УДК 004.3'1

В.М. Амербаев, Р.А. Соловьев, Д.В. Тельпухов, П.С. Поперечный, В.С. Рухлов, А.Н. Щелоков, А.С. Михмель

РАЗРАБОТКА УСТРОЙСТВА ДЛЯ ВЫЧИСЛЕНИЯ РЕЗУЛЬТАТА ОПЕРАЦИИ СКАЛЯРНОГО ПРОИЗВЕДЕНИЯ ВЕКТОРОВ НА БАЗЕ ИНТРАМОДУЛЯРНОГО РАЗЛОЖЕНИЯ КОМПЛЕКСНЫХ ЧИСЕЛ В МОДУЛЯРНОЙ АРИФМЕТИКЕ

Одной из ключевых операций цифровой обработки сигналов (ЦОС) является операция скалярного произведения векторов (СПВ), используемая при построении сверток и КИХ-фильтров. В позиционном варианте эта операция хорошо изучена и для неё разработаны эффективные реализации микроэлектронных устройств. Однако при больших размерностях элементов вектора, производительность позиционных устройств существенно снижается. В данной статье предлагается использовать систему остаточных классов (СОК) для выполнения этой операции. СОК обладает внутренним параллелизмом, который позволяет избежать существенного роста временных затрат на выполнение операций при увеличении размерности элементов векторов. В отличие от традиционного распараллеливания по модулярным каналам СОК, в данной статье предлагается использовать ещё один уровень параллелизма, так называемый интрамодулярный параллелизм комплексных чисел на базе теоремы Гаусса об изоморфизме. В статье рассмотрен метод реализации скалярного умножителя векторов комплексных чисел с помощью аппарата модулярной арифметики над полем Галуа. Предложен подход, связанный с использованием интрамодулярного разложения модулярных каналов для комплексных чисел на базе теоремы Гаусса об изоморфизме. Реализовано устройство, выполняющее операцию скалярного произведения предложенным методом. Приведено подробное описание устройства, а также произведено сравнение с аналогичными устройствами, построенными в двоичном базисе с помощью современных САПР на заказных СБИС и ПЛИС.

Целое комплексное число; система остаточных классов; скалярное произведение векторов; конечное поле; свертка.

V.M. Amerbaev, R.A. Solovyev, D.V. Telpukhov, P.S. Poperechny, V.S. Rukhlov, A.N. Schelokov, A.S. Mihmel

DEVELOPMENT OF A MICROELECTRONIC DEVICE FOR DOT PRODUCT CALCULATION BASED ON RNS INTRAMODULAR DECOMPOSITION OF COMPLEX NUMBERS

One of the key operations in digital signal processing (DSP) is the dot product operation used in the construction of convolutions and FIR filters. In the positional representation, this operation has been well studied and many effective implementations of microelectronic devices have been developed. However, for large dimensions of vector elements performance of positional devices significantly decreases. In this paper we propose to use residue number system (RNS) to perform this operation. RNS has internal parallelism that helps to avoid significant delay growth when dimensions of vector elements increase. As opposed to traditional parallelization for residue channels of RNS, we use another level of parallelism, the so-called complex numbers intramodular parallelism based on the Gauss's theorem on isomorphism. The paper describes the method of implementing of dot product calculation for vectors of complex integers using RNS arithmetic over a Galois field. We present an approach related to the use of modular decomposition in intramodular channels for complex numbers based on the Gauss's theorem on isomorphism. A device calculating dot product by the proposed method was implemented. Detailed description of the device is presented, as well as the results of its comparison to similar devices built in binary basis using modern ASIC and FPGA CADs.

Complex integer; residue number system; dot product (scalar product); finite field; convolution.

Модулярная арифметика и теорема Гаусса об изоморфизме. Модулярная арифметика определяется набором N целых попарно взаимно простых чисел $\{p_1, p_2, \dots, p_N\}$, которые принято называть модулями. Произведение этих чисел $M = p_1 \cdot p_2 \cdot \dots \cdot p_N$ определяет динамический диапазон системы. Это означает, что любое число в пределах диапазона $[0; M)$ имеет уникальное представление в модулярной арифметике, т.е. любое произвольное целое число X меньше, чем M может быть представлено в модулярной арифметике набором $\{x_1, x_2, x_3, \dots, x_N\}$, где $x_i = |X|_{p_i}$ [1–2].

