

## Аппаратная реализация алгоритмов цифровой обработки сигналов в рекуррентном потоковом процессоре на ПЛИС

*Ю. А. Степченков, Н. В. Морозов, Ю. Г. Дьяченко, Д. В. Хилько,  
Д. Ю. Степченков, Ю. И. Шикунов*

*Федеральный исследовательский центр «Информатика  
и управление» Российской академии наук, г. Москва, Россия*

*YStepchenkov@ipiran.ru*

**Аннотация.** Потоковая вычислительная архитектура является альтернативой традиционной вычислительной фон-неймановской архитектуре. Однако существующие варианты потоковой вычислительной архитектуры имеют ряд серьезных проблем, которые к настоящему времени еще не решены. В работе описаны результаты верификации аппаратной реализации архитектуры гибридного рекуррентного сигнального процессора (ГРСП), представленной аппаратной моделью уровня регистровых передач. Макетный образец реализован на отладочной плате с программируемой логической интегральной схемой Intel Arria10. ГРСП включает в себя фон-неймановский процессор в качестве управляющего уровня и потоковое рекуррентное обрабатывающее устройство с четырьмя вычислительными ядрами в качестве операционного уровня. Тестирование ГРСП на типовом приложении цифровой обработки данных – распознавателе изолированных слов – на отладочной плате подтвердило его битэкзактность имитационной модели и исходной C++-модели распознавателя изолированных слов. Верификация аппаратной реализации ГРСП на синтетических тестах показала, что его производительность в среднем на 5 % превышает производительность цифрового сигнального процессора DSP TMS320C55x фирмы Texas Instruments. Оптимизация аппаратной поддержки быстрого преобразования Фурье ускоряет его расчет, существенно уменьшает размер капсулы, сокращает требуемые аппаратные ресурсы и упрощает его масштабирование.

**Ключевые слова:** рекуррентный сигнальный процессор, гибридная потоковая архитектура, аппаратная модель, ПЛИС, быстрое преобразование Фурье

**Финансирование работы:** работа выполнена в рамках государственного задания № 0063-2019-0010.

**Для цитирования:** Аппаратная реализация алгоритмов цифровой обработки сигналов в рекуррентном потоковом процессоре на ПЛИС / Ю. А. Степченков, Н. В. Морозов, Ю. Г. Дьяченко и др. // Изв. вузов. Электроника. 2022. Т. 27. № 3. С. 356–366. doi: <https://doi.org/10.24151/1561-5405-2022-27-3-356-366>

Original article

## Hardware implementation of the digital signal processing algorithms in recurrent signal processor on FPGA

Yu. A. Stepchenkov, N. V. Morozov, Yu. G. Diachenko, D. V. Khilko,  
D. Yu. Stepchenkov, Yu. I. Shikunov

Federal Research Center "Computer Science and Control"  
of the Russian Academy of Sciences, Moscow, Russia

YStepchenkov@ipiran.ru

**Abstract.** Dataflow architecture is an alternative to traditional von Neumann computing architecture. However, known variants of dataflow architecture have a range of serious problems with no effective solution up to the present day. This paper represents Hybrid Recurrent Signal Processor's (HRSP) hardware verification results. It describes HRSP's register transfer level model implementing its architectural specification, and hardware prototype on HAN Pilot Platform demo-board with field-programmable gate array Intel Arria10. HRSP consists of a von Neumann master processor on a control layer and a recurrent dataflow unit on an operational layer. Data-flow unit includes four computing cores. HRSP's hardware model combines soft or hardware implementation of the control processor and hardware model of the operational layer. Testing the HRSP's hardware prototype on demo-board using an isolated word recognizer (IWR) as a typical data processing application has proved that the hardware model is bit-exact with both HRSP's imitation model and the original IWR C++ model. The HRSP's hardware prototype's achieved performance ensures IWR's operation in real-time mode on demo-board. It is slightly better than the performance of the TMSC55x (Texas Instruments) digital signal processor. Verification of the HRSP's hardware implementation on synthetic tests showed that its performance is, on average, 5 % higher than the performance of the DSP TMSC55x digital signal processor. The results of proposed optimization of hardware support for Fast Fourier Transform (FFT) in HRSP prove that such the optimization speeds up FFT calculation, significantly reduces the capsule size, reduces the required hardware resources and simplifies FFT scaling.

**Keywords:** recurrent signal processor, hybrid dataflow architecture, hardware model, FPGA, fast Fourier transform, FFT

**Funding:** the work has been carried out within the framework of the state task No. 0063-2019-0010.

**For citation:** Stepchenkov Yu. A., Morozov N. V., Diachenko Yu. G., Khilko D. V., Stepchenkov D. Yu., Shikunov Yu. I. Hardware implementation of the digital signal processing algorithms in recurrent signal processor on FPGA. *Proc. Univ. Electronics*, 2022, vol. 27, no. 3, pp. 356–366. doi: <https://doi.org/10.24151/1561-5405-2022-27-3-356-366>

**Введение.** Поточковые вычислительные архитектуры (ПВА) [1, 2] являются альтернативой традиционной вычислительной фон-неймановской архитектуре. Однако в существующих вариантах ПВА к настоящему времени не решены следующие проблемы:

реализация рекурсий, петель (циклов) и итераций, работа с константами и др. Эффективная реализация ПВА основана на использовании ассоциативной памяти большого объема, что целесообразно только для систем массового параллелизма [1].

