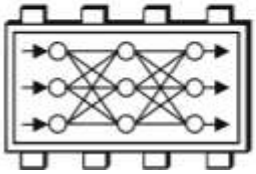
ISSN 1999-8554

НЕЙРОКОМПЬЮТЕРЫ



разработка применение

Международный научно-технический журнал
Включен в перечень ВАК

№ 10, 2016 г.

По материалам II Международной конференции «Параллельная компьютерная алгебра и ее приложения в новых инфокоммуникационных системах» (г. Ставрополь, 24–25 октября 2016 г.)

Содержание

Теория нейронных сетей, нейро-нечеткие модели и сети

Contents

Neuromathematic and Intellectual Computation

Theory of Neural Nets

Все статьи, представленные в данном выпуске журнала, соответствуют номенклатуре специальностей научных работников (Приказ Минобрнауки РФ от 11.08.2009 № 294) по отраслям технических, физико-математических и медико-биологических наук.

"Neirokompiutery" (Neurocomputers) is a scientific and technical journal devoted to development and applications of artificial neural networks and neurocomputers. Established in 1999.

Необходимую информацию Вы найдете на нашем сайте www.radiotec.ru

РАДИОТЕХНИКА

Учредитель: ЗАО «Издательство «Радиотехника». Лицензия – ЛР № 0652229 от 20 июня 1997 г.
Свидетельство о регистрации средств массовой информации ПИ № 77-1109 от 18 ноября 1999 г.

Сдано в набор 17.10.2016. Подписано в печать 16.10.2016. Печ. л. 9. Тираж 250 экз. Изд. № 58.
107031, Москва, К-31, Кузнецкий мост, д. 20/6. Тел./Факс (7-495)621-48-37. E-mail: info@radiotec.ru
Дизайн и допечатная подготовка: ООО «САЙНС-ПРЕСС»
Отпечатано в ФГУП Издательство «Известия» УД ПРФ. 127254, Москва, ул. Добролюбова, д. 6. Контактный телефон: 650-38-80. Заказ № .

ISSN 1999-8554

УДК 004.896

Разработка нового нейросетевого метода вычисления модульного умножения в системе остаточных классов*

Н.И. Червяков – *д.т.н., профессор, зав. кафедрой прикладной математики и математического моделирования, Институт математики и естественных наук, Северо-Кавказский федеральный университет (г. Ставрополь).*
E-mail: k-fmf-primath@stavsu.ru
М.Г. Бабенко – *к.ф-м.н., доцент, кафедра прикладной математики и математического моделирования, Институт математики и естественных наук, Северо-Кавказский федеральный университет (г. Ставрополь)*
E-mail: mgbabenko@ncfu.ru
А.Н. Черных – *профессор, директор лаборатории параллельных вычислений, Центр научных исследований и высшего образования Энсенады (Мексика)*
E-mail: chernykh@cicese.mx
В.А. Кучуков – *аспирант, кафедра прикладной математики и математического моделирования, Институт математических и естественных наук, Северо-Кавказский федеральный университет (Ставрополь)*
E-mail: viktor-kuchukov@yandex.ru
М.А. Дерябин – *ассистент, кафедра прикладной математики и математического моделирования, Институт математических и естественных наук;*
мл. науч. сотрудник, лаборатория математического моделирования и теоретико-числовых систем, Северо-Кавказский федеральный университет (Ставрополь)
E-mail: maderiabin@ncfu.ru
Н.Н. Кучукова – *аспирант, кафедры прикладной математики и математического моделирования Институт математических и естественных наук, Северо-Кавказский федеральный университет (Ставрополь)*
E-mail: knn.storage@yandex.ru

Предложен новый нейросетевой метод вычисления модульного умножения, основанный на нахождении приближенным методом остатка от деления произведения на заданный модуль в системе остаточных классов (СОК) и нейронной сети конечного кольца. Показано, что использование приближенного метода для нахождения остатка от деления не требует дорогостоящих модульных операций, которые заменяются быстрыми битовыми операциями сдвига вправо и взятия младших бит; эффективная реализация предложенного метода с использованием нейронной сети конечного кольца позволяет увеличить скорость выполнения операций.

Ключевые слова: *система остаточных классов, приближенный метод, нейронная сеть конечного кольца, алгоритм Монтгомери, ПЛИС.*

In the paper, we propose a new neural network method of modular multiplication computation, based on Residue Number System. We use an approximate method to find the approximate method a residue from division of a multiplication on the given module. We substitute expensive modular operations, by fast bit right shift operations and taking low bits.

