

## СПЕКТРАЛЬНЫЙ АНАЛИЗ СИГНАЛОВ НА МНОГОЯДЕРНЫХ ПРОЦЕССОРАХ

*Мусаев М.М., д.т.н., профессор кафедры «Компьютерные системы» Ташкентского Университета Информационных Технологий, e-mail: mm.musaev@rambler.ru*

*Кардашев М.С. ассистент кафедры «Компьютерные системы» Ташкентского Университета Информационных Технологий, e-mail: mihail.kardashev@gmail.com*

**Ключевые слова:** спектральный анализ, сигнальный процессор, многоядерный, базисные системы, алгоритмы, аппаратные средства, параллельная обработка.

### Введение

Ускорение алгоритмов ЦОС долгое время заключалось в уменьшении общего количества вычислительных операций за счёт оптимизации существующих алгоритмов или упрощения численных методов. Примерами могут служить рекурсивные вычисления в быстром преобразовании Фурье или применение двоично-рациональных базисных систем. Однако технологические достижения последних лет в области производства процессоров позволяют разделить нагрузку между несколькими вычислительными узлами – ядрами процессора. Применение многоядерных процессоров дает существенное увеличение скорости обработки, однако еще очень мало создано алгоритмов и прикладных программ, способных эффективно работать на многоядерных процессорах в режиме потоковой обработки [1]. Необходима разработка прикладных программ ЦОС, реализуемых с высокой скоростью на новых параллельных системах.

Потоковая обработка в задачах ЦОС должна рассматриваться как новая технология, включающая следующие элементы подготовки и исполнения программ:

- анализ численного метода и соответствующего алгоритма на предмет создания по возможности независимых вычислительных потоков;
- выбор эффективных языковых средств написания программ;
- освоение технологий формирования потоков с помощью современных инструментальных средств;
- оценка эффективности разработанного параллельного решения с помощью соответствующего инструментария.

### Алгоритмы и аппаратные средства

Превращение последовательного алгоритма в параллельный может представлять собой как нахождение большого числа повторяющихся операций над независимыми данными, так и изменения в алгоритмической структуре, поиск других подходов к решению задачи. Приёмы, используемые для организации параллельных вычислений, включают в себя:

- параллельные циклы, итерационное пространство которых делится на число вычислительных потоков;

*Статья посвящена реализации процедур спектрального анализа сигналов на многоядерных процессорах Intel Core i3, i5. Для анализа получаемого ускорения взято несколько базисных систем Фурье-преобразования: БПФ, дискретное косинусное преобразование и преобразование Уолша-Адамара. Рассмотрены параметры ускорения параллельных алгоритмов, выполняемых при различном числе потоков. В качестве инструментария для реализации созданных параллельных версий алгоритмов спектрального анализа используются библиотеки Open MP, Intel TBB и прикладной программный интерфейс ОС Windows Win32 Native Threads.*

- алгоритмы редукции, т.е. выполнение множества однотипных операций с накоплением итоговой суммы, произведения или иной функции;

- выделение процессов (задач), которые можно исполнять одновременно;

- рекурсивные вызовы одного метода внутри самого себя с назначением разных вычислительных потоков на их выполнение;

- использование параллельных шаблонов матрично-векторных операций.

В итоге один и тот же последовательный алгоритм в параллельной версии может быть оптимизирован разными численными методами. Эффективность разработанного параллельного алгоритма напрямую зависит от среды программной реализации: параметров процессора, механизмов создания вычислительных потоков в операционной системе, числа потоков.

Максимальное ускорение параллельных алгоритмов всегда достигается при выполнении в вычислительных потоках, число которых обеспечивается спецификацией процессора. В табл. 1 приведены технические данные процессоров Intel Core i3-2120 и Intel Core i5-2310, на которых проверялась степень ускорения рассматриваемых алгоритмов.

Таблица 1. Спецификация процессоров Intel Core i3 и Core i5

Название	Число ядер	Число потоков	Частота ядра, ГГц	Гиперпоточность
Intel Core i3-2120	2	4	3.30	Да
Intel Core i5-2310	4	4	3.20	Нет

В качестве первого примера рассмотрим параллельные версии реализации алгоритма быстрого преобразования Фурье (БПФ), предложенного Кули и Тьюки (Cooley–Tukey) [2]. Операция «бабочки» (рис. 1) представляет собой функцию, на входе которой два числа:  $A$

и  $B$ . На выходе получаются числа  $A + BW$  и  $A - BW$ . Число  $W$  – коэффициент поворота – вычисляется по формуле:

$$W_N^{l(k)} = e^{-i \frac{2\pi}{N} l(k)}$$

где  $N$  – размерность БПФ,  $l(k)$  – целое число, зависящее от номера  $k$  операции «бабочка» алгоритма БПФ.