В работе [3] утверждается, что известным потоковым процессорам требуется в 2–3 раза больше команд для выполнения программы по отношению к процессорам традиционной архитектуры. В работе [2] отмечается, что ориентация разработчиков на господствующую методологию проектирования (парадигму «сбора») привела в конечном счете к утрате конкурентных преимуществ ПВА.

Попытки использования потоковой парадигмы в области цифровой обработки сигналов также имеют многолетнюю историю [4]. Принципы ПВА и требования со стороны алгоритмов цифровой обработки сигналов хорошо сочетаются друг с другом в приложениях, для которых характерна высокая степень внутреннего параллелизма. Основной сдерживающей причиной широкого практического применения такой интеграции является фактор стоимости, который делает нецелесообразным прямолинейное использование в них решений из области ПВА-систем массового параллелизма [4, 5].

В работе [6] рассматривается реализация принципов рекуррентности и самодостаточности данных за счет объединения в одном операнде собственно данных и необходимой информации: кода операции, выполняемой над ними, и места назначения результата обработки. Совокупность таких операндов образует капсулу, предназначенную для выполнения определенного алгоритма аналогично программному коду для фон-неймановской архитектуры и представляющую собой сжатый потоковый граф вычислительного процесса. При выполнении капсулы в ПВА операнды подвергаются рекуррентной развертке, характер которой определяется как исходным видом операндов, так и логикой работы рекуррентных преобразователей в составе аппаратных средств ПВА.

Разработанный в Федеральном исследовательском центре «Информатика и управление» Российской академии наук (г. Москва) гибридный рекуррентный сигнальный процессор (ГРСП) [5, 7] является одним из представителей семейства ПВА. Он позволяет обойтись без ассоциативной памяти большого объема. Для разработки и отладки капсул, реализующих конкретные алгоритмы цифровой обработки сигналов в ГРСП, используются его имитационная модель на языке C# [8] и аппаратная модель на языке VHDL [9]. Первая облегчает и ускоряет процесс разработки и отладки капсул, вторая обеспечивает адекватность реализации ГРСП на отладочной плате с ПЛИС и позволяет оценить эффективность реализации алгоритмов цифровой обработки сигналов на ГРСП в режиме реального времени. Опытная эксплуатация ГРСП на подмножестве типовых алгоритмов цифровой обработки сигналов продемонстрировала высокую эффективность использования ГРСП для решения задач цифровой обработки сигналов, обеспечивающую большую производительность по сравнению с передовыми цифровыми процессорами [9]. В то же время она выявила необходимость расширения аппаратной поддержки некоторых типовых алгоритмов цифровой обработки сигналов, в частности быстрого преобразования Фурье (БПФ) и ряда цифровых фильтров.

В настоящей работе описываются особенности архитектуры ГРСП, реализованной на отладочной плате HAN Pilot Platform с ПЛИС Intel Arria10 SoC [10], рассматривается ее модернизация с целью расширения аппаратной поддержки типовых алгоритмов цифровой обработки сигналов, обсуждаются технические характеристики ГРСП на ПЛИС и результаты тестирования.

**Аппаратная модель ГРСП.** Аппаратная модель ГРСП включает в себя управляющий процессор и рекуррентный обработчик сигналов, представленный контроллером буферной памяти, буферной памятью, распределителем, интерфейсом обмена данными и  $N$  параллельными вычислительными секциями (рис. 1). Каждая вычислительная секция включает в себя память совпадений и вычислитель, имеющий в своем составе умножитель с 40-разрядным аккумулятором (УА40) и аппаратным устройством сдвига вправо и влево на 1–15 разрядов, 16-разрядное арифметико-логическое устройство (АЛУ16) и преобразователь тегов. Вместе с распределителем и интерфейсом обмена данными

вычислительные секции составляют рекуррентное обрабатывающее устройство. Контроллер буферной памяти содержит арбитр обращений к буферной памяти со стороны управляющего процессора (чтение и запись) и рекуррентного обрабатывающего устройства (запись).

Роль управляющего процессора в архитектуре ГРСП может играть процессор любого типа. Главное требование к его характеристикам – достаточное быстродействие, обеспечивающее требуемые темп и характер обработки промежуточных результатов рекуррентного обрабатывающего устройства и записи данных в рекуррентный обработчик сигналов. В текущей реализации функции управляющего процессора выполняет программный процессор NIOS-II, интегрируемый в проект на этапе логического синтеза в САПР Quartus Prime SE 18 [11], или процессор ARM Cortex-A9, реализованный аппаратно в ПЛИС.