При условии, что результат выполнения операции не выходит за выбранный диапазон M, арифметические операции сложения, вычитания и умножения выполняются над вычетами параллельно, т.е. независимо по каждому модульному основанию [3]:

$$|X + Y|_M = \{|x_1 + y_1|_{p_1}, |x_2 + y_2|_{p_2}, \dots, |x_N + y_N|_{p_N}.$$

Введем понятие вычета для комплексных чисел. Пусть w – фиксированное целое комплексное число $w = a + bj \in \mathbb{CZ}$ с нормой $p = a^2 + b^2$. Пусть $z = x + yj$ – произвольное целое комплексное число. Тогда для вычета числа z по комплексному модулю w вычет можно определить следующим образом [4]:

$$\langle z \rangle_w = \frac{|z \cdot \overline{w}|_p \cdot w}{p}.$$

Известно, что для вычета комплексной переменной так же, как и для вычетов над кольцом целых чисел выполняются следующие свойства:

$$\langle z_1 \pm z_2 \rangle_w = \langle \langle z_1 \rangle_w \pm \langle z_2 \rangle_w \rangle_w,$$
$$\langle z_1 \cdot z_2 \rangle_w = \langle \langle z_1 \rangle_w \cdot \langle z_2 \rangle_w \rangle_w.$$

Для распараллеливания затратной операции умножения комплексных чисел используется теорема Гаусса об изоморфизме, которая позволяет перейти от операций над целыми комплексными числами к операциям над изоморфными образами (вычетами) этих чисел в кольце неотрицательных целых чисел. Иными словами, данная теорема помогает перейти от арифметики в комплексной плоскости к арифметике действительных чисел.

Теорема Гаусса (об изоморфизме). Если целое комплексное число $w = A + jB$ удовлетворяет условию $(A, B) = 1$, то кольцо вычетов $\mathbb{CZ}_w$ кольца целых комплексных чисел $\mathbb{CZ}$ по комплексному модулю w изоморфно кольцу вычетов $\mathbb{Z}_p$ целых действительных чисел $\mathbb{Z}$ по модулю p, где $p = A^2 + B^2$.

Поскольку $(A, B) = 1$, то в силу китайской теоремы об остатках согласно теореме Гаусса каждый элемент $z = x + jy$ из G_p однозначно кодируется парой $(\langle z|_w, \langle z|_{\bar{w}})$ или (v, v'), где

$$v = |x + (Bm_0 - An_0)y|_p,\ v' = |x - (Bm_0 - An_0)y|_p,$$

m_0 и n_0 – некоторое решение уравнения $\mathrm{Am}_0 + \mathrm{Bn}_0 = 1$, которое всегда существует в силу $(\mathrm{A}, \mathrm{B}) = 1$.

При этом $\forall z_1, z_2 \in G_p$:

$$\langle z_1 \pm z_2|_p \leftrightarrow \left(|v_1 \pm v_2|_p, |v'_1 \pm v'_2|_p\right),$$
$$\langle z_1 \cdot z_2|_p \leftrightarrow \left(|v_1 \cdot v_2|_p, |v'_1 \cdot v'_2|_p\right).$$

Зачем нужно это дополнительное преобразование? Очевидно, что для выполнения умножения двух комплексных чисел необходимо применить четыре умножения действительных чисел:

$$z_1 \cdot z_2 = (x_1 + y_1 j) \cdot (x_2 + y_2 j) = x_1 \cdot x_2 - y_1 \cdot y_2 + j \cdot (x_1 \cdot y_2 + y_1 \cdot x_2). \quad (1)$$

Применяя описанные выше преобразования, будем каждый элемент $z = x + yj$ кодировать парой (v, v'), т.е. (см. рис. 1):

$$\langle z_1 \pm z_2 \mid_p \leftrightarrow (\mid v_1 \pm v_2 \mid_p, \mid v'_1 \pm v'_2 \mid_p),$$
$$\langle z_1 \cdot z_2 \mid_p \leftrightarrow (\mid v_1 \cdot v_2 \mid_p, \mid v'_1 \cdot v'_2 \mid_p).$$

Очевидно, что после преобразования для умножения двух комплексных чисел потребуется лишь два умножения. Сложность сложения остается неизменной.

Рис. 1. Схема перевода комплексных чисел в Гауссово представление по нескольким модулям

После вычислений для получения искомого значения требуется обратное преобразование (рис. 2) из вида (v, v') в $z = x + jy$, которое может быть выполнено по формулам: $x = \left|\frac{v+v'}{2}\right|_p$ $y = \left|\frac{v-v'}{2q^+}\right|_p$, где $q^+ = |Bm_0 - An_0|_p$.

