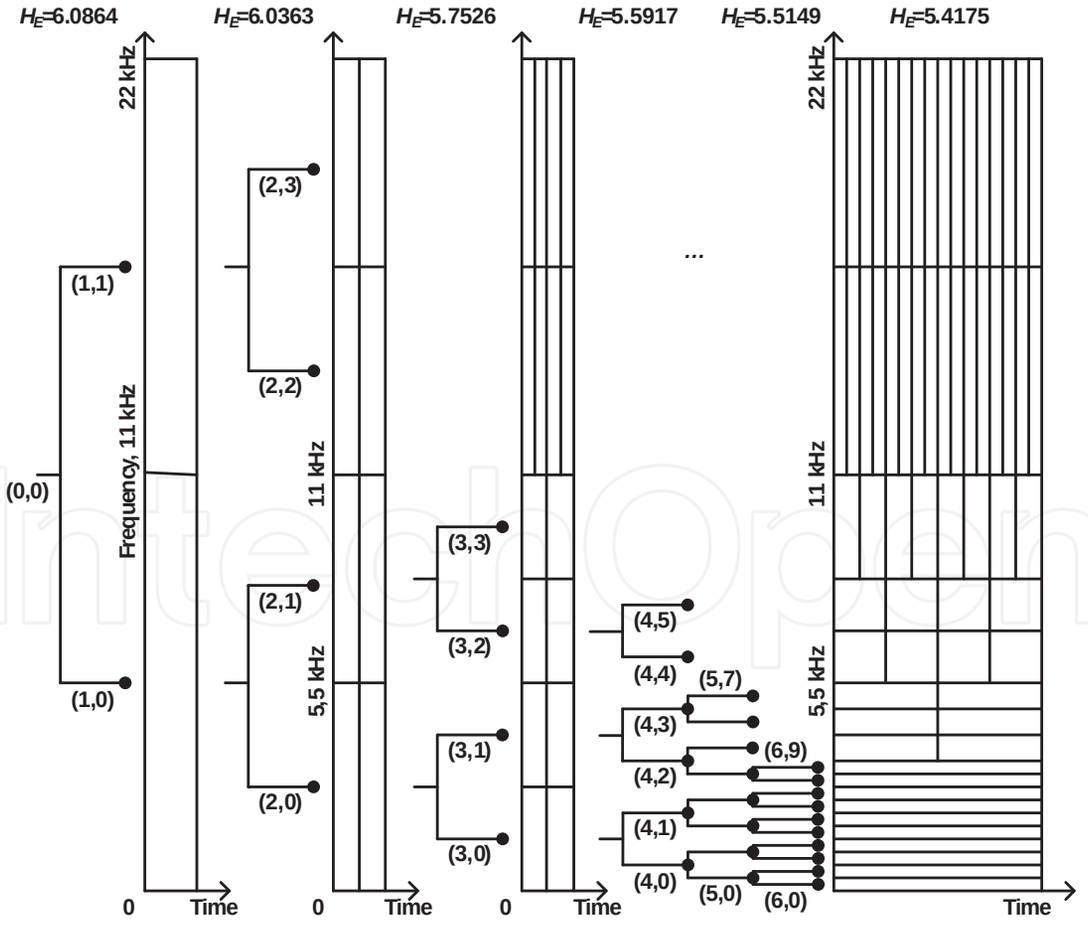


Figure 4. DWPT tree structure creation and corresponding time-frequency tilling map

where $SMR_{l,n,k}$ is a ration between the absolute value of the wavelet coefficients $x_{l,n,k}$ in a subbabnd of tree E_i (node $(l,\ n)$), and the corresponding masking threshold $T_{l,n}$, which is linearly spread among the $K_{l,n}$ coefficients $x_{l,n,k}$, $k = \{0,\ K_{l,n}\}$ of node $(l,\ n)$. The large magnitude of $SMR_{l,n,k}$ determines node $(l,\ n)$ significance for PE formation.

Each allowed parent node $(l,\ n)$ is split on two child nodes $(l+1,2n)$ and $(l+1,2n+1)$, if and only if the sum of $PE_{l+1,2n}$ and $PE_{l+1,2n+1}$ in the child nodes less than in the current node $PE_{l,n}$, that can written as.

$$PE_{l,n} > PE_{l+1,2n} + PE_{l+1,2n+1}. \tag{5}$$

Schematically the DWPT tree growing process is shown in the figure 3. The example of dynamic DWPT tree structure growing level by level based on H and corresponding time-frequency tilling map are demonstrated in the figure 4.

Applying the information density H, the perception entropy PE, the limited WP tree structure $CB - WPD$ and the maximum allowed computation resource together in DWPT growing procedure allows us to found suboptimal solution for input signal analysis on the given hardware architecture.

4. DWPT implementation based on lifting scheme

4.1. Factoring wavelet filters in to lifting steps

In the tree-based scheme of the DWPT, each node of the tree consists of a two-channel filter bank. Each node can be broken down into a finite sequence of simple filtering steps, which are called lifting steps or ladder structures. In [20] for two-channel filter bank proposed a method of transition from the implementation on the basis of FIR filters to architecture at the based on lifting scheme. The decomposition is essentially a factorization of the polyphase matrix of the wavelet filters into elementary matrices. As discussed in [20], the lifting steps scheme consists of three phases: the first step splits the data into two subsets: even and odd; the second step recalculates the coefficients (high-pass) as the failure to predict the odd set based on the even; finally the third step updates the even set using the wavelet coefficients to compute the scaling function coefficients (low-pass). This method allows to reduce on halve the number of multiplications and summations. In terms of the z-transform transition to the implementation of the filter bank based on the lifting scheme can be viewed in two steps.

The first step is to move towards the implementation of polyphase filtering algorithm [20]. The process of calculating the LF and HF components of the signal $x_{l,n,k}$ in any node of the tree can be written as the following expression:

$$\left[X_{l+1,2n}(z) \quad X_{l+1,2n+1}(z)\right] = \left[X^{e}_{l,n}(z) \quad z^{-1}X^{o}_{l,n}(z)\right] \cdot \tilde{\mathbf{P}}, \tag{6}$$

http://dx.doi.org/10.5772/51604

where $X_{l+1,2\,n}(z)$ and $X_{l+1,2\,n+1}(z)$ are z-representation in the field of low and high frequency components, $X^{e}_{l,\,n}(z)$, and $X^{o}_{l,n}(z)$ are representation of sequences, respectively, consisting of the even and odd samples the input sequence $x_{l,n,k}$, $\tilde{\mathbf{P}}$ is a polyphase matrix that can be written as

$$\tilde{\mathbf{P}} = \begin{bmatrix} \tilde{h}_e(z) & \tilde{g}_e(z) \\ \tilde{h}_o(z) & \tilde{g}_o(z) \end{bmatrix}. \tag{7}$$

The elements $\tilde{h}_e(z)$, $\tilde{h}_o(z)$ and $\tilde{g}_e(z)$, $\tilde{g}_o(z)$ of polyphase matrix from (7) are in the following dependence to the original coefficients of low-pass $\tilde{h}(z)$ and high-pass $\tilde{g}(z)$ wavelet filters correspondingly:

$$\tilde{h}(z) = \tilde{h}_e(z^2) + z^{-1}\tilde{h}_o(z^2), \tag{8}$$

$$\tilde{g}(z) = \tilde{g}_e(z^2) + z^{-1}\tilde{g}_o(z^2). \tag{9}$$

This approach does not give the gain to the computational cost, but the hardware implementation allows us to reduce the operation frequency in double in compare with input data rate due to parallel computation (see figure 5).

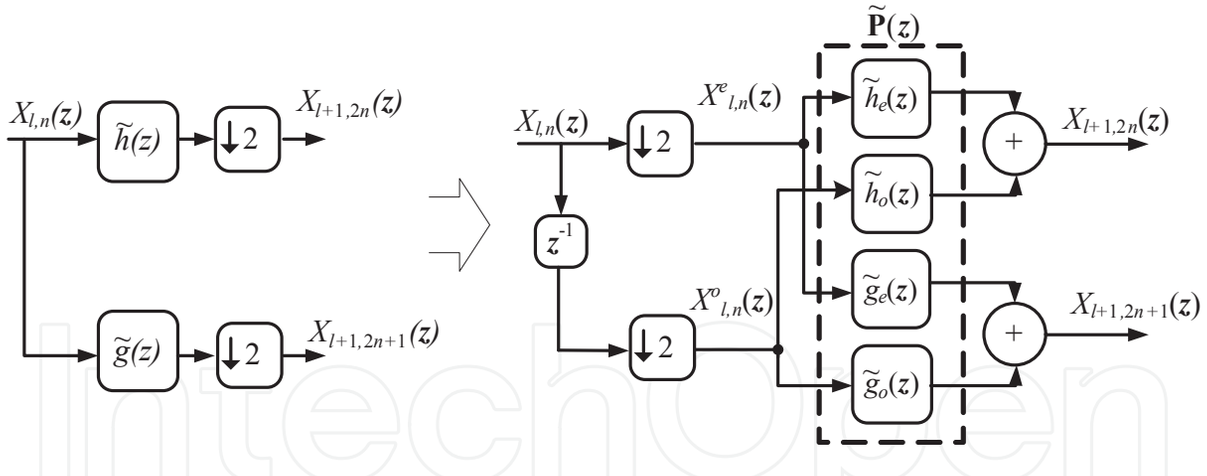


Figure 5. The transition to the polyphase implementation of analysis filter bank

	$s_1(z)$	$s_2(z)$	$s_3(z)$	$t_1(z)$	$t_2(z)$
b_0	3.1029	-5.1995	0.3141	0.0763	-3.1769
b_1	0	1.6625	0	-0.2920	-0.0379
u	0	-1	-3	1	3

Table 1. The parameters of lifting scheme for db4 (8 taps)

The second step is a factorization of the polyphase matrix into simpler triangular matrices. The result is, in a general, the original matrix $\tilde{\mathbf{P}}$ that can be expressed as

$$\tilde{\mathbf{P}}=\prod_{i=1}^{I/2}\left(\begin{bmatrix}1 & \tilde{s}_i(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \tilde{t}_i(z) & 1\end{bmatrix}\right)\begin{bmatrix}c_1 & 0\\ 0 & c_2\end{bmatrix}, \tag{10}$$

where I is the number of elementary triangular matrices derived from the factorization of polyphase matrices; $\tilde{s}_i(z)$ and $\tilde{t}_i(z)$ are low-order polynomials; c_1, c_2 are real coefficients. In general, the polynomials $\tilde{s}_i(z)$ and $\tilde{t}_i(z)$ can represented as $\left(b_0+b_1z^{-1}\right)z^{u}$, where b_0, b_1 are constants, u is the integer exponent. For example, the b_0, b_1 and u parameters of lifting scheme for db4 wavelet mother function are presented in table 1. K_1 and K_2 are equal -0.1202 and -8.3192 correspondingly. For fixed point DWPT implementation an arithmetic with an arbitrary number of integer and fractional bits is used as proposed in [21],[22]. The advantage of this number representation is the fact that it can be realized using conventional integer arithmetic resource.

