УДК 004.272.2 DOI: 10.31114/2078-7707-2022-3-102-107

Реализация обратного преобразователя модулярной арифметики для произвольного базиса на основе таблиц поиска с опорными точками

А.Л. Стемпковский, Д.В. Тельпухов, И.А. Мкртычан, Р.А. Соловьев

Институт проблем проектирования в микроэлектронике РАН, г. Москва, ilia.colour@yandex.ru

***Аннотация* — В данной работе описывается новый способ аппаратного преобразования чисел из системы остаточных классов в двоичную систему счисления. Предложенный для этой цели метод опорных точек основывается на использовании небольшого арифметического блока и сокращенных таблиц поиска (Look up Table (LUT)), что позволяет пользоваться быстродействием LUT и в то же время решать проблему большой площади на кристалле. Метод опорных точек сравнивается с реализацией, основанной на модифицированной Китайской теореме об остатках. Результаты свидетельствуют о том, что новый метод имеет существенный выигрыш в задержке на критическом пути, а также обладает гибкостью, позволяющей подстраивать аппаратную реализацию под нужды разработчика.**

***Ключевые слова* — система остаточных классов, СОК, обратное преобразование, Китайская теорема об остатках, КТО, аппаратная реализация, опорные точки, LUT.**

I. Введение

Позиционные системы счисления наиболее привычны для человека – десятичная используется в повседневной жизни, двоичная в цифровой схемотехнике. На практике могут также применяться системы с другими основаниями – восьмеричная, шестнадцатеричная и т.д. При подобной записи числа, значение каждой цифры зависит от ее позиции – разряда.

В ряде задач позиционная арифметика оказывается недостаточно эффективна, во многом из-за длинных цепей переноса, которые возникают при выполнении арифметических операций. Решить эту проблему можно за счет перехода в непозиционные системы, в частности, в систему остаточных классов (СОК), основанную на модулярной арифметике. Она применяется как в криптографии [1]-[3], так и для реализации цифровых схем [4]-[6]. Среди преимуществ СОК для микроэлектроники можно выделить возможность более эффективной реализации параллельных вычислений [7], а также контроль и коррекцию ошибок в процессе вычислений [8]-[10].

Однако, у СОК есть и недостатки, ограничивающие её более широкое применение, основным из которых является сложность реализации немодульных операций. Алгоритмы таких операций как правило сложны для аппаратной реализации и требуют одновременной обработки всех параллельных модульных каналов. По этой причине, нахождение эффективных методов их имплементации является актуальной задачей.

Одними из важнейших и наиболее часто используемых немодульных операций являются перевод из позиционной системы в СОК (прямое преобразование) и наоборот (обратное преобразование). В данной работе будет рассмотрена операция обратного преобразования. Существует большое количество вариантов ее реализации, однако в основном для них используется специальный набор модулей [11][12], что существенно сокращает гибкость в выборе реализации.

Предложенный метод основывается на использовании сокращенных LUT, а в качестве базисных оснований могут использоваться любые взаимно простые модули.

II. Метод опорных точек

В СОК максимальное количество чисел, которое можно представить с помощью выбранного базиса, определяется произведением всех модулей и называется динамическим диапазоном. Теоретически, все возможные представления чисел можно получить с помощью одной таблицы перекодировок – LUT. У такого подхода есть очевидный плюс – большая скорость работы, и минус – быстро растущая площадь, зависящая от количества записанных значений [13].

Для того чтобы использовать преимущества LUT и в то же время нивелировать его недостатки, предлагается использовать метод опорных точек. Он заключается в следующем:

1. Один из модулей базиса выбирается в качестве опорного.
2. В LUT записываются все значения, для которых остаток от деления по опорному модулю равен 0.
3. Для получения входных данных для LUT, производится сдвиг входных значений с помощью модулярного вычитания.

4. Результат из таблицы складывается со значением, которое было вычтено ранее – таким образом, компенсируется сдвиг.
5. Получившееся число – результат в позиционной системе счисления.

