УДК 004.02:004.2

В. В. Гранкин

АНАЛИТИЧЕСКАЯ ФОРМА МЕТОДА ГАРНЕРА ДЛЯ РАСШИРЕНИЯ БАЗИСА СИСТЕМЫ ОСТАТОЧНЫХ КЛАССОВ

Проведен анализ основных методов расширения базиса системы остаточных классов, получены аналитические оценки аппаратных затрат. Сделан вывод о большей эффективности классического варианта метода Гарнера при аппаратной реализации с позиций аппаратных и временных затрат. По существующему алгоритмическому описанию построена аналитическая форма метода Гарнера, которая позволяет упростить синтез вычислителей расширения базиса системы остаточных классов, в особенности аппаратных реализаций, благодаря тому, что алгоритмический способ построения заменен его рекуррентной формулой. Сравнение аппаратной реализации модулей расширения базиса системы остаточных классов на основе предложенной аналитической формы и классического варианта метода Гарнера показало их эквивалентность. Аналитическую форму метода Гарнера предложено использовать для осуществления преобразования непозиционных кодов системы остаточных классов в позиционный код традиционной системы счисления. Показана эффективность такого решения при аппаратной реализации.

Ключевые слова: система остаточных классов, метод Гарнера, расширение базиса, преобразование в позиционный код.

Введение

Рост быстродействия вычислительных машин в настоящее время не может быть осуществлен только за счет совершенствования технологического процесса, увеличения быстродействия элементарных вентилей и тактовой частоты вычислителей в целом. В связи с этим возникает острая необходимость в проведении вычислений параллельными средствами. Известно, что не все алгоритмы могут быть эффективно реализованы на параллельных вычислительных устройствах. Одним из путей решения этой проблемы является проведение вычислений в системе остаточных классов (СОК). Данная непозиционная система счисления позволяет, за счет естественного параллелизма, увеличить быстродействие некоторых операций [1]. В СОК существует ограничение на диапазон M представления чисел, равный произведению оснований $m_1\, m_2 \ldots m_n = M$ и в общем случае $\forall x,\, x \in [0,\, M-1]$, где $\{m_i\}$ – набор взаимно простых чисел [1]. При решении некоторых задач возникает необходимость изменения данного диапазона во время вычислений [2]. Изменение диапазона СОК возможно благодаря методам расширения базиса без осуществления преобразования кодов СОК в коды традиционной позиционной системы счисления (ПСС) с последующим обратным преобразованием в СОК. Уменьшение временных и аппаратных затрат позволяет считать расширение базиса важной немодулярной операцией. Аппаратные затраты – это важный критерий, отражающий количество элементарных логических вентилей, необходимых для построения аппаратных вычислителей.

Методы расширения базиса системы остаточных классов

В работе [3] показан *метод Гарнера*, являющийся основным для расширения базиса СОК. Суть метода состоит в последовательном вычитании из числа остатка модуля и последующем умножении на обратный элемент к этому модулю, последовательность выполняется для всех модулей [4].

В работах [5, 6] предложен так называемый *метод матриц и констант*. Метод основан на построении матрицы размера $n \times n$, с инициализацией предвычисленных констант и последующим умножением остатков на константы, сложением величин, и вычислении результирующего остатка. Недостатком данного метода является необходимость в реализации таблицы большого размера.

Методы, предложенные в [7, 8], построены *на основе ортогональных базисов и вычисления ранга числа*. Суть данных методов состоит в неявном вычислении базиса и ранга числа (как при преобразовании кодов СОК в позиционный код). В случае явного преобразования новый остаток вычисляется как $\sum x_i \cdot B_i - a \cdot M \bmod m_{n+1} = X \bmod m_{n+1}$, что эквивалентно

$\sum x_i \cdot B_i \bmod M \bmod m_{n+1} = X \bmod m_{n+1}$. Существенным недостатком методов, описанных в [7, 8], являются вычисления с числами, значительно превышающими размер модуля СОК, в том числе при вычислении ранга числа *a*, что увеличивает затраты и задержки распространения сигналов при аппаратной реализации.

В работе [9] рассмотрен *метод расширения, основанный на промежуточном переходе через полиадический код*. Недостатком такого подхода является вычисление промежуточных величин (константы полиадического кода), что требует размещения аппаратных многоразрядных вычислителей.

При аппаратной реализации метода расширения базиса основными критериями выбора являются время вычисления и аппаратные затраты. На основании асимптотик времени вычисления и асимптотик аппаратной сложности возможно попытаться сделать вывод о предпочтительном методе или алгоритме при построении вычислителя. Для обеспечения высокой скорости вычисления необходимо, чтобы комбинационный путь распространения сигнала был как можно короче. Для сравнения методов можно сопоставить асимптотики комбинационного пути и аппаратных затрат (табл. 1).

