

А. И. ГРУШИН, М. Л. РЕМИЗОВ, А. В. РОСТОВЦЕВ,  
Д. Д. НИКОЛАЕВ, ЧИНЬ КУАНГ КИЕН

## ВЫЧИСЛИТЕЛЬНОЕ УСТРОЙСТВО ДЛЯ ОБРАБОТКИ РАДИОЛОКАЦИОННОЙ ИНФОРМАЦИИ

Рассмотрена аппаратная реализация алгоритма рекурсивного вычисления комплексной матрицы  $64 \times 64$ . При обработке радиолокационной информации требуется производить такое вычисление в режиме реального времени. Проанализирована и преобразована исходная формула, разработаны функциональные модели, проведено исследование точности вычислений, предложен способ представления информации, облегчающий проектирование, разработаны и верифицированы узлы устройства, разработана структура специализированного вычислительного устройства, проведена отладка устройства, синтезирован прототип.

*Ключевые слова:* вычислительное устройство, матрица, комплексное умножение с накоплением, числа с плавающей запятой, ПЛИС.

**Введение.** В статье [1] было показано, что при выделении сигналов на фоне интенсивных помех наиболее трудоемкой является процедура вычисления в режиме реального времени комплексной обратной ( $64 \times 64$ )-матрицы  $R^{-1}_n$ , зависящей от выборки векторов  $y_n$  и целой положительной константы  $\mu$ :

$$\mathbf{R}_0^{-1} = \mathbf{I}; \mathbf{R}_n^{-1} = \mathbf{R}_{n-1}^{-1} - \frac{1}{\mu + \tilde{y}_n \mathbf{z}_n} \mathbf{z}_n \tilde{\mathbf{z}}_n, \quad z_n = R_{n-1}^{-1} y_n, \quad n = 1, 2, \dots, 128, \quad (1)$$

где  $\mathbf{I}$  — единичная матрица,  $y_n$  — 64-мерный комплексный вектор, коэффициенты действительной и мнимой частей которого — 12-разрядные целые числа со знаком, „ $\tilde{\phantom{x}}$ “ обозначает комплексное сопряжение и транспонирование, количество векторов  $y_n$  ( $n = 1, N$ ) в одной выборке — 128.

Значение матрицы  $R^{-1}$  вычисляется на выборке из 128 векторов  $y_n$ , затем она передается в другое устройство для принятия решения об обнаружении полезного сигнала. За 5 с (рабочий цикл радиолокационной системы) необходимо вычислить матрицу для 576 различных выборок векторов с тремя значениями  $\mu$ .

Вычисления на компьютере (Core 2 Duo, 2,66 ГГц, 1 Гбайт) занимают более 43 мин, т.е. не обеспечивают требуемой скорости, поэтому было принято решение реализовать вычисления на аппаратуре, имеющей параллельную архитектуру с созданием прототипа на программируемой логической интегральной схеме (ПЛИС).

Анализ формулы (1) и результатов моделирования позволил установить, что  $\tilde{y}_n z_n$  — положительная величина, поэтому вычисления в формате single [2] обеспечивают необходимые диапазон и точность.

**Вычислительное устройство. Схема вычисления и формат чисел.** Для вычисления нового значения матрицы нужно провести 128 итераций по формуле (1). Каждую итерацию разделим на этапы (см. табл. 1).

Таблица 1

№	Этап	Операции и объем вычислений
1	$z_n = R_{n-1}^{-1} y_n$	64×64 MAC
2	$\alpha = \mu + \tilde{y}_n z_n$	1×64 MAC
3	$k = 1 / \alpha$	1 DIV
4	$w_n = -k z_n$	64 MUL
5	$R_n^{-1} = R_{n-1}^{-1} - w_n \tilde{z}_n$	64×64 MAC

Операции деления DIV и умножения MUL выполняются над действительными числами, а умножения с накоплением MAC — над комплексными.

Моделирование показало, что использование 62-разрядных чисел с фиксированной запятой обеспечивает необходимый диапазон вычислений и точность до 10-го двоичного знака по сравнению с вычислениями программным способом в формате single, но производительность при этом меньше требуемой.

