

## **БЫСТРОЕ ПРОТОТИПИРОВАНИЕ ВСТРАИВАЕМЫХ ПОТОЧНЫХ ПРОЦЕССОРОВ ВЫЧИСЛЕНИЯ ПАКЕТНОГО ДИСКРЕТНОГО ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ.**

М.М. Родионов, Ал.А. Петровский.

Белорусский государственный университет информатики  
и радиоэлектроники, Минск.

*Представленная работа посвящается вопросам реализации архитектур поточных процессоров для вычисления алгоритма пакетного дискретного вейвлет-преобразования (ПДВП). Предлагаются аппаратные решения для блоков процессора в зависимости от мультимедиа приложений. Для увеличения скорости обработки данных вычислительный блок процессора ПДВП (двухканальный банк фильтров) реализуется на лестничных структурах. Предлагаются среда прототипирования архитектур процессора на элементную базу ПЛИС. В качестве примера рассматривается процессор ПДВП для обработки речевых сигналов, аппроксимирующий шкалу барков.*

### **Введение**

В данной работе представлен подход по проектированию прототипа поточного процессора ПДВП, ориентированного на обработку звуковых и речевых сигналов. Проведен анализ алгоритма с точки зрения реализации на арифметике с фиксированной запятой, предложен ряд аппаратных решений. Авторами разработаны две схемы построения поточного процессора: для вычисления фиксированного дерева преобразования и для реализации динамически перестраиваемого алгоритма ПДВП [1]. Вычислительное ядро представленных архитектур – двухканальный банк фильтров, реализованный на основе лестничных структур [2, 3] на арифметике с фиксированной запятой. В качестве практического результата представлена среда проектирования для генерирования синтезируемого VHDL-описания процессора на основе технических требований для конкретного приложения. В конце рассмотрен пример по реализации поточного процессора ПДВП для шкалы барков.

Особенностью использования ПДВП в приложениях обработки звука и речи являются большое число уровней декомпозиции и высокие порядки вейвлет-фильтров по сравнению с приложениями для обработки графики и видео. Ряд исследований, проведенный в данной области, в той или иной степени удовлетворяет представленным требованиям, хотя имеются определенные недостатки, связанные прежде всего с направленностью существующих решений на другой класс приложений.

Так, в работе [4] предлагается архитектура на основе фильтров с конечной импульсной характеристикой (КИХ) для вычисления ПДВП с поиском лучшего дерева на основе [5]. Данный подход предполагает вычисление полного пакета преобразования, что приводит к увеличению вычислительных затрат по сравнению с предлагаемой архитектурой, реализующей динамически перестраиваемый алгоритм ПДВП [1]. В работе [6] представлена поточная архитектура для вычисления ПДВП на основе лестничных структур с использованием целочисленной арифметики, обладающая высокой пропускной способностью и использующая малый ресурс памяти. К недостаткам работы можно отнести тот факт, что разработанная архитектура способна вычислять только полный пакет вейвлет-преобразования, а это делает ее затратной с точки зрения аппаратного и вычислительного ресурса для приложений обработки звука и речи с большим числом уровней декомпозиции, не требующих вычисления полного дерева.

# 1. Алгоритм ПДВП

Алгоритм ПДВП является обобщением дискретного вейвлет-преобразования и может быть представлен в виде банка фильтров с древовидной структурой [7] (рис. 1). В отдельно взятом узле с номером  $n$  любого уровня дерева  $l$  ( $n = 0..(2^l - 1)$ ,  $l \in Z$ ) входной сигнал  $x_{l,n,k}$ , ( $k$  номер отсчета сигнала) сепарируется на низкочастотную (НЧ)  $x_{l+1,2l,k}$  и высокочастотную (ВЧ)  $x_{l+1,2n+1,k}$  компоненты при помощи пары вейвлет-фильтров  $\tilde{h}(z)$  и  $\tilde{g}(z)$  с конечной импульсной характеристикой, после чего каждый субполосный сигнал децимируется с понижением частоты дискретизации в два раза. Функциональный блок, реализующий данное сепарирование входного сигнала, называется двухканальным банком фильтров анализа.