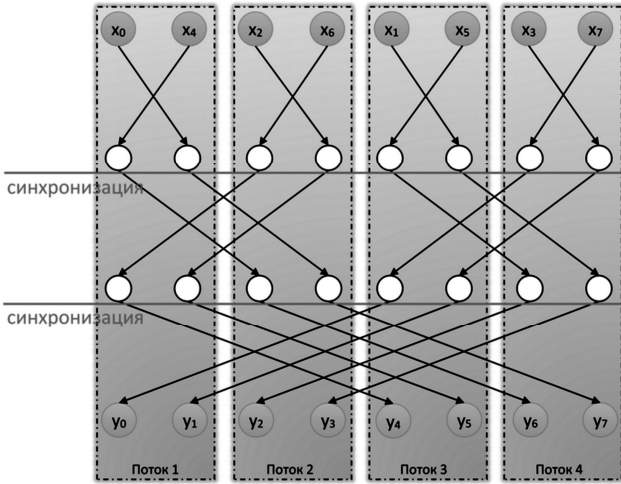


Рис. 1. Параллельная реализация алгоритма БПФ в 4-х потоках

Рассмотрим два алгоритма, один из которых реализует параллельные вычисления базовой операции «бабочки» в параллельном цикле, а другой – с помощью рекурсивных вызовов данной функции, выполняющихся параллельно.

Суть рекурсивного вычисления БПФ заключается в том, что вычисление БПФ от входного сигнала есть вычисление БПФ от каждой из его половин и так далее. На вход функции поступают массив отсчётов и его размер  $N$ . Данный массив делится на две части, каждая из которых, размером  $N/2$ , вновь отправляется в данную функцию, где полученный массив вновь будет разделен на две половины для независимого вычисления БПФ каждой из них. Условием выхода из рекурсии является значение размера массива  $N = 1$ . В самом теле функции выполняется вычисление операции «бабочка» алгоритма БПФ с коэффициентом поворота  $W$ , зависящим от размера отправленного массива.

В параллельной версии рекурсивного алгоритма вместо рекурсивного вызова создаются две подзадачи вычисления БПФ, которые выполняются параллельно.

Для сравнения эффективности данных алгоритмов рассмотрим графики ускорения каждого параллельного алгоритма относительно последовательной версии того же алгоритма, а также одного параллельного алгоритма относительно другого (рис. 2). По горизонтали отложены значения  $n$  (при числе отсчётов сигнала, равном  $2^n$ ).

Графики ускорений наглядно демонстрируют эффективность рекурсивного алгоритма, а также значение параметра **зернистости** данного алгоритма: минимальное число отсчётов сигнала, при котором создаются рекурсивные подзадачи, составляет  $2^{10}$ , т.к. при создании более мелких подзадач программа будет выпол-

няться медленнее последовательной версии. Последовательное выполнение недостаточно крупных задач позволяет не терять производительность из-за внедрения параллелизма на небольшом количестве отсчетов сигнала.

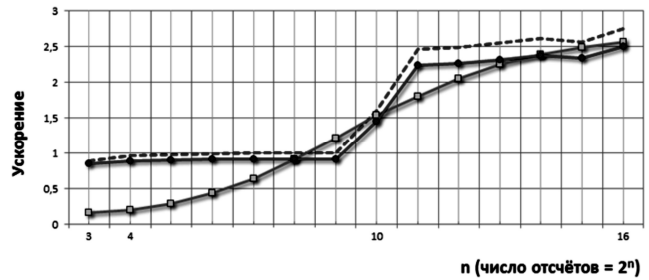


Рис. 2. Ускорение итерационного и рекурсивного параллельных алгоритмов БПФ, выполняющихся в 4 вычислительных потоках на процессоре Intel Core i5:  
 - - - - - ускорение рекурсивного алгоритма относительно последовательного рекурсивного;  
 —■— ускорение итерационного алгоритма относительно последовательного итерационного;  
 —●— ускорение рекурсивного алгоритма относительно последовательного итерационного.

