



УДК 681.3

А. М. Сергиенко, канд. техн. наук
Национальный технический университет Украины «КПИ»
(Украина, 03056, Киев, пр. Победы, 37,
тел.: 4549337; E-mail: aser@comsys.ntu-kpi.kiev.ua)

Спецпроцессоры для авторегрессионного анализа сигналов

(Статью представил д-р техн. наук В. П. Симоненко)

Рассмотрены структуры процессоров для авторегрессионного анализа сигналов на основе алгоритма Дарбина и лестничного фильтра, имеющие высокое быстродействие вследствие параллельной реализации в программируемой логической интегральной схеме. Внедрение процессоров дает возможность расширить область применения авторегрессионного анализа, например в ультразвуковой диагностике, радиосвязи.

Розглянуто структури процесорів для авторегресійного аналізу сигналів на основі алгоритму Дарбіна і драбинкового фільтру, які мають високу швидкість в результаті паралельної реалізації в програмованій логічній інтегральній схемі. Впровадження процесорів дає можливість розширити галузь авторегресійного аналізу, наприклад в ультразвуковій діагностиці, радіозв'язку.

К л ю ч е в ы е с л о в а: программируемая логическая интегральная схема, авторегрессионный анализ, адаптивный фильтр.

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 программируемой логической интегральной схемы (ПЛИС) дает возможность анализировать сигналы в диапазоне частот до сотен мегагерц, но при этом в алгоритмах анализа необходимо использовать арифметику с фиксированной запятой. В то же время, при использовании алгоритмов АР анализа требуется повышенная точность вычислений, и они часто выполняются с плавающей запятой. Рассмотрим структуры спецпроцессоров на основе ПЛИС, позволяющие выполнять АР анализ сигналов в широком диапазоне частот с применением целочисленной арифметики.

Авторегрессионный анализ основан на предположении, что исследуемый физический объект можно представить в виде модели 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 $r_{xx}(0), \dots, r_{xx}(p-1)$, $a_0 = 1$.

Система уравнений (1) эффективно решается по алгоритму Дарбина [1] с 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 являются коэффициенты предсказания a_i и 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 , приближающимися по абсолютной величине к единице, 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 реализации в ПЛИС [5, 6].

Поскольку в (2) приходится выполнять деление на ошибку предсказания E_i , которая может стремиться к нулю, 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.

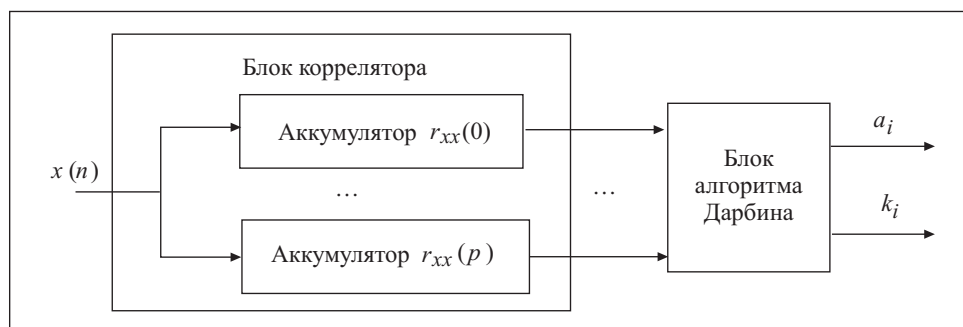


Рис. 1. Структура процессора АР анализа на основе алгоритма Дарбина

Рассмотрим два варианта процессоров для АР анализа, в которых в полной мере используются 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 превалирует в балансе времени. В этом случае процессор для АР анализа целесообразно 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.

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 составит от девяти до 20-ти тактов, т.е. блок будет простаивать большую часть времени.