Keywords: *Residue Number System, Approximate method, Neural network of finite field, Montgomery algorithm, FPGA.*

Нахождения остатка от деления числа на фиксированный модуль является базовой операцией при реализации большого числа алгоритмов защиты информации [1–3], цифровой обработки сигналов [4], систем беспроводной связи [5] и т.д. При разработке аппаратных решений для современных информационных систем особое внимание уделяется техническим характеристикам: скорости работы, площади и т.д. Использование системы остаточных классов (СОК) позволяет выполнять сложение и умножение чисел по параллельным вычислительным каналам без переноса разрядов между каналами, что позволяет повысить скорость выполнения арифметических операций.

* Работа представлена на II Международной конференции «Параллельная компьютерная алгебра и ее приложения в новых инфокоммуникационных в системах», г. Ставрополь, 24–25 октября 2016 г.

Цель работы – разработка нового нейросетевого метода вычисления модульного умножения, основанный на нахождении остатка от деления произведения на заданный модуль в системе остаточных классов.

Благодаря использованию приближенного метода из работы [6], для нахождения остатка от деления не требуется выполнения дорогостоящих модульных операций, которые заменяются быстрыми битовыми операциями сдвига вправо и взятия младших бит.

Система остаточных классов

В основе СОК лежат теория сравнений и Китайская теорема об остатках. Арифметические операции (сложение, умножение) в СОК выполняются параллельно независимо по L каналам и без переноса между вычислительными каналами. Заметим, что в каждом отдельном вычислительном канале СОК ведется работа с числами меньшей размерности – с остатками от деления числа на модуль СОК, что ведет к уменьшению числа переносов и повышению надежности и скорости выполнения арифметических операций с числами. Система остаточных классов задается попарно взаимно простыми числами $m_1, m_2, m_3,..., m_L$, называемыми модулями. Диапазон СОК вычисляется по формуле $M = \prod_{i=1}^{L} m_i$. Любое целое число A, принадлежащее отрезку $[0, M-1]$, однозначно представляется в СОК кортежем $(a_1, a_2, ..., a_L)$, где для всех $i = \overline{1,n}$ выполняется сравнение $a_i = A \bmod m_i$.

Согласно Китайской теореме об остатках, число A может быть восстановлено с использованием формулы

$$A = \left| \sum_{i=1}^{L} \frac{M}{m_i} \left| M_i^{-1} \right|_{m_i} a_i \right|_M , \quad (1)$$

где $M_i = \frac{M}{m_i}$ и $\left| M_i^{-1} \right|_{m_i}$ – мультипликативная инверсия M_i относительно m_i.

Если разделить (1) на константу M, то получим приближенное значение

$$\frac{A}{M} = \left| \sum_{i=1}^{L} \frac{\left| M_i^{-1} \right|_{m_i}}{m_i} a_i \right|_1 = \left| \sum_{i=1}^{L} k_i a_i \right|_1 , \quad (2)$$

где $k_i = \frac{\left| M_i^{-1} \right|_{m_i}}{m_i}$ – константы выбранной системы; a_i – остатки числа A, представленного в СОК, при этом значение выражения (2) будет в интервале $[0,1)$.

Конечный результат суммы определяется после суммирования и отбрасывания целой части числа с сохранением дробной части суммы. Дробная часть может быть записана как $A \bmod 1$, потому что $A = \lfloor A \rfloor + A \bmod 1$.

Из формулы (2) следует, что перевод числа A из СОК в позиционную систему счисления выполняется по формуле

$$A = M \left| \sum_{i=1}^{L} k_i a_i \right|_1 . \quad (3)$$

Обзор работ вычисления модульного умножения в СОК

Алгоритмы вычисления модульного умножения можно разбить на два класса: 1) использующие операцию нахождения остатка от деления произведения на фиксированный модуль; 2) использующие операции нахождения остатка в процессе вычисления умножения. Рассмотрим алгоритмы,

позволяющие выполнить модульное умножение без использования операции нахождения остатка от деления в общем случае, но требующие предвычисления и хранения констант.