Таблица 1

Оценка асимптотик времени вычисления и аппаратных затрат основных методов расширения базиса СОК

Метод расширения	Асимптотика		Оценка	
	времени вычислений	аппаратных затрат	числа таблиц	числа сумматоров
	n **– число бит**		***n*** **– число модулей**	
Гарнера	$O(n)$	$O(n^2)$	$2n$	$0.5\ (n^2+n-2)$
Матриц и констант	$O(n)$	$O(n^2)$	n^2+4n	$3n+1$
Shenoy A. P., Kumaresan R. [8]	$O(n)$	$O(n^2)$	n^{2*}	$2n$ и n^*
Полиадический код	$O(n)$	$O(n^2)$	n^{2*}	n^2 и n^*

* Полноразрядный умножитель или таблица, разрядность которого эквивалентна диапазону СОК (разрядности произведения всех модулей).

Результаты анализа позволяют сделать вывод, что для аппаратной реализации с модулями малой разрядности следует предпочесть метод Гарнера, для модулей большой разрядности – метод матриц и констант.

Аналитическое описание метода Гарнера для расширения базиса системы остаточных классов

Практическое применение алгоритмической записи метода Гарнера для расширения базиса СОК характеризуется высокой сложностью алгоритмизации для решения частной задачи [3]. Построим аналитическое описание методом индукции.

1. Приведем формулу для частного решения с одним модулем: $x_2 = \left|x_1\right|_{m_2}$.

2. Построим формулу для частного решения с двумя модулями (для примера СОК с набором оснований (3, 5) решение рассмотрено в табл. 2).

Таблица 2

Пример расширения базиса методом Гарнера, до нового набора оснований (3, 5, 7) по исходным основаниям (3, 5)

Операция	Модуль		
	3	5	7 (новый)
Исходное число 7	1	2	$x_7 = 0$
–1	0	1	$x_7 - 1 = 0$
1/3		2	$1/3 \cdot x_7 - 5 = 0$
–2		0	$1/3 \cdot x_7 - 0 = 0$
			$x_7 = (0+0)\ 3 \rightarrow x_7 = 0$

Можно заметить, что в данном случае $x_7 = x_3 + 3\left|1/3\cdot(x_5 - x_3)\right|_5$, и для общего случая получим $x_3 = \left|x_1 + m_1 \cdot \left|1/m_1\ (x_2 - x_1)\right|_{m2}\right|_{m3}$.

3. Аналогично, для большего числа оснований и путем обобщения результатов установлено, что для расширения базиса СОК можно использовать следующие выражения:

$$\begin{aligned}
x_2 &= \left|f_1(x_1)\right|_{m_2},\\
x_3 &= \left|f_1(x_1) + m_1 f_2(x_1)\right|_{m_3},\\
x_4 &= \left|f_1(x_1) + m_1 f_2(x_1,x_2) + m_1 m_2 f_2(x_1,x_2,x_3)\right|_{m_4},\\
x_5 &= \left|f_1(x_1) + m_1 f_2(x_1,x_2) + m_1 m_2 f_3(x_1,x_2,x_3) + m_1 m_2 m_3 f_4(x_1,x_2,x_3,x_4)\right|_{m_5},\\
&\ldots,
\end{aligned} \tag{1}$$

где x_{n+1} искомый новый остаток от модуля m_{n+1}; m_n – существующие модули СОК; $x_1, x_2, \ldots, x_n$ – существующие остатки СОК; под операцией $\left|x\right|_m$ подразумевается взятие по модулю m, т. е. $\left|x\right|_m = x \bmod m$; в качестве функций f_i используются:

$$\begin{aligned}
f_1(x_1) &= x_1,\\
f_2(x_1,x_2) &= \left|\frac{1}{m_1}(x_2 - x_1)\right|_{m2},\\
f_3(x_1,x_2,x_3) &= \left|\frac{1}{m_1 m_2}(x_3 - x_1) - \frac{1}{m_2} f_2(x_1,x_2)\right|_{m3},\\
f_4(x_1,x_2,x_3,x_4) &= \left|\frac{1}{m_1 m_2 m_3}(x_4 - x_1) - \frac{1}{m_2 m_3} f_3(x_1,x_2,x_3) - \frac{1}{m_3} f_2(x_1,x_2)\right|_{m4}.
\end{aligned} \tag{2}$$

В общем виде можно предложить рекуррентную формулу:

$$f_n(x) = \left|\prod_{i=1}^{n-1}\frac{1}{m_i}(x_n - x_1) - \sum_{i=2}^{n-1}\left(\prod_{j=i}^{n-1}\frac{1}{m_j}\right) f_i(x)\right|_{m_n}, \tag{3}$$

где под операцией $\left|\frac{1}{k}\right|_m = k^{-1} \bmod m$ подразумевается умножение на обратный элемент по данному модулю m.