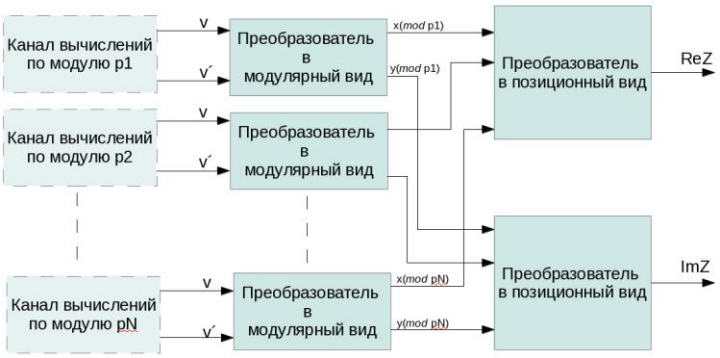

Рис. 2. Схема обратного преобразования из представления по Гауссу в позиционный вид

Проектирования устройства, выполняющего операцию скалярного произведения комплексных векторов в модулярном и позиционном базисах. Для скалярного произведения векторов формула умножения (1) может быть записана в следующем виде:

$$a_1 \cdot b_1 + a_2 \cdot b_2 + \cdots + a_n \cdot b_n = (x_{a1} + jy_{a1}) \cdot (x_{b1} + jy_{b1}) +$$
$$+(x_{a2} + jy_{a2}) \cdot (x_{b2} + jy_{b2}) + \cdots + (x_{an} + jy_{an}) \cdot (x_{bn} + jy_{bn}) =$$
$$= \sum_{i=1}^{n}(x_{ai} \cdot x_{bi}) - \sum_{i=1}^{n}(y_{ai} \cdot y_{bi}) + j(\sum_{i=1}^{n}(x_{ai} \cdot y_{bi}) + \sum_{i=1}^{n}(y_{ai} \cdot x_{bi})). \qquad (2)$$

Формула (2) может быть представлена в виде конвейерной вычислительной схемы, состоящей из 4 подобных участков с сумматором для мнимой части и вычитателем для действительной части на конце тракта вычислений. Подробнее структура схемы приведена на рис. 3.

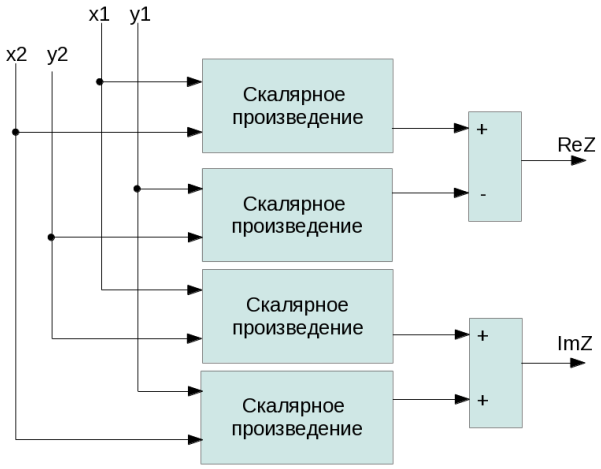

Рис. 3. Схема выполнения операции скалярного умножения комплексных чисел в позиционном варианте

Внутренняя часть вычислений имеет канонический вид, приведенный на рис. 4.

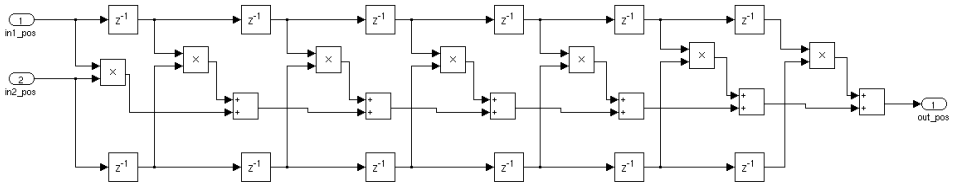

Рис. 4. Канонический вид скалярного произведения

Таким образом, для ускорения моделирования, для расчета задержки и площади достаточно промоделировать только один из четырех критических участков схемы. Задержка на нем будет определять максимальную тактовую частоту устройства, а для расчета площади достаточно будет умножить полученное значение на 4.

При переходе к модулярной арифметике для того чтобы избежать затратной операции округления обычно выбирают динамический диапазон таким образом, чтобы он был больше чем максимально возможный результат вычислений [5–6].

Зададим разрядность входных данных в позиционном виде, $z_1 = x_1 + y_1 j$, $z_2 = x_2 + y_2 j$, равном 12 бит: $x_1, y_1, x_2, y_2 < 2^{12}$. Тогда результат выполнения скалярного произведения в 256 отчетов не будет превышать 32 бит (рассчитывается как 12+12+8).

Один из возможных наборов набор модулей [7], который покрывает выбранный диапазон и удовлетворяет условию НОД(a, b)=1, приведен в табл. 1.