Использование чисел с плавающей запятой формата single позволяет автоматически решить проблему обеспечения необходимого диапазона и точности вычислений.

В табл. 2 проведено сравнение двух реализаций вычислителя.

Таблица 2

Характеристика	Фиксированная запятая (62 разряда)	Плавающая запятая (формат single)
Занимаемая площадь кристалла, %	12	55
Рабочая частота, МГц	300	200—250
Время вычисления $R^{-1}$ для одного $\mu$ , мс	5,6	1
Трудозатраты на разработку, чел.×мес.	~6	~15

Использование чисел с фиксированной запятой предполагает возможность применения более простых алгоритмов, обеспечения меньших трудозатрат, меньшей площади кристалла, но требование по производительности повлекло за собой выбор метода чисел с плавающей запятой.

Исследование алгоритма вычисления обратной матрицы показало, что для обеспечения необходимого диапазона вычислений достаточно 7-разрядного порядка, поэтому в работе предложен формат чисел с плавающей запятой, в котором скрытый бит мантиссы представлен в явном виде (рис. 1, *a* — формат single IEEE Standard 754, *b* — предложенный формат чисел с плавающей запятой).

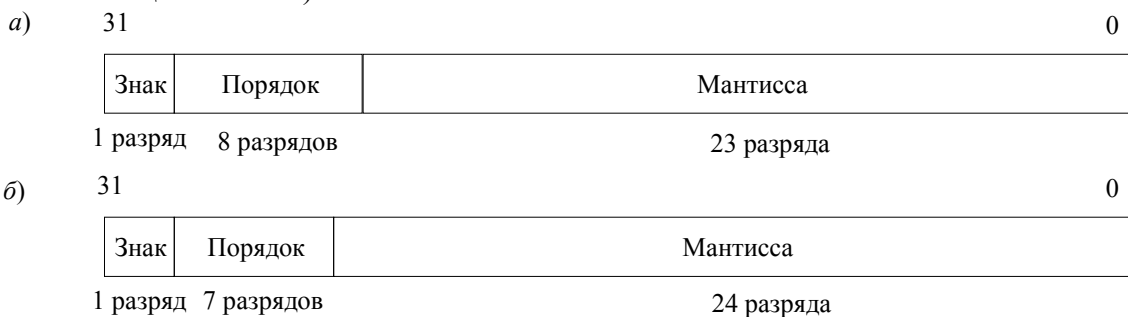


Рис. 1

В устройстве не поддержаны некоторые требования стандарта [2], это упростило аппаратуру и повысило быстродействие, при этом результат совпадает с данными, вычисленными с соблюдением всех требований стандарта.

Основные узлы вычислительного устройства.

— Узел преобразования целого числа в вещественное CI2F. На входе — 12-разрядное целое число  $Y$  в прямом коде со знаком. На выходе — вещественное число, содержащее знак, 7-разрядный порядок и 24-разрядную мантиссу, младшие 13 разрядов мантиссы всегда нулевые.

— Узел комплексного умножения с накоплением MAC.

Узел вычисляет следующую функцию (рис. 2, а):

$$AC - BD + E + (AD + BC + F)i,$$

где  $A + Bi$ ,  $C + Di$ ,  $E + Fi$  — комплексные операнды, коэффициенты  $A, B, C, D, E, F$  — числа с плавающей запятой.

