

# Реализация блочного КИХ-фильтра в потоковом рекуррентном сигнальном процессоре

Д.В. Хилько

Федеральный исследовательский центр «Информатика и Управление» РАН, г. Москва, dhilko@yandex.ru

**Аннотация** — В статье рассматриваются аспекты апробации прототипа потокового рекуррентного сигнального процессора на одном из ключевых алгоритмов цифровой обработки сигналов – фильтре с конечной импульсной характеристикой. Первая попытка реализации блочного КИХ-фильтра показала высокий уровень производительности рассматриваемого прототипа. Однако избыточность потоковой программы оказалась слишком высокой. Был осуществлен анализ методов программной и аппаратной оптимизации реализации КИХ-фильтров. По результатам данного анализа определены основные направления для усовершенствования прототипа рекуррентного сигнального процессора. Средства аппаратной поддержки алгоритма Быстрого преобразования Фурье, созданные на более ранних этапах разработки прототипа, были успешно доработаны и использованы для реализации КИХ-фильтра. Данное решение позволило снизить избыточность потоковой программы реализации блочного КИХ-фильтра почти на 80% и повысить скорость загрузки отсчетов обрабатываемого сигнала.

**Ключевые слова** — потоковая архитектура, рекуррентность, цифровая обработка сигналов, цифровые фильтры, КИХ-фильтры, БПФ.

## I. ВВЕДЕНИЕ

Актуальность разработки вычислительных архитектур на базе потоковой парадигмы не вызывает сомнений, т.к. потенциально они могут обеспечить существенно большую производительность по сравнению с традиционной фон-неймановской архитектурой. Но эффективная реализация потоковых архитектур (ПА) оказалась сопряжена с целым рядом проблем, отмеченных в ранних работах по их разработке [1].

В России серьезные результаты в области разработки универсальных высокопроизводительных систем, использующих принцип управления потоком данных, были достигнуты коллективом под руководством академика Бурцева В.С. После его кончины и перехода большей части его коллектива из ИПИ РАН в ИППМ РАН работы по совершенствованию ПА были продолжены уже там. Данный коллектив ведет разработку параллельной потоковой вычислительной системы «Буря» [2], которая предназначена для реализации мелкозернистого параллелизма в суперкомпьютерных системах.

Также существенные научно-практические результаты в области разработки вычислительных систем

(ВС) на основе ПА были получены коллективом ФИЦ ИУ РАН. Данный коллектив исследует и разрабатывает концепцию многоядерной потоковой рекуррентной архитектуры. Архитектура базируется на рекуррентно-динамической парадигме вычислений, которая расширяет потоковую модель вычислений принципами самодостаточности и рекуррентности.

Для экспериментальной апробации рассматриваемой архитектуры разработан ее прототип: рекуррентный обработчик сигналов (РОС), исполняемый в гибридном, двухуровневом варианте с ведущим фон-неймановским процессором на управляющем (верхнем) уровне (УУ) и рядом потоковых процессоров на нижнем уровне – рекуррентном операционном устройстве (РОУ). Прототип был назван «Гибридная архитектура рекуррентного обработчика сигналов» (ГАРОС). Детальное описание ключевых принципов архитектуры и ее прототипа ГАРОС приводятся в работе [3].

Реализация ГАРОС осуществлялась в несколько этапов, в ходе которых были разработаны элементы методического обеспечения. В том числе, программная имитационная и аппаратная VHDL-модели. На основе отлаженной VHDL-модели был осуществлен синтез ПЛИС прототипа и проведена его апробация, результаты которых представлены в работе [4]. Полученный ПЛИС прототип был верифицирован в соответствии с методикой, представленной в работе [5].

ГАРОС является специализированной архитектурой для области цифровой обработки сигналов (ЦОС). Поэтому объективным способом доказательства ее эффективности является реализация комплекта бенчмарков, используемых ведущими производителями цифровых сигнальных процессоров (ЦСП). Таким комплектом бенчмарков, согласно книге [6], является VDTMark2000. В табл. 1 приводится список алгоритмических ядер VDTMark2000. В статье [7] приводятся результаты оценки производительности прототипа ГАРОС на комплекте данных ядер в количестве циклов, полученные в результате модельных испытаний.

Важнейшим алгоритмом ЦОС является Быстрое преобразование Фурье (БПФ), эффективная реализация которого является индикатором состоятельности ЦСП. Поэтому для ГАРОС были разработаны средства аппаратной поддержки БПФ [8], которые позволили эффективно реализовать Point-In-Place алгоритм.

Таблица 1

Алгоритмические ядра *BDTMark2000*

Функция	Описание функции
Real Block FIR	Фильтр с конечной импульсной характеристикой (КИХ), который выполняется на блоке действительных данных.
Complex Block FIR	КИХ-фильтр, который выполняется на блоке комплексных данных.
Real Single-Sample FIR	КИХ-фильтр, который выполняется на одиночном действительном отсчете.
LMS Adaptive FIR	Адаптивный фильтр наименьших квадратов; работает с одиночным действительным отсчетом.
IIR	Фильтр с бесконечной импульсной характеристикой (БИХ), работающий с одиночным действительным отсчетом.
Vector Dot Product	Сумма поточечного умножения двух векторов.
Vector Add	Поточечное сложение двух векторов, в результате чего получается третий вектор.
Vector Maximum	Поиск значения и расположения максимального значения в векторе.
Viterbi Decoder	Декодирование блока битов, который был закодирован сверточным кодированием.
Control	Последовательность управляющих операций (тест, ветвление, push, pop и манипулирование битами).
256-Point In-Place FFT	Быстрое преобразование Фурье преобразует сигнал временной области в частотную.
Bit Unpack	Распаковывает данные переменной длины из битового потока.