Таблица 1

Набор простых модулей

p	w=a+j*b	
	a	b
61	6	5
73	8	3
89	8	5
97	9	4
113	8	7

Таким ообразом

$P = p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5 = 61 \cdot 73 \cdot 89 \cdot 97 \cdot 113 = 4344030637 > 2^{32}$

Схема арифметических операций над комплексными числами с применением гауссовой арифметики приведена на рис. 5, количество выбранных модулей равно пяти. Каждый из модулей имеет разрядность 7 бит, в сравнении с 12 битными каналами, реализующими СПВ в позиционной арифметике.

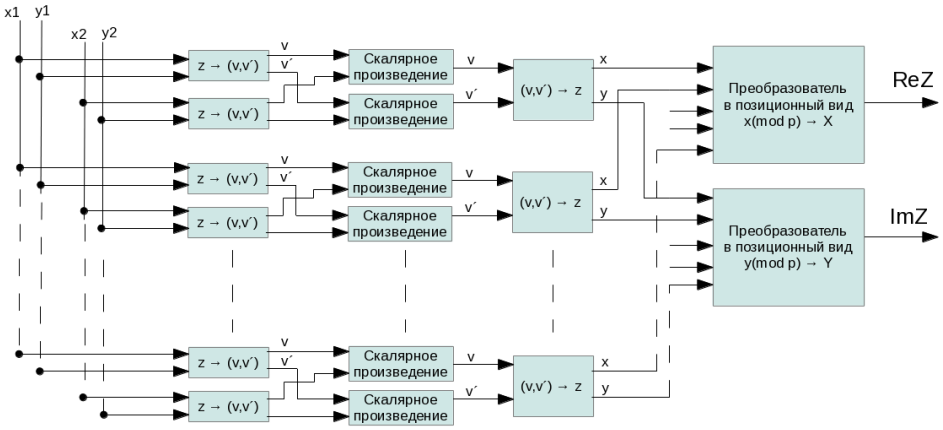

Рис. 5. Общая структурная схема вычисления по пяти модулям

Первый каскад схемы, приведенный на рис. 6, переводит поток комплексных чисел $z_1 = x_1 + y_1 j$, $z_2 = x_2 + y_2 j$, подаваемый на вход схемы, в код Гаусса $(v_1,\ v'_1)$ и $(v_2,\ v'_2)$ соответственно.

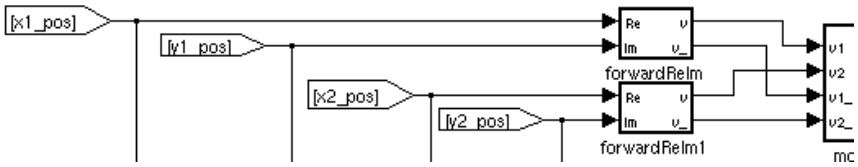

Рис. 6. Блок forwardReIm переводит двоичное комплексное число $z_1 = x_1 + y_1 j$ в код Гаусса $(v_1,\ v'_1)$

Во втором каскаде (рис. 7), выполняются заданные операции, в нашем примере скалярное произведение. Структура устройства аналогична скалярному умножителю в позиционном базисе, за исключением сумматоров и умножителей, которые в данном случае являются операциями по модулю [8]. Операции сложения и умножения выполняются независимо по каждому модулю p, в нашем случае по одному из пяти, а значения на выходе находятся в диапазоне $[0, p)$.

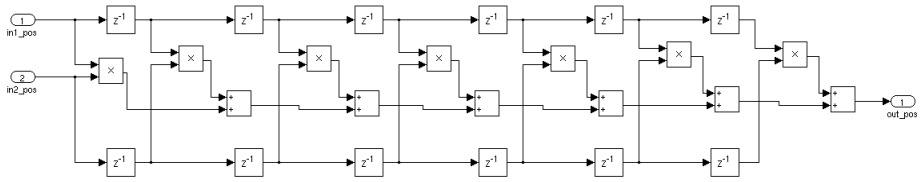

Рис. 7. Схема КИХ-фильтра (частный вид скалярного произведения)

В третьем и четвертом каскадах выполняются преобразования из гауссового кода в комплексные числа и далее в каскаде дельта-преобразования в комплексное число диапазона D [9]. Здесь, как и в позиционном варианте для расчета задержки можно считать только один модулярный канал, с наиболее «сложным» модулем.

Оптимизация операции модулярного умножения с накоплением. В процессе разработки данной схемы выяснилось, что основной вклад в задержку всей схемы вносит умножитель по модулю и многовходовый сумматор по модулю. Прямые и обратные преобразователи имеют конвейерную структуру и не являются критическими участками схемы [10, 11].

В качестве умножителя целых чисел по модулю был выбран высокопроизводительный индексный умножитель [12–14], RTL-описание которого приведено в [15]. Он крайне эффективен для модулей малой разрядности (до 8 бит) за счет использования таблиц и замены умножения по модулю на операцию сложения по модулю.