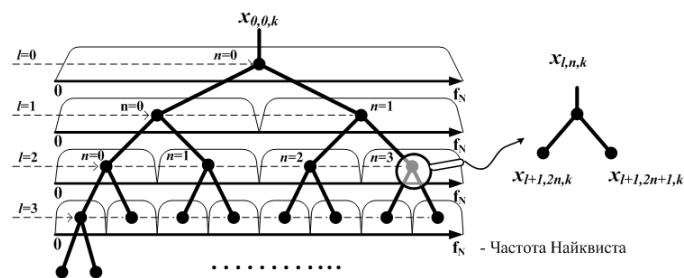


Рис. 1. Древовидная структура алгоритма ПДВП

Таким образом, конкретному узлу  $(l, n)$  соответствует частотный диапазон  $(n \cdot 2^{-l}, (n + 1)2^{-l})$ , нормированный к частоте Найквиста ( $f_N$ ). На каждом последующем уровне декомпозиции разрешающая способность по частоте возрастает в два раза, но при этом в два раза уменьшается разрешающая способность по времени.

Преимущество ПДВП заключается в возможности достаточно гибкого подбора дерева декомпозиции (рис. 2), базируясь на природе исследуемого сигнала. Причем выбор структуры дерева может как осуществляться на основе заранее известных особенностей сигнала, так и выполняться динамически, «подстраиваясь» под текущий фрейм обработки [1].

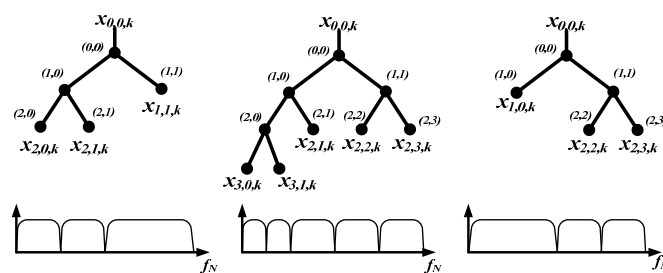


Рис. 2. Примеры структур деревьев ПДВП

## 2. Архитектура поточного процессора ПДВП

### 2.1. Структурная схема процессора

Для реализации вычисления ПДВП в работе предлагается поточный процессор, обобщенная блок-схема которого представлена на рис. 3. Поточное вычисление алгоритма организовано с использованием одного двухканального банка на уровне (степе-

ни конвейера). При этом фильтрация в разных субполосах на одном уровне осуществляется путем мультиплексирования по времени при подаче входных данных с корректной подстановкой соответствующей данной полосе предыстории.

Вычислительный блок (ВБ) выполняет функцию двухканального банка фильтров анализа. Блоки памяти (БП) хранят промежуточные и конечные результаты вычислений на каждом уровне декомпозиции. Блоки коммутации (БК) обеспечивают корректное сопряжение между ВБ на смежных уровнях декомпозиции. Блок управления (БУ) выполняет следующий набор функций: прием и обработка команд от ведущего процессора; организация обмена данными; формирование управляющих сигналов.

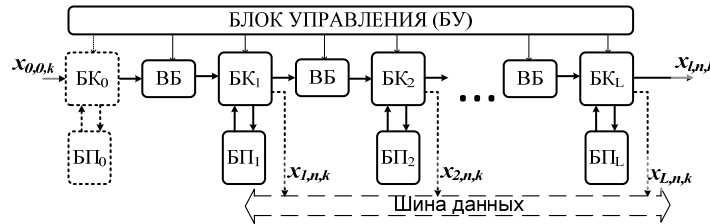


Рис. 3. Блок-схема поточного процессора ПДВП

## 2.2. Двухканальный банк фильтров на основе лестничных структур

Для двухканального банка фильтров анализа предлагается аппаратная реализация на основе лестничных структур [2] (в англ. литературе lifting scheme либо ladder structure). Данная архитектура может быть получена как результат факторизации полифазной матрицы исходных КИХ фильтров банка анализа  $\tilde{h}(z)$  и  $\tilde{g}(z)$  [3].

