

УДК 681.324

РАЗРАБОТКА УСТРОЙСТВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ НА ОСНОВЕ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМ, ИСПОЛЬЗУЯ АРИФМЕТИКУ СИСТЕМЫ ОСТАТОЧНЫХ КЛАССОВ

¹Червяков Н.И., ¹Аникуева О.В., ²Аникуев С.В.

¹ФГАОУ ВПО «Северо-Кавказский федеральный университет»,
Ставрополь, e-mail: k-fmf-primath@stavs.ru;

²Филиал ФГБОУ ВО «Московский государственный университет информационных технологий,
радиотехники и электроники», Ставрополь, e-mail: ser-anikuev@yandex.ru

Статья посвящена разработке устройств цифровой обработки сигналов повышенной пропускной способности на основе программируемых логических интегральных схем, используя арифметику системы остаточных классов. Применение программируемых логических интегральных схем позволяет повысить производительность модульных арифметических процессоров цифровой обработки сигналов. Это приводит к слиянию систем остаточных классов и программируемых логических интегральных схем и позволяет достигнуть требуемой точности вычислений, а также позволяет ускорить основные операции цифровой обработки сигналов, таких как свертка, и за счет этого повысить производительность и обработку сигналов в режиме реального времени. Разработанные устройства программируемых логических интегральных схем обеспечивают потенциальные решения для системы остаточных классов, базирующейся на MAC-алгоритмах. В качестве иллюстрации возможностей ПЛИС в данной области рассмотрен пример реализации на программируемой логике ускоренной обработки XML-запросов.

Ключевые слова: система остаточных классов, программируемые логические интегральные схемы, цифровая обработка сигналов, свертка, КИХ-фильтр

DEVELOPMENT OF DEVICES OF DIGITAL PROCESSING OF SIGNALS ON THE BASIS OF PROGRAMMABLE LOGICAL INTEGRATED SCHEMES, USING ARITHMETICS OF SYSTEM OF RESIDUAL CLASSES

¹Chervyakov N.I., ¹Anikueva O.V., ²Anikuev S.V.

¹North Caucasian Federal University, Stavropol, e-mail: k-fmf-primath@stavs.ru;

²Moscow State University of information technologies, radio engineering
and electronic engineers, branch, Stavropol, e-mail: ser-anikuev@yandex.ru

Article is devoted to development of devices of digital processing of signals to the increased capacity on the basis of programmable logical integrated schemes, using arithmetics of system of residual classes. Application of programmable logical integrated schemes allows to increase productivity of modular arithmetic processors of digital processing of signals. It leads to merge of systems of residual classes and programmable logical integrated schemes and allows to reach the demanded accuracy of calculations, and also allows to accelerate the main operations of digital processing of signals, such as convolution, and at the expense of it to increase productivity and processing of signals in real time. The developed devices of programmable logical integrated schemes provide potential decisions for the system of residual classes which is based on MAC-algorithms. By way of illustration opportunities programmable logical integrated schemes is in the field reviewed a realization example on programmable logic of the accelerated processing of XML inquiries.

Keywords: system of residual classes, programmable logical integrated schemes, digital processing of signals, parcel, КИХ-filter

На сегодняшний день с применением программируемой логики успешно реализуются архитектуры, типичные для супер-ЭВМ – лидеров мирового компьютеростроения. Программируемая логика служит прекрасной элементной базой для организации специфических вычислений, в частности нейросетевых, для реализации алгоритмов нечеткой логики и т.п. Отдельного внимания заслуживают широкий выбор использования вычислений: систем остаточных классов, арифметики рациональных дробей, различных систем счисления и много другого. Без особых сложностей на программируемых логических интегральных схемах (ПЛИС)

аппаратно реализуются некоторые функции и алгоритмы с большим числом нестандартных операций, например с применением специфической битовой обработки. На многопроцессорных вычислительных средствах подобные операции реализуются крайне неэффективно [1].

Совместное использование ПЛИС (программируемые логические интегральные схемы) и СОК (система остаточных классов) позволяет ускорить основные операции цифровой обработки сигналов (ЦОС), таких как свертка, и за счет этого повысить производительность и обработку сигналов в режиме реального времени.