В [9] предложено использовать для решения задач линейной алгебры арифметику рациональных дробей, которая при реализации в ПЛИС может иметь преимущества перед плавающей запятой. При этом число x представляется двумя целыми числами: числителем n_x и знаменателем d_x дроби, т.е. $x = n_x/d_x$. Все вычисления по алгоритму Дарбина следует вы-

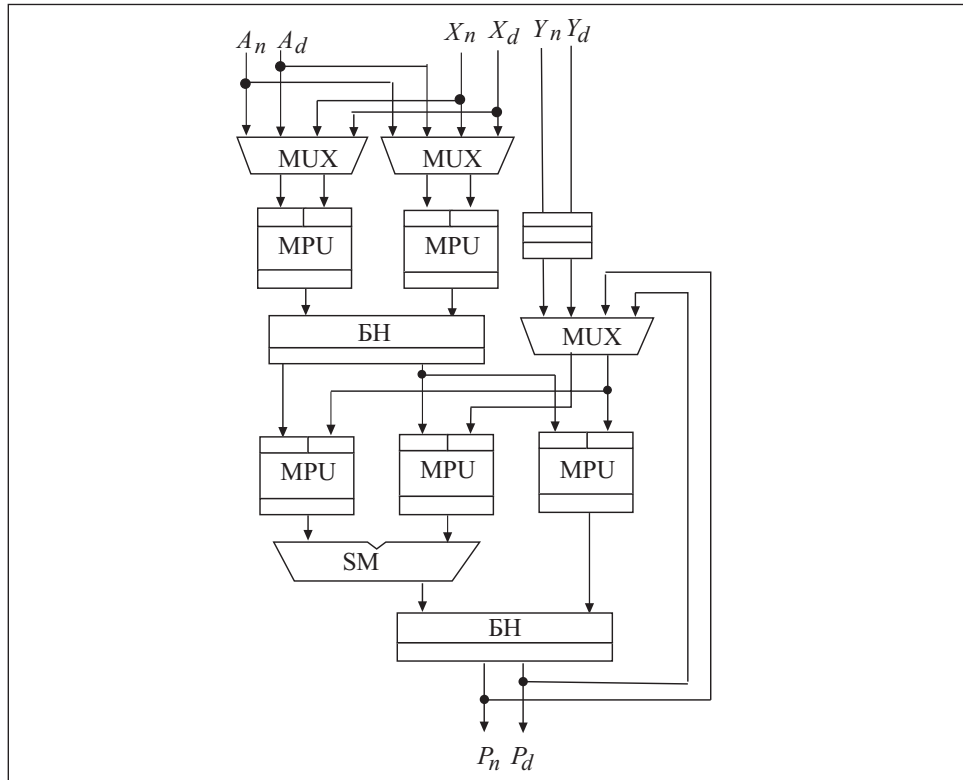


Рис. 2. Структурная схема арифметического устройства блока алгоритма Дарбина: узкие прямоугольники соответствуют ступеням регистров; MUX — мультиплексоры; MPU — блоки умножения; SM — сумматор

полнять с данными в виде таких дробей. Умножение, деление и сложение выполняются по формулам

$$x * y = n_x n_y / (d_x d_y);$$

$$x / y = n_x d_y / (d_x n_y); x + y = (n_x d_y + n_y d_x) / (d_x d_y).$$

Такие вычисления поддерживаются в современных ПЛИС десятками и сотнями блоков умножения с накоплением (например, блок DSP48 в ПЛИС Xilinx Virtex).

Использование этой арифметики для выполнения алгоритма Дарбина обеспечивает малую погрешность вычислений и расширенный динамический диапазон по сравнению с арифметикой целых чисел, а также простоту 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, и такое деление не увеличивает цикл работы процессора АР анализа.

Арифметическое устройство блока алгоритма Дарбина (рис. 2) выполняет такие операции: $P = A + Y$, $P = A X + Y$, $P = A X + P$, $P = A/X$. Блоки нормализации (БН) выполняют нормализацию числителей и знаменателей дробей сдвигом на одинаковое число разрядов влево.