Таким образом, в терминах z-преобразования сепарирование входного сигнала  $X_{l,n}(z)$  на НЧ и ВЧ компоненты  $X_{l+1,2n}(z)$  и  $X_{l+1,2n+1}(z)$  в узле  $(l, n)$  может быть произведено на основе выражения:

$$[X_{l+1,2n}(z) \quad X_{l+1,2n+1}(z)] = [X_e(z) \quad X_o(z)] \prod_{i=1}^{I/2} \left( \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \right) \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}, \quad (1)$$

где  $X_e(z)$ ,  $X_o(z)$  – представления в z-области последовательностей, состоящих соответственно из четных (even) и нечетных (odd) отсчетов входной последовательности  $x_{l,n,k}$ ;  $s_i(z)$  и  $t_i(z)$  – полиномы малого порядка;  $I$  – число элементарных треугольных матриц, полученных в результате факторизации полифазной матрицы;  $K_1$ ,  $K_2$  – вещественные коэффициенты. Ниже приведены блок-схемы для прямой реализации банка анализа на основе КИХ фильтрации и реализации на основе лестничных структур (рис. 4).

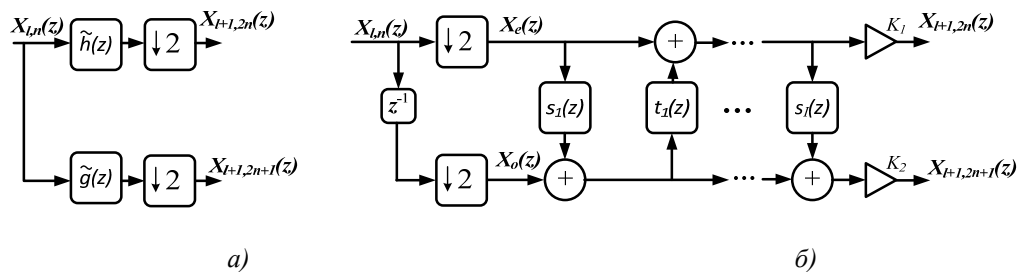


Рис. 4. Блок-схема двухканального банка фильтров: а) прямая реализация; б) реализация на основе лестничных структур

В результате анализа алгоритма факторизации с использованием среды MATLAB общий вид для полиномов  $s_i(z)$  и  $t_i(z)$  был сведен к выражению вида  $(b_0^i + b_1^i z^{-1})z^u$ ; где  $b_0^i, b_1^i$  – вещественные коэффициенты,  $u$  – целое значение степени  $z$ .

В итоге для банка анализа на лестничных структурах число умножений составило  $N_{mul} = N + 2$  и число сложений  $N_{add} = N$ , где  $N$  – число коэффициентов исходного фильтра  $\tilde{h}(z)$ , что почти вдвое меньше по сравнению с прямой реализацией.

Для вычисления алгоритма на арифметике с фиксированной запятой использована методология [8, 9], на основании которой по ходу выполнения алгоритма варьируется число бит, отводимых под целую и дробную части слова.

В соответствии с данным подходом любое число, представленное в формате с фиксированной запятой в дополнительном коде, задается в виде выражения:

$$a = ma \cdot 2^{exp_a}, \text{ где } ma = (-1)^s + \sum_{i=0}^{wl-2} a_i \cdot 2^{i-wl+1}. \quad (2)$$

Здесь  $ma$  – само значение числа, представленное в дополнительном коде, интерпретируемое как дробное в диапазоне  $[-1, 1)$ ;  $exp_a$  – порядок масштабирующего множителя  $2^{exp_a}$ ;  $a_i$  – значения соответствующих битов числа, равные 0 либо 1;  $s$  – знаковый бит;  $wl$  – разрядность слова (рис. 5).