Рассмотрим метод опорных точек более подробно на примере обратного преобразователя в базисе (2, 3, 5). Его динамический диапазон [0, 30), а значит при использовании простой реализации с помощью LUT, в таблице должно храниться 30 значений. Возьмем в качестве опорного наибольший модуль базиса – 5. В сокращенном LUT, (LUT Reference Points) будет всего 6 значений – они представлены в табл. 1.

Таблица 1

Значения в LUT для базиса (2, 3, 5) с опорным модулем 5

input module 2	input module 3	output
0	0	0
1	2	5
0	1	10
1	0	15
0	2	20
1	1	25

Легко видеть, что количество хранимых значений определяется произведением не опорных модулей или, другими словами, можно сказать, что первоначальный LUT был сокращен в 5 раз – на величину опорного модуля.

Возьмем число 8 в качестве примера. В СОК оно запишется как (0, 2, 3). Так как на вход LUT Reference Points будут поданы значения по модулю 2 и 3, легко понять, что без дополнительных преобразований правильный результат не получится. Необходимо выполнить сдвиг входных значений для того, чтобы попасть в подходящую строку таблицы. Ранее было сказано, что в LUT Reference Points записываются только числа, для которых остаток от деления по опорному модулю равен 0. В данном случае это будет значение по модулю 5, которое в представлении числа 8 равняется 3. Для того, чтобы попасть в одну из строк таблицы, необходимо отнять 3 от всех вычетов, задающих число 8: ([0-3]%2, [2-3]%3, [3-3]%5) = = (1, 2, 0) = 5. Используя только не опорные вычеты (1, 2) можно однозначно получить табличное значение 5. Так как ранее был произведен сдвиг на 3, необходимо компенсировать его, прибавив данное число обратно. Таким образом, получим искомое значение 5+3=8. С визуализацией данного примера можно ознакомиться на рис. 1. Схема аппаратной реализации для данного примера представлена на рис. 2.

Рассмотренный метод позволяет сократить количество хранимых значений на величину выбранного опорного модуля за счет добавления небольшого арифметического блока. Однако, если взять более широкий базис, например, (7, 11, 13, 17, 19, 23), результат уже не будет столь впечатляющим. При использовании 23 в качестве опорного модуля, количество значений для записи в LUT сократится с 7436429 до 323323, что хоть и является серьезным улучшением, но все еще слишком велико.

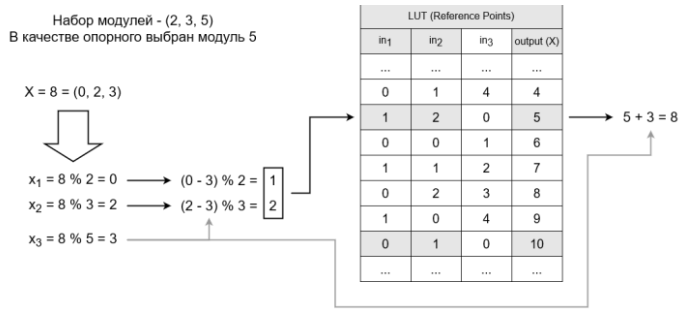

Рис. 1. Пример обратного преобразования

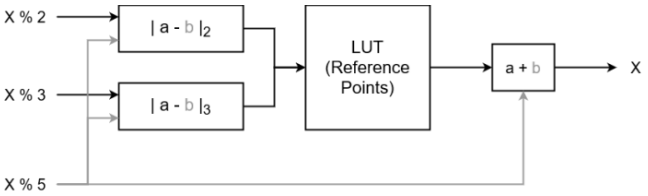

Рис. 2. Аппаратная реализация обратного преобразователя

A. Модификация с дополнительным LUT

Причиной проблемы, описанной выше, является то, что в качестве опорного модуля можно выбрать только один из элементов базиса. В качестве решения можно использовать несколько модулей, однако возникает задача вычисления значения, на которое необходимо произвести последующую коррекцию.