Алгоритм Монтгомери для модульного умножения без деления предложен в работе [8]. Эффективная систолическая реализация алгоритма Монтгомери рассмотрена в работе [2] и представлена далее следующим алгоритмом:

$\mathrm{MMM}^4(A_R, B_R, p)$

1: if $(b_0 = 1)$ then $B_R = B_R - 1$; $T^{(0)} = A_R$;

2: for $i = 0$ to $n+1$;

3: for $j = 0$ to $n+2$;

4: $t' = \left(t_{i,j} \oplus C0_{i,j}\right) \oplus \left(a_i \wedge b_j\right)$;

5: $C0_{i,j+1} = \left[\left(t_{i,j} \oplus C0_{i,j}\right) \wedge \left(a_i \wedge b_j\right)\right] \vee \left(t_{i,j} \wedge C0_{i,j}\right)$;

6: $t_{i+1,j-1} = t' \oplus \left[\left(t_{i,0} \wedge n_j\right) \oplus \left(D0_{i,j} \vee D1_{i,j}\right)\right]$;

7.1: $D0_{i,j+1} = t' \wedge \left[\left(t_{i,0} \wedge n_j\right) \oplus \left(D0_{i,j} \vee D1_{i,j}\right)\right]$;

7.2: $D1_{i,j+1} = \left(t_{i,0} \wedge n_j\right) \wedge \left(D0_{i,j} \vee D1_{i,j}\right)$;

где p – n-битный нечетный модуль; $R = 2^{n+2}$; $A_R = AR \bmod p$, $B_R = BR \bmod p$ и результат $T = \left[t_{n+2,n}, ..., t_{n+2,0}\right]$ удовлетворяют условию $A_R, B_R, T \in [0, 2p)$; в начале цикла $T^{(0)}$, $C0$, $D0$ и $D1$ равны нулю.

В работе [2] говорится, что критический путь задержки данного алгоритма равен $3T_{\mathrm{XOR}}$, где T_{XOR} – задержка двухвходового элемента XOR.

Алгоритм модульного умножения Монтгомери в двух полях и обобщение алгоритма конвертации набора модулей для двух полей показаны в работе [7]. Алгоритм умножения, где $\oplus$ – операция сложения/вычитания в двух полях и $\otimes$ – операция умножения в двух полях, задан в следующем виде:

1: $s_\tau = A_\tau \otimes B_\tau$;

2: $c_\beta = s_\beta \otimes \left(-p^{-1}\right)_\beta$;

3: $c_\alpha = c_\beta$ (конвертация набора модулей);

4: $u_\alpha = c_\alpha \otimes p_\alpha$;

5: $v_\alpha = s_\alpha \oplus u_\alpha$;

6: $T_\alpha = v_\alpha \otimes Q_\alpha^{-1}$;

7: $T_\beta = T_\alpha$ (конвертация набора модулей).

В данном алгоритме заданы наборы модулей $\alpha = \left(p_1, p_2, ..., p_L\right)$ и $\beta = \left(q_1, q_2, ..., q_L\right)$, так что $\gcd\left(p_i,\ q_j\right) = 1,\ \forall i,\ j \in [1, L]$. На вход поступают числа A и B, представленные двумя наборами модулей СОК, то есть A_τ и B_τ, константа $\left(-p^{-1}\right)_\beta$ в СОК β, константы Q_α^{-1} и p_α в СОК α, где $A, B < 2p$, $Q = \prod_{i=1}^{L} q_i$. На выходе данного алгоритма получаем T_τ, где $T < 2p$ и $T \equiv ABQ^{-1} \bmod p$. При выполнении умножения дважды необходимо переходить между системами оснований α и β.

Важным аспектом приведенного алгоритма является то, что он сводится к простому умножению с накоплением, где длина слова равна длине модуля n. Это позволяет реализовать устройство с полностью параллельной архитектурой, где каждый модуль сопоставляется одному из модулей системы оснований СОК.

Эффективная реализация вычисления остатка от деления в СОК

Рассмотрим подход, когда вычисление модульного умножения сводится к двум этапам: 1) нахождению произведения двух чисел $P = A \times B$; 2) нахождению остатка от числа по модулю p, то есть $P \bmod p$.