Как видно из рис. 2, длительность итерации процесса накопления суммы парных произведений составляет всего три такта. Это означает, что выполнение основного цикла алгоритма происходит в три—шесть раз быстрее, чем в аналогичном устройстве с плавающей запятой. Кроме того, операция деления выполняется всего за шесть тактов, а не за 17—27 тактов, как в делителе с плавающей запятой [8]. Следовательно, выполнение алгоритма Дарбина происходит, как минимум, в три раза быстрее. Кроме того, такое арифметическое устройство при разрядности числителей и знаменателей 18 занимает 360 эквивалентных логических блоков (ЭКЛБ) и пять 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 четыре блока DSP-48 [8], т.е. его аппаратные затраты существенно больше.

Процессор на основе лестничного фильтра. Фильтр с конечной импульсной характеристикой (КИХ) с коэффициентами a_j 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 соответственно вперед и назад в n -м такте; 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]

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

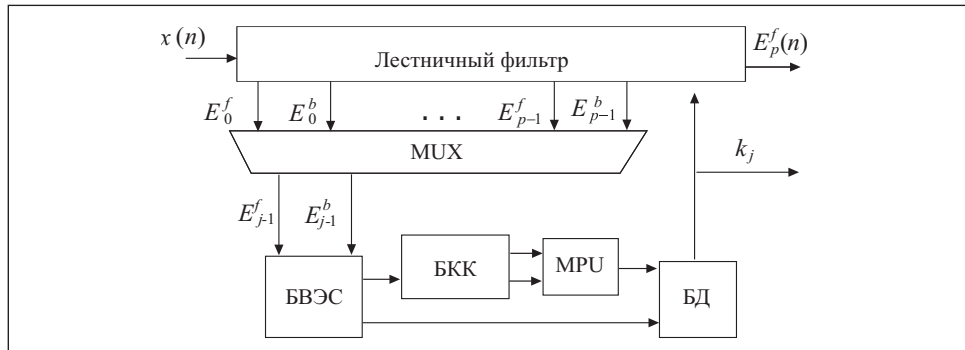


Рис. 3. Структурная схема процессора АР анализа на основе лестничного фильтра

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 решения уравнения Юла—Уолкера, в данной схеме 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 необходимость вычисления знаменателя формулы (4) с повышенной точностью ограничивает применение такого алгоритма.

Процессор АР анализа несложно реализовать в ПЛИС, хотя для этого необходима, по крайней мере, удвоенная разрядность при вычислении коэффициентов (рис. 3).

Блок вычисления энергии сигнала (БВЭС) вычисляет четыре суммы ошибок предсказания вперед и назад в формуле (3) в течение pN тактов, принимая ошибки предсказания из j -й ступени фильтра через мультиплексор MUX. Блок извлечения квадратного корня (БКК), множитель MPU и

блок деления (БД), которые завершают вычисления по формуле (4), могут быть выполнены по параллельно-последовательной схеме. Поскольку число pN сравнительно велико, их невысокое быстродействие незначительно ухудшит быстродействие процессора в целом. В результате при 16-тиразрядных входных данных на одну ступень лестничного фильтра в ПЛИС Xilinx Virtex-4 приходится 90 ЭКЛБ и два блока DSP-48, а на вычислитель коэффициента отражения — около 550 ЭКЛБ и четыре блока DSP-48.

Преимущество процессора на основе лестничного фильтра состоит 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 на каждую ступень фильтра блок вычисления коэффициентов, но тогда аппаратные затраты существенно возрастут.