Буферная память состоит из четырех банков памяти общей разрядностью 64 бит. В отдельном регистре хранятся биты готовности операндов капсулы, разрешающие чтение операндов из буферной памяти, банки которой являются двухпортовыми ОЗУ. Первый порт служит для записи данных в буферную память со стороны управляющего процессора и рекуррентного обрабатывающего устройства, а также для чтения данных из буферной памяти в управляющий процессор. Он управляется системной частотой шины адреса / данных. Второй порт используется для чтения операндов из буферной памяти в рекуррентное обрабатывающее устройство и управляется внутренней частотой последнего.

Функциональные модули рекуррентного обработчика сигналов образуют кольцевой конвейер. Интерфейс обмена данными отправляет промежуточные и окончательные результаты обработки капсулы в буферную память для рекурсивного использования.

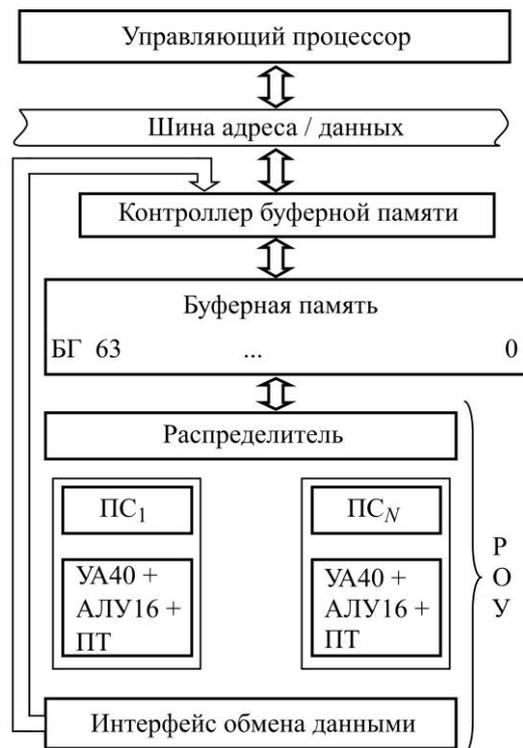


Рис. 1. Структура аппаратной модели ГРСП (ПС – память совпадений; БГ – биты готовности; ПТ – преобразователь тегов; РОУ – рекуррентное обрабатывающее устройство)

Fig. 1. HRSP hardware model structure (ПС – memory of coincidences; БГ – ready bits; ПТ – tag converter; РОУ – recurrent processing device)

**Реализация ГРСП на ПЛИС.** Для аппаратной апробации разработанной архитектуры реализован единственно возможный (быстрый и доступный по цене) вариант в виде ПЛИС-реализации. Он принципиально не может иметь более высокое быстродействие по сравнению с другими видами реализации (заказными и полузаказными ИС на базовом матричном кристалле), но удобен для отладки и позволяет оценить производительность архитектуры ГРСП в тактах частоты рекуррентного обрабатывающего устройства. Аппаратные ресурсы, потребовавшиеся для реализации ГРСП в ПЛИС Intel Arria10 SoC 10AS066K3F40E2SG в режиме автоматического синтеза с помощью САПР Quartus Prime SE 18.0 [6], представлены в табл. 1 для варианта реализации управляющего процессора в виде программного процессора NIOS-II. Аппаратные затраты определены в количестве адаптивных логических модулей, регистров, битов памяти и блоков DSP.

Таблица 1

**Аппаратные ресурсы для реализации ГРСП с NIOS-II**

Table 1

**Hardware resources for the HRSP implementation with NIOS-II**

Функциональные блоки ГРСП	Ресурсы ПЛИС			
	АЛМ*	Регистры	Память, бит	DSP
Управляющий уровень	3641	4048	33788672	4
Буферная память	15262	10223	141900	0
Распределитель	16666	4397	0	0
Вычислительные секции	36826	26788	5248	4
Интерфейс обмена данными	2042	1242	0	0
<b>Итого</b>	<b>74440</b>	<b>46698</b>	<b>33935820</b>	<b>8</b>

\*АЛМ – адаптивный логический модуль.

**Тестирование аппаратной модели.** Цели тестирования ГРСП на отладочной плате следующие:

- верификация аппаратной модели на соответствие имитационной модели путем проверки идентичности получаемых результатов;
- сравнение производительности ГРСП и современных процессоров цифровой обработки сигналов на типовых алгоритмах обработки данных.

Для верификации аппаратной модели выбрано типовое приложение процессора цифровой обработки сигналов – распознаватель изолированных слов. Прогон капсул, сгенерированных для всех алгоритмов распознавателя изолированных слов по исходным данным, извлеченным из тестовых произнесений 100 слов английского языка, в имитационной и аппаратной моделях показал полное совпадение всех промежуточных результатов. Суммарная точность распознавания (95 %), показанная макетным образцом, совпала с точностью распознавания, обеспечиваемой исходной программой C++-распознавателя изолированных слов на персональном компьютере.