Дискретное косинусное преобразование (ДКП) является одним из ортогональных преобразований, которое применяется в алгоритмах сжатия информации с потерями, таких как MPEG и JPEG. Прямое ДКП описывается в виде произведения матрицы коэффициентов  $H$  на вектор исходных отсчётов сигнала  $f$ :

$$\vec{C} = \hat{H} \cdot \vec{f}$$

Базисная функция преобразования носит периодический характер, а элементы матрицы коэффициентов задаются текущими значениями функции «косинус». Обратное преобразование описывается в виде произведения транспонированной матрицы коэффициентов на вектор отсчётов преобразованного сигнала:

$$\vec{f} = \hat{H}^T \cdot \vec{C}$$

Параллельное выполнение преобразования осуществляется построчным разбиением матрицы преобразования [2]. При входе в параллельный цикл создаётся определённое число вычислительных потоков (обычно равное числу ядер процессора). Вектор исходного сигнала и матрица преобразования являются общими для всех потоков, в то время как переменные, отвечающие за индексацию, копируются в локальный пул памяти каждого потока, оказываясь тем самым защищёнными от изменения другими потоками. Итерационное пространство параллельного цикла равномерно распределяется между всеми потоками: начальное и конечное значения для переменной  $i$ , означающей номера строк матрицы и элементов итогового сигнала, определяются номером потока.

Параллельное вычисление самой матрицы дискретного косинусного преобразования представляет собой вычисление элементов по рекуррентной формуле в параллельном цикле с разбиением матрицы по строкам. Такой алгоритм является ещё одним примером идеаль-

но распараллеливаемых вычислений и теоретически должен иметь максимальные показатели ускорения и эффективности.

В эффективности параллельного алгоритма до того, как он будет реализован с помощью той или иной специализированной библиотеки программирования, можно убедиться с помощью программы Intel Advisor [3].

Разработанный алгоритм обладает хорошей масштабируемостью. Прогнозируемое ускорение при использовании библиотеки Intel TBB (Intel Threading Building Blocks) [4] может быть рассчитано при соблюдении следующих условий:

- запланированные параллельные задачи должны исполняться в одних и тех же параллельных потоках, создаваемых единственным раз за время работы программы;

- необходимо обеспечить оптимальный параметр «зернистости» и не создавать слишком много мелких параллельных задач для выполнения итераций циклов.

Алгоритм ДКП продемонстрировал лучшее ускорение по сравнению с алгоритмом БПФ. Результаты реализации ДКП с помощью библиотеки TBB для выбранных процессоров приведены на рис. 3. Данный алгоритм выявляет более существенные различия в скорости работы выбранных процессоров. Четырёхъядерный Intel Core i5 показывает большее ускорение как на 2, так и на 4 потоках.

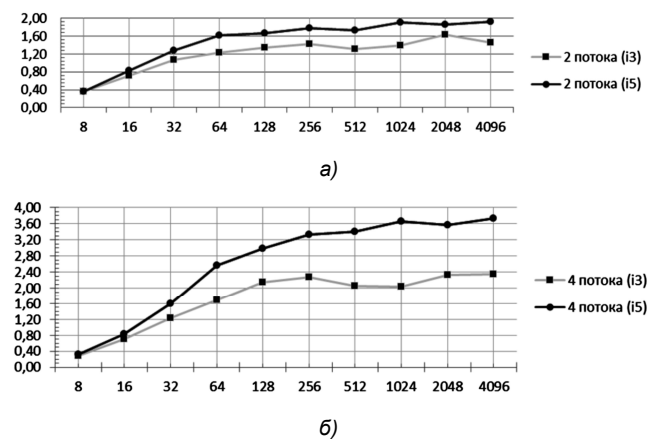


Рис. 3. Графики ускорения ДКП при выполнении на процессорах Intel Core i3 и Intel Core i5: а) в двух потоках; б) в четырех потоках

Ортогональная система функций Уолша занимает особое место среди линейных ортогональных преобразований, позволяющих эффективно сократить избыточность вычислительных операций. Преобразование Уолша-Адамара отличается относительной простотой реализации для преобразования в реальном масштабе времени и характеризуется возможностью применения быстрых алгоритмов вычислений, реализованных на многоядерном процессоре.

Функциями Уолша называется семейство функций, образующих ортогональную систему, принимающих значения только 1 и -1 на всей области определения. Группа из функций Уолша образует матрицу.