The scaling $X_{l+1,2n}(z)$ and wavelet $X_{l+1,2n+1}(z)$ coefficients relative to the input signal $X_{l,n}(z)$ in z domain are two-channel analysis filter bank results in according to (6) and (10) can be written as follows:

$$\left[X_{l+1,2n}(z)\quad X_{l+1,2n+1}(z)\right]=\left[X_{l,n,e}(z)\quad X_{l,n,o}(z)\right]\times\prod_{i=1}^{I/2}\left(\begin{bmatrix}1 & \tilde{s}_i(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \tilde{t}_i(z) & 1\end{bmatrix}\right)\begin{bmatrix}c_1 & 0\\ 0 & c_2\end{bmatrix}, \tag{11}$$

where $X_{l,n,e}(z)$ and $X_{l,n,o}(z)$ are z-representation of two sequences consisting of even and odd samples of the input signal $x_{l,n,k}$. The block diagram for the direct implementation of two-channel analysis filter bank based on lifting scheme is shown on figure 6.

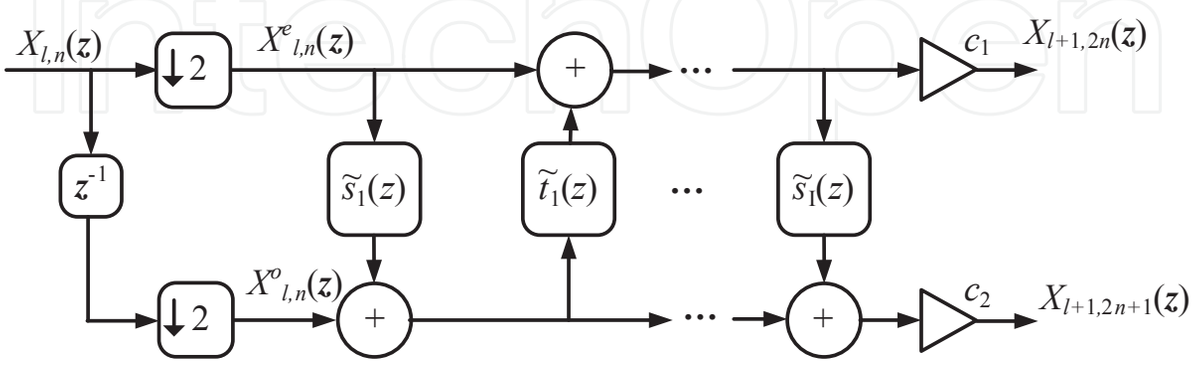


Figure 6. Block diagram of two-channel analysis filter bank based on the lifting scheme

The inverse decomposition of a two-channel filter bank in the same terms can be expressed as

http://dx.doi.org/10.5772/51604

$$\left[\hat{X}^{e}_{l,n}(z) \quad z^{-1}\hat{X}^{o}_{l,n}(z)\right]=\left[X_{l+1,2n}(z) \quad X_{l+1,2n+1}(z)\right]\begin{bmatrix}1/c_1 & 0\\ 0 & 1/c_2\end{bmatrix}\prod_{i=I/2}^{1}\left(\begin{bmatrix}1 & 0\\ -\tilde{t}_i(z) & 1\end{bmatrix}\begin{bmatrix}1 & -\tilde{s}_i(z)\\ 0 & 1\end{bmatrix}\right), \tag{12}$$

and corresponding block diagram of two-channel synthesis filter bank based on the lifting scheme shown in a figure 7.

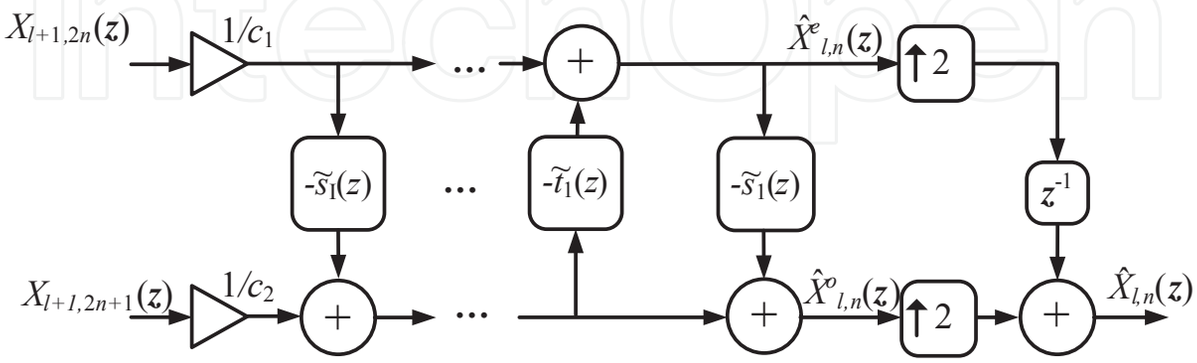


Figure 7. The block diagram of two-channel synthesis filter bank based on lifting scheme

According to the block diagram (see figure 7), the synthesis procedure is implemented as follows: at first, the input coefficients $X_{l+1,2n}(z)$ and $X_{l+1,\,2n+1}(z)$ of each channel are multiplied on the coefficients $1/c_1$, $1/c_2$ correspondingly, at second, two-channel synthesis filter bank implements the inverse operations of algorithm analysis. This implementation uses the same polynomial $\tilde{s}_i(z)$ and $\tilde{t}_i(z)$ from (10) with opposite signs. Reconstructed signal $X_{l,\,n}(z)$ is obtained from the calculating sequences $X^{e}_{l,n}(z)$, $X^{o}_{l,n}(z)$.

Together the analysis and synthesis filter bank implementation based on lifting scheme is required the same number of operations (summation and multiplication) in compare with analysis only implementation based on regular FIR filter implementation.

4.2. The algorithm implementation based on fixed-point variable format arithmetic

A number of the target application requirements (work in real time, greater throughput, and other) make it necessary to use fixed-point arithmetic to perform the specified computation. With the implementation of two-channel filter bank based on lifting structures using integer arithmetic, the number of difficulties arise, related to the fact that values of the coefficients of the polynomials $s_i(z)$ and $t_i(z)$ can take both fractional and great integer values (that is well-known negative effect of polyphase matrix factorization). This feature causes to an increase of the arithmetic units and the word length of internal registers. Therefore, in this paper for the implementation of the algorithm DWPT on fixed-point arithmetic the approach based on [21], [22] is used, according to which the format of the numbers involved in the intermediate computation, is variable. This method assumes that the number of bits to be allocated under the integer and fractional parts of numbers in the different nodes of the algorithm is different.

In accordance with this approach, any number represented in two's complement fixed-point format, is given in the form of expression:

$$a = ma \cdot 2^{ex\,p_a}, \quad \text{where } ma = (-1)^s + \sum_{i=0}^{wl-2} a_i \cdot 2^{i-wl+1}. \tag{13}$$

Here, ma– value of the number presented in two's complement code that is interpreted as a fraction in the range $[-1,1)$; $ex\,p_a$ – the order of the scaling factor $2^{ex\,p_a}$; a_i - value of the i-th bit of the number equal to 0 or 1; s – the sign bit; wl – the word length. Thus for intermediate data in different nodes of the algorithm its value $ex\,p_a$is determined. So, depending on the $ex\,p_a$ value the redistribution of bits for fractional and integer part of number at different sites of the algorithm is produced (see figure 8).

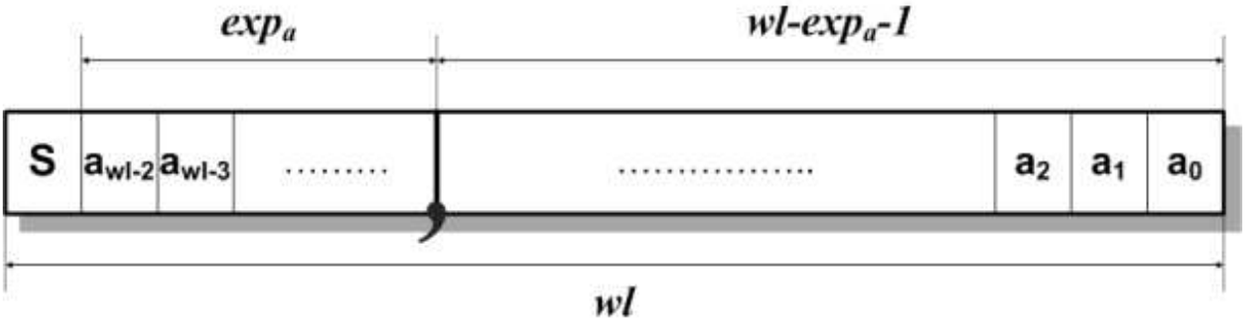


Figure 8. Data format in fixed-point arithmetic with variable word length

For a given in (13) format, the operations of addition and multiplication of a and b ($ex\,p_b \geq exp_a$) are defined as

$$c = a + b = mc \cdot 2^{ex\,p_c} = \left(ma \cdot 2^{ex\,p_a - ex\,p_b} + mb\right) \cdot 2^{ex\,p_b}, \tag{14}$$

$$c = a \cdot b = mc \cdot 2^{ex\,p_c} = ma \cdot mb \cdot 2^{ex\,p_a + ex\,p_b}. \tag{15}$$

The figure 9 schematically illustrates the process of performing operations described above in (14) and (15).