Другим ключевым классом задач ЦОС является цифровая фильтрация, а именно, фильтры с конечной импульсной характеристикой (КИХ-фильтры). Данный вид фильтров обладает рядом существенных характеристик, которые делают его применение более предпочтительным по сравнению с фильтрами с бесконечной импульсной характеристикой (БИХ-фильтры).

Реализация блочного КИХ-фильтра имеет хорошую оценку производительности. Однако первоначальная версия реализации данного фильтра имела высокую степень избыточности исполняемой в ГАРОС программы (называемой капсулой). Кроме того, новые средства аппаратной поддержки БПФ также не были использованы. Данная статья посвящена анализу проблем реализации блочного КИХ-фильтра в ГАРОС. Предлагается ряд усовершенствований средств поддержки БПФ, которые позволили использовать данные средства для эффективной реализации КИХ-фильтра.

## II. АНАЛИЗ МЕТОДОВ ЭФФЕКТИВНОЙ РЕАЛИЗАЦИИ КИХ-ФИЛЬТРОВ

### A. Свойства КИХ-фильтра

Фильтром с конечной импульсной характеристикой называют цифровой фильтр, импульсная характеристика которого ограничена по времени (т.е., начиная с некоторого момента времени, сходится к нулю). Данный вид фильтра не имеет обратной связи и потому называется *нерекурсивным*.

Фильтром с бесконечной импульсной характеристикой называют цифровой фильтр, импульсная характеристика которого расходится при стремлении време-

ни к бесконечности. Данный вид фильтра образует обратную связь и называется *рекурсивным*.

Типовой КИХ-фильтр является сверткой входного сигнала и импульсного отклика и может быть представлен формулой (1), которая называется *прямой формой* [9,10].

$$y[n] = \sum_{k=0}^N h_k * x[n - k] \quad (1).$$

где  $x[n]$  - отсчет входного сигнала,  $y[n]$  - отсчет выходного сигнала,  $N$  - порядок фильтра (количество коэффициентов),  $h_k$  -  $k$ -ый коэффициент фильтра.

Главным преимуществом БИХ-фильтров является их небольшой порядок и, следовательно, высокая скорость вычисления (т.к. необходимо вычислять малое количество умножений). В тоже время, проектировать БИХ-фильтры существенно труднее, т.к. они требуют осуществлять проверку на устойчивость.

По сравнению с БИХ-фильтрами, КИХ-фильтры обладают следующими свойствами [11]:

- 1) Всегда устойчивы.
- 2) Не используют обратную связь.
- 3) Могут иметь симметричную структуру.
- 4) Могут гарантировать линейную фазочастотную характеристику, если имеют симметричную структуру.
- 5) Могут быть вычислены с использованием арифметики с фиксированной точкой одинарной точности.

Таким образом, КИХ-фильтры являются незаменимыми инструментами для решения целого ряда задач ЦОС. Поэтому ГАРОС должен предоставлять набор инструментов для эффективной реализации данной разновидности фильтров.

В рамках данной статьи рассматриваются аспекты реализации блочного КИХ-фильтра средствами ГАРОС. Блочным данный фильтр называется потому, что за один запуск он должен обработать блок входных отсчетов. То есть данный фильтр ставит в соответствие множеству входных отсчетов размера  $M$  множество выходных отсчетов размера  $M$ . Поэтому он не может быть использован в реальном масштабе времени.

### B. Методы эффективной реализации КИХ-фильтров

Существует два основных пути повышения производительности ВС на конкретной задаче. Это программная и аппаратная оптимизация. Под программной оптимизацией реализации КИХ-фильтра понимается такая модификация исполняемой программы, которая наилучшим образом задействует имеющиеся аппаратные ресурсы вычислительной системы. Под аппаратной оптимизацией понимается ввод в состав архитектуры вычислительной системы специализированных блоков, которые обеспечивают эффективное исполнение отдельных фрагментов алгоритма.

Современные ЦСП оснащены множеством аппаратных блоков, предназначенных для оптимизации различных алгоритмов ЦОС. Например, наиболее распространенными являются блоки аппаратной под-

держки БПФ. Кроме того, в состав ЦСП вводятся дополнительные блоки умножения с накоплением (Multiply and Accumulate – MAC), механизмы циклической адресации, аппаратная поддержка циклов и др.

В частности, ЦСП компании Texas Instruments из линейки TMS320C55x имеет два MAC-блока. Как показано в техническом отчете [12] программная оптимизация реализации КИХ-фильтра для TMS320C55x заключается в специальной структуризации исполняемого алгоритма, которая позволяет использовать оба MAC-блока. Тогда при вычислении блочного КИХ-фильтра появляется возможность одновременного вычисления двух последовательных выходных отсчетов фильтрации, путем пересылки одной и той же константы в разные MAC-блоки.

Максимального повышения скорости вычислений КИХ-фильтра можно достичь путем аппаратной оптимизации вычислительной системы. Причем степень модернизации аппаратных средств может варьироваться от ввода нескольких блоков до полной аппаратной реализации фильтра. Основным критерием модернизации является универсальность полученной ВС.