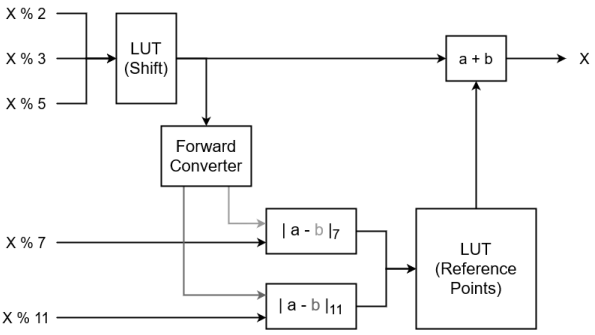

Рис. 3. Обратный преобразователь с дополнительным LUT

Предлагается использовать дополнительный LUT Shift. Его целью будет подсчет нового сдвига, который до этого задавался просто выходным значением по опорному модулю. Таким образом, можно выбирать сразу несколько опорных модулей. Пример подобной реализации представлен на рис. 3. LUT Shift, по сути, представляет собой такой же обратный преобразователь. В данном примере в нем хранятся значения для динамического диапазона, заданного модулями (2, 3, 5) – от 0 до 29 включительно.

Произведение 2, 3 и 5 можно рассматривать как новый опорный модуль 30.

В некоторых случаях размерность числа *b*, поступающего на модулярный вычитатель, будет существенно больше модуля по которому производится расчет разности. В рассмотренном примере максимальное число $b = 29$, в то время как максимальное a равно 6 или 11. Данная разница может быть еще больше, что отрицательно скажется на задержке на критическом пути. В качестве решения предлагается использовать операцию взятия остатка по модулю для приведения *b* к необходимой размерности. Прямой преобразователь (схематично представлен на рис. 4) оперирует в базисе не опорных модулей – в примере это (7, 11).

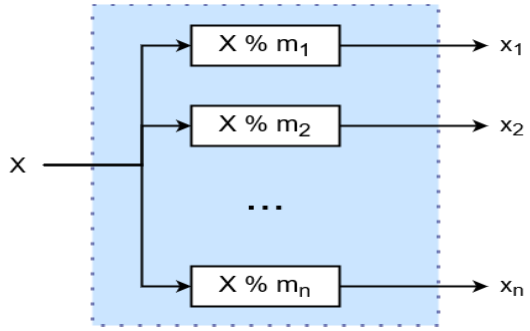

Рис. 4. Общий вид прямого преобразователя

B. Расширенный LUT Shift

При необходимости максимально сократить задержки можно отказаться от прямого преобразователя и записывать предрассчитанные значения остатков по всем не опорным модулям в LUT Shift. Это увеличит размерность хранимых данных. Так, для модулей (7, 11), в таблице будут храниться данные вида, как на рис. 5, занимающие 12 бит. Без использования расширенного LUT Shift их размерность составит 5 бит. Схема реализации расширенного варианта представлена на рис. 6.