Процессоры АР анализа, реализованные в ПЛИС. Две описанные выше вычислительные схемы для АР анализа выполнены в виде спецпроцессоров, конфигурированных в ПЛИС фирмы Xilinx. Процессор, реализующий алгоритм Дарбина, имеет арифметико-логическое устройство, выполняющее параллельно операции умножения (деления), накопления в арифметике рациональных дробей с 18-тиразрядными числителем и знаменателем, а также преобразование дроби в целый 18-тиразрядный результат. Операции умножения (деления) выполняются в конвейерном режиме с периодом один такт с латентной задержкой семь тактов. Операция накопления выполняется за четыре такта, т.к. ее результат приходится использовать как исходное данное этой операции.

Результаты сравнения представленных в таблице данных свидетельствуют о том, что процессоры на основе алгоритма Дарбина имеют большую тактовую частоту, меньшие аппаратные затраты и меньший цикл полу-

Результаты реализации процессоров в ПЛИС XC4VSX35 при $q = 10$

Процессор	Аппаратные затраты, ЭКЛБ + DSP-48, при p			Максимальная тактовая частота, МГц, при p			Длительность цикла получения k_1, \dots, k_p тактов при p		
	10	30	90	10	30	90	10	30	90
На основе алгоритма Дарбина	1330+15	1850+35	4753+95	172	169	160	610	1830	5490
На основе лестнично- го фильтра	1416+24	3280+64	8311+184	151	147	145	1650	10950	86850

чения 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 малом значении p и $q < 10$, 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 АР анализа на базе ПЛИС [10—12] при одинаковой пропускной способности предлагаемые процессоры имеют существенно меньшие аппаратные затраты. Так, при одинаковом числе блоков умножения (эквивалентных DSP-48) фильтр описанный в работе [11], имеет почти в 20 раз бóльшие аппаратные затраты, чем процессор, построенный на основе алгоритма Дарбина. В работе [12] описан адаптивный лестничный фильтр на основе логарифмической системы счисления, который реализует ступени фильтра последовательно, в связи с чем имеет существенно меньшее быстродействие.

Выводы. Предложенные два типа процессоров для АР анализа, при реализации в ПЛИС дают возможность в реальном масштабе времени анализировать сигналы, дискретизированные с частотой до 300 МГц, и имеют незначительные аппаратные затраты. Внедрение процессоров позволяет существенно расширить область применения АР анализа, например, при ультразвуковой диагностике в медицине, 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 speed up to hundreds of Megahertz. The processors use permits extending the application area of autoregression analysis, e.g., in ultrasonic diagnosis, radiocommunication.

1. Марпл С. Л. Цифровой спектральный анализ и его приложения. — М. : Мир. — 1990. — 584 с.
2. Уидроу Б., Ступиз С. Адаптивная обработка сигналов. — М. : Радио и связь. 1989. — 440 с.
3. Wojanczyk 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. — P. 95—104.
4. Brent R. P. Old and New Algorithms for Toeplitz Systems //Proc. SPIE. Advanced Algorithms and Architectures for Signal Processing, III. — SPIE; Bellingham, Washington, 1989. — V. 975. — P. 2—9.
5. Клименко О. М., Сергиенко А. М., Шевченко Ю. В., Овраменко С. Г. Конфігурована обчислювальна система для вирішення задач лінійної алгебри // Електрон. моделювання. — 2005. — 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. — P. 453—459.
7. *Ортега Дж.* Введение в параллельные и векторные методы решения линейных систем. — М. : Мир. — 1991. — 365 с.
8. *DS335.* Floating-Point Operator v1.0. // Xilinx Product Specification. — July 27. — 2005. — 20 p. — <http://www.xilinx.com>
9. *Сергиенко А. М.* Применение арифметики рациональных дробей для реализации метода сопряжения градиентов. //Электрон. моделирование. — 2006. — **28**, № 1. — С. 33—41.
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. — P. 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.

Поступила 25.11.09

СЕРГИЕНКО Анатолий Михайлович, канд. техн. наук, ст. науч. сотр. Национального технического университета Украины «КПИ», который окончил в 1981 г. Область научных исследований — отображение алгоритмов в структуры вычислительных средств, цифровая обработка сигналов.