ЦСП являются универсальными ВС для решения задач ЦОС. Поэтому ЦСП реализуют небольшое количество механизмов, позволяющих эффективно вычислять КИХ-фильтры. В статье [13] подробно рассматриваются механизмы аппаратной поддержки вычисления сверток сигналов, к которым относятся:

- 1) Циклическая адресация. Это специализированный вид адресации, который используется в ЦСП для организации цифровых линий задержек входных отсчетов. Контроллер памяти, который реализует механизм циклической адресации, осуществляет модернизацию адреса на заданную величину смещения и автоматический возврат к началу буфера при достижении его границы. Это позволяет избавиться от необходимости постоянной проверки достижения границы буфера.
- 2) Аппаратная поддержка циклов. Данный механизм позволяет сохранить небольшое количество команд тела цикла в отдельном буфере и организовывать их выборку без обращения к памяти команд. Так как тело цикла вычисления свертки невелико, это позволяет существенно сократить издержки доступа к памяти.
- 3) Оптимальное размещение данных. Для наиболее эффективного выполнения MAC-операции за один такт ЦСП требуется считать одну команду и два данных. Достигается это путем расслоения памяти на два параллельных банка, из которых одновременно осуществляется чтение двух требуемых операндов.
- 4) Very Long Instruction Word (VLIW), суперскалярность или Single Instruction Multiple Data (SIMD) инструкции. Использование VLIW-архитектуры или суперскалярной микроархитектуры позволяет одновременно исполнять несколько инструкций. А использование SIMD инструкций позволяет исполнять векторную инструкцию над разными наборами данных, что особенно эффективно для многоядерных ЦСП.

Другим направлением реализации эффективных КИХ-фильтров, широко освещенным как в отечественной, так и в зарубежной литературе, является разработка аппаратных КИХ-фильтров. В статье [14] приводится обширный обзор различных подходов к построению аппаратных КИХ-фильтров. К данным подходам относятся:

- 1) Параллельное или блочное вычисление фильтра.

Например, в статье [15] авторы предлагают структуру параллельного КИХ-фильтра 4-го порядка в прямой форме, который реализуется на ПЛИС и предназначен для System-On-Chip (SoC) применений. Данная реализация не использует конвейеризацию и позволяет за 1 цикл для одного входного сигнала вычислить одно выходное значение сигнала, что подходит для применения в реальном времени. Недостатком такой реализации являются высокие аппаратные расходы и низкая энергоэффективность.

Также в статье [16] предлагается структура параллельного симметричного КИХ-фильтра в прямой форме, реализуемый на ПЛИС. В отличие от фильтра из [14], рассматриваемый фильтр использует конвейеризацию, что позволяет существенно экономить аппаратные ресурсы и повысить энергоэффективность.

В статье [17] авторы предлагают структуру параллельного четырехполосного КИХ-фильтра в транспонированной форме, коэффициенты которого рассчитаны оптимальным образом в формате с фиксированной точкой. Реализация в транспонированной форме позволяет одновременно вычислять для четырех входных отсчетов четыре выходных отсчета. Кроме того, использование конвейеризации позволяет реализовать фильтр любого порядка.

- 2) Оптимизация микроархитектуры умножителей.

Данный метод может сочетаться с предыдущим для: повышения общей производительности аппаратного фильтра, снижения аппаратных расходов и повышения энергоэффективности. В работах [18-21] рассматриваются различные способы реализации блоков умножений: Ведический умножитель; дерево Уоллеса; умножитель Бута; усовершенствованный умножитель Бута; сдвигатель-сумматор; различные варианты оптимизации переноса при накоплении.

- 3) Представление коэффициентов фильтра в виде суммы степеней двойки (signed-powers-of-two SPT).

Как показано в работах [21-24], представление коэффициентов в виде SPT позволяет использовать вместо дорогостоящих и потребляющих много энергии блоков MAC серию сдвигателей и сумматоров. Это позволяет значительно сократить аппаратные затраты и повысить энергоэффективность.

Другим этапом развития SPT стало каноническое представление чисел со знаком (Canonical signed digit representation – CSD), что фактически является трюковой логикой. В работах [21, 22] показано, что CSD является более эффективным по сравнению с бинарным

представлением, т.к. содержит на 33% меньше ненулевых битов, что снижает аппаратные затраты.

4) Использование распределенной арифметики и умножения нескольких констант.

Метод распределенной арифметики описан в работе [24] и заключается в дальнейшем усовершенствовании КИХ-фильтра без умножителей, коэффициенты которого представлены в SPT. В силу того, что КИХ-фильтр в прямой форме является конвейеризованным, то каждый новый отсчет должен быть последовательно умножен на все коэффициенты. Каждый бит в SPT коэффициентов индицирует требуемый сдвиг отсчета. Сохраняя результаты сдвига для конкретного бита в специальную Look-up table (LUT), можно значительно уменьшить количество операций сдвига для КИХ-фильтров достаточно высоких порядков. Таким образом, по мере продвижения по линии задержки, в LUT будут сохранены значения сдвинутого входного отсчета. А дальнейшие вычисления сведутся к чтению LUT и накоплению результата.

В основе метода умножения нескольких констант (multiple constant multiplication – MCM), который описан в работе [25], лежит идея аналогичная распределенной арифметике. Суть метода заключается в снижении количества сложений путем сохранения результатов накопления одинаковых подвыражений в SPT разных коэффициентов (имеется в виду, что различные константы могут иметь идентичные группы битов в разложении по степеням 2, что и позволяет один раз вычислить сдвиги и накопления для все группы битов). Полученное значение используется при вычислении умножений входного отсчета на все константы фильтра (по аналогии с распределенной арифметикой). Метод MCM используется для реализации транспонированных КИХ-фильтров большого порядка с постоянными коэффициентами.

### III. РЕАЛИЗАЦИЯ КИХ-ФИЛЬТРА В ГАРОС

#### A. Описание ГАРОС

Ранее было отмечено, что ГАРОС реализует расширенную потоковую модель вычислений и базируется на двух ключевых принципах: рекуррентность и самодостаточность. Самодостаточными называются данные, которые включают в свою структуру как непосредственно данные, так и инструкции (теги) для их обработки. Другими словами, два традиционных потока – данных и инструкций – объединяются в единый самодостаточный поток. Рекуррентностью называется свойство динамического развития траектории вычислительного процесса путем вычисления новых значений теговых полей самодостаточных данных с помощью преобразования тегов.