Прямое преобразование Уолша – произведение вектора исходного сигнала на матрицу коэффициентов HW:

$$C(n) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} f(k) \cdot H_w(n, k)$$

Обратное преобразование Уолша – это произведение транспонированной матрицы коэффициентов HWT на вектор спектральных коэффициентов:

$$\vec{f} = \hat{H}_w^T \cdot \vec{C}$$

Алгоритм вычисления матрицы HW носит рекурсивный характер. Количество отсчётов  $N$  исходного сигнала должно быть целым числом  $2n$ , где  $n$  – натуральное число. Матрица коэффициентов составляется из четырёх квадратов по следующему принципу:

$$\hat{H}_w(N) = \begin{array}{c|c} H_w(N/2) & H_w(N/2) \\ \hline H_w(N/2) & -H_w(N/2) \end{array}$$

Например, матрицы для  $N=2$  и  $N=4$  будут выглядеть следующим образом:

$$\hat{H}_w(2) = \begin{array}{cc} 1 & 1 \\ 1 & -1 \end{array}$$

$$\hat{H}_w(4) = \begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}$$

Таким образом, чтобы составить матрицу коэффициентов для числа отсчётов  $N$ , следует вычислить матрицу для числа отсчётов  $N/2$ , присвоить эти значения элементам каждой из трёх четвертей, а для четвертой части матрицы взять эти элементы с противоположным знаком.

Чтобы обеспечить параллельное выполнение данного алгоритма, необходимо превратить рекурсивный вызов процедуры в задачу, которая сможет выполняться параллельно с другими подобными задачами. Однако, на данном этапе производится только один вызов рекурсивной процедуры «Матрица Уолша», а дальнейшие операции зависят по данным от результатов выполнения данной процедуры. К тому же операции, выполняемые в цикле, несут в себе основную вычислительную нагрузку и их последовательное выполнение не обеспечит должной масштабируемости. Решение этой проблемы – в переносе операций из цикла в дополнительные рекурсивные вызовы, которые могут выполняться параллельно.

Две из трёх операций в цикле, которые позволяли заполнить вторую и третью четверти матрицы размера  $N$  значениями матрицы размера  $N/2$ , в представленном параллельном алгоритме превращены в вызовы процедуры «Матрица Уолша» с указанием соответствующих размеров матрицы и адресов элементов, к которым нужно обращаться. Кроме того, исходная матрица представлена здесь одномерным массивом для удобства её разбиения при рекурсивных вызовах указанной процедуры.

Другой подход к распараллеливанию алгоритма состоит в избавлении от рекурсивных вызовов и выполнении данных операций в циклах, отдельные итерации которых не имеют зависимостей по данным и могут вы-

полняться одновременно. Данный алгоритм сначала заполняет матрицу с наименьшим возможным размером ( $N=2$ ), а затем, постоянно удваивая область копирования, заполненную часть копирует «вправо», «вниз» и «по диагонали» с обратным знаком. Таким образом, заранее известно количество итераций, и во время их выполнения нет информационных зависимостей.

Рассмотренный пример наглядно показывает, как для создания параллельной версии алгоритма иногда приходится менять саму структуру алгоритма. В то время как рекурсивный подход является самым очевидным решением исходной задачи, параллельная реализация циклами проще в осуществлении и, в зависимости от конечной технологии, может работать эффективней.

Прямое и обратное преобразования Уолша осуществляются перемножением полученной матрицы (транспонированной в случае обратного преобразования) на вектор исходного сигнала. Используется алгоритм ленточного разбиения матрицы по строкам [5].

### Языковые средства и библиотеки

На стадии программной реализации параллельного алгоритма перед разработчиком стоит выбор: создавать вычислительные потоки вручную, напрямую обращаясь к средствам операционной системы, или воспользоваться специальной библиотекой программирования, расширением языка, прикладным программным интерфейсом или иным средством, автоматизирующим создание низкоуровневого кода и позволяющим проектировать параллельные области программы на более абстрактном уровне [3]. Программы на языке C++ сегодня могут использовать такие стандарты и библиотеки, как OpenMP, Intel Threading Building Blocks (TBB), Intel Math Kernel Library (MKL), Intel Integrated Performance Primitives (IPP), OpenCL, Intel Cilk Plus [5]. В то время как библиотеки MKL и IPP содержат набор готовых функций в разных областях вычислений (в том числе алгоритмы цифровой обработки сигналов), другие библиотеки и языковые расширения содержат определения для типов, функций и операторов языка, позволяющих строить собственные параллельные алгоритмы.