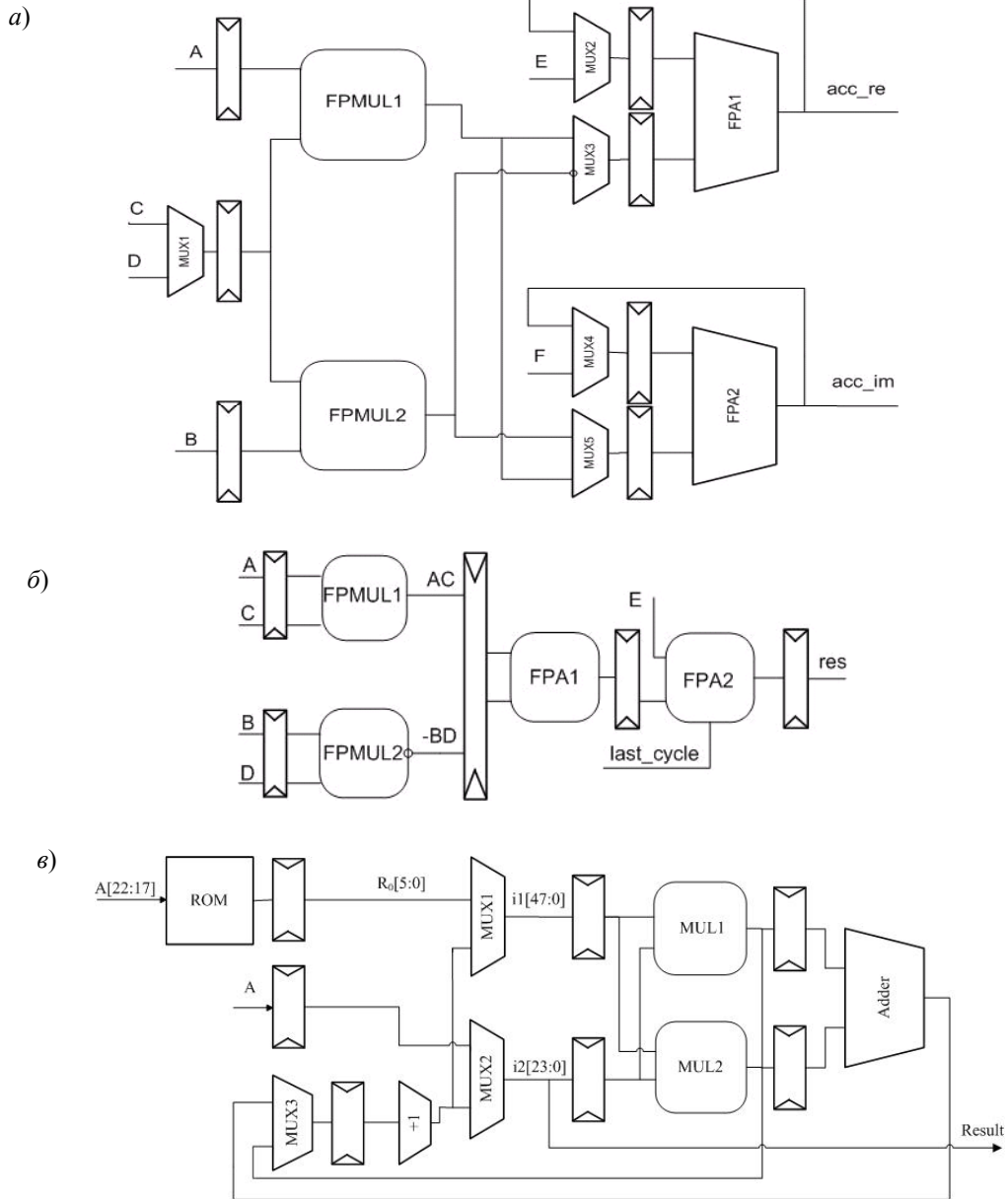


Рис. 2

Сначала коммутатор MUX1 пропускает коэффициент  $C$  во входной регистр. Узел умножения чисел с плавающей запятой FPMUL1 вычисляет произведение  $AC$ , а узел FPMUL2 —  $BC$ .  $AC$  проходит через MUX3 на вход узла сложения чисел с плавающей запятой FPA1, на второй вход которого через коммутатор MUX2 проходит  $E$ .  $BC$  через MUX5 попадает на вход FPA2, где складывается с  $F$ .

Затем через MUX1 в узлы умножения проходит  $D$ , в FPMUL1 вычисляется произведение  $AD$ , в FPMUL2 —  $BD$ .  $AD$  через MUX5 попадает в FPA2, где складывается с  $BC + F$ , которое через MUX4 подается с выхода асс\_ге.  $BD$  через MUX3 попадает в FPA1, где вычитается из  $AC + E$ , которое через MUX2 подается с выхода асс\_им. В режиме накопления через внешние коммутаторы на входы  $E$  и  $F$  подается ранее вычисленный комплексный результат.