В табл. 2 приведена оценка производительности ГРСП по сравнению с DSP TMSC55x фирмы Texas Instruments при выполнении синтетических тестовых алгоритмов [12, 13], являющихся типовыми для широкого класса задач цифровой обработки сигналов и аналогичных тестам из набора [14]. При этом быстродействие DSP TMSC55x принято за эталонное. Анализ данных показывает, что ГРСП по сравнению с DSP TMSC55x лучше выполняет алгоритмы фильтрации (за исключением адаптивного фильтра) и некоторые другие. Например, фильтры с  $K_c = 6$  и  $K_s = 3$  ГРСП считает в 1,14–1,50 раза быстрее, чем DSP TMSC55x. Однако на некоторых задачах ГРСП работает в 1,6–3,2 раза медленнее, чем DSP TMSC55x. Это связано с наличием в DSP TMSC55x аппаратной

Таблица 2

Показатели производительности ГРСЦ и DSP TMS55x при выполнении синтетических тестов алгоритмов

Table 2

HRSP performance compared to DSP TMS55x on synthetic tests

Алгоритм		DSP TMS55x	ГРСЦ
Фильтр с конечной импульсной характеристикой	для реальных данных	1	$1 + (1/(1 + K_c))$
	для одиночных отсчетов	1	1
	для комплексных данных	1	$1 + (3/(1 + K_c))$
Адаптивный фильтр наименьших среднеквадратичных значений		1	$1 - (K_c - 1)/(4 + 3K_c)$
Биквадратный фильтр с бесконечной импульсной характеристикой		1	$1 + (3 + 10K_s)/(4 + 21K_s)$
Сумма поразрядных произведений двух векторов		1	1
Поразрядная сумма двух векторов		1	1,50
Поиск максимума в векторе		1	0,60
Декодер Витерби		1	0,31
256-точечное БПФ		1	1,25

Примечание:  $K_c$  – число коэффициентов фильтра;  $K_s$  – число секций фильтра.

поддержки соответствующих алгоритмов, существенно ускоряющей их выполнение. Этот недостаток может быть устранен за счет расширения функциональности ГРСЦ, в частности более широкой аппаратной поддержки суперскалярных операций в вычислительных секциях; увеличения числа рабочих регистров в вычислителе; внедрения механизмов быстрой загрузки данных в регистры компьютерных блоков, минуя некоторые этапы конвейерной обработки.

**Расширение аппаратной поддержки типовых алгоритмов цифровой обработки сигналов.** Текущая версия реализует 256-точечный БПФ-алгоритм (Radix-2) в формате фиксированной точки с записью результатов на место входных данных и имеет следующие особенности:

- четыре входных 16-битных данных упаковываются в один операнд, мнимые и действительные части хранятся в разных разделах капсулы в битреверсивном порядке;
- поворотные коэффициенты хранятся в секционной памяти констант в полном объеме в каждой вычислительной секции;
- специальная инструкция *Butt*, введенная в систему команд, обеспечивает четырехцикловое вычисление Radix-2 «бабочки».

На рис. 2 показана схема выполнения инструкции *Butt* в установленном режиме. Эффективность такого вычисления БПФ на одной секции ГРСЦ сравнима с DSP TMS55x. Но ее можно повысить за счет аппаратной оптимизации.

Перечислим основные проблемы аппаратной поддержки БПФ в ГРСЦ:

- 1) избыточные накладные расходы хранения поворотных коэффициентов;
- 2) хранение отсчетов в капсуле в упакованных операндах;
- 3) нециклическая схема считывания действительных и мнимых частей отсчетов;
- 4) цикл № 2 инструкции *Butt* требует специального режима распаковки и рассылки упакованных данных;
- 5) изменение числа отсчетов БПФ требует переработки капсулы.

Анализ текущей версии средств аппаратной поддержки БПФ свидетельствует о большой степени аппаратной избыточности. Можно выделить два основных направления оптимизации архитектуры: оптимизация инструкции *Butt* и модификация буферной памяти.