Необходимо отметить, что при каскадном построении схем задержка увеличивается с возрастанием числа каскадов, в нашем случае общая задержка вычислительного модуля складывается из задержки умножителя и задержки сумматора множенная на количество каскадов:

$$\tau = t_{\otimes} + n \cdot t_{\oplus}.$$

Задержка последовательно включенных модулярных сумматоров может быть существенна, поэтому была применена другая схема построения КИХ-фильтра, а именно с применением принципа пирамидального суммирования (рис. 8).

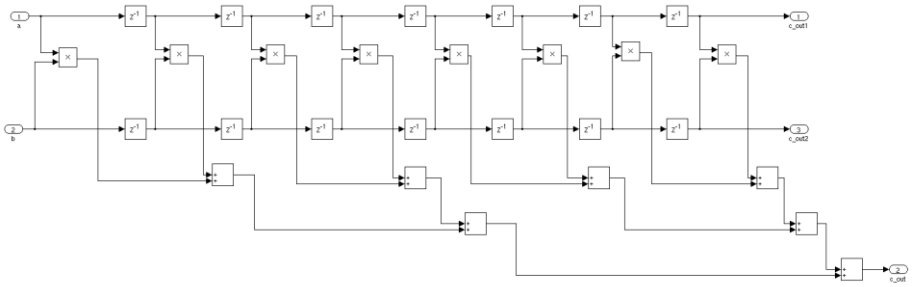

Рис. 8. Схема скалярного произведения с пирамидальным суммированием, 6 каскадов

При таком построении общая задержка существенно снижается:

$$\tau = t_{\otimes} + t_{\oplus} \cdot \log_2 n.$$

Существует также подход, в котором суммирование происходит на многовходовом позиционном сумматоре и модуль берется один раз на выходе позиционного сумматора [16]. Как показали эксперименты, этот подход позволяет ещё больше сократить задержку даже по сравнению с пирамидальным сумматором, а также уменьшить площадь вычислительного каскада. Это связано с тем, что существующие САПР как для разработки устройств на базе ПЛИС, так и для проектирования заказных СБИС, оптимизированы для проектирования традиционных двоичных структур, в то время как модулярные структуры требуют большего внимания со стороны разработчика [17].

Экспериментальные результаты. При проектировании данного устройства был использован пакет MatLab. Разработанные блоки были переведены в RTL-описание при помощи возможностей этого пакета. Создана библиотека стандартных операций над числами для модулярной арифметики (умножитель, многовходовый сумматор), а также немодульных операций (преобразователь в код Гаусса, обратный преобразователь, дельта-преобразователь), по произвольно-задаваемым модулям [18–19].

Проведены сравнения по занимаемой площади (количество логических вентилей), а также по задержкам на критических путях. Ниже представлена таблица полученных характеристик для блоков арифметических операций (в данном случае скалярного произведения), как в модулярном, так и в двоичном виде. В сравнение не попали блоки прямого преобразования в гауссовый код и блоки обратного преобразования в позиционный. Созданные таким образом Verilog описания блоков были синтезированы в пакете Atera-Quartus под заданное семейство ПЛИС – MAXV (как наиболее объективная оценка, так как в данном семействе отсутствуют аппаратные умножители), а также для СБИС технологии 45 нм.

Таблица 2

Площадь всей системы без прямых/обратных преобразователей в ПЛИС (кол-во ячеек)

Кол-во каскадов		8	16	32	64
Сумматор пирамидальный	10	23000	44200	85300	167380
Сумматор с корр. на выходе	10	22370	38050	67820	99850
Сумматор последовательный	10	24150	45760	89830	168910
Позиционный КИХ, 32 бита	4	46800	77800	143120	275480

Таблица 3

Задержка всей системы без прямых/обратных преобразователей в ПЛИС (нс)

Кол-во каскадов	8	16	32	64
Сумматор пирамидальный	9,2	12,6	15,0	17,1
Сумматор с корр. на выходе	9,1	12,8	14,2	16,9
Сумматор последовательный	18,0	37,1	68,0	125,0
Позиционный КИХ, 32 бита	8,6	11,2	12,4	14,5

Созданная библиотека Simulink доступна по ссылке [20]. Данная библиотека проста в использовании, все параметры задаются с помощью интуитивно-понятного интерфейса пользователя. Удобство заключается не только в том, что можно создать сколь угодно сложную систему простым добавлением и соединением блоков в схему, но и генерацией из составленной схемы RTL-описания, подходящее для синтеза.

