

УДК 681.3

А.М.Сергиенко, канд. техн. наук,

Национальный технический университет Украины "КПИ", Киев,

03056, пр.Победы, 37

тел.: 4549337

E-mail: aser@comsys.ntu-kpi.kiev.ua

Спецпроцессоры для авторегрессионного анализа сигналов

Рассмотрены структуры процессоров для авторегрессионного (АР) анализа сигналов на основе алгоритма Дарбина и лестничного фильтра, которые, благодаря своей параллельной реализации в программируемой логической интегральной схеме (ПЛИС) имеют высокое быстродействие. Внедрение процессоров дает возможность расширить область применения АР анализа, например, в ультразвуковой диагностике, радиосвязи.

Розглянуто структури процесорів для авторегресійного (АР) аналізу сигналів на основі алгоритму Дарбіна і драбинкового фільтру, які, завдяки їхній паралельній реалізації в програмованій логічній інтегральній схемі (ПЛИС) мають високу швидкодію. Впровадження процесорів дає можливість розширити галузь АР аналізу, наприклад, в ультразвуковій діагностиці, радіозв'язку.

Ключевые слова: ПЛИС, авторегрессионный анализ, адаптивный фильтр, FPGA.

Введение. Авторегрессионный (АР) анализ – это эффективный способ определения характеристик физических объектов. Поэтому АР анализ вошел в практику экономических расчетов, сейсморазведки, медицинской диагностики, мобильной связи [1]. Применение ПЛИС дает возможность анализировать сигналы в диапазоне частот до сотен мегагерц. Но при этом алгоритмы анализа должны использовать арифметику с фиксированной запятой. В то же время, алгоритмы АР анализа требуют вычисления с повышенной точностью и часто выполняются с плавающей запятой. Ниже предлагаются структуры спецпроцессоров на базе ПЛИС, позволяющие выполнять АР анализ сигналов в широком диапазоне частот с применением целочисленной арифметики.

АР анализ основан на предположении, что исследуемый физический объект представим моделью АР фильтра, выполняющего фильтрацию сигнала возбуждения, например, белого шума, с получением выходного сигнала $x(n)$. В некоторых системах, например, вокодерах, сигнал возбуждения АР фильтра имеет специальную форму. Коэффициенты АР фильтра – коэффициенты предсказания a_i ($i=0, \dots, p$) – находят в два этапа [1]. Вначале вычисляют оценки автокорреляционной функции $r_{xx}(i)$, а затем решают нормальную систему уравнений Юла-Уолкера

$$R_{xx} (a_1, \dots, a_p)^T = - (r_{xx}(1), \dots, r_{xx}(p))^T, \quad (1)$$

где R_{xx} – симметричная теплицева матрица с первой строкой $r_{xx}(0), \dots, r_{xx}(p-1)$, $a_0=1$.

Систему уравнений эффективно решается по алгоритму Дарбина, описываемый гнездом циклов:

$$\begin{aligned} & \text{for } i = 1 \text{ to } p \\ & \quad a_0 = 1; \\ & \quad k_i = -(a_0 * r_i + \dots + a_{i-1} * r_i) / E_{i-1}; \\ & \quad a_i = k_i; \\ & \quad \text{for } j = 1 \text{ to } i-1 \\ & \quad \quad a_j = a_j + k_i * a_{i-j}; \\ & \quad \text{end} \\ & \quad E_i = (1 - k_i^2) * E_{i-1}; \\ & \text{end;} \end{aligned} \quad (2)$$

где $E_0 = r_0$ – начальное значение ошибки предсказания. Результатами анализа выступают коэффициенты предсказания a_i и коэффициенты отражения k_i . В практике АР анализа на основе коэффициентов a_i выполняют спектральный анализ сигнала с высоким разрешением, а дистанцию между коэффициентами k_i , приближающимся по абсолютной величине к 1, трактуют как расстояние между слоями и другими неоднородностями среды, в которой распространяется входной сигнал [1,2].

Как видим, алгоритм (2) требует порядка p^2 операций и его трудно распараллелить. Поэтому часто рекомендуют вместо алгоритма Дарбина использовать более трудоемкие, но линейно распараллеливаемые алгоритмы, такие как алгоритмы QR-факторизации [3,4]. В этой связи алгоритм Гивенса или обращение теплицевой матрицы также может иметь преимущества при его реализации в ПЛИС [5,6].