Эффективная реализация операции нахождения остатка от деления числа A по модулю простого числа p в СОК может быть получена из формулы (2):

$$\frac{A}{M} = \left| p \sum_{i=1}^{L} \frac{\left| M_i^{-1} \right|_{m_i}}{p m_i} a_i \right|_1 = \left| p \sum_{i=1}^{L} \overline{\overline{k_i}} a_i \right|_1 = \left| \sum_{i=1}^{L} k_i a_i \right|_1 = p \sum_{i=1}^{L} \overline{\overline{k_i}} a_i - \left[\sum_{i=1}^{L} k_i a_i \right], \tag{4}$$

где $\overline{\overline{k_i}} = \dfrac{\left| M_i^{-1} \right|_{m_i}}{p m_i}$ и $[x]$ – целая часть от числа x.

Из формулы (4) следует, что A можно представить в виде

$$A = \left(p \sum_{i=1}^{L} \overline{\overline{k_i}} a_i - \left[\sum_{i=1}^{L} k_i a_i \right] \right) M. \tag{5}$$

Используя формулу (5), вычислим значение $\dfrac{A}{p}$, получим

$$\frac{A}{p} = \left(\sum_{i=1}^{L} \overline{\overline{k_i}} a_i - \left[\sum_{i=1}^{L} k_i a_i \right] \frac{1}{p} \right) M. \tag{6}$$

Следовательно, значение $A \bmod p$ можно найти по формуле

$$A \bmod p = A - \left\lfloor \frac{A}{p} \right\rfloor p = A - \left\lfloor \left(\sum_{i=1}^{L} \overline{\overline{k_i}} a_i - \left[\sum_{i=1}^{L} k_i a_i \right] \frac{1}{p} \right) M \right\rfloor p. \tag{7}$$

Отсутствие необходимости производить вычисления с целыми частями вещественных чисел делает возможным произвести переход от вычислений с дробными частями к целочисленным вычислениям. Это можно осуществить следующим образом:

каждую вещественную константу умножить на 2^N, где N – число двоичных знаков после запятой, обеспечивающее необходимый уровень точности вычислений;

округлить каждое полученное число вверх, то есть до следующего целого числа;

производить все вычисления в кольце классов вычетов по модулю 2^N.

Используя оценку N из работы [8], получим, что $N = \left\lceil \log_2 \left(\sum_{i=1}^{L} (m_i - 1) \right) M \right\rceil$. Учитывая тот факт, что в ПЛИС работа с вещественными числами является дорогостоящей, то перейдем от вещественных чисел к целым, умножив $\overline{\overline{k_i}}$ и k_i на 2^N. Тогда формула (7) примет следующий вид:

$$A \bmod p = A_{RNS} - K \cdot p_{RNS}, \; K = \left\lfloor \left(\sum_{i=1}^{L} \tilde{k}_i a_i - \mu \left\lfloor \sum_{i=1}^{L} \overline{k_i} a_i / 2^N \right\rfloor \right) \frac{M}{2^N} \right\rfloor, \tag{8}$$

где $\tilde{k}_i = \left\lfloor 2^N \dfrac{\left| M_i^{-1} \right|_{m_i}}{m_i p} \right\rfloor$; $\overline{k_i} = \left\lceil 2^N \dfrac{\left| M_i^{-1} \right|_{m_i}}{m_i} \right\rceil$; $\mu = \left\lfloor \dfrac{2^N}{p} \right\rfloor$, $N = \left\lceil \log_2 \left(\sum_{i=1}^{L} (m_i - 1) \right) M \right\rceil$.

В результате работы алгоритма вычисления $T = A \bmod p$ по формуле (8) получится итог, удовлетворяющий неравенству $0 \le T < 2p$, аналогично алгоритму Монтгомери из работ [1–3, 7–9].

Эффективная нейросетевая реализация

Аппаратная реализация нейронной сети (НС) позволяет проводить массовое параллельное выполнение простейших операций, причем, чем большая степень параллельности вычислений достигается, тем лучше. ПЛИС – идеальная элементная база для реализации таких параллельных структур, как нейронная сеть. ПЛИС позволяет реализовать n параллельно работающих нейросетей, при этом обмен данными между нейронами осуществляется внутри той же ПЛИС с высокой скоростью. Кроме того, использование нейронной сети конечного кольца (НСКК) сокращает число каскадируемых схем, что позволяет повысить скорость обработки информации. При ограничении на арифметические операции арифметика НСКК обеспечивает их более эффективную реализацию по сравнению с двоичной арифметикой. С появлением ПЛИС большой площади стало возможным построение единой системы обработки данных на одном кристалле, что открывает новые возможности применения НСКК в системах обработки и защиты больших объемов графической информации.