Таким образом, на выходе асс\_ге имеем  $AC - BD + E$ , на асс\_им —  $AD + BC + F$ . Время выполнения операции MAC 14 тактов, новые операнды на вход MAC можно подавать каждые 8 тактов.

— Узел комплексного умножения с накоплением MACR.

На вход узла MACR (см. рис. 2, б) поступают два комплексных числа  $A + Bi$  и  $C + Di$  и действительное число  $E$ . Результат является положительным числом с плавающей запятой  $res = AC - BD + E$ . Мантиссы чисел  $A$  и  $B$  содержат 24 разряда, а  $C$  и  $D$  — 11 разрядов.

MACR состоит из двух узлов умножения чисел с плавающей запятой FPMUL1 и FPMUL2 и двух узлов сложения чисел с плавающей запятой FPA1 и FPA2. Для повышения точности разность двух полных 35-разрядных произведений  $AC - BD$  округляется до 35 разрядов к ближайшему. К полученному значению прибавляется  $E$ , и сумма округляется до 35 разрядов. Когда сигнал  $last\_cycle = 1$ , происходит округление до 24 разрядов, т.е. большего из форматов исходных чисел.

Время выполнения операции MACR 12 тактов, новые операнды на вход MACR можно подавать каждые 4 такта.

— Узел вычисления обратной величины Resipr.

Узел использует алгоритм Ньютона — Рафсона [3], итерации проводятся по формуле:

$$R_m = R_{m-1}(2 - MR_{m-1}),$$

где  $R_{m-1}$  — предыдущее приближение для матрицы,  $M$  — мантисса числа, обратная величина которого вычисляется.

В качестве нулевого приближения  $R_0$  используется значение из таблицы, хранящейся в памяти ROM (см. рис. 2, в), которое выбирается по шести старшим разрядам дробной части мантиссы числа, подающегося на вход узла.

Коммутатор MUX1 пропускает множимое  $R_0$ , MUX2 пропускает  $M$ , в умножителе MUL1 в следующем такте вычисляется произведение  $R_0M[23:0]$ . Разность  $2 - MR_0$  вычисляется как дополнительный код произведения  $MR_0$ .

Второе умножение первой итерации выполняется сразу на двух умножителях MUL1 и MUL2, два произведения складываются в сумматоре Adder. Произведение  $R_0(2 - MR_0)$  округляется до 24 разрядов с помощью инкрементора. Таким образом, в результате первой итерации получается новое приближение мантиссы результата  $R_1$ .

Последующие итерации выполняются аналогичным образом. Устройство выдает результат через 19 тактов после прихода операнда.

*Структура вычислительного устройства.* На рис. 3, а приведена структурная схема специализированного вычислительного устройства.

Основной объем вычислений происходит в блоке из 64 узлов MAC1—MAC64, каждый из которых вычисляет  $AC - BD + E + (AD + BC + F)i$ . Такой подход позволяет быстро выполнять действия с матрицами, предусмотренные табл. 1.

Блок MAC совершает вычисления одновременно над всеми компонентами одного столбца матрицы  $R^{-1}_n$ , что обуславливает оптимальную организацию памяти для хранения матрицы  $R^{-1}_n$ . Это память с глубиной 64 (количество столбцов) и разрядностью шины данных  $64 \times 2 \times 32$ .

В состав устройства также входят: узел преобразования целого числа в вещественное C2F, узел MACR, узел вычисления обратной величины, узел управления Control Unit и др.