Так как в (2) приходится выполнять деление на ошибку предсказания E_i , которая может стремиться к нулю, то для решения системы уравнений необходимы вычисления с повышенной точностью. Аналогичные свойства имеют и другие алгоритмы [7]. Поэтому чаще всего задачу АР анализа решают на микропроцессорах с плавающей запятой.

В статье рассматриваются два варианта процессоров для АР анализа, которые в полной мере используют преимущества и особенности архитектуры ПЛИС.

Процессор, реализующий алгоритм Дарбина. Для получения устойчивой оценки параметров a_i необходимо накопить значения $r_{xx}(i)$ для не менее, чем $qp = 10p$ отсчетов данных [1]. При p , ограниченном десятками, сложность этапа вычисления корреляции приблизительно в p раз выше сложности решения системы уравнений. Если данные поступают последовательно, распараллеливание решения системы уравнений с применением многопроцессорной системы не дает заметного выигрыша в производительности, так как накопление корреляции будет превалировать в балансе времени. В этом случае предлагается процессор для АР анализа выполнить в виде блока коррелятора, состоящего из p процессорных элементов, работающих с фиксированной запятой и блока выполнения алгоритма Дарбина с арифметикой повышенной точности (рис.1). При этом оба блока могут выполнять свои задачи приблизительно за одинаковое время.

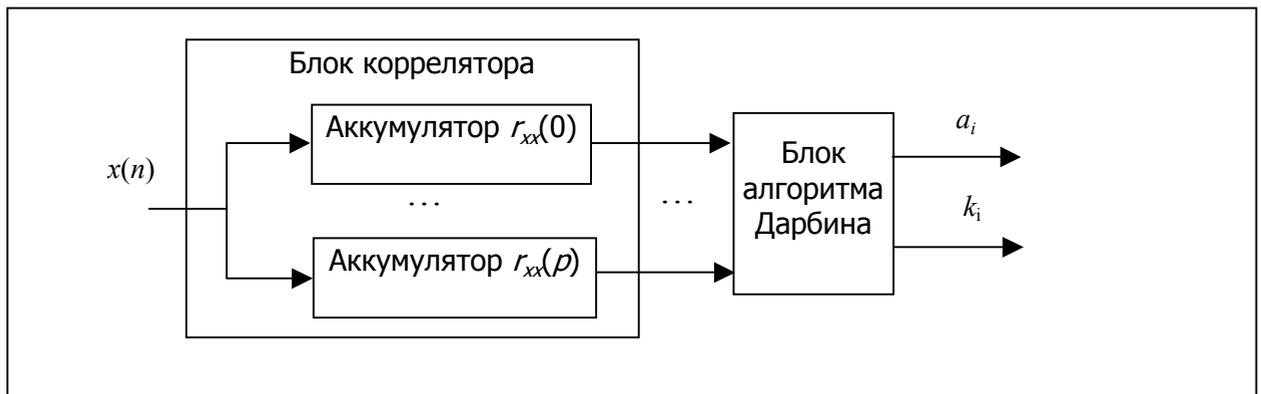


Рис.1. Структура процессора АР-анализа на основе алгоритма Дарбина

В алгоритме (2) критический путь проходит через циклический участок получения коэффициента k_i . При конвейерной реализации операций длительность итерации равна глубине конвейеров сумматора и умножителя. Если применять арифметические устройства с плавающей запятой, реализованные в ПЛИС [8], то длительность итерации будет равна 9–20 тактов, т.е. блок будет простаивать бо́льшую часть времени.

В [9] предложено использовать для решения задач линейной алгебры арифметику рациональных дробей, которая может иметь преимущества перед плавающей запятой при своей реализации в ПЛИС. При этом число x представляется двумя целыми числами: числителем n_x и знаменателем d_x дроби, то есть $x = n_x/d_x$. Все вычисления алгоритма Дарбина предлагается выполнять с данными в виде таких дробей. Умножение, деление и сложение вычисляются по формулам:

$$x * y = n_x \cdot n_y / (d_x \cdot d_y); \quad x / y = n_x \cdot d_y / (d_x \cdot n_y); \quad x + y = (n_x \cdot d_y + n_y \cdot d_x) / (d_x \cdot d_y).$$

Такие вычисления поддерживаются в современных ПЛИС десятками и сотнями блоков умножения с накоплением, такими как блок DSP48 в ПЛИС Xilinx Virtex.