Система остаточных классов

Предположим, что система остаточных классов определена рядом L с положительными и попарными относительно главных целыми числами $\{m_1, m_2, \dots, m_i\}$. Эти числа называются модулями. Пусть $M = \prod_{i=1}^L m_i$,

тогда кольцо модуля целых чисел $Z(M)$ представляет собой динамический диапазон системы остаточных классов. Следовательно, любое положительное целое число $X \in Z(M)$ имеет единственное представление $L = [x_1, x_2, \dots, x_L]$, где $x_L = X \bmod m_L$. Учитывая, что два целых числа $X, Y \in Z(M)$, и их соответствующие представления остатками $[x_1, x_2, \dots, x_L]$ и $[y_1, y_2, \dots, y_L]$, то арифметика в системе остаточных классов определена следующим образом:

$$X \circ Y \leftrightarrow [|x_1 \circ y_1|_{m_1}, |x_2 \circ y_2|_{m_2}, \dots, |x_L \circ y_L|_{m_L}], \quad (1)$$

где \circ операция сложения, вычитание или умножение. Таким образом, арифметика СОК определена по прямой сумме кольца модуля целого числа $m_i, Z(m_i)$ и выполняется параллельно без зависимости между остатками чисел [6].

Преобразование от представления остатка $[x_1, x_2, \dots, x_L]$ к X – целому числу основано на китайской теореме об остатке.

Теорема. Если числа a_1, a_2, \dots, a_n попарно взаимно просты, то для любых остатков r_1, r_2, \dots, r_n таких, что $0 \leq r_i < a_i$ при всех $i = 1, 2, \dots, n$, найдётся число N , которое при делении на a_i даёт остаток r_i при всех $i = 1, 2, \dots, n$.

Таким образом, СОК обеспечивает фундаментальную методологию для разделения больших диапазонов динамических систем на множество меньших независимых каналов, по которым вычисления выполняются параллельно [5].

Линейная свертка

Существуют многочисленные алгоритмы ЦОС как общего типа для сигналов в их классической временной форме (телекоммуникации, связь, телевидение и пр.), так и специализированные в самых различных отраслях науки и техники (геоинформатике, геологии и геофизике, медицине, биологии, военном деле, и пр.). Однако все эти алгоритмы, как правило – блочного типа, построены на сколь угодно сложных комбинациях достаточно небольшого набора типовых цифровых операций, к основным из которых относятся свертка, корреляция, фильтрация, функциональные преобразования, модуляция.

Линейная свертка – основная операция ЦОС, особенно в режиме реального времени. Для двух конечных причинных последовательностей $h(n)$ и $y(k)$ длиной соответственно N и K свертка определяется выражением

$$s(k) = h(n) * y(k) = \sum_{n=0}^N h(n)y(k-n), \quad (2)$$

где $*$ – символьные обозначения операции свертки. Как правило, в системах обработки одна из последовательностей $y(k)$ представляет собой обрабатываемые данные (сигнал на входе системы), вторая $h(n)$ – оператор (импульсный отклик) системы, а функция $s(k)$ – выходной сигнал системы. В компьютерных системах с памятью для входных данных оператор $h(n)$ может быть двусторонним от $-N_1$ до $+N_2$, например – симметричным $h(-n) = h(n)$, с соответствующим изменением пределов суммирования в (2), что позволяет получать выходные данные без сдвига фазы частотных гармоник относительно входных данных. При строго корректной свертке с обработкой всех отсчетов входных данных размер выходного массива равен $K + N_1 + N_2 - 1$ и должны задаваться начальные условия по отсчетам $y(k)$ для значений $y(0-n)$ до $n = N_2$ и конечные для $y(K+n)$ до $n = N_1$.

Рассмотрим пример выполнения дискретной свертки в среде программирования MATHCAD.

Преобразование свертки однозначно определяет выходной сигнал для установленного значения входного сигнала при известном импульсном отклике системы. Обратная задача деконволюции – определение функции $y(k)$ по функциям $s(k)$ и $h(n)$, имеет решение только при определенных условиях. Это объясняется тем, что свертка может существенно изменить частотный спектр сигнала $s(k)$ и восстановление функции $y(k)$ становится невозможным, если определенные частоты ее спектра в сигнале $s(k)$ полностью утрачены.

КИХ-фильтры

Одним из видов линейных цифровых фильтров, характерной особенностью которого является ограниченность по времени его импульсной характеристики (с какого-то момента времени она становится точно равной нулю) является фильтр с конечной импульсной характеристикой (КИХ-фильтр).

Из всего вышесказанного следует, что КИХ-фильтры могут быть осуществлены с помощью свертки сигнала, т.е.

$$y(n) = \sum_{k=0}^{N-1} h_k x(n-k), \quad (3)$$

где $x(n)$ и $y(n)$ – вход и выход фильтра; h_k – коэффициенты фильтра; N – длина фильтра.