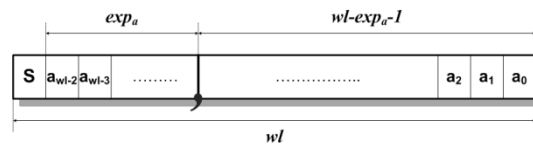


Рис. 5. Представление числа в арифметике с фиксированной запятой

Для заданного в (2) формата данных операции сложения и умножения (рис. 6) чисел  $a$  и  $b$  ( $exp_b \geq exp_a$ ) задаются как

$$c = a + b = mc \cdot 2^{exp_c} = (ma \cdot 2^{exp_a - exp_b} + mb) \cdot 2^{exp_b}, \quad (3)$$

$$c = a \cdot b = mc \cdot 2^{exp_c} = ma \cdot mb \cdot 2^{iwl_a + iwl_b}. \quad (4)$$

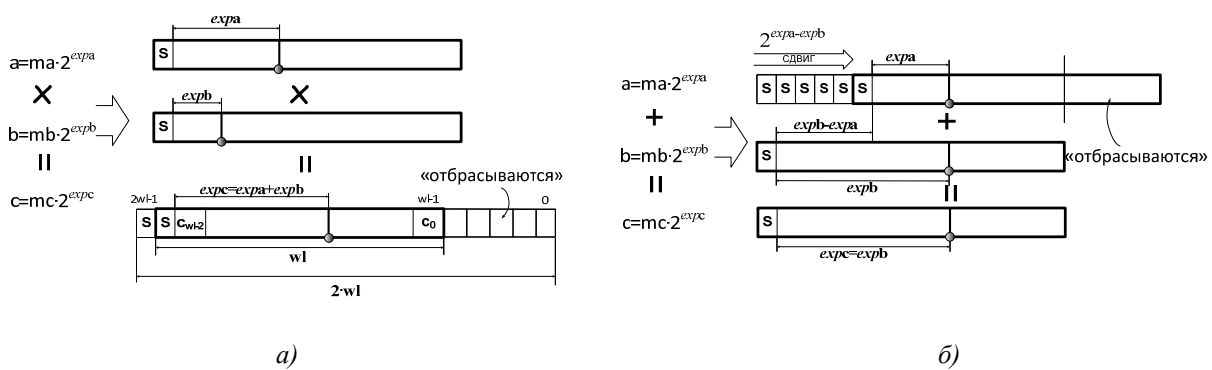


Рис. 6. Выполнение операций: а) умножения; б) сложения

В качестве примера рассмотрим расчет параметров банк фильтров на основе лестничных структур для материнской вейвлет-функции Db-4 [10]. В табл. 1 приведены рассчитанные параметры  $u, b_0, b_1$  для полиномов  $s_i(z)$  и  $t_i(z)$  из (1). Кроме рассчитанных значений данных параметров в таблице в соответствии с (2) приведены параметры

$mb_0, expb_0$  и  $mb_1, expb_1$  коэффициентов  $b_0, b_1$  для реализации алгоритма на арифметике с фиксированной запятой. Коэффициенты  $K_1$  и  $K_2$  (1) для рассматриваемого примера равны 2,6131 и 0,3827 соответственно.

Таблица 1

Параметры лестничной структуры для исходных вейвлет-фильтров Db-4 (8 коэффициентов)

| Номер шага ЛС, $k$ | Тип      | $u$ | $b_0^k$  |         |          | $b_1^k$  |         |          |
|--------------------|----------|-----|----------|---------|----------|----------|---------|----------|
|                    |          |     | значение | $mb_0$  | $expb_0$ | значение | $mb_1$  | $expb_1$ |
| 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       | -        |

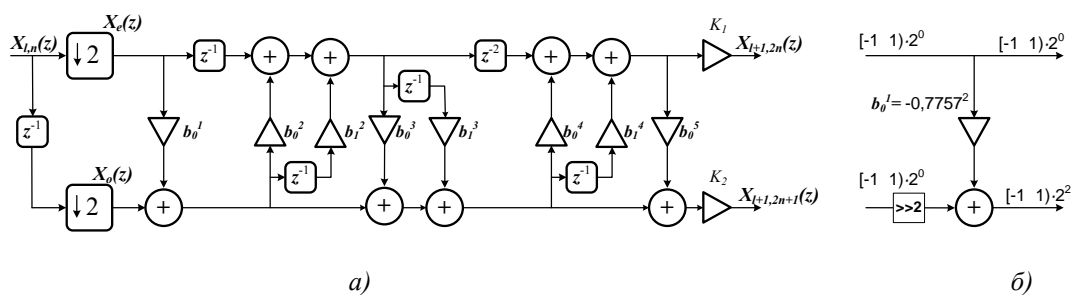


Рис. 7. Реализация двухканального банка фильтров на основе Db-4 (табл. 1.): а) блок-схема всей архитектуры; б) первый шаг лестничной структуры.