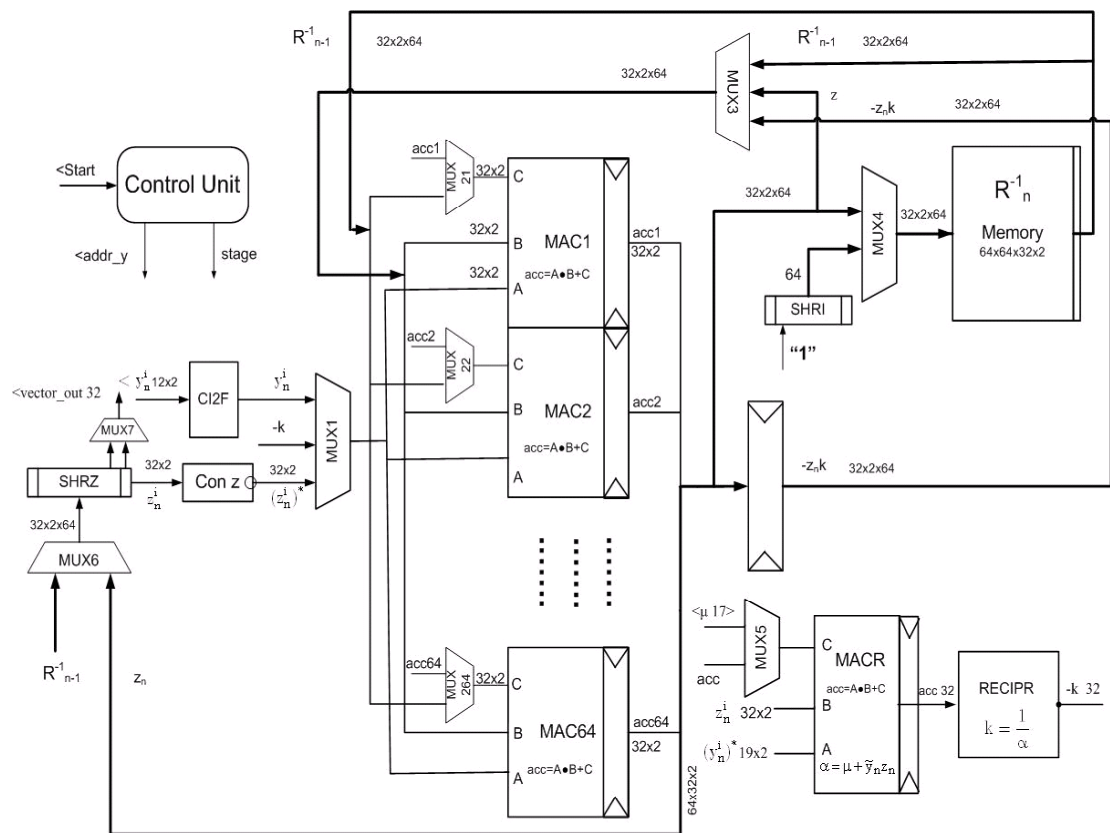
**Интерфейс вычислительного устройства.** Входной сигнал Start запускает работу устройства, параметр  $\mu$  определяет возможность различать близко находящиеся цели, от него зависит точность вычислений, вектор  $y_n$  получают аналого-цифровым преобразованием сигнала с фазированной антенной решетки.

Адрес  $addr\_y$  используется для чтения компоненты вектора  $y_n$  из внешней памяти, младшие 6 разрядов определяют номер компоненты, старшие 7 разрядов определяют номер вектора. В памяти хранятся 128 векторов выборки.

32-разрядный выход  $vector\_out$  попеременно выдает коэффициент действительной и мнимой частей элемента матрицы, через него во внешнюю память для дальнейшей обработки передается вычисленная матрица.

**Описание работы вычислителя.** Вычисления начинаются с подачи сигнала Start, который вырабатывается один раз для обсчета всех 128 векторов  $y_n$  выборки. По сигналу Start обнуляются счетчики узла управления, и устройство начинает работать в соответствии с временной диаграммой (см. рис. 3, б).

а)



б)



Рис. 3

— Фаза  $Load\_I$  (начальная загрузка).

В память обратной матрицы  $R^{-1}_n$  записывается вещественная единичная матрица  $I$ , для этого на соответствующий вход коммутатора MUX4 подаются нуль (коэффициент мнимой части), вещественная единица (коэффициент действительной части). В течение 64 тактов прописываются все 64 столбца памяти. В первом столбце вещественная единица записывается в первый (верхний) элемент, во втором столбце — во второй элемент и так далее. Для этого в сдвиговом регистре SHRI каждый такт на один разряд сдвигается единица, указывающая на нужный элемент столбца.