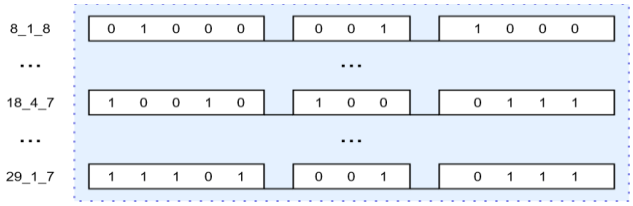

Рис. 5. Пример бинарной записи данных в расширенном LUT Shift

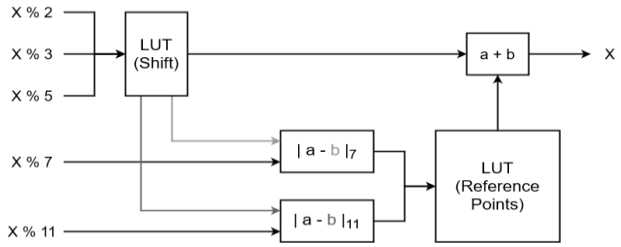

Рис. 6. Обратный преобразователь с расширенным LUT Shift

C. Рекурсивная модификация

Как уже было сказано ранее, LUT Shift выполняет функцию обратного преобразователя. Отсюда следует, что метод опорных точек можно использовать рекурсивно. Это позволит разработчику иметь еще больший набор возможных конфигураций для лучшего соответствия имеющимся ограничениям. Примеры двух реализаций для базиса (2, 3, 5, 7, 11, 13, 17) представлены на рис. 7 и рис. 8.

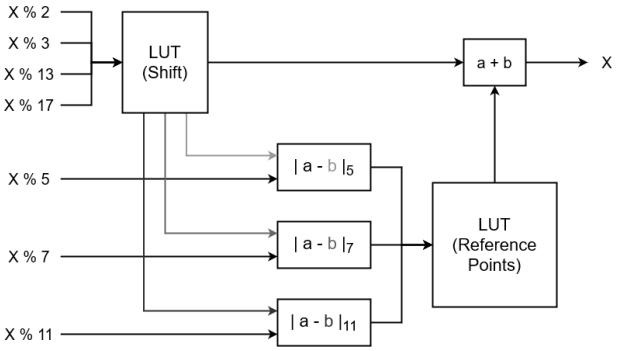

Рис. 7. Обратный преобразователь, реализованный без рекурсивного использования метода опорных точек

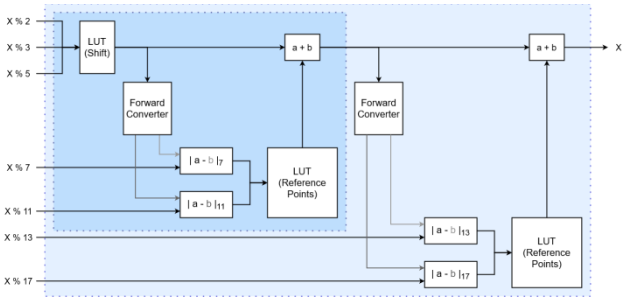

Рис. 8. Обратный преобразователь, реализованный рекурсивным использованием метода опорных точек

На данном этапе вводится понятия слоя – обратного преобразователя, где его номер обозначает позицию в схеме. Преобразователь, в котором не используется рекурсия, обозначается как двухслойный, где LUT Shift – слой под номером 0. Схема на рис. 8 состоит из трех слоев.

В первом примере суммарное количество записанных в таблицу значений равно 1711, когда во втором всего 328. Однако критический путь во втором случае длиннее, исходя из чего, уже на данном этапе можно сказать, что чем больше слоев используется, тем меньше площадь (из-за уменьшения LUT), но и тем больше задержка.

Данный вариант реализации обратного преобразователя является финальной версией метода опорных точек. Разработчик может использовать все рассмотренные модификации для достижения необходимых параметров схемы [14].

III. Сравнение нового метода с аналогом, основанным на Китайской Теореме об Остатках

Для определения эффективности разработанного метода опорных точек, необходимо сравнить его с другим, реализованным ранее и зарекомендовавшим себя методом. На эту роль был выбран способ обратного преобразования, основанный на одной из модификаций Китайской теоремы об остатках, который будет обозначаться в дальнейшем как КТО 2 [15]. Выбор обосновывается тем, что в данном методе также может применяться любой набор взаимно простых модулей.

При тестировании рассмотренных методов сравнивались два параметра:

1. Площадь, занимаемая на кристалле
2. Задержка на критическом пути

Синтез схем производился с помощью средств Cadence с использованием библиотеки Nangate. Всего было подготовлено 3 тестовых набора схем.