В структурной схеме банка анализа из рассматриваемого примера (рис. 7 а) в верхнем канале добавлены элементы задержки  $z^{-1}$  с целью удовлетворения условию казуальности системы. При реализации первого шага банка (рис. 7 б) на основе предложенной арифметики предполагается, что на вход устройства поступают числа в формате с фиксированной запятой в дополнительном коде из диапазона  $[-1, 1)$ , поэтому согласно (3) перед операцией сложения арифметическому сдвигу вправо на два разряда подвергается операнд из нижнего канала (последовательность нечетных отсчетов).

### 2.3. Организация блоков памяти и блоков коммутации

Блок памяти (рис. 3) для хранения промежуточных вейвлет-коэффициентов может быть реализован двумя способами: при помощи регистров либо на основе памяти.

Первый способ применим для задач, где необходима организация поточного вычисления ПДВП для произвольной фиксированной структуры дерева. Для этого была модифицирована архитектура [6] на основе регистровых файлов с организацией FIFO и достаточно простой схемой управления с использованием коммутаторов  $K_l$  (рис. 8) для правильной подачи данных на входы ВБ в соответствии с алгоритмом. При этом размер регистрового файла на уровне  $l$  равен числу субполос, получаемых на данном уровне.

Для правильной работы устройства в каждом ВБ необходимо заменить регистры, выполняющие роль элементов задержки  $z^{-1}$ , на регистровые файлы, размер которых равен числу узлов в структуре дерева на данном уровне декомпозиции, для организации корректного хранения предыстории в каждой субполосе. Достоинства представленной архитектуры: высокая пропускная способность, простота организации схемы управления и коммутации.

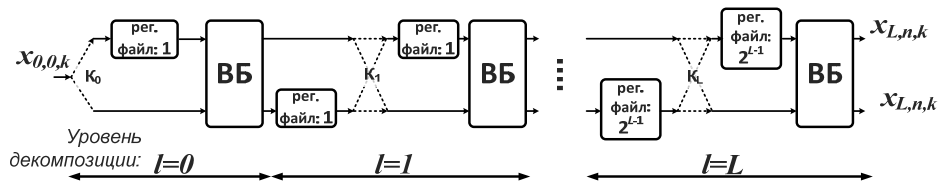


Рис. 8. Поточная архитектура процессора ПДВП на основе регистровых файлов с организацией FIFO

Второй способ построения поточной архитектуры процессора основан на использовании памяти (рис. 9). Данный подход применяется в случаях, когда необходимо достаточно большой ресурс для хранения промежуточных результатов, а также для построения динамически реконфигурируемых систем с произвольной сигнализависимой структурой дерева [11]. На каждый уровень декомпозиции ставится блок памяти, объем которой зависит от размера обрабатываемого фрейма, а также от предельной структуры дерева (в случае динамически реконфигурируемой системы).