Самодостаточный поток состоит из элементов самодостаточных данных (ЭСД), называемых также операндами. Каждый операнд несет в: себе данное; инструкцию, которая должна быть выполнена; начальное состояние теговых полей для последующей рекуррентной развертки.

ГАРОС является двухуровневой архитектурой. На верхнем управляющем уровне прототипа используется фон-Неймановский процессор общего назначения, который исполняет функции процессора ввода-вывода и управляет подготовкой потока ЭСД. Кроме того, данный процессор используется для реализации последовательных фрагментов исполняемого алгоритма.

На нижнем операционном уровне РОУ принимает на вход посредством двух-портовой буферной памяти (БП) сформированные на верхнем уровне ЭСД, а также размещает в этой памяти выходные данные. Операционный уровень состоит из селектирующей среды, названной «Распределитель», а также четырех идентичных вычислительных ядер, каждое из которых содержит модули Памяти совпадений (ПС), МАС-блоки, регистровый файл, а также блоки Преобразователей тегов (ПТ).

На рис. 1 представлена структура ГАРОС.

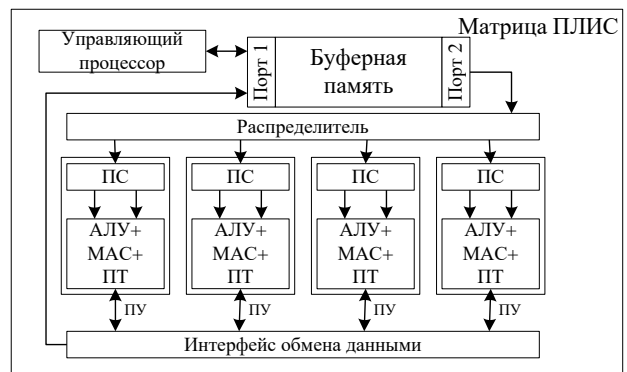


Рис. 1. Структура ГАРОС

В текущей версии прототипа ЭСД имеют размер равный 56 битам. Вычислительные модули реализуют арифметику с фиксированной точкой, а разрядность обрабатываемых данных составляет 16 бит.

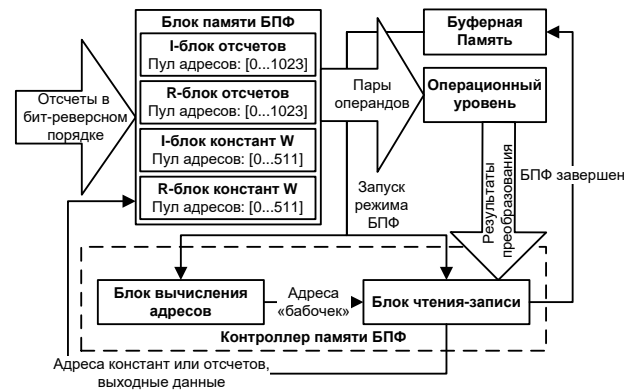


Рис. 2. Структура средств поддержки БПФ (рис. 8 из [8])

В процессе разработки средств аппаратной поддержки БПФ компонент БП был значительно модифицирован. Был добавлен дополнительный блок памяти, предназначенный для хранения отсчетов БПФ и поворотных коэффициентов. Новый блок памяти состоит из 4 банков памяти: банки действительных и мнимых частей отсчетов, банки действительных и мнимых час-

тей поворотных коэффициентов. Также был добавлен вспомогательный контроллер памяти, обеспечивающий вычисление адресов и обработку выходных данных для In-Place реализации. На рис. 2 (рис. 8 из [8]) приводится структура средств поддержки БПФ.

### *В. Проблемы реализации КИХ-фильтра*

В ходе оценки производительности прототипа ГАРОС на комплекте бенчмарков VDTIMark200 была осуществлена реализация первой версии блочного КИХ-фильтра. Данный фильтр имеет следующую спецификацию:  $nh=16$  констант в формате Q.15 и блок из 256 входных отсчетов в формате Q.15. Полученная оценка производительности на одном МАС-блоке составила  $px*(1+nh) + 12$ , то есть 4364 цикла. Однако размер капсулы составил 360 операндов. С учетом того, что каждый операнд имеет размер 56 битов, расширенный до 64-битного слова в памяти, избыточность данной реализации по памяти оказалась колоссальной.

В результате анализа первой версии реализации фильтра и функциональных возможностей прототипа ГАРОС были выявлены следующие проблемы:

- 1) Высокая степень избыточности капсулы (программы КИХ-фильтра). Входные отсчеты и константы не удалось упаковать по 4 штуки в 64 разрядное слово.
- 2) БП не поддерживает циклическую адресацию из-за специфики реализации потоковой архитектуры.
- 3) Данные фильтра оказались размещены в капсуле неоптимальным образом – на физическом уровне отсчеты и коэффициенты хранятся в одном банке памяти. Следствием этого стала невозможность упаковать их (см. п.п. 1)). Тем не менее, банк памяти имеет расслоенную структуру на 8 банков, что позволяет считывать из БП до 4-х 64-разрядных операндов за цикл.
- 4) Дополнительные средства аппаратной поддержки БПФ не использовались для реализации КИХ-фильтра. Существующие средства настройки не предусматривали использования данных средств в других сценариях.