При использовании этой арифметики для исполнения алгоритма Дарбина обеспечивается как малая погрешность вычислений, так и расширенный динамический диапазон в сравнении с арифметикой целых чисел, а также простота реализации в ПЛИС и высокое быстродействие. Кроме того, операция деления, входящая в цикл алгоритма, выполняется максимально кратко и с минимизированной погрешностью. Если необходимо естественное представление результатов, то в конце вычислений числители дробей результатов a_i или k_i делятся на знаменатели обычным делением. При этом такое деление не увеличивает цикл работы процессора AP анализа.

Структура арифметического устройства блока алгоритма Дарбина (рис.2) выполняет такие операции, как $P=A +Y$, $P=AX + Y$, $P=AX + P$, $P = A/X$. На рис. 2 узкими прямоугольниками показаны ступени регистров. Блоки нормализации БН выполняют нормализацию числителей и знаменателей дробей сдвигом на одинаковое число разрядов влево.

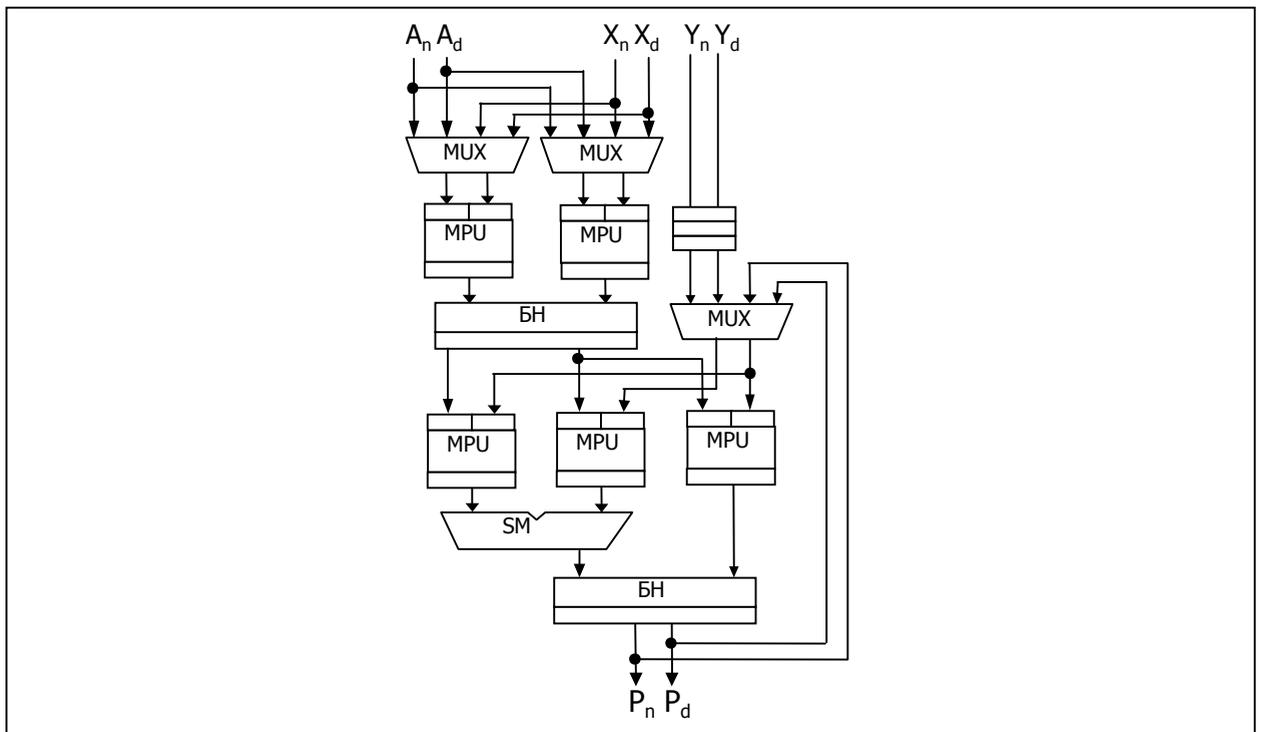


Рис.2. Арифметическое устройство блока алгоритма Дарбина

Как видно из рис.2, длительность итерации операции накопления суммы парных произведений составляет всего 3 такта. Это означает, что выполнение основного цикла алгоритма происходит в 3–6 раз быстрее, чем в аналогичном устройстве с плавающей запятой. Кроме того, операция деления выполняется всего за 6 тактов, а не за 17–27 тактов, как в делителе с плавающей запятой [8]. В результате, выполнение алгоритма Дарбина происходит, как минимум, втрое быстрее. Кроме того, такое арифметическое устройство при разрядности числителей и знаменателей 18 занимает 360 эквивалентных логических блоков (ЭКЛБ) и 5 блоков DSP-48 в ПЛИС Xilinx Virtex-4. Арифметическое устройство с плавающей запятой одинарной точности, выполняющее такие же функции с той же тактовой частотой, содержит 1154 ЭКЛБ и 4 блока DSP-48 [8], т.е. оно имеет существенно большие аппаратные затраты.

Процессор на основе лестничного фильтра. Фильтр с конечной импульсной характеристикой (КИХ) с коэффициентами a_i является обратным к АР фильтру. Поэтому адаптивный КИХ-фильтр часто используют для получения коэффициентов авторегрессии [1,2]. Однако медленная и не всегда надежная сходимость адаптации таких фильтров препятствует их применению для анализа быстропротекающих процессов.