Таблица 4

Площадь всей системы без прямых/обратных преобразователей для СБИС (мкм^2)

Кол-во каскадов		8	16	32	64
Сумматор пирамидальный	10	146590	282250	544730	1030040
Сумматор с корр. на выходе	10	143040	254780	472120	724580
Сумматор последовательный	10	143190	276750	543380	1060530
Позиционный КИХ, 32 бита	4	256940	497352	975080	1927652

Таблица 5

Задержка всей системы без прямых/обратных преобразователей в СБИС (нс)

Кол-во каскадов	8	16	32	64
Сумматор пирамидальный	2,5	3,0	3,7	4,4
Сумматор с корр. на выходе	2,6	2,9	3,4	4,1
Сумматор последовательный	4,2	7,6	14,6	29,2
Позиционный КИХ, 32 бита	6,4	6,8	7,2	8,1

Заключение. В данной статье предложен путь построения ключевых систем для обработки большого объема арифметических данных в задачах цифровой обработки сигналов над комплексными числами при помощи модулярной арифметики. Для упрощения операций над комплексными числами применена теорема Гаусса, что позволило перейти от операций над комплексными числами к параллельным операциям над действительными числами. Показано, что при некоторых условиях, например, большой размерности данных или большой длине каскада, реализация в рамках модулярной арифметики может давать преимущества в сравнении с двоичной реализацией.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Abdelgawad A., Bayoumi M.* High Speed and Area-Efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications // IEEE International Symposium on Circuits and Systems. – 2007. – P. 3199-3202.
2. *Preethy A.P. and Radhakrishnan D.* A 36-bit Balanced Moduli MAC Architecture // 42nd Midwest Symp. on Circuits and Systems (MWSCAS99), Las Cruces, NM. – Aug. 1999. – Vol. 1. – P. 380-383.
3. *Амербаев В.М., Соловьев Р.А., Тельпухов Д.В., Щелоков А.Н.* Исследование эффективности модулярных вычислительных структур при проектировании аппаратных однотактных умножителей // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 248-254.
4. *Амербаев В.М., Стемпковский А.Л., Соловьев Р.А.* Параллельные вычисления в кольце гауссовых чисел над полем Галуа GF(P) // Всероссийская научно-техническая конференция "Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)": сборник трудов. – 2012. – № 1. – С. 517-520.
5. *Dugdale M.* VLSI implementation of residue adders based on binary adders // IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 1992.
6. *Amerbaev V.M., Solovyev R.A., Telpukhov D.V.* Hardware Implementation of Fir Filter based on Number-Theoretic Fast Fourier Transform in Residue Number System // Open Sciences Journal. – 2014. – P. 1-6.
7. *Амербаев В.М., Тельпухов Д.В., Константинов А.В.* Бивалентный дефект модулярных кодов. Выбор технологических модулей, понижающих бивалентный дефект // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). – 2008. – С. 462.
8. *Амербаев В.М., Соловьев Р.А., Тельпухов Д.В.* Реализация библиотеки модульных арифметических операций на основе алгоритмов минимизации логических функций // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 221-225.

9. *Амербаев В.М., Соловьев Р.А., Тельпухов Д.В.* Метод вычисления циклической свертки на базе БПФ с использованием чисел Прота // Информационные технологии. – 2014. – № 10. – С. 22-27.
10. *Соловьев Р.А., Тельпухов Д.В.* Аппаратная реализация операции нахождения остатка целочисленного деления для входных данных большой разрядности в модулярной арифметике // Известия высших учебных заведений. Электроника. – 2013. – № 4. – С. 75-83.
11. *Соловьев Р.А., Тельпухов Д. В., Амербаев В.М., Балака Е.С.* Построение обратных преобразователей модулярной арифметики с коррекцией ошибок на базе полиадического кода // Нейрокомпьютеры: разработка, применение. – 2014. – № 9. – С. 30-35.
12. *Амербаев В.М., Пак И.Т.* Параллельные вычисления в комплексной плоскости. – Алма-Ата.: Изд-во «Наука», 1984. – 183 с.
13. *Виноградов И.М.* Основы теории чисел. – М.-Л.: Гостехиздат, 1952. – 180 с.
14. *Omondi A. and Premkumar B.* Residue Number System: Theory and Implementation // Imperial College Press. – 2007. ISBN 978-1-86094-866-4.
15. *Соловьев Р.А.* Генератор Verilog для индексных умножителей по модулю: vscripts. 2012. URL: http://vscripts.ru/2012/index-modulo-multiplication.php (дата обращения: 16.02.2015).
16. *Piestrak S.J.* Design of residue generators and multioperand modular adders using carry-save adders // IEEE Trans. Comput. – 1994. – Vol. 423, no. 1. – P. 68-77.
17. *Балака Е.С., Тельпухов Д.В.* Принципы построения специализированного вычислителя для задач матричной алгебры с применением параллельной арифметики // Нейрокомпьютеры: разработка, применение. – 2010. – № 9. – С. 46-49.
18. *Амербаев В.М., Тельпухов Д.В.* Обратный преобразователь модулярной арифметики с использованием неточного ранга для задач ЦОС // Известия высших учебных заведений. Электроника. – 2013. – № 1 (99). – С. 41-46.
19. *Амербаев В.М., Балака Е.С., Константинов А.В., Тельпухов Д.В.* Методы построения прямых преобразователей модулярной логарифметики, ориентированных на ЦОС // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). – 2010. – С. 374.
20. *Поперечный П.С.* Matlab Residue Library: vscripts. 2014. URL: http://vscripts.ru/w/Matlab_residue_library (дата обращения: 16.02.2015).