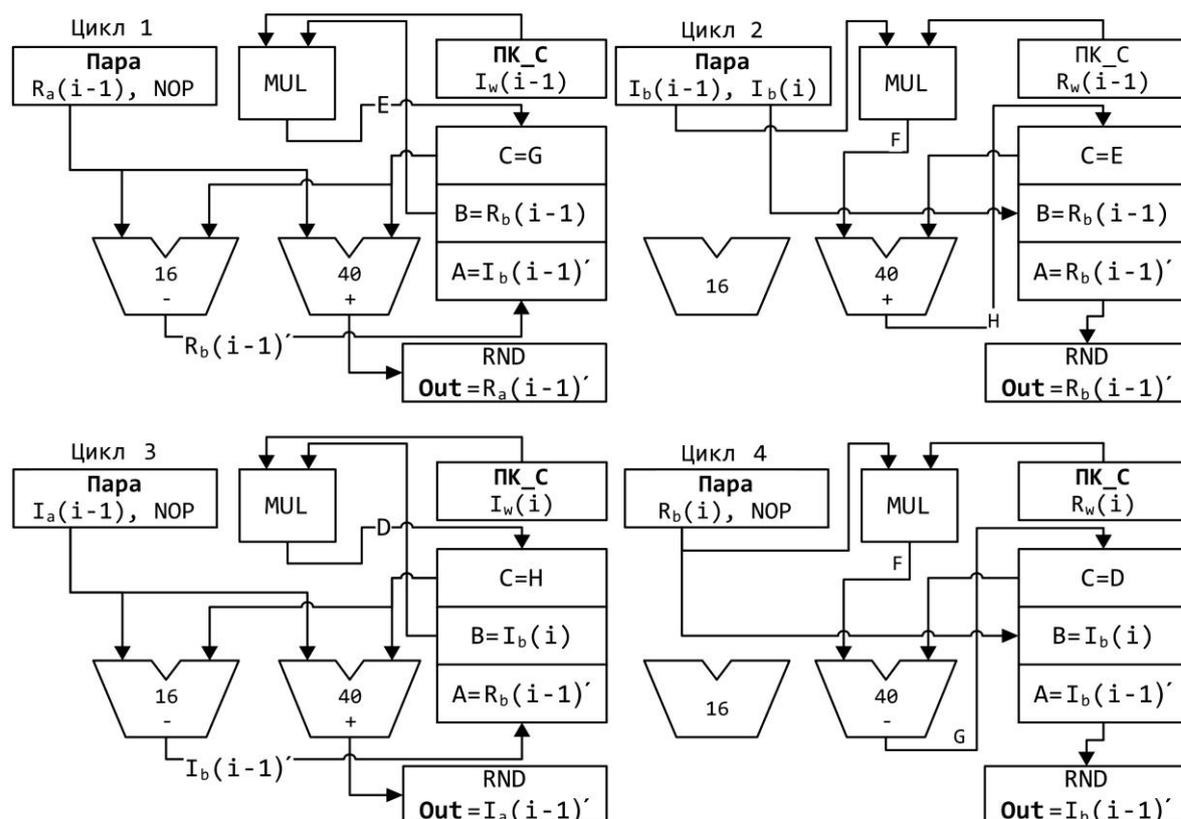


Рис. 2. Существующая схема инструкции Butt  
 Fig. 2. Current Butt instruction algorithmic diagram

Новая инструкция Butt, четырехцикловая реализация которой показана на рис. 3, решает проблемы 3 и 4 за счет добавления в секционный вычислитель двух регистров W2 и W3, а также регистровой памяти для четырех 16-битных констант. В состав буферной памяти введены автономные банки памяти БПФ для хранения действительных и мнимых значений отсчетов и коэффициентов БПФ и блок управления этой памятью, обеспечивающий генерацию адресов и чтение-запись ее содержимого. Особенности нового алгоритма заключаются в следующем:

- банки действительных и мнимых значений адресуются одним адресом;
- адреса самых первых «бабочек» не зависят от размера БПФ, поэтому их можно задавать в процессе инициализации аппаратуры;
- адреса компонент предыдущей «бабочки» хранятся для последующей записи по ним результата расчета «бабочки»;
- считанные значения отсчетов и коэффициентов записываются в упакованные операнды и передаются в рекуррентный обработчик сигналов стандартным путем.

В табл. 3 показаны результаты сравнения двух версий реализации БПФ – старой и новой (данные приведены для  $N = 256$  точек). Новая версия инвариантна к размерности БПФ, требует меньшего объема памяти и почти в 6 раз сокращает длину капсулы.

Таким образом, реализованная модификация архитектуры ГРСП требует существенно меньших аппаратных ресурсов при одинаковом числе поддерживаемых точек БПФ, более производительна и масштабируема при выполнении алгоритма БПФ.