Широко известна лестничная форма адаптивного КИХ-фильтра, j -я ступень которого выполняет вычисления [1]:

$$E_j^f(n) = E_{j-1}^f(n) + k_j E_{j-1}^b(n-1); \quad E_j^b(n) = E_{j-1}^b(n-1) + k_j E_{j-1}^f(n), \quad (3)$$

где $E_j^f(n)$, $E_{j-1}^b(n)$ – ошибки предсказания вперед и назад в n -м такте, соответственно, k_j – коэффициент отражения, $E_0^f(n) = E_0^b(n) = x(n)$, $j=1, \dots, p$.

В фильтрах с последовательной адаптацией коэффициенты k_j вычисляются по рекурсивному градиентному алгоритму. В фильтре для обработки блоков $N = qp$ данных эти коэффициенты наиболее точно вычисляются по формуле [1]:

$$k_j = - \sum_n E_{j-1}^f(n) E_{j-1}^b(n) / \left(\sqrt{\sum_n (E_{j-1}^f(n))^2} \sqrt{\sum_n (E_{j-1}^b(n))^2} \right). \quad (4)$$

Здесь суммы в числителе и знаменателе являются оценками коэффициентов частичной корреляции. В такой вычислительной схеме коэффициенты k_j находят последовательно, начиная с первого, p раз пропуская один и тот же массив данных через схему фильтра. Если сигнал стационарный в пределах pN отсчетов, то данные могут следовать непрерывно.

Структура процессора для АР анализа состоит из лестничного фильтра, каждая ступень которого вычисляет формулы (3) и блока вычисления коэффициентов k_j . В процессе вычислений блок вычисления коэффициентов на j -м шаге адаптации подключается к входам j -й ступени фильтра, накапливает оценки коэффициентов частичной корреляции, вычисляет k_j по формуле (4) и подставляет его как множитель для умножителей j -й ступени фильтра. Таким образом, в данной схеме совмещены этапы накопления корреляции и вычисления коэффициентов отражения.

В отличие от традиционных схем адаптивных фильтров, здесь процесс адаптации надежно сходится менее, чем за pN тактов. По сравнению со схемой, решающей уравнения Юла-Уолкера, потенциально здесь нет накопления ошибок вычислений и всегда получаются стабильные результаты (т.е. $|k_j| < 1$). Поэтому для реализации блока вычисления коэффициентов достаточно использовать арифметику целых чисел. Однако, из-за необходимости вычислений знаменателя формулы (4) с повышенной точностью, такой алгоритм до сих пор не находил широкого применения.

Такой процессор АР анализа несложно реализовать в ПЛИС, хотя для этого необходима, по крайней мере, удвоенная разрядность при вычислении коэффициентов. Структура процессора показана на рис.3.