REFERENCES

1. *Abdelgawad A., Bayoumi M.* High Speed and Area-Efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications, *IEEE International Symposium on Circuits and Systems*, 2007, pp. 3199-3202.
2. *Preethy A.P. and Radhakrishnan D.* A 36-bit Balanced Moduli MAC Architecture, *42nd Midwest Symp. on Circuits and Systems (MWSCAS99), Las Cruces, NM*., Aug. 1999, Vol. 1, pp. 380-383.
3. *Amerbaev V.M., Solov'ev R.A., Tel'pukhov D.V., Shchelokov A.N.* Issledovanie effektivnosti modulyarnykh vychislitel'nykh struktur pri proektirovanii apparatnykh odnotaktnykh umnozhiteley [A survey on efficiency of modular computing structures for single-cycle hardware multiplier design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 248-254.
4. *Amerbaev V.M., Stempkovskiy A.L., Solov'ev R.A.* Parallel'nye vychisleniya v kol'tse gaussovykh chisel nad polem Galua GF(P) [Parallel computations in the ring of Gaussian integers over Galois field GF(P)], *Vserossiyskaya nauchno-tekhnicheskaya konfe-rentsiya "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES)". Sbornik trudov* [All-Russian scientific-technical conference "problems of development of perspective micro- and nanoelectronic systems (MES)". Proceedings of], 2012, No. 1, pp. 517-520.
5. *Dugdale M.* VLSI implementation of residue adders based on binary adders, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 1992.*
6. *Amerbaev V.M., Solovyev R.A., Telpukhov D.V.* Hardware Implementation of Fir Filter based on Number-Theoretic Fast Fourier Transform in Residue Number System, *Open Sciences Journal*, 2014, pp. 1-6.
7. *Amerbaev V.M., Tel'pukhov D.V., Konstantinov A.V.* Bivalentnyy defekt modulyarnykh kodov. Vybor tekhnologicheskikh moduley, ponizhayushchikh bivalentnyy defekt [Bivalent defect modular codes. The choice of process modules, lowering bivalent defect], *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES)* [Problems of development of perspective micro- and nanoelectronic systems (MES)], 2008, pp. 462.