Первый набор включал в себя 5 схем обратного преобразователя в базисе (2, 3, 5, 7, 11, 13, 17), одна из которых была основана на КТО 2, а остальные 4 – на методе опорных точек. Схемы для метода опорных точек были скомпонованы следующим образом:

1. 3 слоя с использованием прямого преобразователя на первом слое
2. 3 слоя с использованием расширенного LUT Shift
3. 2 слоя с использованием прямого преобразователя на первом слое
4. 2 слоя с использованием расширенного LUT Shift

Результаты синтеза данных схем можно видеть на рис. 9.

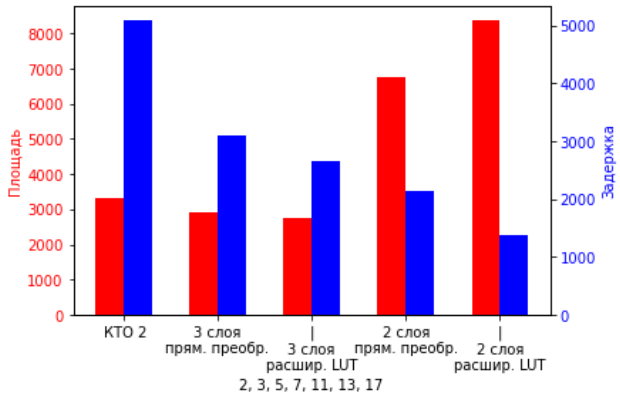

Рис. 9. Результаты для первого набора тестов

Легко видеть, что метод опорных точек, для всех рассмотренных компоновок, показывает меньшие значения задержки на критическом пути, чем КТО 2. Величина задержки уменьшается с уменьшением количества слоев. Она также уменьшается при использовании расширенного LUT Shift за счет незначительного увеличения площади. Площадь растет при увеличении количества хранимых в таблицах данных.

Второй набор включал в себя 5 схем обратного преобразователя в базисе (23, 29, 31, 37, 41, 43), одна из которых была основана на КТО 2, а остальные 4 – на методе опорных точек. Схемы для метода опорных точек были скомпонованы следующим образом:

1. 3 слоя с использованием прямого преобразователя на первом слое
2. 3 слоя с использованием расширенного LUT Shift
3. 2 слоя с использованием прямого преобразователя на первом слое
4. 2 слоя с использованием расширенного LUT Shift

Результаты синтеза можно видеть на рис. 10.

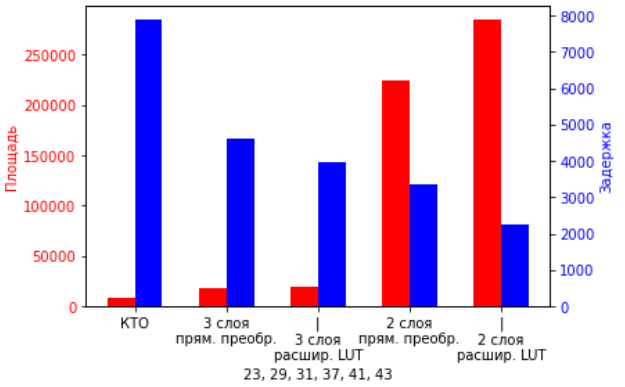

Рис. 10. Результаты для второго набора тестов

По результатам синтеза данных схем можно заметить тенденции, схожие с зафиксированными на прошлом наборе. Рост площади, в свое время, гораздо более быстрый. Метод опорных точек все еще выигрывает у КТО 2 по быстродействию, но теперь уступает по площади во всех рассмотренных случаях.