In this paper a generic set of processing elements is proposed to implement of analysis and synthesis banks on the lifting structures using variable arithmetic format (see table 2). In this table $x_e[k]$, $x_o[k]$ – input values, and $y_e[k]$, $y_o[k]$ – output values, respectively, in the upper and lower channels filter bank analysis (synthesis) in the k -th time value. The parameters $s0$, $s1$, $s2$ define the arithmetic shift values. These parameters are computed according to (14) and (15) for each node of the algorithm.

http://dx.doi.org/10.5772/51604

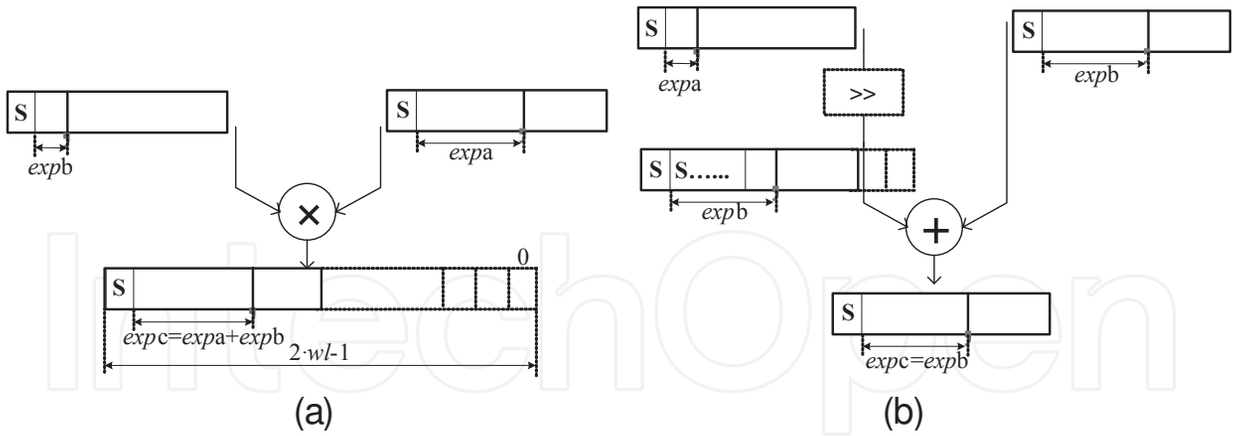


Figure 9. Performing operations of a) addition and b) multiplication

Analysis bank		**Syntesis bank**	
Symbol	Computational operations	Symbol	Computational operations
S*1*	$y_e[k]=x_e[k]$; $y_o[k]=x_o[k]\gg s0+(x_e[k]\cdot mb^0)\gg s1$	$S1^{-1}$	$y_e[k]=x_e[k]$; $y_o[k]=\left(x_o[k]-(x_e[k]\cdot mb^0)\gg s1\right)\ll s0$
S*2*	$y_e[k]=x_e[k]$; $y_o[k]=x_o[k]\gg s0+(x_e[k]\cdot mb^0)\gg s1+$ $+(x_e[k-1]\cdot mb^1)\gg s2$;	$S2^{-1}$	$y_e[k]=x_e[k]$; $y_o[k]=\left(x_o[k]-(x_e[k]\cdot mb^0)\gg s1-\right.$ $\left.-(x_e[k-1]\cdot mb^1)\gg s2\right)\ll s0$
T1	$y_e[k]=x_e[k]\gg s0+(x_o[k]\cdot mb^0)\gg s1$ $y_o[k]=x_o[k]$;	$T1^{-1}$	$y_e[k]=(x_e[k]-(x_e[k]\cdot mb^0)\gg s1)\ll s0$ $y_o[k]=x_o[k]$;
T2	$y_e[k]=x_e[k]\gg s0+(x_o[k]\cdot mb^0)\gg s1+$ $+(x_o[k-1]\cdot mb^1)\gg s2$; $y_o[k]=x_o[k]$;	$T2^{-1}$	$y_e[k]=\left(x_e[k]-\left(x_e[k]\cdot mb^0\right)\gg s1-\right.$ $\left.-\left(x_o[k-1]\cdot mb^1\right)\gg s2\right)\ll s0$ $y_o[k]=x_o[k]$;

Table 2. A set of processing elements for the based on the lifting structures algorithm using arithmetic variable format

For an explanation of the practical aspects related to the use of the proposed arithmetic, below an example of the two-channel analysis bank realization on the mother wavelet function db4 [23] is considered. In result of polyphase matrix factorization for a given wavelet basis in the MATLAB environment the following expression was obtained:

$$\tilde{\mathbf{P}}=\begin{bmatrix}1 & b_1^0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \left(b_2^0+b_2^1z^{-1}\right)z & 1\end{bmatrix}\begin{bmatrix}1 & \left(b_3^0+b_3^1z^{-1}\right)z^{-1}\\ 0 & 1\end{bmatrix}\cdot\begin{bmatrix}1 & 0\\ \left(b_4^0+b_4^1z^{-1}\right)z^3 & 1\end{bmatrix}\begin{bmatrix}1 & b_5^0z^{-3}\\ 0 & 1\end{bmatrix}\begin{bmatrix}c_1 & 0\\ 0 & c_2\end{bmatrix}. \tag{16}$$

The coefficients b_m^n, and also their parameters *mb* and *expb*, calculated in accordance with (13), are presented in table 3.

Lifting step number, i	Type	u	b_i^0			b_i^1		
			Value	*mb*	*expb*	**Value**	*mb*	*expb*
1	$s_1(z)$	0	-3,1029	-0,7757	2	0	0	-
2	$t_2(z)$	1	-0,0763	-0,6104	-3	0,2920	0,5840	-1
3	$s_3(z)$	-1	5,1995	0,6499	3	-1,6625	-0,8313	1
4	$t_4(z)$	3	3,1769	0,7942	2	0,0379	0,6064	-4
5	$s_5(z)$	-3	0,3141	0,6282	-1	0	0	-

Table 3. The lifting structures parameters calculated for the wavelet filters db4 (8 taps)

In the figure 10 shown a block diagram of the implementation of two-channel filter bank analysis for this example. In this scheme, apart from computing elements $S1$, $S2$, $T2$ (see table 2) in the upper channel of the bank to satisfy the condition of causality delay registers are inserted (elements z^{-l}, $l \in \mathbb{Z}$).

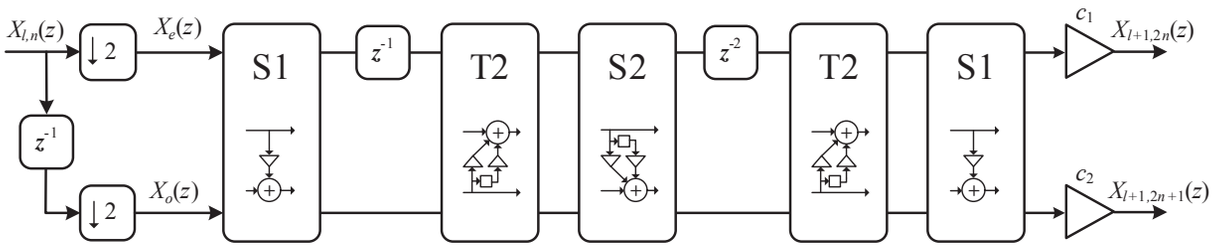


Figure 10. Block diagram of two-channel filter bank based on the lifting structures for db4 (see table 2)

In the figure 11a in more detail the first step realization of the analysis bank is considered and in the figure 11b the last step realization of synthesis bank is shown.

As can be seen from the figure 10 and figure 11, the computing units of analysis and synthesis procedures in terms of implementation differ only in the signs of constant multiplying coefficients and the arithmetic shifts positions and directions.

Based on the materials described above, a concrete realization of two-channel bank can be represented as a vector of parameters containing a set of multiplier constants, shift parameters and some additional information regarding the delay elements in the intermediate nodes of the algorithm.

4.3. Accuracy analysis of the algorithm for fixed-point variable format

To analyse of the proposed approach in MATLAB function library was written, which simulates the process of fixed-point calculating in filter bank with specified structure of the tree. In the figure 12 is shown the estimation error variance signal recovery, depending on the

http://dx.doi.org/10.5772/51604

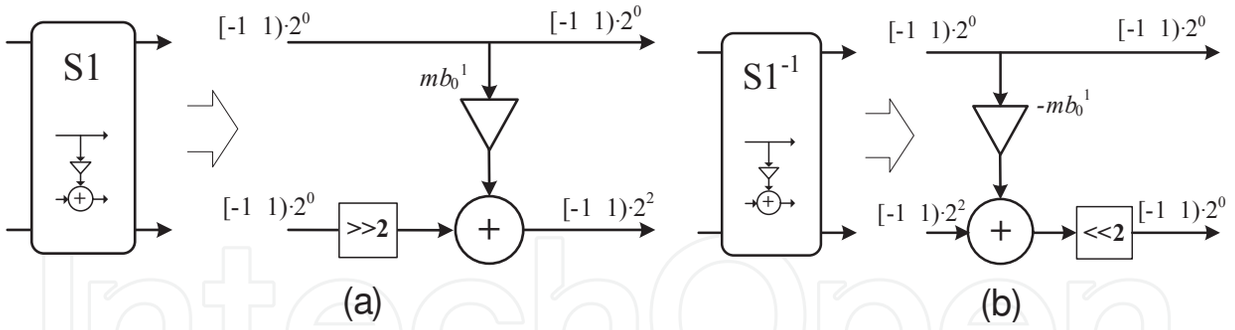


Figure 11. First step of the analysis (a) and the last step of the synthesis (b) bank implementation

choice of bit internal registers as a result of passing through the two-channel filter bank analysis / synthesis (in the example used wavelet filters db8). This figure also shows the results of an experiment using FIR filters, the underlying of the algorithm DWPT. It can be noted that FIR filter implementation gives better results while using the same registers word length, but requires twice as many calculations. So in order to achieve the level of error variance in the -70 dB for based on lifting structures implementation requires 16-bit, which are approximately two bits more than the realization based on FIR. But this drawback is compensated by a significant reduction of arithmetic operations compared to the direct implementation. Thus, we conclude that the proposed approach is more efficient in hardware implementation compared with the bank on the basis of FIR filters.