На рис. 4 представлено сравнение роста ускорения алгоритма ДКП. Реализация параллельных версий использует такие популярные средства, как директивы компилятора OpenMP, библиотека Intel TBB, а также ручное создание потоков с помощью стандартного прикладного программного интерфейса ОС Windows (Win32 Native Threads).

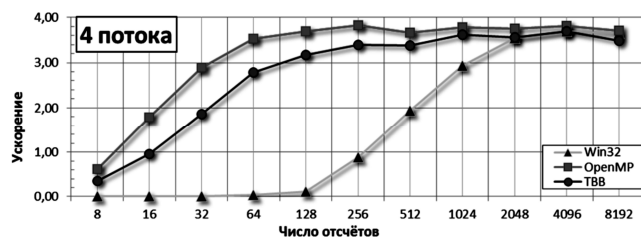


Рис. 4. Сравнение ускорения алгоритма ДКП, реализованного с помощью библиотек TBB, OpenMP, Win32

Результаты проведенных экспериментов по анализу параллельных алгоритмов спектральных преобразова-

ний в базисных системах БПФ, ДКП и системы Уолша-Адамара отражает сводная таблица (табл. 2) значений отсчетов сигнала, при которых в реализованных параллельных решениях появляется ускорение, т.е. использование параллельного алгоритма имеет смысл.

Таблица 2. Эффективные значения числа отсчетов для различных многопоточных инструментов

Инструментарий	БПФ	ДКП	Адамара
OpenMP	64	16	64
TBB	1024	16	128
Win32	2048	512	2048

Таким образом, для эффективного выполнения параллельных алгоритмов необходимо обеспечить объем вычислений, превышающий заданную границу эффективности.

### Выводы

1. Основной параметр, влияющий на масштабируемость алгоритмов спектральных преобразований и конечное ускорение работы приложения при переносе на многоядерную платформу – это число исходных отсчетов сигнала, которое и составляет вычислительную сложность представленных алгоритмов. Зависимость ускорения от числа отсчетов обычно носит характер степенной функции с горизонтальной осью; ускорение перестает расти, достигнув пороговой величины (при числе отсчетов  $2^8$ - $2^{12}$  и более для разных алгоритмов). При оптимальном решении данная максимальная величина ускорения стремится к числу процессоров в системе.

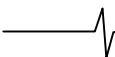
2. Другой важный параметр при реализации параллельных вычислений спектральных преобразований – минимальный объем данных (или количество итераций разделяемого цикла), который еще может обрабатываться параллельно без потери производительности. При распараллеливании итерационного пространства в большинстве случаев оптимальным значением этого параметра является частное от деления общего количества итераций на количество вычислительных потоков.

3. Оптимальное число потоков при параллельной обработке должно равняться числу вычислительных потоков, обеспечиваемых спецификацией процессора. При этом два физических ядра могут представлять собой четыре логических и обеспечивать работу четырех вычислительных потоков на уровне, близком к показателям реального четырехъядерного процессора.

4. Сравнение библиотек программирования для реализации поточного параллелизма показывает разный характер роста ускорения с увеличением числа отсчетов  $N$ . При малом числе отсчетов сигнала более эффективным представляется использование директив компилятора OpenMP (ускорение появляется при числе отсчетов от 16 до 64). При числе отсчетов больше 128 алгоритмы TBB ведут себя стабильнее.

5. Можно разделить области применения разработанных параллельных алгоритмов спектральной обработки на две группы:

– вызов параллельных алгоритмов с передачей чис-



ла отсчетов, превышающим границы эффективности для данного алгоритма (см. табл. 2). Алгоритм применим в большинстве случаев обработки стандартного звукового сигнала при работе с достаточно большими участками (построение АЧХ сигнала, спектрограммы, фильтры, эквалаизация);

– параллельное применение последовательного алгоритма к разным участкам сигнала (алгоритмы обработки речи, построение спектров очень малых участков сигнала, спектрограмма с очень малым интервалом обновления).