Третий набор включал в себя 7 схем обратного преобразователя в базисе (2, 3, 5, 7, 11, 13, 17, 19, 23, 29), одна из которых была основана на КТО 2, а остальные 6 – на методе опорных точек. Схемы для метода опорных точек были скомпонованы следующим образом:

1. 5 слоев с использованием прямого преобразователя на первом слое
2. 5 слоев с использованием расширенного LUT Shift
3. 4 слоя с использованием прямого преобразователя на первом слое
4. 4 слоя с использованием расширенного LUT Shift
5. 3 слоя с использованием прямого преобразователя на первом слое
6. 3 слоя с использованием расширенного LUT Shift

Результаты синтеза данных схем можно видеть на рис. 11.

Полученные результаты в очередной раз подтверждают, что при уменьшении количества слоев увеличивается площадь и уменьшается задержка. Данный набор тестов также демонстрирует, что при большом количестве слоев, метод опорных точек начинает проигрывать по задержке КТО 2, почти сравнявшись при этом по площади для рассмотренного случая.

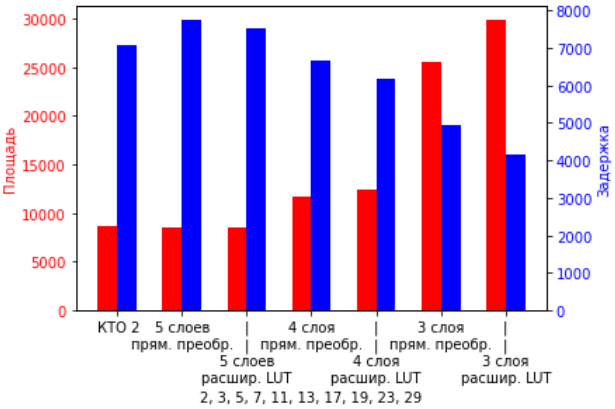

Рис. 11. Результаты для третьего набора тестов

IV. Заключение

Рассмотрев результаты синтеза всех схем из трех тестовых наборов можно заключить, что метод опорных точек, в большинстве рассмотренных конфигураций, показывает лучшие значения задержки на критическом пути, нежели аналогичный ему по функционалу метод обратного преобразования, основанный на КТО 2, а значит, может быть эффективно использован для аппаратной реализации в схемах, основанных на СОК. Новый метод, как и предполагалось ранее, позволяет модифицировать схему в зависимости от того, что разработчику важнее минимизировать – площадь (путем увеличения количества слоев) или задержки (путем минимизации количества слоев и использования расширенного LUT Shift).

Из-за большого количества вариантов компоновки разработанного блока, особенно актуальной становится возможность выбора наиболее подходящей реализации для каждого конкретного случая [16]. В дальнейшей работе будет представлен вариант решения данной проблемы.

Поддержка

Исследование выполнено при финансовой поддержке Гранта Президента Российской Федерации № МД-1414.2021.4.

Литература

[1] D. K. Shaji and V. Jacob. Efficient random number generator using novel modulo 2n-2k-1 adder for RNS // 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). 2016. pp. 1659-1663.

[2] S. K. Singh, V. P. Gopi and P. Palanisamy. Image security using DES and RNS with reversible watermarking // 2014 International Conference on Electronics and Communication Systems (ICECS). 2014. pp. 1-5.

[3] C. S. Kumar, A. Prathiba and V. S. K. Bhaskaran. Implementation of RNS and LNS based addition and subtraction units for cryptography. 2016 International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA). 2016. pp. 1-5.

[4] B. Arambepola. VLSI residue arithmetic architecture for signal processing applications // IEE Colloquium on VLSI Signal Processing Architectures. 1990. pp. 1-6.

[5] E. B. Olsen. RNS Hardware Matrix Multiplier for High Precision Neural Network Acceleration: “RNS TPU” // 2018 IEEE International Symposium on Circuits and Systems (ISCAS). 2018. pp. 1-5.

[6] Telpukhov D.V., Sovovyev R.A., Balaka E.S., Rukhlov V.S., Mikhmel A.S. Design Features Of The Rns-Based Multipliers Using Advanced Cad // Проблемы Разработки Перспективных Микро- и Наноэлектронных Систем (МЭС). 2017. №1. С. 66-70.