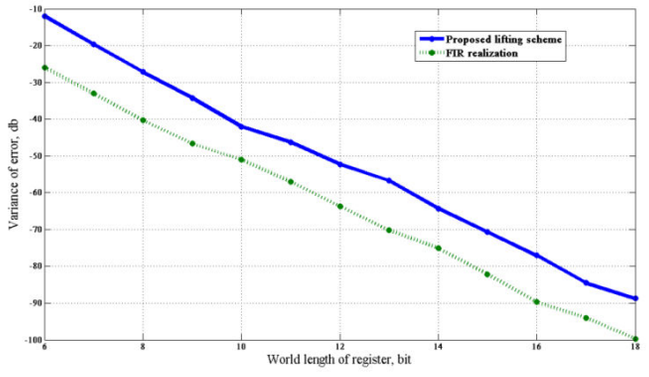


Figure 12. The variance of the reconstruction signal error dependence on the registers bit capacity in analysis/synthesis two-channel filter bank systems (wavelet filters are db8): the solid line indicates the results for the proposed algorithm on the lifting structures, dotted - integer implementation of the same system using a FIR filter

Below we consider another experiment for demonstration of the energy localization properties by using our fixed-point DWPT algorithm realization. For this a polyharmonic signal was generated and passed through a five-level decomposition tree fast wavelet transform (division of the tree is carried out only in the low-frequency components). As an example, the wavelet functions db2 family was chosen. Thus all range of amplitudes of the wavelet coefficients was divided by 40 thresholds (these values are plotted on the X-axis of figure 13). Each threshold

has been mapped to a vector of the obtained analysis wavelet coefficients on condition that these coefficients are greater than this threshold. Otherwise, the values of the coefficients were replaced by zeros (i.e. in each vector were discarded unimportant relative to a given threshold values). For all vectors was performed reconstruction procedure by synthesis filter bank. In figure 13a, figure 13b for the floating-and fixed-point implementations, respectively, are shown: solid line – the reconstructed to original signal energy relation (in percentage) depending on the threshold values; dotted line – the percentage of "discarded" wavelet coefficients depending on the chosen threshold.

Based on these results, we note almost complete compliance of floating-point model with the proposed fixed-point approach. Thus, the fixed-point variable format algorithm DWPT implementation preserves the energy localization inherent to the wavelet packets.

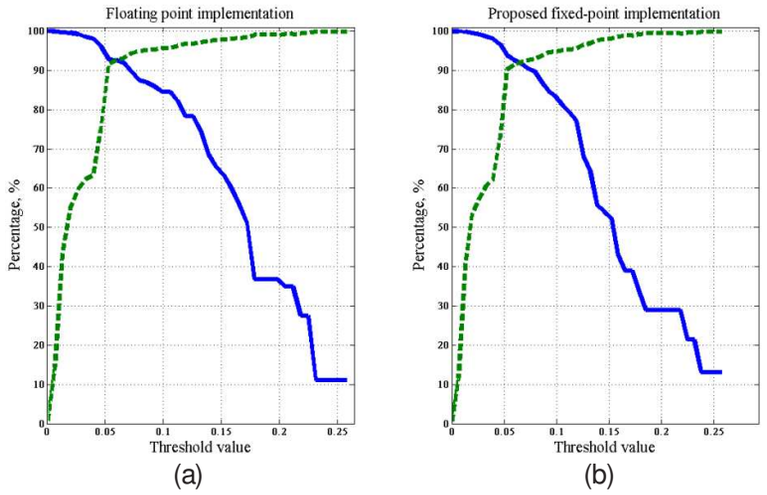


Figure 13. Energy estimates of signal reconstruction, depending on the threshold of significant wavelet coefficients for the based on a floating-point (a) and proposed fixed-point (b) DWPT algorithm implementations

5. DWPT pipeline processor with dynamic reconfigurable architecture

5.1. DAT based reconfigurable signal processing system

The structure of reconfigurable DSP system for signal analysis based on DAT approach consists of the specific microprocessor oriented on the signal processing (DSP microprocessor) and DWPT processor itself with the reconfigurable architecture. The DSP microprocessor perform several task, such as: processing wavelet coefficients $X_{l,n,k}$ in subbands $(l,\ n)$ that corresponds to the current DWPT tree structure E_i; estimate H_{E_i} and $PE_{l,n}$; obtain the reconfiguration vector for DWPT processor $r_{l,n}$, $(l,\ n) \in E_i$. DWPT processor is realized on pipeline architecture with

http://dx.doi.org/10.5772/51604

dynamic reconfiguration for implementing adaptive DWPT. The length of the pipeline is obtained from the limited DWPT tree structure (*CB - WPD*). A great dependence of the process on the DWPT structure grows leads to the necessity of introducing an easily reconfigurable parallel-pipeline structure with computation resource *C*. Thus, the DSP system for audio processing based on DAT-approach consists as shown on figure 14.

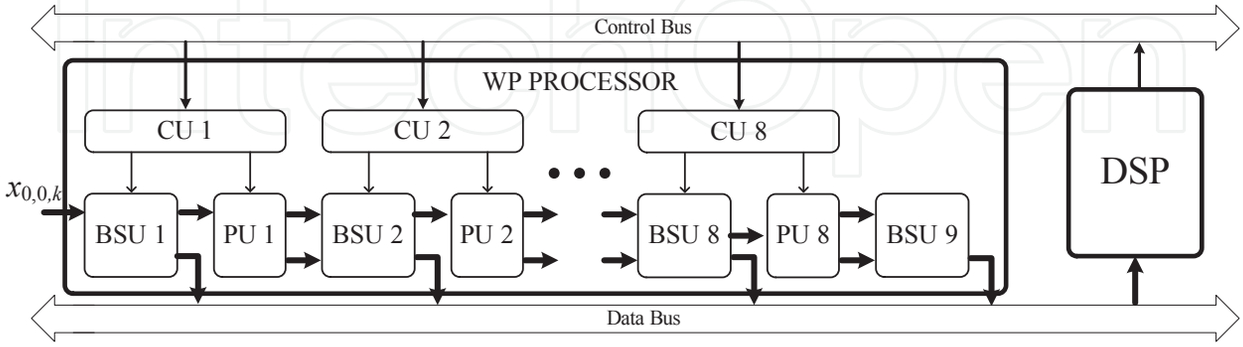


Figure 14. DAT-based reconfigurable signal processing system

The pipeline architecture is applied for effective implementation of the DWPT algorithm. We suggest the pipeline architecture for constructing the DWPT lifting based processor. This architecture integrates the sequential connection of the homogeneous block (buffer/switch unit (BSU) and processing unit (PU)) that implement a two-channel filter bank that allows parallel calculating DWPT with an arbitrary tree structure. The maximum number of decomposition level that can be realize is 8, it is associate with the depth of *CB - WPD*. The basic decomposition of DWPT expressed as PU which acts a two-channel filter bank based on the lifting scheme. The reconfiguration vector $r_{l,n}$ decoding, memory address generation, PU enabling, data exchange controlling and the pipe-line synchronizing are performed by the control units (CU). All this functions in CU at each DWPT processor stages is carried out in parallel. The pipe-line DWPT processor stages are synchronized according to the DAT's techniques.

5.2. DWPT lifting based pipeline Processing Unit (PU)

The block diagram of PU is shown in the figure 15. The input sequence $x_{l,n,k}$ is split into even X_e and odd X_o samples in PU before the processing is started according to the lifting scheme. The structure of PU has the following abbreviations (see figure 15): wl is a bit capacity; I is a number of elementary steps of the lifting scheme; V_{PE} is a vectors, each element of it is a set of the parameters of the same elementary step of the lifting scheme; V_{BUF} is a vector, each element of it specify the number of delays, respectively in the upper and lower channels after the same elementary step. The present elements corresponds to FIFO registers that, on the one hand are delay elements z^{-1} in the algorithm and, on the other hand, makes possible a pipelined realization in architecture for throughput performance increasing. The coefficients c_1 and c_2 applies to the result of lifting scheme as it described in (11). The estimated hardware resources required for PU implementation are shown in a table 4.

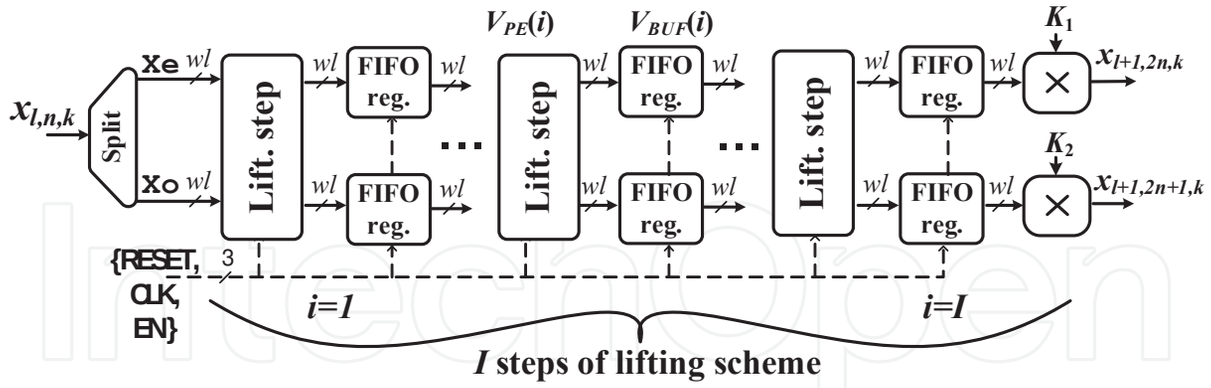


Figure 15. Block diagram of the PU

Resource type	Utilized
Multipliers wl×wl	*N*+2
Adders(wl – bit capacity)	N
Registers(wl – bit capacity)	*N*+1
Multiplexers 2-in-1 (wl – bit capacity)	1

Table 4. Estimation of hardware resources for PU implementation (N – is the number of filter taps)

5.3. Buffer/Switch Unit (BSU)

The BSU realizes double buffering scheme known as "ping-pong" for providing parallel access to the data for storing results and getting source data from/for PU. The additional channel is for outputting the result data. The two output streams of samples $x_{l+1,2n,k}$ and $x_{l+1,2n+1,k}$ from l-th PU are stored in BSU and simultaneously $l+1$-th PU can get the samples for the next processing stage. Unified block diagram of BSU is represented in the figure 16. Each BSU in parallel-pipeline architecture has addressed a different memory size that depends on the DWPT decomposition level.