### *С. Реализация механизмов аппаратной поддержки КИХ-фильтров*

Прототип ГАРОС является универсальным ЦСП. Следовательно, для наиболее эффективной реализации КИХ-фильтров данный прототип должен реализовывать какие-либо механизмы из рассмотренных в разделе II В. Проведем анализ, какие из перечисленных механизмов могут быть реализованы:

- 1) Циклическая адресация. Может быть реализована при помощи модификация контроллера памяти БПФ. Связано это со структурой I- и R- блоков отсчетов БПФ. Каждый блок позволяет хранить линейным образом до 1024 16-разрядных данных. Следовательно, каждый из этих блоков может адресоваться при помощи механизма циклической адресации.
- 2) Аппаратная поддержка циклов. Не может быть реализована в ГАРОС, т.к. самодостаточный поток данных не подразумевает буферизацию некоторого подмножества инструкций. Кроме того, в вычислительных блоках ГАРОС уже реализован механизм ап-

паратного ускорения циклических процедур (с точки зрения организации перехода при достижения счетчиком нулевого значения).

- 3) Оптимальное размещение данных. Может быть достигнуто путем размещения памяти и отсчетов фильтра в I-блоке памяти БПФ, а коэффициентов фильтра – в R-блоке памяти БПФ.
- 4) Суперскалярность и SIMD. Вычислительные блоки ГАРОС имеют суперскалярную архитектуру. Кроме того, селектирующая среда Распределителя совместно с режимом распаковки данных позволяют рассылать во все МАС-блоки операнды, несущие в себе одинаковые инструкции. Данное поведение аналогично SIMD инструкции, хотя и не является ей.
- 5) Параллельное или блочное вычисление фильтра. Первая версия реализации КИХ-фильтра уже является блочной, параллельной и конвейеризованной в транспонированной форме. Отсчеты и коэффициенты представлены в формате Q.15.
- 6) Оптимизация микроархитектуры умножителей. В рамках используемого инструментария синтеза ПЛИС прототипа существует возможность конфигурации аппаратного блока умножения на заданную оптимизированную архитектуру.
- 7) Представление коэффициентов фильтра в SPT-виде. Не может быть реализован в ГАРОС, т.к. это универсальный ЦСП, содержащий 4 МАС-блока.
- 8) Использование распределенной арифметики и умножения нескольких констант. Также не может быть реализовано в прототипе ГАРОС в силу п.п. 7).

Таким образом, для решения рассмотренных проблем реализации были введены или модифицированы следующие механизмы и элементы архитектуры:

- 1) Введена конфигурация режима FIR для контроллера памяти БПФ.
- 2) В состав контроллера памяти БПФ введены следующие параметры:
  - FS (Filter Size) – размер блока отсчетов,
  - FO (Filter Order) – порядок фильтра (число коэффициентов),
  - FL (Filter Length) = (FS + FO – 1) – суммарная длина блока отсчетов с учетом памяти фильтра,
  - FU (Filter Unfolding) – степень параллельности фильтра (число одновременно вычисляемых выходных отсчетов),
  - SD (Sample Delta) – величина приращения адреса текущего отсчета,
  - CD (Coefficient Delta) – величина приращения адреса текущего коэффициента.
- 3) В состав контроллера памяти БПФ введены следующие индексные регистры:
  - SW (Sample Window) – адрес начала текущего «окна» отсчетов, используемых для вычисления свертки,

- CW (Coefficient Window) – адрес начала «окна» коэффициентов фильтра (не изменяется в рамках вычисления одного фильтра).

4) Для вычисления адресов считываемых на текущем шаге отсчетов и константы применяются массивы индексных регистров bFlyAdrs и constAdrs соответственно. Данные регистры используются в режиме FFT для вычисления адресов считываемых действительных и мнимых отсчетов, а также действительных и мнимых констант. Адреса в массиве bFlyAdrs отличаются друг от друга на величину SD. Приращение constAdr осуществляется на величину CD.

5) На основе значений SW и FL в контроллере памяти БПФ реализована циклическая адресация по I-блоку, в котором хранятся отсчеты фильтра.

6) На основе значений CW и FO в контроллере памяти БПФ реализована циклическая адресация по R-блоку, в котором хранятся коэффициенты фильтра.

7) В режиме FIR контроллер памяти БПФ каждый цикл вычислений формирует на выходе два упакованных операнда по 4 данных каждое. Первый операнд содержит значения отсчетов, а второй – разное значение коэффициента.

8) Для записи блока отсчетов и коэффициентов фильтра используется механизм прямого доступа в блоки памяти БПФ. Данный механизм также используется для записи отсчетов в режиме FFT.

9) Для чтения памяти фильтра используется механизм прямого доступа в блоки памяти БПФ, который также используется для чтения отсчетов в режиме FFT.

10) Результаты фильтрации наполняются в выходной раздел капсулы в соответствии с уже существующим в ГАРОС механизмом.

11) Для конфигурации новых параметров используется специальный Acm: операнд, который определяет режим работы БП. В режиме FIR контроллер памяти БПФ ставит в соответствие полям данного операнда иную семантическую нагрузку. В табл. 2 приводится интерпретация полей Acm: операнда в разных режимах. Значение FU извлекается из поля операнда глобальной конфигурации Acg:, которое устанавливает количество задействованных секций (степень параллельности).

12) Модификация SW и уменьшение счетчика итераций капсулы осуществляется всякий раз, когда контроллер памяти БПФ фиксирует циклический возврат в самому первому коэффициенту. Величина модификации SW соответствует значению FU.

Таблица 2

*Интерпретация полей Acm: операнда*

Поле	Семантика (не FIR)	Семантика (FIR)
%Ci	Количество итераций	Не изменилась
%COs	Начальное значение IR0	SW
%C1s	Начальное значение IR1	CW
%C0d	Приращение IR0 за итерацию	SD
%C1d	Приращение IR1 за итерацию	CD

%Cdm	Режим работы БП и Распределителя	Не изменилась
%C0m	Модуль модификации значения адреса регистра IR0	FS
%C1m	Модуль модификации значения адреса регистра IR1	FO

#### IV. ЗАКЛЮЧЕНИЕ

Анализ различных методов повышения производительности при вычислении блочных КИХ-фильтров позволил модифицировать средства аппаратной поддержки БПФ. Эта модификация позволила конфигурировать и использовать данные средства для эффективного вычисления блочных КИХ-фильтров.