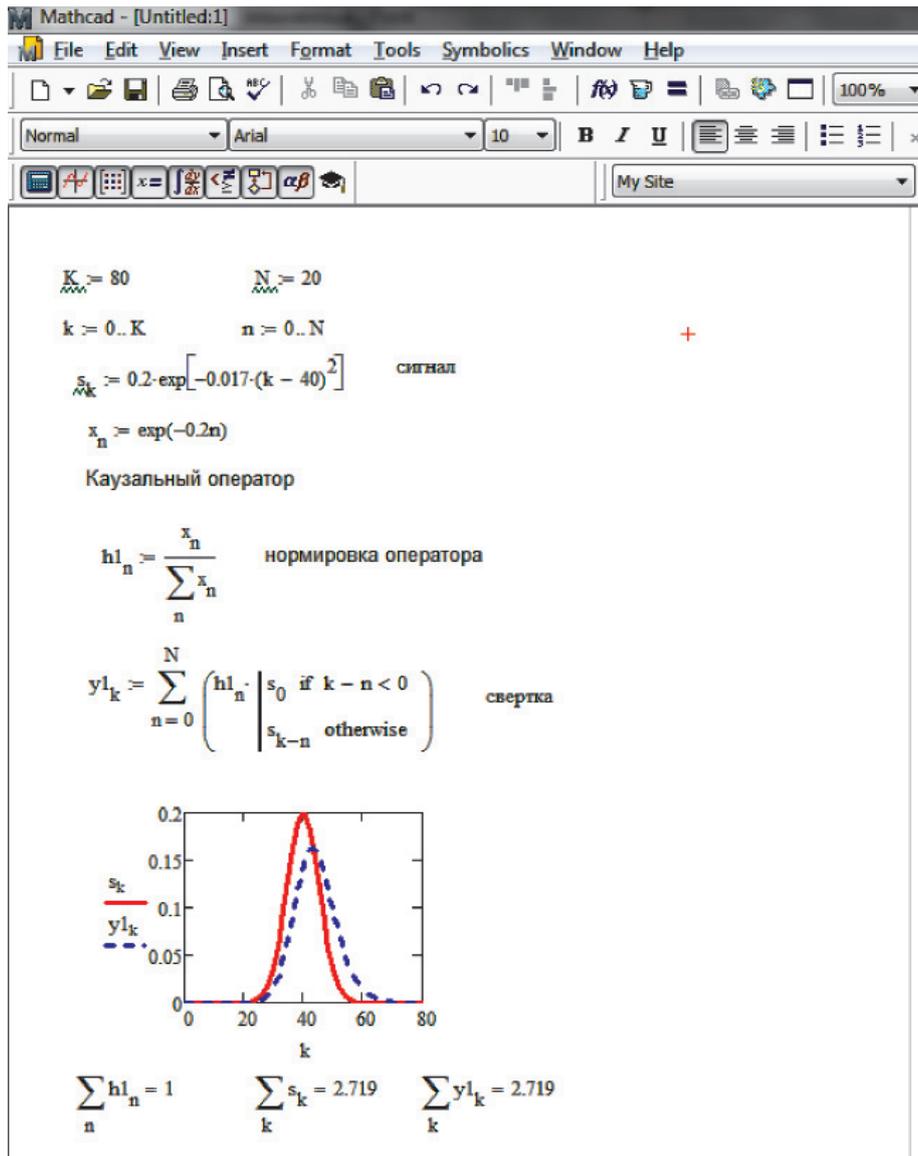


Рис. 1. Пример выполнения дискретной свертки с каузальным оператором в среде программирования MATHCAD

К основным операциям фильтрации информации относят операции сглаживания, прогнозирования, дифференцирования, интегрирования и разделения сигналов, а также выделение информационных (полезных) сигналов и подавление шумов (помех). Основными методами цифровой фильтрации данных являются частотная селекция сигналов и оптимальная (адаптивная) фильтрация.

Итак, если (3) перевести в СОК, то уравнение примет вид

$$|y(n)|_{(m-k)} = \left| \sum_{k=0}^{N-1} h_{k=0}^j x_j(n-k) \right|_{m_j},$$

$$j = 1, 2, \dots, L, \quad (4)$$

где h_k^j и $x_j(n-k)$ обозначают остатки h_k и $x_j(n-k)$ по модулю m_j . Каждый модуль m_j в (4) для $j = 1, 2, \dots, L$ функционирует независимо. Таким образом, для реализации каждого фильтра по модулю необходимо N множителей и $N-1$ сумматоров. Использование распределенной арифметики позволяет уменьшить количество конфигурируемых логических блоков программируемых логических интегральных схем за счет использования системы остаточных классов [6].