8. *Amerbaev V.M., Solov'ev R.A., Tel'pukhov D.V.* Realizatsiya biblioteki modul'nykh arifmeticheskikh operatsiy na osnove algoritmov minimizatsii logicheskikh funktsiy [Library implementation of modular arithmetic operations, based on logic functions minimization algorithms], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 221-225.
9. *Amerbaev V.M., Solov'ev R.A., Tel'pukhov D.V.* Metod vychisleniya tsiklicheskoy svertki na baze BPF s ispol'zovaniem chisel Prota [Method for computing the cyclic convolution based on FFT using Proth numbers], *Informatsionnye tekhnologii* [Information technologies], 2014, No. 10, pp. 22-27.
10. *Solov'ev R.A., Tel'pukhov D.V.* Apparatnaya realizatsiya operatsii nakhozhdeniya ostatka tselochislennogo deleniya dlya vkhodnykh dannykh bol'shoy razryadnosti v modulyarnoy arifmetike [Hardware implementation of the operation of finding the remainder of integer division to input data high-capacity in modular arithmetic], *Izvestiya vysshikh uchebnykh zavedeniy. Elektronika* [Izvestiya of Higher Educational Institutions. Electronics], 2013, No. 4, pp. 75-83.
11. *Solov'ev R.A., Tel'pukhov D. V., Amerbaev V.M., Balaka E.S.* Postroenie obratnykh preobrazovateley modulyarnoy arifmetiki s korrektsiey oshibok na baze poliadicheskogo koda [Build inverters modular arithmetic error correction on the basis politicheskogo code], *Neyrokomp'yutery: razrabotka, primenenie* [Neurocomputers: Development, Application], 2014, No. 9, pp. 30-35.
12. *Amerbaev V.M., Pak I.T.* Parallel'nye vychisleniya v kompleksnoy ploskosti [Parallel computations in the complex plane]. Alma-Ata.: Izd-vo «Nauka», 1984, 183 p.
13. *Vinogradov I.M.* Osnovy teorii chisel [Fundamentals of the theory of numbers]. Moscow-Leningrad: Gostekhizdat, 1952, 180 p.
14. *Omondi A. and Premkumar B.* Residue Number System: Theory and Implementation, *Imperial College Press*. 2007. ISBN 978-1-86094-866-4.
15. *Solov'ev R.A.* Generator Verilog dlya indeksnykh umnozhiteley po modulyu [Generator Verilog for index modulo multipliers]: vscripts. 2012. Available at: http://vscripts.ru/2012/index-modulo-multiplication.php (accessed 16 February 2015).
16. *Piestrak S.J.* Design of residue generators and multioperand modular adders using carry-save adders, *IEEE Trans. Comput*., 1994, Vol. 423, No. 1, pp. 68-77.
17. *Balaka E.S., Tel'pukhov D.V.* Printsipy postroeniya spetsializirovannogo vychislitelya dlya zadach matrichnoy algebry s primeneniem parallel'noy arifmetiki [Principles of construction of a specialized computer tasks for matrix algebra with application of parallel arithmetic], *Neyrokomp'yutery: razrabotka, primenenie* [Neurocomputers: Development, Application], 2010, No. 9, pp. 46-49.
18. *Amerbaev V.M., Tel'pukhov D.V.* Obratnyy preobrazovatel' modulyarnoy arifmetiki s ispol'zovaniem netochnogo ranga dlya zadach TsOS [Return Converter modular arithmetic using inaccurate rank for DSP tasks], *Izvestiya vysshikh uchebnykh zavedeniy. Elektronika* [Izvestiya of Higher Educational Institutions. Electronics], 2013, No. 1 (99), pp. 41-46.
19. *Amerbaev V.M., Balaka E.S., Konstantinov A.V., Tel'pukhov D.V.* Metody postroeniya pryamykh preobrazovateley modulyarnoy logarifmetiki, orientirovannykh na TsOS [Methods of construction of direct converters of modular logarithmica-oriented DSP], Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES) [Problems of development of perspective micro - and nanoelectronic systems (MES)], 2010, pp. 374.
20. *Poperechnyy P.S.* Matlab Residue Library: vscripts. 2014. Available at: http://vscripts.ru/w/Matlab_residue_library (accessed 16 February 2015).

Статью рекомендовал к опубликованию д.т.н., профессор А.Л. Глебов.

Щелоков Альберт Николаевич – Институт проблем проектирования в микроэлектронике РАН; e-mail: schan@ippm.ru; 124681, Зеленоград, ул. Советская, 3; тел.: +74997299890; зам. директора; к.ф.-м.н.

Амербаев Вильжан Мавлютинович – e-mail: ippm@ippm.ru; отдел методологии вычислительных процедур; д.т.н; г.н.с.

Соловьев Роман Александрович – e-mail: turbo@ippm.ru; отдел методологии вычислительных процедур; руководитель отдела; к.т.н.

Тельпухов Дмитрий Владимирович – e-mail: nofrost@inbox.ru; отдел методологии вычислительных процедур; н.с.; к.т.н.

Поперечный Павел Сергеевич – e-mail: ppoperechny@elvees.com; аспирант.

Рухлов Владимир Сергеевич – e-mail: do1p@ya.ru; отдел методологии вычислительных процедур; м.н.с.

Михмель Артем Сергеевич – e-mail: rf170c@gmail.com; инженер.

Schelokov Albert Nikolaevich – The Institute for Design Problems in Microelectronics (IPPM RAS); e-mail: schan@ippm.ru; 3, Sovetskaya street, Zelenograd, 124681, Russia; phone: +74997299890; deputy director; cand. of phis.-math. sc.

Amerbaev Viljan Mavlutinovich – e-mail: ippm@ippm.ru; department of computing procedure methodology; chief researcher; dr. of eng. sc.

Solovyev Roman Alexandrovich – e-mail: turbo@ippm.ru; department of computing procedure methodology; head of department; cand. of eng. sc.

Telpukhov Dmitry Vladimirovich – e-mail: nofrost@inbox.ru; department of computing procedure methodology; researcher; cand. of eng. sc.

Poperechny Pavel Sergeevich – e-mail: ppoperechny@elvees.com; postgraduate.

Rukhlov Vladimir Sergeevich – e-mail: do1p@ya.ru; department of computing procedure methodology; junior scientist.

Michmel Artem Sergeevich – e-mail: rf170c@gmail.com; engineer.