Нейронная сеть представляет собой высокопараллельную динамическую систему с топологией направленного графа, которая может получать выходную информацию посредством реакции ее состояния на входные воздействия. Узлами в НС называются процессорные элементы и направленные графы [10, 11]. Структура алгоритма обработки данных, представленных в системе остаточных классов, также же как и структура НС, обладает естественным параллелизмом, что позволяет использовать НС в качестве формального аппарата описания алгоритма.

С этой точки зрения алгоритмы модулярных вычислений соответствуют алгоритмам вычислений с помощью базовых процессорных элементов (искусственных нейронов). Искусственные НС и основные модулярные структуры представляют собой коннекционные устройства, полученные последовательным соединением между собой базовых элементов. Нейронные и модулярные образования будут послойно определены, если задан алгоритм соединения базовых элементов.

Рассмотрим общий подход применения НС к вычислениям в конечных кольцах и формированию модели НСКК. При этом нейроны являются арифметическими элементами, которые имеют характеристики оператора по модулю, а не обычные нелинейные функции активации, применяемые при обучении НС. Анализ арифметики конечного кольца показал, что вычислительная модель, основанная на итеративном механизме сокращения по модулю, является основной операцией при модулярной обработке данных.

Общая интерпретация архитектуры НС – это массово-параллельная взаимосвязанная сеть простых элементов и ее иерархическая организация. Структура НС в некоторой степени моделирует биологическую нервную систему. Мощностью НС является ее способность использования начальной базы знаний для решения существующей проблемы. Все нейроны работают конкурентно, а на непосредственные вычисления влияют знания, зашифрованные в связях сети [11, 12, 14, 16].

Взаимодействие нейронов учитывается в трехуровневой иерархии сети, состоящей из следующих слоев [11, 13, 15]: 1) отображение параметра (этот слой содержит остаток, связанный со взвешенной величиной каждого вычислительного разряда); 2) отображение разрядного вычисления (определяет функцию конечного кольца, применяемую к каждому вычислительному разряду); 3) отображение операции конечного кольца (определяет основные операции, использующиеся для реализации арифметики конечного кольца).

Рассмотрим архитектуру НСКК из работы [17]. На основании вычислительной модели конечного кольца, главным оператором в которой является оператор извлечения отдельных разрядов двоичного представления преобразуемого числа, могут быть построены многослойные подсети. Структура подсети показана на рис. 1,

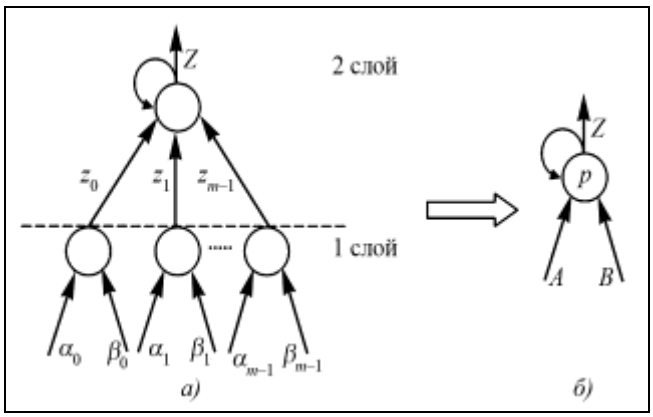


Рис. 1. Структура подсети (*а*) и ее символьное обозначение (*б*)

где синаптические веса равны $w_i = |2^i|_p$, $i = 0, 1, ..., n-1$.

1. Результат операций преобразования двоичного числа к остатку, умножения, сложения, вычисляемых при помощи НСКК, является функцией суммы взвешенных входных разрядов.

2. Результат вычисления определяется положительной логикой. Конечный результат НСКК будет иметь устойчивую форму.

Моделирование проведено на плате Kintex7 XC7K70T.

Из рис. 2 видно, что предложенный метод нахождения остатка от числа позволяет получить выигрыш во времени в среднем на 75%, а по площади – в среднем на 80% относительно модифицированного метода из работы [2].