The momory amount M_V (taking into account the requirement of double buffering) and the number of processing units of L, can be expressed as

$$M_V = 2 \cdot \sum_{j=1}^{J} \frac{K}{2^{l_j}},\ L = \max_{j=1..J} l_j \tag{17}$$

where J is amount of all nodes CB-WPD, l_j is a decomposition level, K is a initial frame length of input signal.

http://dx.doi.org/10.5772/51604

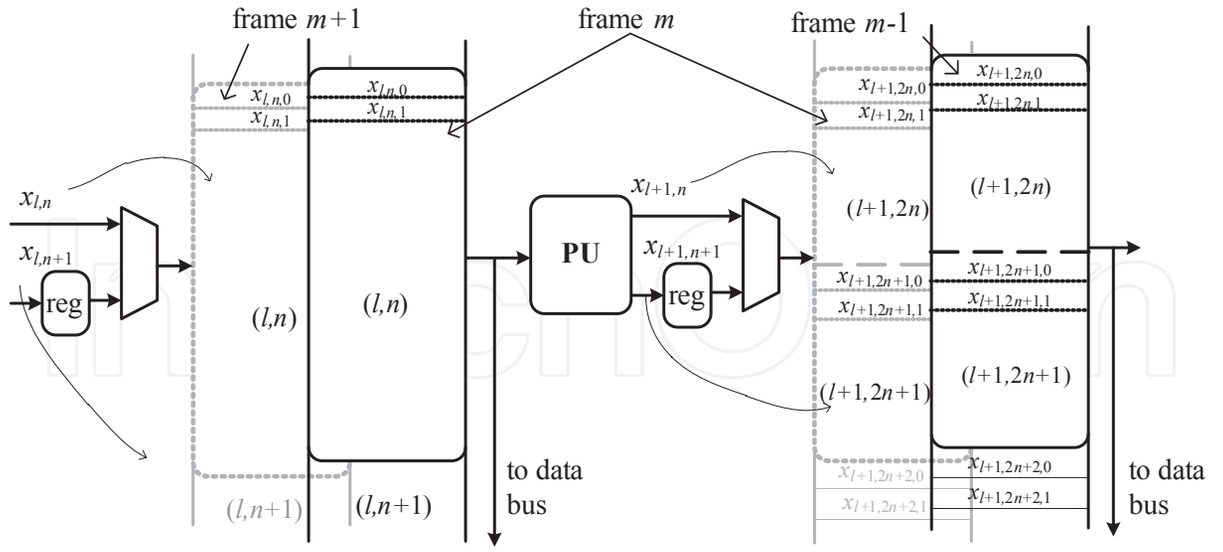


Figure 16. Unified block diagram of the BSU

In the figure 17 is shown an example of the tree decomposition, given by the set of nodes {(0,0), (1,0), (1,1), (2,0), (2,1), (2,2), (2.3), (3,0), (3,1)}. It also schematically illustrates the principle of distribution of blocks of memory for the structure of the tree.

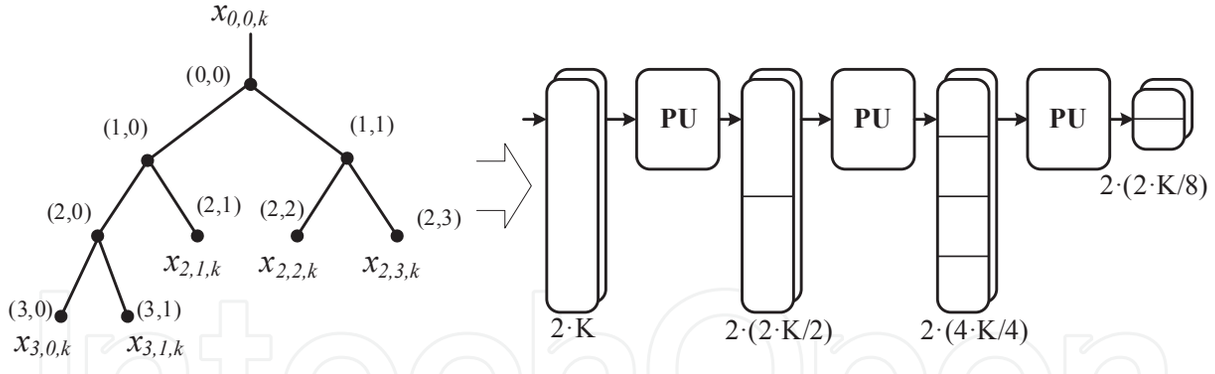


Figure 17. The parallel pipeline architecture for three-level DWPT tree structure

5.4. Rapid prototyping algorithm of pipeline DWPT processor

The prototype of the DWPT processor can be specified as parameters describing the structure of two-channel filter bank and the vector that defines the limit tree decomposition.

The method of rapid prototyping can be described by the following sequence of actions.

1. Calculating the lifting structure of the dual filter bank based on the original wavelet basis functions.

2. Translating the mathematical model for fixed-point arithmetic with the requirements of accuracy, and limitation of hardware resources (registers and bit computing units).
3. Forming a parameters vector for configure a DWPT processor prototype.
4. Estimating the cost of hardware prototype implementation.
5. Estimating computation characteristics of the DWPT procressor prototype.
6. Generating the output files of the synthesized VHDL-description of the DWPT processor.

5.5. FPGA based hardware implementation of the pipeline DWPT processor

For estimation of performance and resource utilization the present architecture has been implemented on Xilinx FPGA XC3s2000. The realized pipeline DWPT processor has following features. The number of decomposition levels is limited by eight. The mother wavelet function Db8 (16 taps) transformed into nine lifting steps is used. The input and output data has the 16 bits word length, the capacity of internal computing is 18 bits. The present implementation doesn't have FIFO stages in PU that allows minimizing hardware resources. The processed frame size can be selected in a range from 128 to 1024 samples. Each BSU contains the pair of two 1024×16 bits block RAMs that is used for realizing double buffering scheme. The PU hardware resources utilization are shown in a table 5 and complete processor implementation resources are presented in table 6.

Resource type	Utilized
4 input look-up tables	788
flip-flops	226
MULT18x18s	18

Table 5. Estimations of hardware resource for FPGA-based PU implementation.

Resource type	Utilized, pcs.	Percentage wise, %
4 input look-up tables	31356	76
flip-flops	3037	7
RAMB16	16	40
MULT18x18s	40	100

Table 6. Hardware resource estimations for WP processor implementation on XC3s2000

In the figure 18 the protopyte board of dynamic reconfigurable pipeline DWPT processor is shown.

The implemented design performance is 8 MSPS. So, if the sample rate of input audio signal is 44100 Hz then the time cost for computation of wavelet coefficients is 0.6% from all time

http://dx.doi.org/10.5772/51604

resource. For example, the 512 samples frame size (~11.6 ms) processing take approximately 0.064 ms on presented DWPT lifting based pipeline processor. The rest time is distributed between the dynamic DWPT tree decomposition algorithms, wavelet coefficients post-processing and transfer operation.

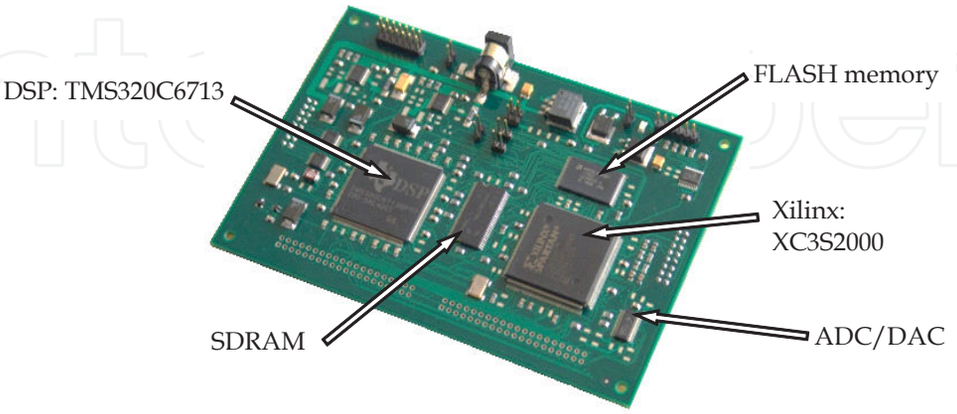


Figure 18. Protopyte board of dynamic reconfigurable pipeline DWPT processor

5.6. DAT based dynamic reconfigurable architecture algorithm

Suppose that for some audio input frame is the space of trees structures E, which is processing a stream-flow or parallel reconfigurable processor $(m, r_{l,n})$, where m is a number of processor stages, $r_{l,n}$ is a processor reconfigurable parameters vector of the structure corresponding to the decomposition of tree DWPT $(l, n) \in E_i$. The limit corresponds to a tree CB - WPD: $(l, n) \in E_{CB}$. Next, on the basis of the growing algorithm, described in section 3, the DWPT tree structures are formed, for example, E_1, E_2, E_3 for which restrictions are checked E_{CB}, as well as the calculated information density H_{E_i}. Based on that, if it turns out $H_{E_3} < H_{E_2} < H_{E_1}$, the structure of the E_3 to the required frequency-time resolution processing of the frame. Reconfigurable processor DWPT determined by the current vector of reconfigurable parameters:

$$r_{l,n} = [(\alpha_1, \beta_0, \beta_1), (\alpha_2, \beta_0, \beta_1, \beta_2, \beta_3), \ldots (\alpha_2, \beta_0, \ldots, \beta_n)] \tag{18}$$

where α_l and β_n takes the values 0 or 1.