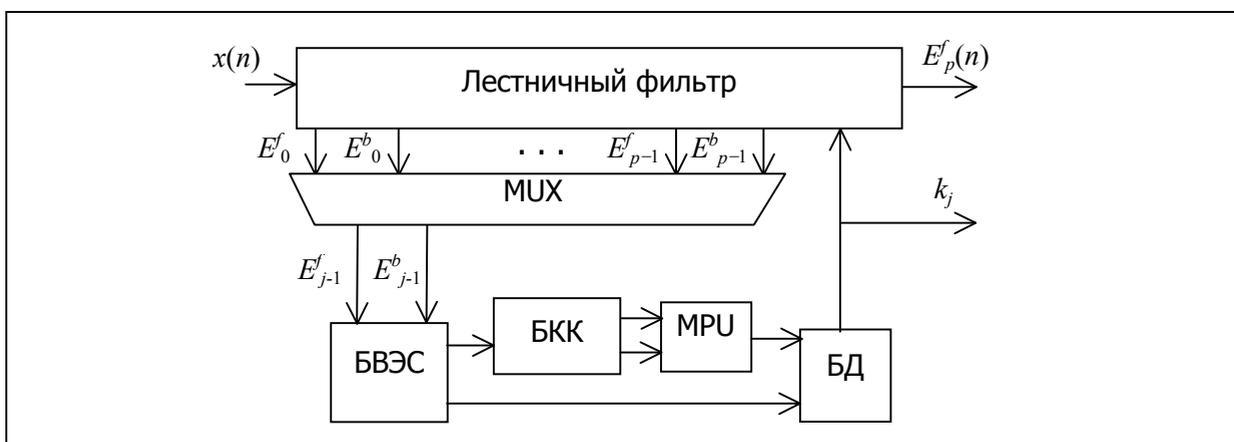


Рис.3. Структура процессора на основе лестничного фильтра

Блок вычисления энергии сигнала (БВЭС) вычисляет четыре суммы ошибок предсказания вперед и назад в формуле (4) в течение pN тактов, принимая ошибки предсказания из j -й ступени фильтра через мультиплексор МUX. Блок извлечения квадратного корня (БКК), умножитель MPU и блок деления, которые завершают вычисления по формуле (4), могут быть выполнены по параллельно-последовательной схеме. Так как число pN сравнительно велико, то их невысокое

быстродействие незначительно ухудшит быстродействие процессора в целом. В результате, при 16-разрядных входных данных на одну ступень лестничного фильтра в ПЛИС Xilinx Virtex-4 приходится 90 ЭКЛБ и 2 блока DSP-48, а на вычислитель коэффициента отражения – около 550 ЭКЛБ и 4 блока DSP-48.

Преимущество процессора на основе лестничного фильтра в том, что после получения коэффициентов отражения он может работать в режиме фильтрации входного сигнала с высоким быстродействием. Скорость схождения процесса адаптации можно увеличить, поставив по блоку вычисления коэффициентов на каждую ступень фильтра, но тогда аппаратные затраты существенно возрастут.

Процессоры AP анализа, реализованные в ПЛИС. Две вышеописанные вычислительные схемы для AP анализа были выполнены в виде спецпроцессоров, конфигурированных в ПЛИС фирмы Xilinx. Процессор, реализующий алгоритм Дарбина, имеет АЛУ, выполняющее параллельно операции умножения/деления, накопления в арифметике рациональных дробей с 18-разрядными числителем и знаменателем, а также преобразование дроби в целый 18-разрядный результат. Операции умножения/деления выполняются в конвейерном режиме с периодом 1 такт с латентной задержкой 7 тактов. Операция накопления выполняется за 4 такта, т.к. приходится ее результат использовать как исходное данное этой операции.

Результаты реализации процессоров для 16-разрядных данных, $q = 10$ и различных значений p представлены в табл.1. Их сравнение показывает, что процессоры с алгоритмом Дарбина имеют бо́льшую тактовую частоту, меньшие аппаратные затраты и меньший цикл получения результатов (период адаптации). Анализ проектов показывает, что можно повысить частоту поступления входных данных вдвое, если тактировать блок вычисления корреляционной функции и лестничный фильтр вдвое большей частотой, чем блок вычисления коэффициентов. Процессор на основе лестничного фильтра может иметь преимущество при p – малом и при $q < 10$, а также при обработке непрерывного потока данных. Кроме того, он обеспечивает устойчивую оценку k_j для любых входных данных.