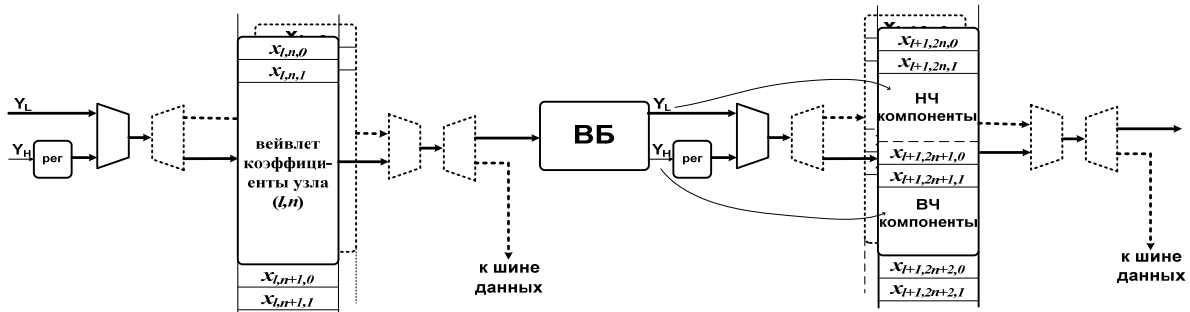


Рис. 9. Организация поточной архитектуры процессора ПДВП на основе памяти

На каждом уровне  $l$  мультиплексор справа от ВБ позволяет организовать запись двух рассчитанных НЧ и ВЧ коэффициентов в блок памяти на уровне  $l + 1$ . Для построения поточной системы, позволяющей обработку на каждом уровне декомпозиции отдельного фрейма, требуется ввод дополнительного блока памяти для устранения конфликтов по данным (пунктирная линия на рис.9). При этом необходимо задействовать дополнительные мультиплексор и демультиплексор для чтения и записи в один из двух блоков памяти на уровнях  $l$  и  $l + 1$ . На каждом уровне также может быть установлен дополнительный демультиплексор для выбора приемника данных (ВБ следующего уровня либо внешняя шина данных).

### 3. Среда быстрого прототипирования архитектур процессоров ПДВП

Для быстрого прототипирования архитектур процессоров вычисления ПДВП в зависимости от конкретного приложения авторами предлагается среда проектирования, представленная на рис. 10. Входными параметрами для среды являются: структура (либо предельная структура для реконфигурируемой системы) дерева ПДВП; коэффициенты базисных вейвлет-фильтров, тип используемой памяти, разрядность вычислений. При помощи программного обеспечения, реализованного в среде MATLAB, рассчитываются параметры банка анализа на основе лестничных структур, осуществляется трансформация алгоритма на арифметику с фиксированной запятой, и далее генерируется VHDL-пакет параметров описания ПДВП процессора. Данный пакет объединяется с описаниями компонентов процессора в синтезируемое VHDL-описание, которое при помощи специализированных САПР реализуется на конкретной микросхеме FPGA.



Рис. 10. Среда разработки

#### 4. Поточный процессор ПДВП аппроксимирующий шкалу барков

В задачах обработки речевых сигналов, основанных на перцептуальном восприятии информации человеком, необходимо применение банка фильтров, который осуществляет частотную декомпозицию в соответствии со шкалой барков. Ниже рассмотрена реализация поточной архитектуры процессора, вычисляющего ПДВП для фиксированного дерева, аппроксимирующего шкалу барков для частотного диапазона сигнала 0-8000 Гц (рис. 11).

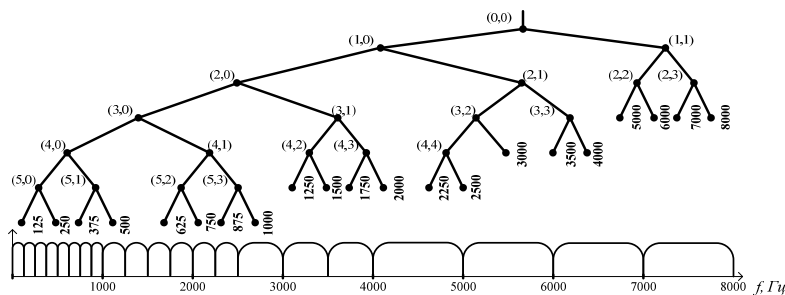


Рис. 11. Структура дерева ПДВП, аппроксимирующая шкалу барков в полосе частот 0-8000 Гц

Процессор состоит из шести уровней декомпозиции. Хранение промежуточных данных организовано на основе регистровых файлов. Разрядность внутренних вычислений 18 бит, разрядность входных данных 16 бит. Латентность получения выходных результатов 12 тактов. Базисная вейвлет-функция – Db10 (20 коэффициентов). Аппаратные затраты на основные типы ресурсов приведены в табл. 2.