Parameters α_l determine the transition to a new level of large-scale tree DWPT l, i.e. include signal processing in the next processor step m:

$$\alpha_l = \begin{cases} 1, & if\ H_{E_i} < H_{E_{i-1}}\ and\ E_i \notin E_{CB} \\ 0, & otherwise \end{cases}. \tag{19}$$

In turn, a group of parameters β_n includes n nodes at the level l:

$$\beta_n = \begin{cases} 1, \ if PE_{l,n} > PE_{l+1,2n} + PE_{l+1,2n+1} \\ 0, \ otherwise \end{cases}. \tag{20}$$

Thus, the transition of signal processing according to the DWPT tree structure E_i on the architecture of the processor $E_{m,i}$ for processing on the architecture of $E_{m+1,i+1}$, in accordance with the tree structure E_{i+1} is the vector according to the reconfigurable parameters $r_{l,n}$, $(l, n) \in E_{i+1}$:

$$E_{m+1,i+1} = r_{l,n} \cdot E_{m,i}. \tag{21}$$

From basic principles of psychoacoustics follows that human perception of acoustic information is quite inert, from 5 ms to 300 ms. Masking forward and backward is approximately 20 ms. With a input audio signal frame length of 5 ms. and processing delay determined by a single stage parallel pipeline processor, we can assume that the delay in processing the input signal (l - 2) th levels of the processor (the maximum value of l =8 for *CB - WPD*) much smaller than the temporal instability signal perceived by man. This allows you to organize multi-frame processing on the basis of parallel pipeline processors, a reconfiguration of the structure DWPT processor to determine the variability of the current signal frame - a frame for which to calculate the cost function H_{E_i}.

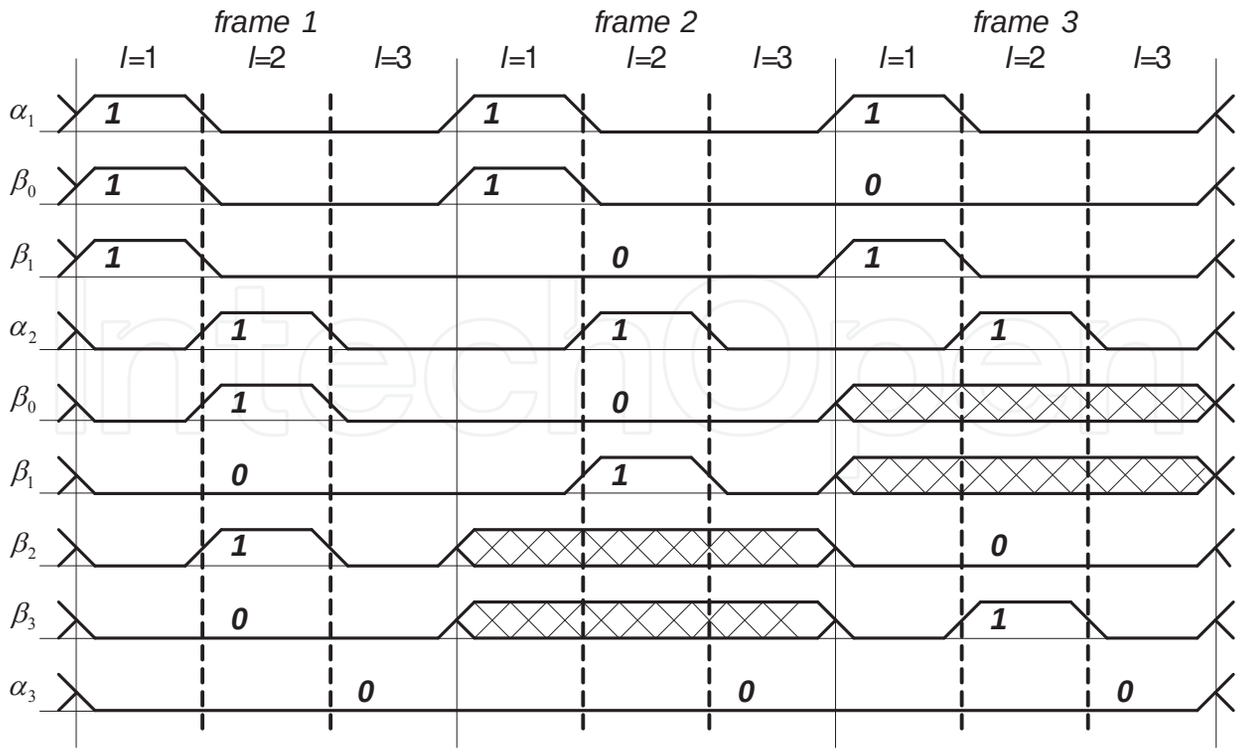


Figure 19. The timing diagram of control signals change for three consecutive frames of the audio

http://dx.doi.org/10.5772/51604

The profile of the time parameters α_l and β_n, the transformation vector processor $r_{l,n}$, in accordance with the tree structure $(l,\ n) \in E_i$ for three consecutive frames of the audio signal is shown in figure 19. DWPT tree structures that you see dotted line in figure 20 for the respective frames, determine options for their future growth in accordance with the obtained values of the perceptual entropy $PE_{l,n}$at each node of the tree, but, for example, a value that indicates the informative density $H_{E_{m,i}}$ of the resulting decomposition tree DWPT shows the ineffectiveness of further growth of the tree structure.

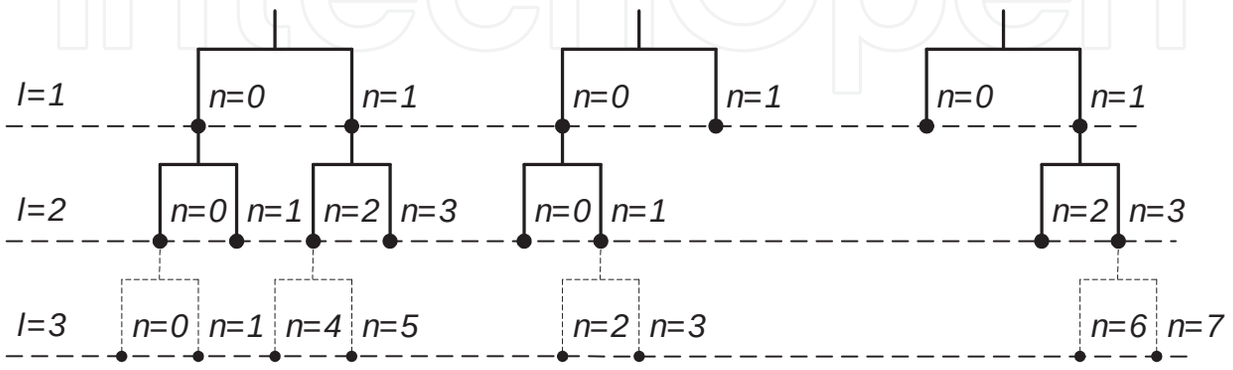


Figure 20. DWPT tree structures for figure 19

Thus, the DWPT trees structure E_i, described by the nodes $(l,\ n)$, as well as the corresponding reconfiguration vector DWPT processor are obtained according to the algorithm of dynamic reconfiguration shown in the figure 21 and can be written as following:

for 1st frame: E_1={[(1,0);(1,1)], [(2,0);(2,1);(2,2)(2,3)]} and vector is r_1=[(1,1, 1), (1,1, 0,1, 0), (0)];

for 2nd frame: E_2={[(1,0);(1,1)], [(2,0);(2,1)]}, and vector is r_2=[(1,1, 0), (1,0, 1, ×, ×), (0)];

for 3rd frame E_3={[(1,0);(1,1)], [(2,2);(2,3)]}, and vector is r_3=[(1,0, 1), (×, ×, 0,1), (0)].

This algorithm of dynamic reconfiguration allows obtaining a suboptimal solution for DWPT analysis. The advantages of the above algorithm can be summarized as: pruning method is a top-down method, DWPT pruning can be viewed as a split process, i.e. we have the temporal construction DWPT tree for each signal frame that is ideal decision for real time processing implemented in a reconfigurable hardware.

The processing of the first nine frames in the pipeline DWPT processor is shown in the figure 22 where j is a number of the frame which was loaded to DWPT processor in the current time for processing according to the DWPT tree structure $E_{m,i}$ of the current frame i. The computation process at each stage of pipeline DWPT processor schematically shows with cubes, where cube mean a frame processing on corresponding DWPT processor stage. The cube "Master" means that on this stage a current frame is used for actual DWPT tree structure creation. The cube "Slave" means that on this stage a current frame is processed according to the actual DWPT tree structure. The cube "Master (suboptimal decomposition)" means that

Suppose, a computation resource $(m, r_{l,n})$ is limited by C, the limiting DWPT tree structure is $CB-WPD$: $(l,n) \in E_{CB}$, the required computation resource of i-th DWPT tree $(l,n) \in E_i$ is defined as value c_i, the split decision of dividing the parent node (l,n) on the two children $(l+1,2n)$ and $(l+1,2n+1)$ is referred as a $split(l,n)$ where l is a transformation scale level, n is n-th node of the level l, m is a stage of DWPT processor and number of the input frame of the audio signal is j.

STEP 1. Let $j=1$, $m=1$, $l=0$ and $split(l,n)=YES$, $r_{l,n}=YES$, DWPT tree root node is (0,0) for the first frame of the input audio signal with perceptual entropy $PE_{0,0}$, information density $H_{E_{0,0}}$ and reconfiguration process is allowed.

STEP 2. $i=j$, the first frame of the input audio signal defines growing process of DWPT tree structure. Making the signal decomposition is based on 1st PU.

STEP 3. Estimate the perceptual entropy in each node $PE_{l,n}$ and information density $H_{E_{1,1}}$ of the DWPT tree structure.

STEP 4. Check the DWPT tree structure information density $H_{E_{1,1}}$ in comparison with the DWPT tree $E_{0,0}$

IF $H_{E_{0,0}} < H_{E_{1,1}}$,

THEN that is not an audio signal, the coefficients are not processed and GOTO STEP 1.

STEP 5. $l=l+1$.