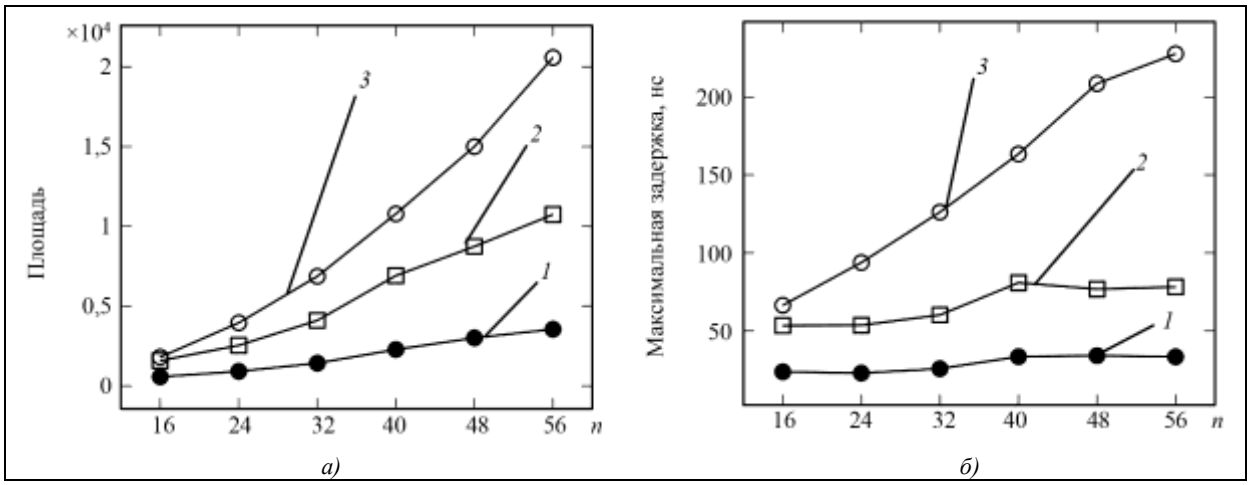


Рис. 2. Технические параметры реализации архитектур модульного умножения на ПЛИС: *1* – предложенный метод; *2* – метод модульного умножения Монтгомери [2]; 3 – метод модульного умножения Монтгомери [7]

- Предложенный новый нейросетевой метод вычисления модульного умножения, для нахождения остатка от деления не требует дорогостоящих модульных операций, которые заменяются быстрыми битовыми операциями сдвига вправо и взятия младших бит. В результате проведенного моделирования на плате Kintex7 XC7K70T предложенный метод позволяет получить выигрыш во времени в среднем на 75%, а по площади – в среднем на 80% относительно модифицированного метода из работы [2], что делает его более применимым для аппаратной реализации криптографических примитивов, построенных над конечным полем.

Работа выполнена при поддержке базовой части государственного задания СКФУ №2563. Работа выполнена при поддержке стипендии Президента РФ молодым ученым и аспирантам СП-1215.2016.5.

Литература

1. *Manochehri K., Sadeghian B., Pourmozafari S.* A modified radix-2 Montgomery modular multiplication with new recoding method // IEICE Electronics Express. 2010. V. 7. № 8. P. 513–519.
2. *Choi S.-H., Lee K.-J.* New systolic modular multiplication architecture for efficient Montgomery multiplication // IEICE Electronics Express. 2014, V. 12. № 2. P. 2014–1051.
3. *Choi S.-H., Lee, K.-J.* Enhancement of a modified radix-2 Montgomery modular multiplication // IEICE Electronics Express. 2014. V. 11. № 19. P. 2014–0782.
4. *Zivaljevic D., Stamenkovic N., Stojanovic V.* Digital filter implementation based on the RNS with diminished-1 encoded channel // Telecommunications and Signal Processing (TSP), 35th International Conference // IEEE. 2012. July. P. 662–666.
5. *Yatskiv V., Jun S., Yatskiv N., Sachenko A., Osolinskiy O.* Multilevel method of data coding in WSN. // IDAACS, IEEE 6th International Conference. 2011. P. 863–866.
6. *Chervyakov N.I., Babenko M.G., Lyakhov P.A., Lavrinenko I.N.* An Approximate method for comparing modular numbers and its application to the division of numbers in residue number systems // Cybernetics and Systems Analysis. 2014. V. 50. № 6. P. 977–984.