В результате капсула, которая реализует блочный КИХ-фильтр со спецификацией 256 отсчетов и 16 коэффициентов, была уменьшена до необходимого минимума с 360 операндов до 73, т.е. примерно на 80%, без потерь производительности. При этом значительный объем неиспользуемой памяти БПФ и ресурсов ее контроллера оказались задействованы. Кроме того, ввиду линейной организации блоков памяти БПФ, запись в них осуществляется быстрее, чем загрузка первой версии капсулы.

Таким образом, усовершенствованный прототип ГАРОС позволяет вычислять блочный параллельный конвейеризованный КИХ-фильтр в транспонированной форме. Теоретический максимальный порядок фильтра, который сможет вычислить ГАРОС, составляет примерно 1020. При этом имеется возможность конфигурации степени параллелизма фильтра от 1 до 4.

Тем не менее, остались нерешенными ряд вопросов, связанных с реализацией двух других КИХ-фильтров из комплекта бенчмарков VDTMark2000. А именно, Single-Sample FIR и Complex Block FIR (см. табл. 1). Однако средства поддержки КИХ-фильтров разрабатывались таким образом, чтобы они могли быть легко расширены для поддержки указанных разновидностей КИХ-фильтров.

#### БЛАГОДАРНОСТИ

Автор статьи выражает благодарность всему составу отдела «Перспективных архитектур компьютерных систем» ФИЦ ИУ РАН за неоценимый вклад в исследование и разработку проблемы эффективной реализации КИХ-фильтров с помощью ГАРОС.

#### ЛИТЕРАТУРА

- [1] Ben Lee, A.R. Hurson. Issues in Dataflow Computing, Editor(s): Marshall C. Yovits, Advances in Computers, Elsevier, Volume 37, 1993, Pages 285-333.
- [2] Стемпковский А.Л., Бобков С.Г., Змеев Д.Н., Левченко Н.Н., Климов А.В. Автоматизированное определение оптимальной конфигурации параллельной потоковой вычислительной системы для решения конкретной задачи // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2021. Выпуск 3. С. 82-93.
- [3] Yu. A. Stepchenkov, Yu. G. Diachenko, D. V. Khilko, V.S. Petrukhin. Recurrent data-flow architecture: features and realization problems // Problems of Perspective Micro- and

- Nanoelectronic Systems Development, 2017, Part II, Moscow, IPPM RAS, pp. 52-58.
- [4] Yury Stepchenkov, Nikolai Morozov, Dmitry Khilko, Yury Shikunov, Georgy Orlov. Hybrid multi-core recurrent architecture approbation on FPGA // 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) Moscow, Russia, January 28-31, 2019. — IEEE, pp. 1705 — 1708.
- [5] Дьяченко Ю.Г., Степченков Ю.А., Морозов Н.В., Хилько Д.В., Степченков Д.Ю., Шикунов Ю.И. Аппаратная верификация рекуррентного обработчика сигналов на ПЛИС // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2021. Выпуск 2. С. 77-82.
- [6] Lavagno L., Martin G., Markov I.L., Scheffer L.K. Electronic Design Automation for IC System Design, Verification, and Testing. CRC Press, 2016. pp. 215–216.
- [7] Yury A. Stepchenkov, Dmitry V. Khilko, Yury I. Shikunov, Georgy A. Orlov. DSP Filter Kernels Preliminary Benchmarking for Recurrent Data-flow Architecture // 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) St. Petersburg, Moscow, Russia, January 26-29, 2021. — IEEE, pp. 2040-2044.
- [8] Хилько Д.В., Степченков Ю.А., Шикунов Ю.И., Дьяченко Ю.Г., Орлов Г.А. Оптимизация аппаратной поддержки быстро преобразования Фурье в рекуррентном сигнальном процессоре. // Системы и средства информатики, 2021. Т. 31. № 4. С. 71-83.
- [9] Naim Dahoun. Multicore DSP: From Algorithms to Real-time Implementation on the TMS320C66x SoC, First Edition. John Wiley & Sons Ltd, 2018. 648 p.
- [10] Водовозов А.М., Полетаев Д.С. Программирование цифрового линейно-фазового фильтра в архитектуре ARMv8-A / Труды ИСП РАН, том 30, вып. 6, 2018 г., С. 305-314.
- [11] Ричард Лайонс. Цифровая обработка сигналов: Второе издание. Пер. с англ. – М.: ООО «Бином Пресс», 2006 г. – 656 с.: ил.
- [12] David M. Alter Efficient implementation of Real-Valued FIR Filters on the TMS320C55x DSP: Application Report SPRA655 – April 2000. Доступно по: <https://web.iitd.ac.in/~saifkm/docs/EEL319/Practical/fir.pdf> (последний вход 30.08.2022).
- [13] Сотников А. Проектирование с использованием процессоров Analog Devices. Цифровой КИХ-фильтр. // Компоненты и технологии. 2010. № 10(111). С. 61-64.
- [14] Abhijit Chandra, Sudipta Chattopadhyay. Design of hardware efficient FIR filter: A review of the state-of-the-art approaches. // Engineering Science and Technology, an International Journal. 2016. Volume 19. Issue 1. P. 212-226.
- [15] Qasim S. M., Bensaleh M. S. Hardware implementation of microprogrammed controller based digital fir filter. // Lecture Notes in Electrical Engineering. 2014. Vol. 247 LNEE. P. 29-40.
- [16] Горочный В.В. Реализация симметричного разделенного КИХ фильтра на ПЛИС // Инженерный вестник Дона. 2022. № 1(85). С. 97-106.
- [17] Baolin Hou, Yuancheng Yao, Mingwei Qin. Design and FPGA Implementation of High-speed Parallel FIR Filters // Proceedings of the 3rd International Conference on Mechatronics, Robotics and Automation. April 2015. P. 975-979.
- [18] Smitha N Mallya, Sneha Revankar. Efficient Implementation of Multiplier for Digital FIR Filters // International Journal of Engineering Research & Technology (IJERT). May 2015. Vol. 04. ISSUE 05. P. 1661-1664.
- [19] K. N. Parvin and M. Z. Hussain. Multiplication techniques for an efficient FIR filter design for hearing aid applications // 2018 2nd International Conference on Inventive Systems and Control (ICISC). 2018. P. 964-968.
- [20] N. Sameeksha Rai, Pannaga Shree B.S., Meghana Y.P., Arunkumar P. Chavan, H.V. Ravish Aradhya. Design and implementation of 16 tap FIR filter for DSP Applications // 2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAEECC). 2018. P. 1-5.
- [21] Mittal A., Nandi A. and Yadav D. Comparative study of 16-order FIR filter design using different multiplication techniques // IET Circuits Devices Syst. March 2017. Vol. 11. P. 196-200.
- [22] Спажакин М.И. Энергоэффективная реализация высокоскоростных КИХ фильтров на ПЛИС // Цифровая обработка сигналов. 2018. № 2. С. 69-74.
- [23] Каплун Д.И. Разработка и исследование нерекурсивных цифровых фильтров без умножений // Современные проблемы науки и образования. 2012. № 4. С. 117.
- [24] Sivakumar V, Venkateshwarlu K, Gopikrishnarao P, Ramesh G. Efficient implementation of FPGA based distributed arithmetic architecture for FIR filter // proceedings of International Journal of Communications and Engineering. July 2012. Vol. 3. ISSUE 1. P. 16-23.
- [25] Ch.Prathyusha, S.Umamaheshwar. Design of a High-Performance FIR Filter // Advances in Computational Sciences and Technology (ACST). 2017. Vol. 10. P. 3103-3109.