— Фаза *MAC1* (вычисление вектора  $z_n$ ).

Длительность фазы 64 цикла работы MAC. Коммутатор MUX1 пропускает компоненту  $y_n$ , которая поступает в MAC1—MAC64 на вход множителя, MUX2 при подаче нулевой компоненты вектора  $y_n$  пропускает нуль, затем результат с выхода MAC (асс), MUX3 — столбец из памяти матрицы  $R^{-1}_n$ . В каждом цикле работы MAC1 меняются разряды адреса чтения  $y_n$   $addr\_y[5:0]$ .

За 64 цикла работы блок MAC вырабатывает 64 компоненты вектора  $z_n$ , которые параллельно поступают в сдвиговый регистр SHRZ, где преобразуются в последовательный код во время фазы MACR.

— Фаза *MACR* (вычисление знаменателя  $\alpha$ ). Длительность фазы 64 цикла работы MACR.

На вход множителя поступает 19 разрядов компоненты вектора  $y_n^i$  (знак, 7 разрядов порядка, 11 старших разрядов мантиссы) с выхода преобразователя целого в вещественное C12F, на вход множимого поступает компонента  $z_n$  из сдвигового регистра SHRZ. Через MUX5 на вход слагаемого в начальный момент времени поступает константа  $\mu$  с входа специализированного вычислителя, затем через этот коммутатор проходит результат предыдущей операции MACR (асс).

В каждом цикле работы MACR изменяются младшие 6 разрядов адреса чтения  $y_n$  и происходит сдвиг на одну компоненту в регистре SHRZ. В конце фазы на выходном регистре определяется величина  $\alpha$ , которая используется в фазе *Recipr*.

— Фаза *Recipr* (вычисление обратной величины). За три итерации вычисляется величина  $k$ , длительность фазы 19 тактов.

— Фаза *MAC2* (вычисление произведения  $-kz_n$ ). Длительность фазы 1 цикл работы MAC.

С выхода узла *Recipr* через MUX1 на вход множителя MAC1—MAC64 поступает значение  $-k$ . На вход множимого через MUX3 поступают компоненты вектора  $z_n$ , на вход слагаемого через MUX2 подается нуль. Произведение записывается в регистр RgZK.

— Фаза *MAC3* (вычисление нового значения матрицы  $R^{-1}_n$ ). Длительность фазы 64 цикла работы MAC.

Через MUX1 в блок MAC1—MAC64 на вход множителя поступает компонента вектора  $z_n$ , она получается в узле комплексного сопряжения ConZ. На входы множимого MAC1—MAC64 через MUX3 поступают компоненты вектора  $-z_n k$ . На вход слагаемого MAC1—MAC64 через MUX2 из памяти матриц поступает столбец предыдущего приближения матрицы  $R^{-1}_{n-1}$ . На выходе MAC1—MAC64 получается новое значение столбца матрицы, которое записывается в память матрицы через MUX4.

Каждый цикл работы MAC в память матрицы записывается новое значение столбца, в сдвиговом регистре SHRZ происходит сдвиг на одну компоненту вектора  $z_n$  (таким образом, в младших 64 разрядах регистра оказывается следующая компонента вектора  $z_n$ ), из памяти считывается новый столбец. За 64 цикла работы MAC в память записывается новое значение матрицы.

По завершении фазы MAC3 значение старших семи разрядов адреса чтения компоненты вектора  $y_n$   $addr\_y[12:6]$  увеличивается на единицу.

— Фаза *Upload* (передача матрицы  $R^{-1}$  во внешнюю память по окончании обработки выборки векторов). Передача всей матрицы занимает 8192 такта.

Из памяти считывается столбец, через MUX6 он записывается в сдвиговый регистр SHRZ, где один раз в 2 такта происходит сдвиг вправо на одну компоненту. С младших 64 разрядов регистра SHRZ через MUX7 каждый такт считывается половина элемента матрицы: в первом такте 32 разряда мнимой части, во втором такте — 32 разряда действительной части. Эти действия повторяются 64 раза по числу столбцов матрицы  $R^{-1}$ . По окончании фазы устройство ожидает нового сигнала Start.