#### Литература

1. Эхтер Ш., Робертс Дж. Многоядерное программирование. – СПб.: «Питер», 2010. – 316 с.
2. Johnson S. G., Frigo M. Implementing FFTs in practice, ch. 11. – Rice University, Houston TX: Connexions, 2008.
3. Blair-Chappell S., Stokes A. Parallel Programming with Intel Parallel Studio XE. – Indianapolis, Indiana: «John Wiley & Sons, Inc.», 2012. – 556 p.
4. Reinders J. Intel Threading Building Blocks. Outfitting

C++ for Multi-Core Processor Parallelism. – Sebastopol, California: «O'Reilly Media, Inc.», 2007. – 334 p.

5. Шпаковский Г.И. Реализация параллельных вычислений: MPI, OpenMP, кластеры, грид, многоядерные процессоры, графические процессоры, квантовые компьютеры. – Минск: Белорусский Государственный Университет, 2010. – 155 с.

#### SPECTRUM ANALYSIS OF SIGNALS ON MULTICORE PROCESSORS

*Musayev M.M., Kardachev M.C.*

This article is devoted to the implementation of spectral analysis of signals on Intel Core i3, i5 multicore processors. Several basic Fourier transform systems were taken to analyze the received acceleration: FFT, DCT and Walsh-Hadamard transform. Acceleration parameters of parallel algorithms that are performed in a different number of threads were considered. The libraries of Open MP, Intel TBB and MS Windows API Win32 Native Threads were used as a tool for the implementation of created parallel algorithms of spectral analysis.

#### Уважаемые коллеги!

*Приглашаем Вас принять участие в формировании тематических выпусков журнала «Цифровая обработка сигналов» и размещению рекламы продукции (услуг) Вашей организации на его страницах. В случае положительного решения просим представить в редакцию журнала Ваши предложения по плановому размещению информационных материалов и макет рекламы продукции (услуг) с указанием желаемого её месторасположения: обложка (2-я, 3-я или 4-я стр.), цветная внутренняя полоса (объем полосы).*

Журнал «Цифровая обработка сигналов» издается с 1999 года. Выходит ежеквартально, тиражом – 700 экз. Распространяется по подписке через агентство «Роспечать» в России (индекс 82185), СНГ и странах Балтии (индекс 20630), а также на Конференции: «Цифровая обработка сигналов и ее применение – DSPA'» и Выставках: «ЕхроElectronica», «СвязьЭкспокомм», «ЭЛЕКТРОНИКА: компоненты, оборудование, технологии» (г. Москва) и др.

Научно-технический журнал «Цифровая обработка сигналов» включен в Список изданий, рекомендуемый ВАК РФ для публикации результатов научных исследований соискателями ученой степени доктора и кандидата технических наук в области радиотехники, связи, вычислительной техники, электроники, информационно-измерительных и управляющих систем.

#### Планируемые сроки издания отдельных номеров журнала:

- № 2 – июнь 2014 г. Тематический выпуск: «Теория и методы цифровой обработки сигналов» (по материалам международной научно-технической конференции «Цифровая обработка сигналов и ее применение – DSPA' 2014»).
- № 3 – сентябрь 2014 г. Тематический выпуск: «Цифровая обработка изображений».
- № 4 – декабрь 2014 г. Тематический выпуск: «ЦОС в радиотехнике и системах телекоммуникаций».

#### Ориентировочная стоимость рекламных услуг:

- 4-я (внешняя) страница цветной обложки – 25 тысяч рублей.
  - 2-я и 3-я (внутренние) страницы цветной обложки – 15 тысяч рублей.
  - 1/2 цветной внутренней полосы – 8 тысяч рублей.
- Ждем Ваших предложений.

С наилучшими пожеланиями, зам. главного редактора  
д.т.н., профессор Витязев Владимир Викторович, телефон 8-903-834-81-81.

Предложения прошу направлять по адресу: E-mail: vityazev.v.v@rsreu.ru или info@dspa.ru

#### Уважаемые коллеги!

Для тех, кто не успел оформить подписку на второе полугодие 2014 года через ОАО «Роспечать», сохраняется возможность приобретения журналов непосредственно в редакции по адресу: г. Москва, ул. Авиамоторная, дом 8, Научный Центр МТУСИ, Российское научно-техническое общество радиотехники, электроники и связи им. А.С. Попова, метро «Авиамоторная» или оформить Заказ в соответствии с требованиями, выставленными на сайте журнала: [www.dspa.ru](http://www.dspa.ru).

Тел.: (+7 903) 221-79-79 (Алексеева Любовь Ильинична) или (8-903) 201-53-33 (Самсонов Геннадий Андреевич).