## Block FIR Filter Implementation with a Data-flow Recurrent Signal Processor

D.V. Khilko

Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, [dhilko@yandex.ru](mailto:dhilko@yandex.ru)

**Abstract** — The article covers aspects of the prototype approbation of a data-flow recurrent signal processor in the subject area of digital signal processing. A brief description of the scientific and practical results obtained during the development of the considered prototype is presented. A set of BDTIMark2000 benchmarks is selected to assess the prototype performance. The successful testing of the most impactful DSP algorithm - FFT with point-in-place implemen-

tation through the introduction of hardware support is especially noted. The next essential algorithm for testing the prototype was a filter with a finite impulse response. The first attempt to implement a block FIR filter showed a high level of performance of the covered prototype.

However, the redundancy of the data-flow program turned out to be too high. Therefore, software and hardware optimi-

zation techniques for FIR filters implementation have been analyzed. Based on the analysis results, the main directions for improving the prototype of the recurrent signal processor have been determined. Following techniques have been considered: cyclic addressing mechanisms; hardware support for cycles; optimal memory placement of samples and coefficients; superscalar calculations; parallel and block implementation of the filter; optimization of the multiplier micro-architecture; representation of coefficients in signed-powers-of-two form for implementation without multipliers; distributed arithmetic; multiple constant multiplications.

Applicability analysis of the studied techniques for a prototype of a recurrent signal processor is covered. It concluded that most of the techniques could be used to improve the FFT hardware support. This hardware has been successfully refined and used to implement the FIR filter. The resulting solution reduced the redundancy of the block FIR filter data-flow program by almost 80% and increased the loading speed of the input signal samples.

In conclusion, unresolved problems with the implementation of other types of FIR filters, such as single-sample FIR and complex block FIR are considered. It is shown that the developed tools were designed with the goal of further development and can be efficiently modified for the effective implementation of these filters.