Фазы Load\_I и Upload выполняются один раз за время обработки выборки векторов  $u_n$ , остальные фазы повторяются 128 раз.

**Верификация и создание прототипа вычислительного устройства.** Все узлы вычислительного устройства прошли верификацию в автономном режиме на случайных направленных тестах. Одинаковые воздействия подавались на RTL-модель узла на языке Verilog и на функциональную модель этого узла, написанную на языке C++, затем реакции обеих моделей сравнивались.

Функциональная модель не описывает всех подробностей алгоритмов обработки чисел с плавающей запятой, используемых в узле, а опирается на операции, выполняемые микропроцессором Pentium 4, что упростило создание функциональной модели и ускорило ее отладку и работу.

После автономной верификации узлы собирались в устройство, которое также верифицировалось на случайных направленных тестах. Устройство синтезировано на ПЛИС типа Virtex-5 xc5v1x330 [4]. Прототип устройства занимает 54 % ресурсов кристалла, рабочая частота 200 МГц. Вычисление матрицы для одной выборки занимает менее  $10^{-3}$  с, производительность — ~6,5 млрд. операций с плавающей запятой в секунду.

**Заключение.** В работе описано вычислительное устройство с параллельной архитектурой для аппаратной реализации алгоритма рекурсивного вычисления комплексной матрицы  $64 \times 64$ . Проанализирована и преобразована исходная формула, разработаны функциональные модели, проведено исследование точности вычислений, предложен способ представления информации, облегчающий проектирование, разработаны и верифицированы узлы устройства, разработана структура специализированного вычислительного устройства, проведена отладка устройства, синтезирован прототип.

Производительность устройства и точность вычислений удовлетворяют требованиям.

Повышение точности вычислений улучшает возможность различать близко расположенные цели, для этого можно увеличить разрядность мантиссы используемого формата и уменьшить количество округлений при вычислениях за счет использования метода MAF (Multiply Add Fused).

Производительность устройства можно увеличить за счет полной конвейеризации узлов, использования более быстрых алгоритмов умножения и вычисления обратной величины и использования нескольких узлов MACR.

Реализация этих возможностей наряду с использованием заказной интегральной схемы позволяет увеличить производительность в 5—10 раз по сравнению с прототипом.

#### СПИСОК ЛИТЕРАТУРЫ

1. Черемисин О. П. Адаптивные алгоритмы обработки сигналов в многоканальных приемных системах с антенными решетками // Радиотехника и электроника. 2006. Т. 51, № 9. С. 1087—1098.
2. IEEE Standard for Binary Floating-Point Arithmetic. ANSI/IEEE Standard N 754. American National Standards Institute. Washington, DC, 1985.
3. Ercegovac M., Lang T. Digital Arithmetic. San Francisco: Morgan Kaufmann Publishers, 2004.
4. [Electronic resource]: <<http://www.xilinx.com/support/documentation/virtex-5.htm>>.

#### Сведения об авторах

- Анатолий Иванович Грушин** — канд. техн. наук; Институт точной механики и вычислительной техники РАН, Москва; вед. научный сотрудник; E-mail: aigrushin@ipmce.ru
- Максим Леонидович Ремизов** — Институт точной механики и вычислительной техники РАН, Москва; инженер-конструктор; E-mail: mlremizov@ipmce.ru

- Артём Вадимович Ростовцев** — Институт точной механики и вычислительной техники РАН, Москва; инженер-конструктор; E-mail: avrostovtsev@ipmce.ru
- Дмитрий Дмитриевич Николаев** — студент; Московский физико-технический институт (государственный университет), факультет радиотехники и кибернетики; E-mail: ddnikolaev@ipmce.ru
- Куанг Киен Чинь** — студент; Московский физико-технический институт (государственный университет); факультет радиотехники и кибернетики; E-mail: kkchin@ipmce.ru

Рекомендована кафедрой  
ЭВМ МФТИ

Поступила в редакцию  
03.10.08 г.