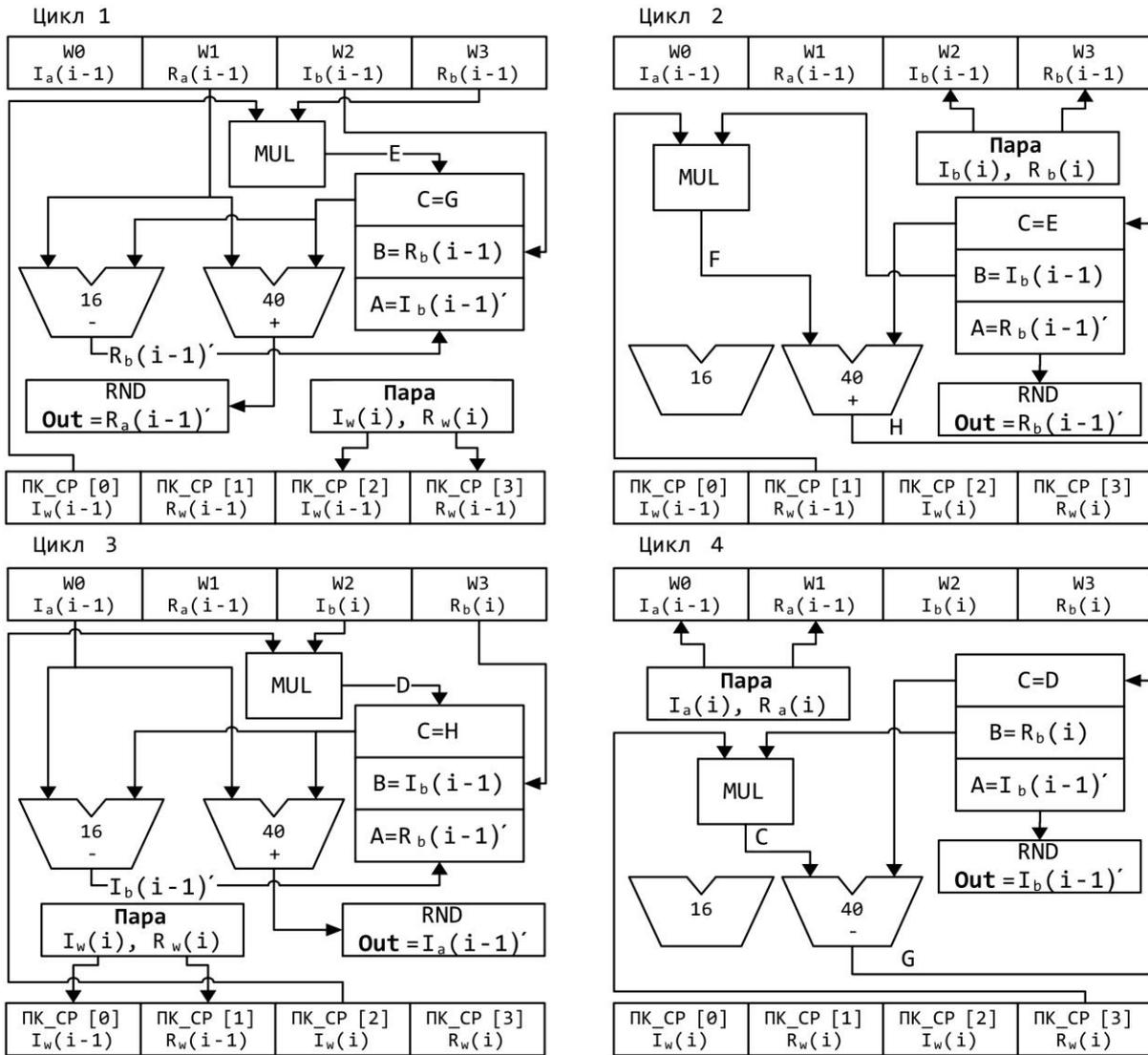


Рис. 3. Усовершенствованная схема инструкции ButT  
 Fig. 3. Advanced ButT instruction algorithmic diagram

Результаты реализации двух версий БПФ на ГРСЦ

Таблица 3

Results of the implementation of two FFT implementation cases on HRSP

Table 3

Параметр	Старая версия	Новая версия	
Поддерживаемые БПФ, число точек	256	от 8 до 256	от 8 до 1024
Память коэффициентов, число 16-битных слов	$256 \times 4 = 1024$	256	1024
Память отсчетов, число 16-битных слов	528	512	
Размер капсулы, число операндов	155	23	
Benchmark, число циклов	1024	1024	
Overhead, число циклов	44	16	
Итого, число циклов	1068	1040	

**Заключение.** Аппаратная VHDL-модель ГРСП обеспечила проверку корректности ее решений на макетном образце на базе отладочной платы HAN Pilot Platform, что подтвердило эффективность ее архитектуры для задач, допускающих распараллеливание вычислений. На синтетических тестах ГРСП продемонстрировал производительность в среднем на 5 % выше, чем DSP TMS320C55x. Тестирование аппаратной реализации ГРСП на типовом приложении процессора цифровой обработки сигнала – распознавателе изолированных слов – подтвердило ее битэкзактность имитационной модели ГРСП и исходной модели C++-распознавателя.

Полученные результаты подтверждают успешность оптимизации средств аппаратной поддержки БПФ в ГРСП. Новая версия средств не только характеризуется большей гибкостью и скоростью, но и позволяет сократить размер капсулы.

*Материалы статьи доложены на X Всероссийской научно-технической конференции с международным участием «Проблемы разработки перспективных микро- и наноэлектронных систем» (МЭС-2021) (1 марта – 1 ноября 2021 г., г. Москва, г. Зеленоград).*