**Keywords** — data-flow architecture, recurrence, digital signal processing, digital filters, FIR filters, FFT.

#### REFERENCES

- [1] Ben Lee, A.R. Hurson. Issues in Dataflow Computing, Editor(s): Marshall C. Yovits, *Advances in Computers*, Elsevier, Volume 37, 1993, Pages 285-333.
- [2] Stempkovsky A.L., Bobkov S.G., Zmejev D.N., Levchenko N.N., Klimov A.V. Automated Optimal Configuration Determination of the Parallel Dataflow Computing System for Solving a Specific Problem // *Problems of Perspective Micro- and Nanoelectronic Systems Development* - 2021. Issue 3. P. 82-93. doi:10.31114/2078-7707-2021-3-82-93
- [3] Yu. A. Stepchenkov, Yu. G. Diachenko, D. V. Khilko, V.S. Petrukhin. Recurrent data-flow architecture: features and realization problems // *Problems of Perspective Micro- and Nanoelectronic Systems Development*, 2017, Part II, Moscow, IPPM RAS, pp. 52-58.
- [4] Yury Stepchenkov, Nikolai Morozov, Dmitry Khilko, Yury Shikunov, Georgy Orlov. Hybrid multi-core recurrent architecture approbation on FPGA // *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) Moscow, Russia, January 28-31, 2019.* — IEEE, pp. 1705 — 1708.
- [5] Diachenko Yu.G., Stepchenkov Yu.A., Morozov N.V., Khilko D.V., Stepchenkov D.Yu., Shikunov Yu.I. Hardware verification of the recurrent signal processor on FPGA // *Problems of Perspective Micro- and Nanoelectronic Systems Development* - 2021. Issue 2. P. 77-82. doi:10.31114/2078-7707-2021-2-77-82
- [6] Lavagno L., Martin G., Markov I.L., Scheffer L.K. *Electronic Design Automation for IC System Design, Verification, and Testing*. CRC Press, 2016. pp. 215–216.
- [7] Yury A. Stepchenkov, Dmitry V. Khilko, Yury I. Shikunov, Georgy A. Orlov. DSP Filter Kernels Preliminary Benchmarking for Recurrent Data-flow Architecture // *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) St. Petersburg, Moscow, Russia, January 26-29, 2021.* — IEEE, pp. 2040-2044.
- [8] Khilko D.V., Stepchenkov Yu.A., Shikunov Yu.I. Diachenko Yu.G., Orlov G.A. Optimizatsiya apparatnoj podderzhki bystrogo preobrazovaniya Fur'e v rekurrentnom signal'nom processore. (Optimization of hardware support for fast Fourier transform in recurrent signal processor.) // *Sistemy i sredstva informatiki*, 2021. T. 31. № 4. S. 71-83.
- [9] Naim Dahnoun. *Multicore DSP: From Algorithms to Real-time Implementation on the TMS320C66x SoC*, First Edition. John Wiley & Sons Ltd, 2018. 648 p.
- [10] Vodovozov A.M., Poletaev D.S. Programmirovaniye cifrovogo linejno-fazovogo fil'tra v arkhitekture ARMv8-A (Programming a Digital Linear Phase Filter in the ARMv8-A Architecture) // *Trudy ISP RAN*, V. 30, Issue. 6, 2018 r., P. 305-314.
- [11] Richard Lajons. *Cifrovaya obrabotka signalov: Vtoroe izdanie (Digital Signal Processing: Second Edition)*. Per. s angl. — M.: OOO «Binom Press», 2006 g. — 656 s.: il.
- [12] David M. Alter Efficient implementation of Real-Valued FIR Filters on the TMS320C55x DSP: Application Report SPRA655 — April 2000. Available at: <https://web.iitd.ac.in/~saifkm/docs/EEL319/Practical/fir.pdf> (access date: 30.08.2022).
- [13] Sotnikov A. Proektirovaniye s ispol'zovaniem processorov Analog Devices. Cifrovoy KIH-fil'tr. (Designing using Analog Devices processors. Digital FIR filter.) // *Komponenty i tekhnologii*. 2010. № 10(111). S. 61-64.
- [14] Abhijit Chandra, Sudipta Chattopadhyay. Design of hardware efficient FIR filter: A review of the state-of-the-art approaches. // *Engineering Science and Technology, an International Journal*. 2016. Volume 19. Issue 1. P. 212-226.
- [15] Qasim S. M., Bensaleh M. S. Hardware implementation of microprogrammed controller based digital fir filter. // *Lecture Notes in Electrical Engineering*. 2014. Vol. 247 LNEE. P. 29-40.
- [16] Gorochnyy V. V. Realizatsiya simmetrichnogo razdelenogo KIH fil'tra na PLIS (Implementation of a symmetrical split FIR filter on FPGA) // *Inzhenernyy vestnik Dona*. 2022. № 1(85). S. 97-106.
- [17] Baolin Hou, Yuancheng Yao, Mingwei Qin. Design and FPGA Implementation of High-speed Parallel FIR Filters // *Proceedings of the 3rd International Conference on Mechatronics, Robotics and Automation*. April 2015. P. 975-979.
- [18] Smitha N Mallya, Sneha Revankar. Efficient Implementation of Multiplier for Digital FIR Filters // *International Journal of Engineering Research & Technology (IJERT)*. May 2015. Vol. 04. ISSUE 05. P. 1661-1664.
- [19] K. N. Parvin and M. Z. Hussain. Multiplication techniques for an efficient FIR filter design for hearing aid applications // *2018 2nd International Conference on Inventive Systems and Control (ICISC)*. 2018. P. 964-968.
- [20] N. Sameeksha Rai, Pannaga Shree B.S., Meghana Y.P., Arunkumar P. Chavan, H.V. Ravish Aradhya. Design and implementation of 16 tap FIR filter for DSP Applications // *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAEECC)*. 2018. P. 1-5.
- [21] Mittal A., Nandi A. and Yadav D. Comparative study of 16-order FIR filter design using different multiplication techniques // *IET Circuits Devices Syst.* March 2017. Vol. 11. P. 196-200.
- [22] Spazhakin M.I. Energoeffektivnaya realizatsiya vysokoskorostnyh KIH fil'trov na PLIS (Energy-efficient implementation of high-speed FIR filters on FPGA) // *Cifrovaya obrabotka signalov*. 2018. № 2. S. 69-74.
- [23] Kaplun D.I. Razrabotka i issledovaniye nerekursivnyh cifrovyyh fil'trov bez umnozhenij (Development and research of non-recursive digital filters without multiplications) // *Sovremennyye problemy nauki i obrazovaniya*. 2012. № 4. S. 117.
- [24] Sivakumar V, Venkateshwarlu K, Gopikrishnarao P, Ramesh G. Efficient implementation of FPGA based distributed arithmetic architecture for FIR filter // *proceedings of International Journal of Communications and Engineering*. July 2012. Vol. 3. ISSUE 1. P. 16-23.
- [25] Ch.Prathyusha, S.Umamaheshwar. Design of a High-Performance FIR Filter // *Advances in Computational Sciences and Technology (ACST)*. 2017. Vol. 10. P. 3103-3109.