IF $l-1 >$ maximum of the scale level of the limiting DWPT tree structure $CB-WPD$,

THEN STOP – the growing process for the DWPT tree structure $E_{i,m}$ is finished.

STEP 6. Check the DWPT tree structure $E_{i,m}$ nodes belonging to $CB-WPD$: E_{CB}:

IF $(l,n) \in E_{m,i} = (l,n) \in E_{CB}$,

THEN $r_{l,n}=NO$.

STEP 7. $m=m+1$. DWPT tree structure $E_{i,m}$ growing is performed as follows

FOR each node n of the level l:

- estimate and check the adequacy of the DWPT tree computation resource $c_{i,m}$:

IF $c_{i,m} > C$,

THEN $r_{l,n}=NO$ and $STOP$ – the growing process for the DWPT tree structure $E_{i,m}$ is finished.

- perform the decomposition of the parent node (l,n).
- calculate the perceptual entropy into the child nodes $PE_{l+1,2n}$ and $PE_{l+1,2n+1}$:

IF $PE_{l,n} \geq PE_{l+1,2n} + PE_{l+1,2n+1}$,

THEN $split(l,n)=YES$, $r_{l,n}=YES$

ELSE $split(l,n)=NO$, $r_{l,n}=NO$.

STEP 8. $j=j+1$. Read the next frame of the input audio signal. It will be processed according to DWPT tree structure $E_{i,m}$.

STEP 9. Estimate the information density $H_{E_{i,m}}$ of the DWPT tree structure $E_{i,m}$:

IF $H_{E_{i,m}} > H_{E_{i,m-1}}$,

THEN $r_{l,n}=NO$ and $STOP$ – the growing process for the DWPT tree structure $E_{i,m}$ is finished.

STEP 10. GOTO STEP 5.

STOP. Suboptimal DWPT tree structure for the ith input frame of the audio signal is $E_{i,m-1}$, $m=m-2$, $l=l-2$, $i=i+1$. GOTO STEP 5.

Figure 21. Algorithm of dynamic reconfiguration

on this stage the suboptimal decomposition is fund for the current frame. The cube "Slave (suboptimal decomposition)" means that on this stage the suboptimal decomposition is fund for the current frame but it is slave. The last cube "Master (no optimal decomposition)" means that on this stage no optimal decomposition is detected. Here, the current frame i, for example i=0, as it show in the figure 22, sets a master and begin from the stage m=0 consequentially processed on each stage in DWPT processor, involving new frames i=1,2, 3… as slave in processing while no optimal decomposition for the master will find at stage m=2. The

http://dx.doi.org/10.5772/51604

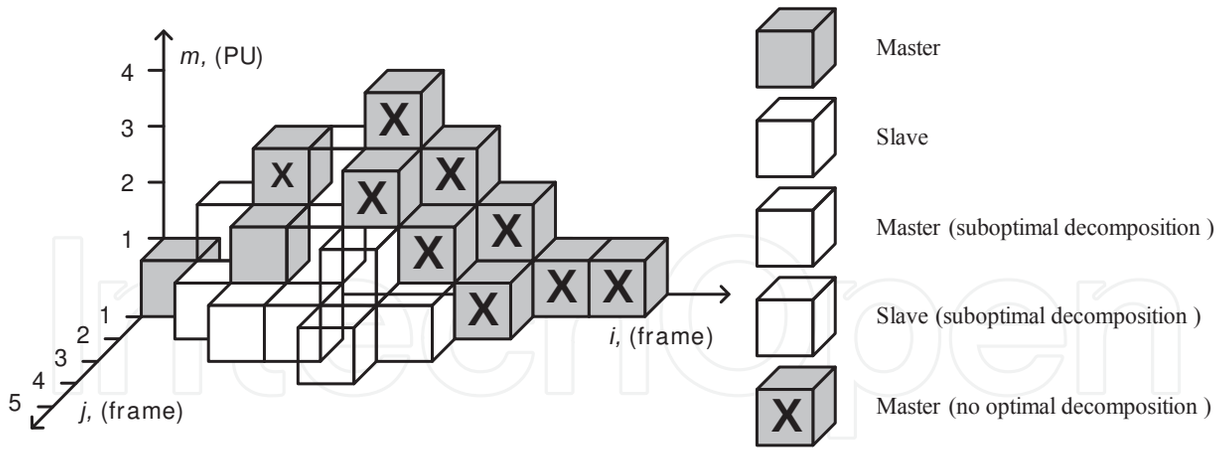


Figure 22. Diagram of the dynamic reconfiguration DWPT tree structures and multi-frame processing in the parallel-pipeline DWPT processor

suboptimal decomposition for master was founded on stage $m=1$. The result DWPT coefficients of the frame $i=0$are removed from DWPT processor and next frame $i=1$ sets as a master and process is repeated while no optimal decomposition for the master will find at stage $m=3$. The suboptimal decomposition for the current frame $i=1$ was at stage $m=2$then the result DWPT coefficients removed from DWPT processor and next frame $i=2$ sets as a master. As we can see, the decomposition is not optimal for the current frame $i=2$ then the suboptimal decomposition is at stage $m=1$ and next frame $i=3$ becomes a master. In the figure 22 is shown how DWPT tree is growing and involving a frames to computation process and how it rolling back with removing frames from the process. The reconfiguration in DWPT processor is based on the formation of transformation vectors $r_{l,n}$ according to the algorithm of dynamic transformation DWPT tree structure mentioned on figure 21 which is formed in DSP processor.

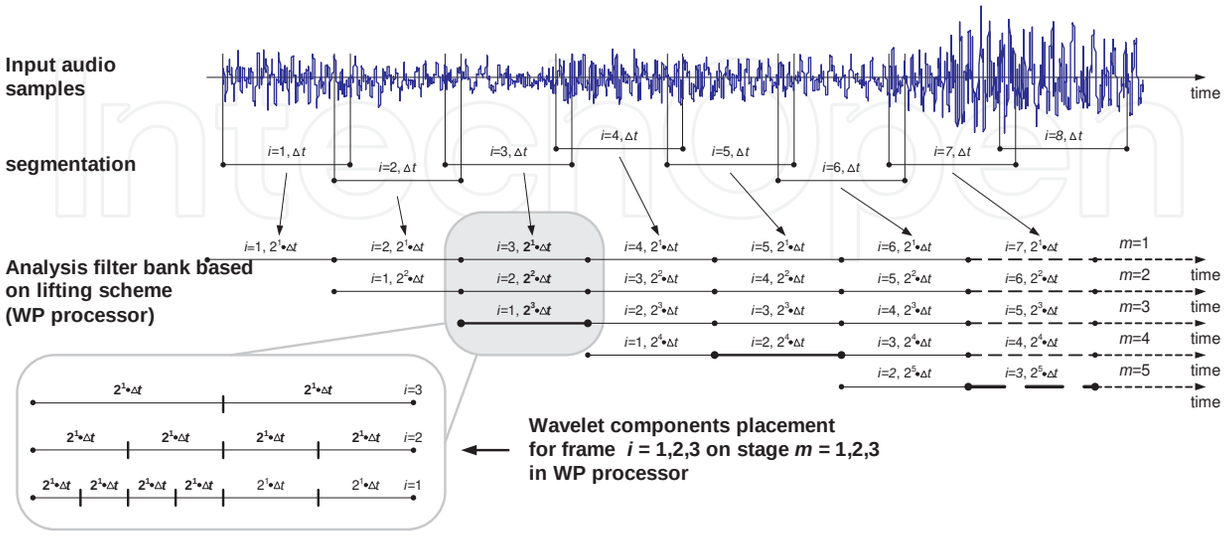


Figure 23. The input signal analysis in DWPT processor

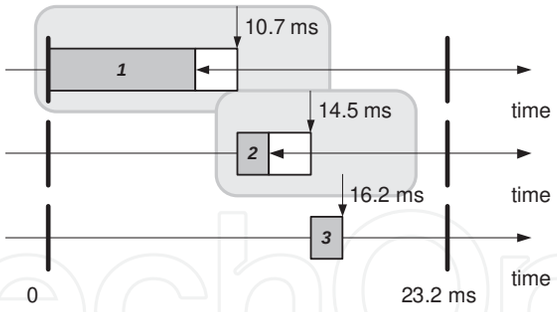


Figure 24. Time schedule operation in the DSP processor

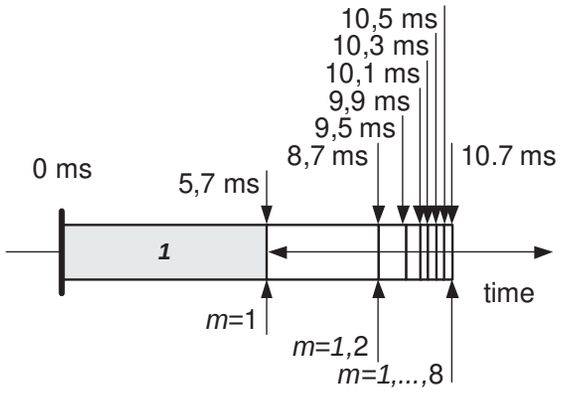


Figure 25. Run-time of the procedure 1, depending on a number of stages *m*

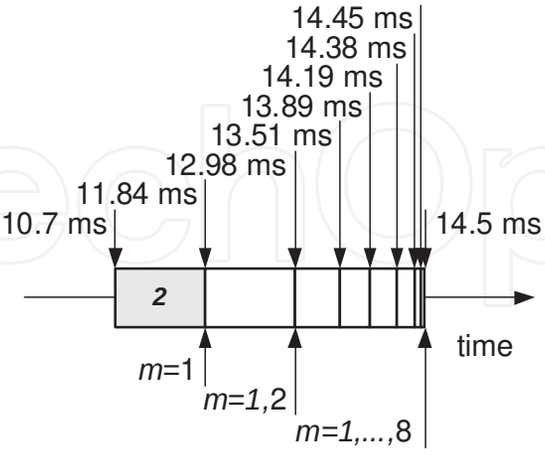


Figure 26. Run-time of the procedure 2, depending on the number of stages *m*

The complete input signal analysis in DWPT processor is demonstrated in the figure 23. The input signal is segmented on a frame with minimal overlapping and analysed. The frame

http://dx.doi.org/10.5772/51604

length in a time is equal 22.3 ms. At the same time for frames to be processed under the current structure of the tree $E_{m,i}$ on the steps m of DWPT processor, a DSP processor shall monitor the implementation of the procedures such as: the masking threshold calculation algorithm as it described in appendix A [24] (procedure 1), perceptual entropy $PE_{l,n}$ assessment based on (4)-(5) (procedure 2) and the entropy of the DWPT tree structure H_E estimation according to (1)-(3) (procedure 3). The time schedule for DSP processor (250 MHz, 32 bit floating point DSP microprocessor) based on the listened above procedures are shown in the figure 24. The run time of procedure 1 and procedure 2 are showed in the figure 25 and figure 26 correspondingly as it is mentioned the computational time is not a constant, it depends on the number of stages m in DWPT processor involving in the input frames processing.