7. *Schinianakis D., Stouraitis T.* Multifunction residue architectures for cryptography // Circuits and Systems I: Regular Papers. IEEE Transactions. 2014. V. 61. № 4. P. 1156–1169.
8. *Montgomery P. L.* Modular multiplication without trial division // Mathematics of computation, 1985. V. 44. № 170. P. 519–521.
9. *Schinianakis D., Stouraitis T.* A RNS Montgomery multiplication architecture // Circuits and Systems (ISCAS). IEEE International Symposium. 2011. P. 1167–1170.
10. *Галушкин А.И.* Теория нейронных сетей. М.: ИНРЖР. 2000. 416 с.
11. *Червяков Н.И., Сахнюк П.А., Шапошников В.А., Макоха А.Н.* Нейрокомпьютеры в остаточных классах. Кн. 11. М.: Радиотехника. 2003. 272 с.
12. *Zhang D.* Parallel designs for Chinese remainder conversion // International Conference on Parallel Processing – ICPP. 1987. P. 557–559.
13. *Zhang D.* Parallel VLSI Neural System Designs. Berlin, Germany: Springer-Verlag. 1998. 257 p.
14. *Zang D. Jullien G.A., Miller W.C.* A neural-like approach to finite ring computation // IEEE Transactions on Circuits and Systems. 1990. V. 37. № 8. P. 1048–1052.
15. *Zhang D., Jullien G.A., Miller W.C.* VLSI implementations of neural-like networks for finite ring computations // Proceedings of the 32nd Midwest Symposium on Circuits and Systems. 1989. V.1. P. 485–488.
16. Патент № 2317584 РФ. Конвейерная нейронная сеть конечного кольца / *Н.И. Червяков.* 2008.
17. Патент № 2279132 РФ. Нейронная сеть конечного кольца / *Н.И. Червяков, В.А. Галкина, Ю.А. Стрекалов, С.В. Лавриненко*. 2006.

Поступила 11 октября 2016 г.

Development of a new method for computing modular multiplication in the residue number system

N.I. Chervyakov – *Dr.Sc. (Eng.), Professor, Head of Department of the Applied Mathematics and Mathematical Modeling, Institute of Mathematics and Natural Sciences, North Caucasus Federal University (Stavropol)*
E-mail: k-fmf-primath@stavsu.ru
M.G. Babenko – *Ph.D. (Phys.-Math.), Associate Professor, Department of the Applied Mathematics and Mathematical Modeling, Institute of Mathematics and Natural Sciences, North Caucasus Federal University (Stavropol)*
E-mail: k-fmf-primath@stavsu.ru
A.N. Tchernykh – *PhD, Full Professor in the Computer Science Department, Head of the Parallel Computing Laboratory in CICESE Research Center, Ensenada, Baja California, Mexico*
E-mail: chernykh@cicese.mx
V.A. Kuchukov – *Post-graduate Student, Department of Applied Mathematics and Mathematical Modeling, Institute of Mathematics and Natural Sciences, North Caucasus Federal University (Stavropol)*
E-mail: viktor-kuchukov@yandex.ru
M.A. Deryabin - *Lecturer Assistant, Department of Applied Mathematics and Mathematical Modeling; Junior Research Scientist, Laboratory of Mathematical Modeling and Theoretical-Numeric Systems, Institute of Mathematics and Natural Sciences, North Caucasus Federal University (Stavropol)*
E-mail: maderiabin@ncfu.ru
N.N. Kuchukova - *Post-graduate Student, Department of Applied Mathematics and Mathematical Modeling, Institute of Mathematics and Natural Sciences, North Caucasus Federal University (Stavropol)*
E-mail: knn.storage@yandex.ru

The paper proposes a new neural network method for computing the modular multiplication based on finding by the approximate method remainder of the division by a given module in residue number system (RNS) and the neural network of the final ring. RNS allows to compute addition and multiplication in parallel computing channels without caries between the channels, which increases the speed of arithmetic operations. The paper reviews the algorithms for computing modular multiplication in RNS, that allow to perform modular multiplication without using the operation of finding the remainder of division in the general case, but require precomputing and storage of constants. Considered in the paper modification of Montg omeryalgorithm of modular multiplication is reduced to a simple multiplication with accumulation, where the word length equals to the length of the module, which allows a device to have a fully parallel architecture where each module is associated with one of the modules