[7] Амербаев В.М., Соловьев Р.А., Тельпухов Д.В., Поперечный П.С., Рухлов В.С., Щелоков А.Н., Михмель А.С. Разработка устройства для вычисления результата операции скалярного произведения векторов на базе интрамодулярного разложения комплексных чисел в модулярной арифметике // Известия ЮФУ. Технические Науки. 2015. №6. С. 95-105.

[8] J. -. Bajard, L. Imbert. A full RNS implementation of RSA // IEEE Transactions on Computers. Vol. 53. No. 6. June 2004. pp. 769-774.

[9] E. D. Di Claudio, G. Orlandi, F. Piazza. Parallel error correction algorithm in RNS VLSI digital circuits // ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing. 1988. pp. 1738-1741.

[10] S. Pontarelli, G. C. Cardarilli, M. Re, A. Salsano. Totally Fault Tolerant RNS Based FIR Filters // 2008 14th IEEE International On-Line Testing Symposium. 2008. pp. 192-194.

[11] K. Karthik, N. C. H. Vun. Efficient reverse converters designs for RNS based digital signal processing systems // 2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE). 2013. pp. 153-154.

[12] A. A. Emrani Zarandi, A. S. Molahosseini, L. Sousa, M. Hosseinzadeh. An Efficient Component for Designing Signed Reverse Converters for a Class of RNS Moduli Sets of Composite Form {2^{k}, 2^{P}-1\} // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Vol. 25. No. 1. Jan. 2017. pp. 48-59.

[13] Telpukhov D.V., Solovyev R.A., Mkrtchan I.A. Hardware Implementation of Scaling in Residue Number System in Application to Convolutional Neural Networks // International Scientific – Practical Conference “Information Innovative Technologies”. 2020. pp 165-169.

[14] Стемпковский А.Л., Бобков С.Г., Змеев Д.Н., Левченко Н.Н., Климов Арк.В. Автоматизированное определение оптимальной конфигурации параллельной потоковой вычислительной системы для решения конкретной задачи // Проблемы Разработки Перспективных Микро- и Наноэлектронных Систем (МЭС). 2021. №3. С. 82-93.

[15] Yuke Wang. Residue-to-binary converters based on new Chinese remainder theorems // IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. Vol. 47. No. 3. March 2000. pp. 197-205.

[16] Жукова Т.Д. Разработка системы автоматизированного проектирования СФК на основе методов избыточного кодирования // Проблемы Разработки Перспективных Микро- и Наноэлектронных Систем (МЭС). 2020. №4. С. 51-57.

Implementation of an RNS Reverse Converter for General Moduli Sets Based on LUTs with Reference Points

A.L. Stempkovsky, D.V. Telpukhov, I.A. Mkrtchan, R.A. Solovyev

Institute for Design Problems in Microelectronics RAS, Moscow, ilia.colour@yandex.ru

***Abstract* — Residue Number System (RNS) is a non-positional number system that can be used to implement parallel arithmetic operations. This feature allows achieving high performance in hardware. However, algorithms for some operations are complex and hard to realize effectively. One of the most common operations in RNS is reverse conversion (from RNS to binary). In this paper we propose a new approach to its implementation, based on a combination of LUTs and simple arithmetic calculations. The main advantage of this method is flexibility – the design can be easily changed to achieve the best solution for given hardware constraints and performance criteria. This approach also does not require any special set of moduli – all integers just are to be coprime. Experimental results show that proposed implementation outperforms the method based on the modification of the Chinese remainder theorem in terms of computational speed while maintaining moderate hardware costs. More compact solutions can also be created at the expense of loss in timing.**

***Keywords* — residue number system, RNS, reverse conversion, Chinese remainder theorem, CRT 2, hardware implementation, reference points, LUT.**

REFERENCES

[1] D. K. Shaji and V. Jacob. Efficient random number generator using novel modulo 2n-2k-1 adder for RNS // 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). 2016. pp. 1659-1663.