На рис. 3 показана блок-схема фильтра, который в таком виде широко известен как трансверсальный (z -задержка на один интервал дискретизации).

Разработка устройств ЦОС повышенной пропускной способности на основе ПЛИС

Устройства программируемых логических интегральных схем обеспечивают потенциальные решения для системы остаточных классов, базирующейся на MAC-алгоритмах. Эти устройства состоят из логических элементов. В зависимости от группы, каждый логический элемент включает один или более переменных на входе программируемых логических интегральных схем.

Сложные логические операции и управление были первыми задачами, для решения которых, собственно, и создавались ПЛИС. В качестве иллюстрации возможностей ПЛИС в данной области рассмотрим пример реализации на программируемой логике ускоренной обработки XML-запросов.

В настоящее время в информационных системах сбора и обработки больших объемов данных для хранения информации в электронном виде стандартом является формат XML (Extensible Markup Language). Для навигации по XML-дереву и обращения к его элементам разработан специальный язык – XPath (XML Path Language). Суть задачи в упрощенном виде может быть сформулирована следующим образом. На сервере хранится большой документ в формате XML. От удаленных пользователей поступает интенсивный поток XPath-запросов. Необходимо для каждого запроса в реальном масштабе времени по известному алгоритму вычислить путь адресации, выбрать из имеющейся базы необходимые данные и сформировать результирующий XML-документ.

Данный процесс можно разделить на два последовательных этапа – синтаксический анализ и фильтрацию. Известное решение данной проблемы основано на кластеризации запросов, то есть на их группировке по сходным признакам с целью сузить область поиска в пространстве элементов. Но реализация данного подхода на кластерной архитектуре приводит к интенсивным обменам между процессорными элементами. Возникает противоречие: параллельная по своей природе задача обработки большого количества независимых запросов становится тесно связанной, из-за чего теряются все преимущества кластеризации. Ситуация выглядит совершенно иначе при решении той же задачи на программируемой логике.

Для реализации вышеупомянутого подхода в реконфигурируемой структуре ПЛИС синтезируются цифровые автоматы, на состояния которых отображаются элементы пользовательских запросов. Собственно кластеризация осуществляется

накапливанием многих запросов в одном цифровом автомате путем анализа элементов XPath-сообщений и обнаружения схожих путей адресации. За счет однообразия, а также использования априорно известной информации о запросах, анализ и кластеризацию удастся выполнить до начала процесса фильтрации. В результате синтаксический анализатор и фильтр, представляющий собой большую комбинационную схему, также могут функционировать как две независимые ступени конвейера. Следует заметить, что схема фильтра, синтезируемая в ПЛИС, оптимизируется специальными программными средствами под конкретный XML-документ, хранящийся на сервере. Таким образом, достигается беспрецедентная степень адаптации вычислительной структуры к решаемой задаче, принципиально невозможная для микропроцессорных систем. Как следствие, в практических разработках удастся достичь ускорения более чем на порядок по сравнению с традиционными подходами [4].

Заключение

Применение программируемых логических интегральных схем позволяет повысить производительность модульных арифметических процессоров цифровой обработки сигналов. Это приводит к слиянию систем остаточных классов и программируемых логических интегральных схем и позволяет достигнуть требуемой точности вычислений [2].

Совместное использование системы остаточных классов и современных программируемых логических интегральных схем обеспечивает высокую точность и пропускную способность выполнения двух операций: сложение и умножение, используемых при цифровой обработке сигналов реального времени.

Список литературы

1. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с.
2. Зотов В.Ю. Проектирование цифровых устройств на основе ПЛИС фирмы XILINX в САПР WebPACK ISE. Горячая Линия – Телеком, 2003.
3. Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А. Модулярные параллельные вычислительные структуры нейропроцессорных систем. – М.: ФИЗМАТЛИТ, 2003. – 288 с.
4. Abhishek Mitra, Marcos R. Vieira, Petko Bakalov, Walid Najjar, Vassilis J. Tsotras Boosting XML Filtering with a Scalable FPGA-based Architecture // 4th Biennial Conference on Innovative Data Systems Research (CIDR) January 4–7, 2009, Asilomar, USA.
5. Actel Corporation, ProASIC 500K Family Data Sheet, 2000.
6. Garcia A., Meyer-Base U., Lloris A., Taylor F. RNS Implementation of FIR Filters based on Distributed Arithmetic using Field-Programmable Logic. Proc. IEEE Int. Symp. on Circuits and Systems. – Orlando, 1999. – vol.1. – P. 486–489, Jun.