По сравнению с аналогичными проектами устройств для AP анализа на базе ПЛИС [10,11,12] при одинаковой пропускной способности предлагаемые процессоры имеют существенно меньшие аппаратные затраты. Так, при одинаковом числе блоков умножения (эквивалентных DSP48) фильтр [11], имеет почти в 20 раз бо́льшие аппаратные затраты в количестве ЛТ, чем процессор с алгоритмом Дарбина. В работе [12] описан адаптивный лестничный фильтр на основе логарифмической системы счисления, который реализует ступени фильтра последовательно, из-за чего имеет существенно меньшее быстродействие.

Таблица 1. Процессоры с порядком p , реализованные в ПЛИС XC4VSX35

Процессор на основе	Аппаратные затраты, ЭКЛБ +DSP48			Максимальная тактовая частота, МГц			Длительность цикла полу- чения k_1, \dots, k_p , тактов		
	$p=10$	$p=30$	$p=90$	$p=10$	$p=30$	$p=90$	$p=10$	$p=30$	$p=90$
алгоритма Дарбина	1330+15	1850+35	4753+95	172	169	160	610	1830	5490
лестничного фильтра	1416+24	3280+64	8311+184	151	147	145	1650	10950	86850

Выводы. Предложены два типа процессоров для AP анализа, которые при их реализации в ПЛИС дают возможность в реальном масштабе времени анализировать сигналы, дискретизированные с частотой до 300 МГц и имеют невысокие аппаратные затраты. Внедрение процессоров дает возможность существенно расширить область применения AP анализа, например, при ультразвуковой диагностике в медицине и машиностроении, радиосвязи.

The autoregressive analysis processors based on Durbin algorithm and adaptive lattice filter are considered. The processors are implemented in FPGA which provides their high throughput up to hundreds of Megahertz.

1. *Марпл С.Л.* Цифровой спектральный анализ и его приложения –М.: Мир. –1990. – 584с.
2. *Уидроу Б., Стуриз С.* Адаптивная обработка сигналов. – М.: Радио и связь. 1989. – 440с.
3. *Bojanczyk A., Brent R.P., Kung H.T.* Numerically stable solution for dense systems of linear equations using mesh-connected processors//SIAM J. Sci Statist. Comput.-1984.-V. 1,-р. 95-104
4. *Brent R.P.* Old and new algorithms for Toeplitz systems //Proc. SPIE, V.975, Advanced Algorithms and Architectures for Signal Processing III, SPIE, Bellingham, Washington, 1989. -Pp.2–9.
5. *Клименко О.М., Сергієнко А.М., Шевченко Ю.В., Овраменко С.Г.* Конфігурована обчислювальна система для вирішення задач лінійної алгебри // Електрон. моделювання. –2005. –Т.27. –№1. –с.109–114.
6. *Sergyienko, O. Maslennikov.O.* Implementation of Givens QR Decomposition in FPGA. Lecture Notes in Computer Science. – Berlin: Springer. –2002. –V.2328. –р. 453–459.
7. *Ортега Дж.* Введение в параллельные и векторные методы решения линейных систем. –М.: Мир. –1991. – 365с.
8. DS335. Floating-Point Operator v1.0. // Xilinx Product Specification. –July 27. – 2005. –20р., доступно на <http://www.xilinx.com>
9. *Сергиенко А.М.* Применение арифметики рациональных дробей для реализации метода сопряжения градиентов. //Электрон. моделирование. –2006. –Т.28. –№ 1. – С. 33–41.
87. *Уидроу Б., Стуриз С.* Адаптивная обработка сигналов. – М.: Радио и связь. 1989. – 440с.
10. *Hwang K. S., Casavant A. E., Chang C. T., d'Abreu M. A.* Scheduling and hardware sharing in pipelined data paths // Proc. Int'l Conf. on Computer Aided Design. –1989. –р. 24–27.
11. *Lin A.Y., Gugel K.S.* Feasibility of fixed-point transversal adaptive filters in FPGA devices with embedded DSP blocks // 3rd IEEE Int. Workshop on System-on-Chip for Real-Time Applications (IWSOC'03). –2003. –P. 157–160.
12. *Pohl Z., Matoušek R., Kadlec J., Tichý M., Licko M.* Lattice adaptive filter implementation for FPGA // Proc. 2003 ACM/SIGDA 11-th Int. Symp. – Monterey, California, USA. –2003. –P.246–250.