from RNS moduli set. Approximate method doesn'trequire expensive modular operations, which are replaced by fast right bit shifts and taking lower bits. The paper presents the implementation of finding the remainder of division by a prime number in RNS, in which there is no need to perform computations with whole parts of real numbers, which makes it possible to make the transition from the calculations with fractional parts to integer calculations. Effective implementation of the proposed method using a neural network of the finite ring can increase the speed operations. Algorithms of modular operations correspond too perations with basic processing elements (artificial neurons). FPGA is the perfect element base for the implementation of such parallel structures as a neural network. FPGA allows to implement n parallel neural networks, and the communication between the neurons takes place within the same FPGA at high speed. In addition, the use of a neural network of finite ring (NNFR) reduces the number of cascaded circuits, which allows to increase the speed of information processing. With restriction to arithmetic operations NNFRis more effective in comparison with the binary arithmetic. Since computations with real numbers in FPGA areresource-consuming, the transition from the real numbers to integers, according to the proposed algorithm can significantly increase the speed of data processing. Results of simulation on Kintex7 XC7K70T showed that the proposed method ison average 75 % faster, and takes 80 % less in the area than the modified method from [2], which makes it more useful for hardware implementation of cryptographic primitives over a finite field.

REFERENCES

1. Manochehri K., Sadeghian B., Pourmozafari S. A modified radix-2 Montgomery modular multiplication with new recoding method // IEICE Electronics Express. 2010. V. 7. № 8. P. 513–519.
2. Choi S.-H., Lee K.-J. New systolic modular multiplication architecture for efficient Montgomery multiplication // IEICE Electronics Express. 2014, V. 12. № 2. P. 2014–1051.
3. Choi S.-H., Lee, K.-J. Enhancement of a modified radix-2 Montgomery modular multiplication // IEICE Electronics Express. 2014. V. 11. № 19. P. 2014–0782.
4. Zivaljevic D., Stamenkovic N., Stojanovic V. Digital filter implementation based on the RNS with diminished-1 encoded channel // Telecommunications and Signal Processing (TSP), 35th International Conference // IEEE. 2012. July. P. 662–666.
5. Yatskiv V., Jun S., Yatskiv N., Sachenko A., Osolinskiy O. Multilevel method of data coding in WSN. // IDAACS, IEEE 6th International Conference. 2011. P. 863–866.
6. Chervyakov N.I., Babenko M.G., Lyakhov P.A., Lavrinenko I.N. An Approximate method for comparing modular numbers and its application to the division of numbers in residue number systems // Cybernetics and Systems Analysis. 2014. V. 50. № 6. P. 977–984.
7. Schinianakis D., Stouraitis T. Multifunction residue architectures for cryptography // Circuits and Systems I: Regular Papers. IEEE Transactions. 2014. V. 61. № 4. P. 1156–1169.
8. Montgomery P. L. Modular multiplication without trial division // Mathematics of computation, 1985. V. 44. № 170. P. 519–521.
9. Schinianakis D., Stouraitis T. A RNS Montgomery multiplication architecture // Circuits and Systems (ISCAS). IEEE International Symposium. 2011. P. 1167–1170.
10. Galushkin A.I. Teoriya nejronny'x setej. M.: INRZhR. 2000. 416 s.
11. Chervyakov N.I., Saxnyuk P.A., Shaposhnikov V.A., Makoxa A.N. Nejrokomp'yutery' v ostatochny'x klassax. Kn. 11. M.: Radiotexnika. 2003. 272 s.
12. Zhang D. Parallel designs for Chinese remainder conversion // International Conference on Parallel Processing – ICPP. 1987. P. 557–559.
13. Zhang D. Parallel VLSI Neural System Designs. Berlin, Germany: Springer-Verlag. 1998. 257 p.
14. Zang D. Jullien G.A., Miller W.C. A neural-like approach to finite ring computation // IEEE Transactions on Circuits and Systems. 1990. V. 37. № 8. P. 1048–1052.
15. Zhang D., Jullien G.A., Miller W.C. VLSI implementations of neural-like networks for finite ring computations // Proceedings of the 32nd Midwest Symposium on Circuits and Systems. 1989. V.1. P. 485–488.
16. Patent № 2317584 RF. Konvejernaya nejronnaya set' konechnogo kol'cza / N.I. Chervyakov. 2008.
17. Patent № 2279132 RF. Nejronnaya set' konechnogo kol'cza / N.I. Chervyakov, V.A. Galkina, Ju.A. Strekalov, S.V. Lavrinenko. 2006.