[2] S. K. Singh, V. P. Gopi and P. Palanisamy. Image security using DES and RNS with reversible watermarking // 2014 International Conference on Electronics and Communication Systems (ICECS). 2014. pp. 1-5.

[3] C. S. Kumar, A. Prathiba and V. S. K. Bhaskaran. Implementation of RNS and LNS based addition and subtraction units for cryptography. 2016 International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA). 2016. pp. 1-5.

[4] B. Arambepola. VLSI residue arithmetic architecture for signal processing applications // IEE Colloquium on VLSI Signal Processing Architectures. 1990. pp. 1-6.

[5] E. B. Olsen. RNS Hardware Matrix Multiplier for High Precision Neural Network Acceleration: “RNS TPU” // 2018 IEEE International Symposium on Circuits and Systems (ISCAS). 2018. pp. 1-5.

[6] Telpukhov D.V., Sovovyev R.A., Balaka E.S., Rukhlov V.S., Mikhmel A.S. Design Features of The Rns-Based Multipliers Using Advanced Cad // Problems of Advanced Micro- and Nanoelectronic Systems Development (MES). 2017. №1. pp. 66-70.

[7] Amerbaev V.M., Sovovyev R.A., Telpukhov D.V., Poperechny P.S., Rukhlov V.S., Schelokov A.N., Michmel A.S. Razrabotka ustrojstva dlya vychisleniya rezul'tata operacii skalyarnogo proizvedeniya vektorov na baze intramodulyarnogo razlozheniya kompleksnyh chisel v modulyarnoj arifmetike (Development of a microelectronic device for dot product calculation based on rns intramodular decomposition of complex numbers). Izvestiya YUFU. Tekhnicheskie Nauki. 2015. №6. pp. 95-105.

[8] J. -. Bajard, L. Imbert. A full RNS implementation of RSA // IEEE Transactions on Computers. Vol. 53. No. 6. June 2004. pp. 769-774.

[9] E. D. Di Claudio, G. Orlandi, F. Piazza. Parallel error correction algorithm in RNS VLSI digital circuits // ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing. 1988. pp. 1738-1741.

[10] S. Pontarelli, G. C. Cardarilli, M. Re, A. Salsano. Totally Fault Tolerant RNS Based FIR Filters // 2008 14th IEEE International On-Line Testing Symposium. 2008. pp. 192-194.

[11] K. Karthik, N. C. H. Vun. Efficient reverse converters designs for RNS based digital signal processing systems // 2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE). 2013. pp. 153-154.

[12] A. A. Emrani Zarandi, A. S. Molahosseini, L. Sousa, M. Hosseinzadeh. An Efficient Component for Designing Signed Reverse Converters for a Class of RNS Moduli Sets of Composite Form {2^{k}, 2^{P}-1\} // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Vol. 25. No. 1. Jan. 2017. pp. 48-59.

[13] Telpukhov D.V., Solovyev R.A., Mkrtchan I.A. Hardware Implementation of Scaling in Residue Number System in Application to Convolutional Neural Networks // International Scientific – Practical Conference “Information Innovative Technologies”. 2020. pp 165-169.

[14] Stempkovskiy A.L., Bobkov S.G., Zmejev D.N., Levchenko N.N., Klimov A.V. Automated Optimal Configuration Determination of The Parallel Dataflow Computing System for Solving a Specific Problem // Problems of Advanced Micro- and Nanoelectronic Systems Development (MES). 2021. №3. pp. 82-93.

[15] Yuke Wang. Residue-to-binary converters based on new Chinese remainder theorems // IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. Vol. 47. No. 3. March 2000. pp. 197-205

[16] Zhukova T.D. Functional Control Circuits CAD System Based on Redundant Coding Methods // Problems of Advanced Micro- and Nanoelectronic Systems Development (MES). 2020. №4. pp. 51-57.