Оценочная частота тактирования для данной реализации, полученная в САПР Xilinx ISE для кристалла семейства Virtex-4 xc4vlx25ff668-12, составила 8 МГц. При фреймовой обработке и наличии предварительной буферизации для данных, поступающих с частотой дискретизации 16 кГц, используется лишь 0,2 % от имеющегося времени, что тем самым высвобождает больше времени для субполосной обработки.

Таблица 2

Аппаратные затраты на реализацию поточного процессора

| Тип ресурса                               | Количество |
|---|------------|
| Регистры, разрядность 18 бит              | 547        |
| Сумматоры, разрядность 18 бит             | 160        |
| Умножители 18×18                          | 176        |
| Мультиплексоры 2-1 (демультиплексоры 1-2) | 17         |

## Заключение

В качестве основных результатов рассмотренной работы можно отметить следующие моменты. Представлены два подхода для организации поточной архитектуры процессора: с использованием блоков памяти и на основе регистровых файлов. Первый подход находит применение в приложениях с большим числом уровней декомпозиции, в динамически перестраиваемых системах, при мультифреймовой обработке. Второй обеспечивает высокую пропускную способность, гибкость реализации под любую структуру дерева, простое устройство управления.

Особенностью предложенных реализаций является высокое быстродействие за счет использования банка анализа на лестничных структурах, что при фреймовой обработке данных обеспечивает хороший запас по времени на субполосную обработку.

В качестве практического результата в работе представлена среда проектирования архитектур поточных процессоров ПДВП под определенный класс приложений, что позволяет осуществить автоматический переход от программной модели в среде MATLAB к проекту в виде синтезируемого VHDL-описания.

## Список литературы

1. Petrovsky, Al. Dynamic algorithm transforms for reconfigurable real-time audio coding processor / Al. Petrovsky, A. Petrovsky // The proc. of the Intern. Conf. on Parallel Computing in Electrical Engineering, PARELEC'02, Warsaw, Poland, Sep. 22-25, 2002, IEEE Computer Society Press. – Los Alamitos, California, 2002. – P.422-424.
2. Sweldens, W. The lifting scheme: A construction of second generation wavelets / W. Sweldens // Siam J. Math. Anal. – 1997. – Vol. 29(2) – P. 511-546.
3. Daubechies, I. Factoring wavelet transforms into lifting steps / I. Daubechies, W. Sweldens, // Journal of Fourier Anal. Appl. – 1998. – Vol. 4, № 3, – P. 247-269.
4. Architecture for wavelet packet transform with best tree searching / M. A. Treans [at al.] // 12th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP'00). – 2000. – P.289-298.
5. Coifman, R. R. Entropy-based algorithms for best basis selection / R.R. Coifman, M.V. Wickerhauser // Trans. Inform. Theory. –1992. – Vol. 38. – P. 1713-1716.
6. Wang, C. Efficient VLSI Architecture for Lifting-Based Discrete Wavelet Packet Transform / C. Wang, W.S. Gan // IEEE Transactions on circuits and system – II: Express briefs. – 2007. – Vol. 54(5). – P. 422-426.
7. Mallat, S. A wavelet tour of signal processing: 2nd edition / S. Mallat. – Academic Press, 1999 – 637 p.
8. Design and DSP implementation of fixed-point systems / M. Coors [at al.] // EURASIP journal on applied signal processing. – 2002– №9 – P. 908-925.
9. Menard, D. Floating-to-fixed-point conversion for digital signal processors / D. Menard, D. Chillet, O. Sentieys // EURASIP J. on Applied Signal Processing. – 2006. – Vol. 2006 –P. 1-19.
10. Daubechies, I. Orthogonal bases of compactly supported wavelets / I. Daubechies // Communications on pure and applied mathematics, – 1988. –Vol. 41. – P. 909-996.
11. Petrovsky, Al. Real-time wavelet packet-based low bit rate audio coding on a dynamic reconfiguration system / Al. Petrovsky, D. Krahe, A.A. Petrovsky.// AES, Convention paper 5778, 114<sup>th</sup> Convention, 2003 March 22-25, Amsterdam, The Netherlands. – Amsterdam, 2003 – 22 p.