### Литература

1. **Бурцев В. С.** Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ: сб. статей. М.: Торус Пресс, 2006. 414 с.
2. Вопросы применения и реализации потоковой модели вычислений / **А. В. Климов, Н. Н. Левченко, А. С. Окунев и др.** // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2016. № 2. С. 100–106.
3. **Дикарев Н. И., Шабанов Б. М., Шмелёв А. С.** Использование мелко гранулярного параллелизма в процессоре с архитектурой управления потоком данных // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2016. № 2. С. 144–150.
4. **Sundararajan S.** Minimizing communication and synchronization overhead in multiprocessors for digital signal processing: PhD dissertation. Berkeley, CA: University of California, 1995. 187 p.
5. **Степченков Ю. А., Дьяченко Ю. Г., Хилько Д. В., Петрухин В. С.** Рекуррентная потоковая архитектура: особенности и проблемы реализации // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2016. № 2. С. 120–127.
6. **Хилько Д. В., Степченков Ю. А., Шикунов Д. И., Шикунов Ю. И.** Рекуррентная потоковая архитектура: технические аспекты реализации и результаты моделирования // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2016. № 2. С. 128–135.
7. **Shikunov Yu., Stepchenkov Yu., Khilko D.** Recurrent mechanism developments in the data-flow computer architecture // 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus). Moscow; St. Petersburg: IEEE, 2018. P. 1413–1418. doi: <https://doi.org/10.1109/EIconRus.2018.8317362>
8. **Хилько Д. В., Степченков Ю. А., Шикунов Ю. И., Орлов Г. А.** Развитие средств капсульного программирования потоковой рекуррентной архитектуры // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2018. № 3. С. 2–9. doi: <https://doi.org/10.31114/2078-7707-2018-3-2-9>
9. Аппаратная верификация рекуррентного обработчика сигналов на ПЛИС / **Ю. А. Степченков, Н. В. Морозов, Ю. Г. Дьяченко и др.** // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2021. № 2. С. 77–82. doi: <https://doi.org/10.31114/2078-7707-2021-2-77-82>
10. HAN pilot platform. Specifications // terasIC [Электронный ресурс]. URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=216&No=1133&PartNo=2> (дата обращения: 28.03.2022).
11. Intel Quartus Prime software // Intel [Электронный ресурс]. URL: <https://www.intel.ru/content/www/ru/ru/software/programmable/quartus-prime/download.html> (дата обращения: 28.03.2022).
12. **Stepchenkov Yu. A., Khilko D. V., Shikunov Yu. I., Orlov G. A.** 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: IEEE, 2021. P. 2040–2044. doi: <https://doi.org/10.1109/EIconRus51938.2021.9396594>

13. TMS320C55x DSP CPU Reference Guide. Literature Number: SPRU371F. URL: [https://www.ti.com/lit/ug/spru371f/spru371f.pdf?ts=1654850280246&ref\\_url=https%253A%252F%252Fwww.google.ru%252F](https://www.ti.com/lit/ug/spru371f/spru371f.pdf?ts=1654850280246&ref_url=https%253A%252F%252Fwww.google.ru%252F).

14. The BDTImark2000™: a measure of DSP execution speed / Berkeley Design Technology, Inc. // Muhammad Shaaban's Home Page [Электронный ресурс]. URL: <http://meseec.ce.rit.edu/eecc722-fall2001/papers/dsp/4/bdtimark2000.pdf> (дата обращения: 28.03.2022).

Статья поступила в редакцию 04.02.2022 г.; одобрена после рецензирования 11.02.2022 г.; принята к публикации 04.05.2022 г.

### **Информация об авторах**

**Степченков Юрий Афанасьевич** – кандидат технических наук, ведущий научный сотрудник, заведующий отделом Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук (Россия, 119333, г. Москва, ул. Вавилова, 44, стр. 2), YStepchenkov@ipiran.ru

**Морозов Николай Викторович** – старший научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской академии наук (Россия, 119333, г. Москва, ул. Вавилова, 44, стр. 2), NMorozov@ipiran.ru

**Дьяченко Юрий Георгиевич** – кандидат технических наук, старший научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской академии наук (Россия, 119333, г. Москва, ул. Вавилова, 44, стр. 2), YDiachenko@ipiran.ru

**Хилько Дмитрий Владимирович** – старший научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской академии наук (Россия, 119333, г. Москва, ул. Вавилова, 44, стр. 2), DKhilko@ipiran.ru

**Степченков Дмитрий Юрьевич** – старший научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской академии наук (Россия, 119333, г. Москва, ул. Вавилова, 44, стр. 2), Stepchenkov@mail.ru

**Шикунов Юрий Игоревич** – инженер-исследователь Федерального исследовательского центра «Информатика и управление» Российской академии наук (Россия, 119333, г. Москва, ул. Вавилова, 44, стр. 2), yishikunov@yandex.ru

### **References**

1. Burtsev V. S. *Parallelism of computing processes and development of the supercomputer's architecture*, collected papers. Moscow, Torus Press Publ., 2006. 414 p. (In Russian).

2. Klimov A. V., Levchenko N. N., Okunev A. S., Stempkovsky A. L. The application and implementation issues of dataflow computing system. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES) = Problems of Advanced Micro- and Nanoelectronic Systems Development (MES)*, 2016, no. 2, pp. 100–106. (In Russian).

3. Dikarev N. I., Shabanov B. M., Shmelev A. S. The use of fine-grained parallelism in dataflow processor. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES) = Problems of Advanced Micro- and Nanoelectronic Systems Development (MES)*, 2016, no. 2, pp. 144–150. (In Russian).