The output signal synthesis in DWPT processor is demonstrated in the Figure 27. The monitoring system loads the input frame i to the appropriate level m of DWPT processor. Move the frame i to the next stage of the processor is executed when the monitoring system takes the next frame $i + 1$, which will need to get involved is to step DWPT processor. To coordinate the work performed at each stage of the processor, it is necessary to introduce a delay, multiple processing time of one frame at one stage, the most rhythmic work will be provided by the parallel pipeline structure of DWPT processor.

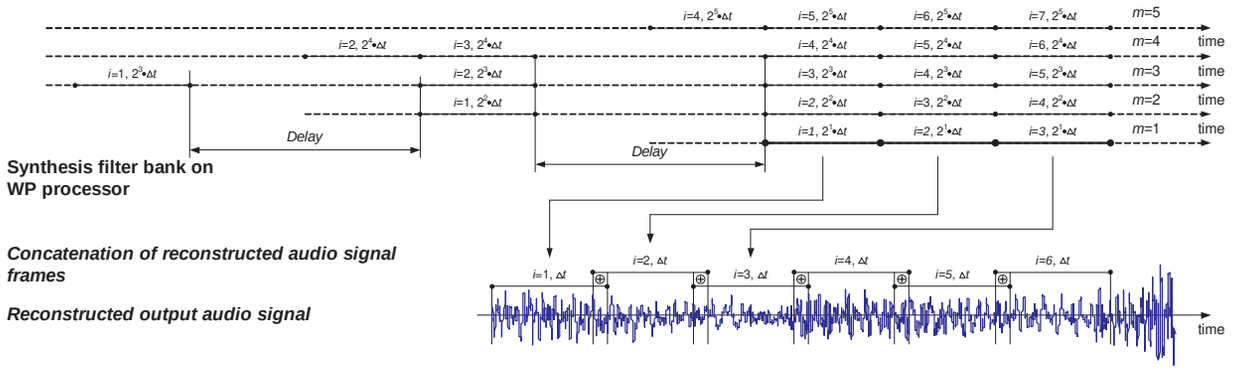


Figure 27. The output signal synthesis in DWPT processor

6. Conclusion

In the given paper the dynamic reconfigurable lifting based adaptive DWPT processor was presented. The lifting scheme allows to reduce on halve the number of multiplications and summations and increase the processing speed. Appling DAT-based approach as the design techniques for time-varying DWPT decomposition allows us to construct dynamically adapted to input signal DWPT analysis. The reconfigurable system offers several advantages over competing alternatives: faster and smaller than general purpose hardware solutions; lower development cost than dedicated hardware solutions; dynamic reconfigurable supports multiple algorithms within a single application; multi-purpose architecture generates volume

demand for a single hardware design. The proposed techniques optimize system performance and, in addition, provide a convenient framework within which on-going research in the areas of non-uniform filter bank applied to speech/audio coding algorithms and reconfigurable architectures can be synergistically combined to enable the design of reconfigurable high-performance DSP systems.

Thus, the proposed dynamic reconfigurable DWPT processor with frame-based psychoacoustic optimized time-frequency tilling is successfully applicable for several application such as monophonic full-duplex audio coding system [18] and scalable audio coding based on hybrid signal decomposition where the transient part of the signal is modelled on psychoacoustic motivated frame based adaptive DWPT in marching pursuit algorithm [24]. The advantages of this DWPT processor is better viewed by considering the DWPT growing as a splitting process, i.e. the temporal construction DWPT tree created for each signal frame presents an ideal decision for real time processing implemented in a reconfigurable hardware.

Acknowledgements

This work was supported in part by Belarusian republican fund for fundamental research under the grants T04-217 and T08MC-040.

Author details

Alexey Petrovsky*, Maxim Rodionov and Alexander Petrovsky*

*Address all correspondence to: {petrovsky,post-rodmax,palex}@bsuir.by

Department of Computer Engineering, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

References

[1] Mallat, S. A Theory of Multiresolution signal decomposition: the wavelet representation. IEEE transactions on pattern analysis and machine intelligence, july (1989). doi:, 11(7), 674-693.

[2] Coifman, R, & Wickerhauser, M. Entropy based algorithms for best basis selection. IEEE transaction of information theory, March (1992). doi:, 38, 712-718.

[3] Wickerhauser, M. V. Adapted wavelet analysis from theory to software. A Peters Wellesley, Massachusetts, (1994). p.

http://dx.doi.org/10.5772/51604

[4] Cohen, I, Raz, S, & Malah, D. Orthonormal shift-invariant adaptive wavelet packet decomposition and representation. Signal processing, 57 (3), march (1997). doi:S0165-1684(97)00007-8., 251-270.

[5] Wu, X, Li, Y, & Chen, H. Programmable wavelet packet transform processor. IEE electronics letters, (1999). doi:el:19990330., 35(6), 449-450.

[6] Trenas, M. A, Lopez, J, Sanchez, M, Arguello, F, & Zapata, E. L. Architecture for wavelet packet transform with best tree searching. Proceedings of IEEE international conference on Application-Specific Systems, Architectures and Processors, ASAP'00, (2000). doi:ASAP.2000.862399., 289-298.

[7] Trenas, M. A, Lopez, J, & Zapata, E. L. A configurable architecture for the wavelet packet transform. Journal of Signal Processing Systems, (2002). doi:A:1020221003822., 32(3), 255-273.

[8] Sweldens, W. The lifting scheme: A construction of second generation wavelets. SIAM: SIAM Journal on Mathematical Analysis, 29 (2) ((1997). , 511-546.

[9] Arguello, F, Lopez, J, Trenas, M. A, & Zapata, E. L. Architecture for wavelet packet transform based on lifting steps. Parallel Computting, (2002). doi:S0167-8191(02)00101-1., 28(7-8), 1023-1037.

[10] Aroutchelvame, S. M, & Raahemifar, K. Architecture of wavelet packet transform for 1-D signal. Proceedings of IEEE Canadian Conference on Electical and Computer Engineering, CCECE'05, may (2005). doi:CCECE.2005.1557216., 1304-1307.

[11] Paya, G, Peiro, M. M, Ballester, F, Herrero, V, & Mora, F. Lifting folded pipelined discrete wavelet packet transform architecture. VLSI Circuits and Systems. Edited by Lopez, Jose Fco.; Montiel-Nelson, Juan A.; Pavlidis, Dimitris. Proceedings of the SPIE, (2003). doi:, 5117, 321-328.

[12] Wang, C, & Gan, W. S. Efficient VLSI Architecture for Lifting-Based Discrete Wavelet Packet Transform. IEEE transactions on circuits and system- II: Express briefs, may (2007). doi:TCSII.2007.892410., 54(5), 422-426.

[13] Parhi, K. K. Algorithm transformation techniques for concurrent processors. Proceedings of the IEEE, dec. (1989). doi:, 77(12), 1879-1895.

[14] Petrovsky Al Petrovsky A. Dynamic algorithm transforms for reconfigurable real-time audio coding processor. Proceedings of the International Conference on Parallel Computing in Electrical Engineering, PARELEC'02, Warsaw, Poland, sep. 22-25, 2002, IEEE Computer Society Press, Los Alamitos, California, (2002). doi:PCEE. 2002.1115317., 422-424.

[15] Ackenhusen, J. G. Real-time signal processing: design and implementation of signal processing systems. Printice Hall, NJ, (1999). p.

[16] Villasenor, J, & Hutchings, B. The flexibility of configurable computing. IEEE Signal Processing Magasine, sep. (1998). doi:, 15(5), 67-84.

[17] Johnston, J. D. Transform coding of audio signals using perceptual noise criteria. IEEE Journal on Selected Areas in Communications, feb. (1988). doi:, 6(2), 314-323.

[18] Petrovsky Al Krahe D., Petrovsky A.A., Real-time wavelet packet-based low bit rate audio coding on a dynamic reconfigurable system. 114th AES Convention preprint 5778, March (2003). Amsterdam, Netherlands, 22 p.

[19] Petrovsky Al A multiresolution auditory model using adaptive WP excitation scalograms. ELEKTRONIKA, PAN, Warsaw, (2008). , 49(4), 65-70.

[20] Daubechies, I, & Sweldens, W. Factoring wavelet transforms into lifting steps. Journal of Fourier Analysis and Applications, (1998). , 4(3), 247-269.

[21] Coors, M, Keding, H, Luethje, O, & Meyr, H. Design and DSP implementation of fixed-point systems. EURASIP journal on advances in signal processing, (2002). doi:S1110865702205065., 908-925.

[22] Menard, D, Chillet, D, & Sentieys, O. Floating-to-fixed-point conversion for digital signal processors, EURASIP journal on applied signal processing, article ID 96421, doi:ASP/2006/96421., 2006, 1-19.

[23] Daubechies, I. Orthonormal bases of compactly supported wavelets II. Variations on theme. Communications on pure and applied mathematics, (1988). doi:, 41, 909-996.

[24] Petrovsky Al Azarov E., Petrovsky A. Hybrid signal decomposition based on instantaneous harmonic parameters and perceptually motivated wavelet packets for scalable audio coding. Special Issue: “Fourier Related Transforms for Non-Stationary Signals”, Signal Processing, june (2011). doi:j.sigpro.2010.09.005., 91(6), 1489-1504.

IntechOpen