4. Sundararajan S. *Minimizing communication and synchronization overhead in multiprocessors for digital signal processing*, PhD dissertation. Berkeley, CA, University of California, 1995. 187 p.

5. Stepchenkov Yu. A., Diachenko Yu. G., Khilko D. V., Petrukhin V. S. Recurrent data-flow architecture: features and realization problems. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES) = Problems of Advanced Micro- and Nanoelectronic Systems Development (MES)*, 2016, no. 2, pp. 120–127. (In Russian).

6. Khilko D. V., Stepchenkov Yu. A., Shikunov D. I., Shikunov Yu. I. Recurrent data-flow architecture: technical aspects of implementation and modeling results. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES) = Problems of Advanced Micro- and Nanoelectronic Systems Development (MES)*, 2016, no. 2, pp. 128–135. (In Russian).

7. Shikunov Yu., Stepchenkov Yu., Khilko D. Recurrent mechanism developments in the data-flow computer architecture. *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. Moscow, St. Petersburg, IEEE, 2018, pp. 1413–1418. doi: <https://doi.org/10.1109/EIConRus.2018.8317362>
8. Khilko D. V., Stepchenkov Yu. A., Shikunov Yu. I., Orlov G. A. Development of capsule programming tools for recurrent data-flow architecture. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES) = Problems of Advanced Micro- and Nanoelectronic Systems Development (MES)*, 2018, no. 3, pp. 2–9. (In Russian). doi: <https://doi.org/10.31114/2078-7707-2018-3-2-9>
9. Stepchenkov Yu. A., Morozov N. V., Diachenko Yu. G., Khilko D. V., Stepchenkov D. Yu., Shikunov Yu. I. Hardware verification of the recurrent signal processor on FPGA. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES) = Problems of Advanced Micro- and Nanoelectronic Systems Development (MES)*, 2021, no. 2, pp. 77–82. (In Russian). doi: <https://doi.org/10.31114/2078-7707-2021-2-77-82>
10. HAN pilot platform, specifications. *terasIC*. Available at: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=216&No=1133&PartNo=2> (accessed: 28.03.2022).
11. Intel Quartus Prime software. *Intel*. Available at: <https://www.intel.ru/content/www/ru/ru/software/programmable/quartus-prime/download.html> (accessed: 28.03.2022).
12. Stepchenkov Yu. A., Khilko D. V., Shikunov Yu. I., Orlov G. A. 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, IEEE, 2021, pp. 2040–2044. doi: <https://doi.org/10.1109/EIConRus51938.2021.9396594>
13. TMS320C55x DSP CPU Reference Guide. Literature Number: SPRU371F. URL: [https://www.ti.com/lit/ug/spru371f/spru371f.pdf?ts=1654850280246&ref\\_url=https%253A%252F%252Fwww.google.ru%252F](https://www.ti.com/lit/ug/spru371f/spru371f.pdf?ts=1654850280246&ref_url=https%253A%252F%252Fwww.google.ru%252F)
14. Berkeley Design Technology, Inc. The BDTImark2000™: a measure of DSP execution speed. *Muhammad Shaaban's Home Page*. Available at: <http://meseec.ce.rit.edu/eccc722-fall2001/papers/dsp/4/bdtimark2000.pdf> (accessed: 28.03.2022).

The article was submitted 04.02.2022; approved after reviewing 11.02.2022;  
accepted for publication 04.05.2022.

#### **Information about the authors**

**Yuri A. Stepchenkov** – Cand. Sci. (Eng.), Leading Researcher, Head of Department, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (Russia, 119333, Moscow, Vavilov st., 44, bld. 2), [YStepchenkov@ipiran.ru](mailto:YStepchenkov@ipiran.ru)

**Nikolay V. Morozov** – Senior Scientific Researcher, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (Russia, 119333, Moscow, Vavilov st., 44, bld. 2), [NMorozov@ipiran.ru](mailto:NMorozov@ipiran.ru)

**Yuri G. Diachenko** – Cand. Sci. (Eng.), Senior Scientific Researcher, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (Russia, 119333, Moscow, Vavilov st., 44, bld. 2), [YDiachenko@ipiran.ru](mailto:YDiachenko@ipiran.ru)

**Dmitri V. Khilko** – Senior Scientific Researcher, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (Russia, 119333, Moscow, Vavilov st., 44, bld. 2), [DKhilko@ipiran.ru](mailto:DKhilko@ipiran.ru)

**Dmitri Yu. Stepchenkov** – Senior Scientific Researcher, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (Russia, 119333, Moscow, Vavilov st., 44, bld. 2), [Stepchenkov@mail.ru](mailto:Stepchenkov@mail.ru)

**Yuri I. Shikunov** – Research Engineer, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (Russia, 119333, Moscow, Vavilov st., 44, bld. 2), [yishikunov@yandex.ru](mailto:yishikunov@yandex.ru)