Используя формулы (1)–(3), можно получить выражение для вычисления нового остатка по имеющемуся набору остатков. Для примера приведем формулу расширения базиса с набором оснований (3, 5, 7):

$$x = \left|x_3 + 3\left|2(x_5 - x_3)\right|_5 + 3\cdot 5\left|3\cdot 5(x_7 - x_3) - 3\left|2(x_5 - x_3)\right|_5\right|_7\right|_{m_n},$$

где m_n – новое основание (новое основание выбирает пользователь, т. е. тот, кто строит вычислитель для решения своей задачи).

Аппаратная реализация

При непосредственной тривиальной аппаратной реализации формул (1)–(3) аппаратные затраты имеют асимптотику $O(2^n)$ и время вычисления $O(n)$.

Обратим внимание на регулярную структуру выражений и построим вычислительный тракт модуля расширения базиса в виде дерева (рис. 1).

На рис. 1 изображены вычислительные устройства расширения базиса СОК с набором оснований (3, 5), (3, 5, 7) и (3, 5, 7, 11), построенные по приведенным аналитическим выражениям (1)–(3). В квадрате с цифрой обозначен сумматор-умножитель, умножение происходит на константу по модулю. Это сделано в предположении, что в аппаратной реализации в каждом вычислительном узле фиксированный набор модулей. Благодаря этому отпадает необходимость в хранении таблиц обратных чисел, в вычислении этих чисел и реализации затратного аппаратного модулярного умножителя. Тем самым, объединив умножитель на константу (сумматор-умножитель) и сумматор, можно добиться экономии ресурсов. Заключительный этап вычисления – сложение – происходит по требуемому целевому модулю.

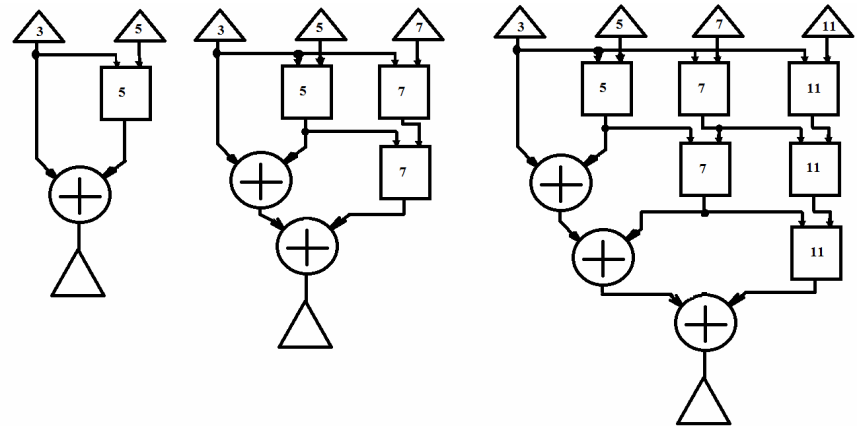

Рис. 1. Схематическое изображение регулярной структуры аппаратной реализации расширения базиса СОК

При реализации в виде дерева (рис. 1) аппаратные затраты имеют асимптотику $O(n^2)$, а время вычислений $O(n)$, где n – число модулей.

Сумматор-умножитель (умножитель на константу) может быть реализован по различным схемам, таким как предварительное суммирование и табулирование (характерна малая задержка, но большой размер таблицы), а также модулярное суммирование и табулирование (меньший размер таблицы, но несколько большая задержка).

На рис. 2 изображена аппаратная реализация расширения базиса с набором оснований (3, 5, 7), в которой реализованы сумматоры по модулю (модуль сумматора указан в обозначении сумматора, сумматор с обозначением *m* является сумматором по целевому модулю). Не трудно заметить, что данная схема состоит из *n* модулярных сумматоров с убывающими младшими модулями. В квадратах буквой «Т» обозначена таблица (таблица поиска, LUT).

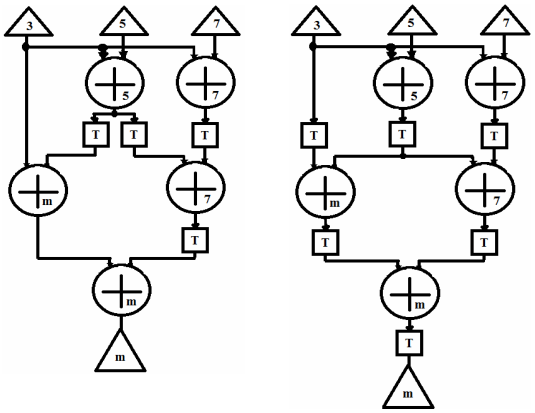

Рис. 2. Аппаратная реализация расширения базиса СОК с набором оснований (3, 5, 7): *а* – по предложенной аналитической записи; *б* – по методу Гарнера

Очевидно, что реализации по аналитическим выражениям и по методу Гарнера требуют эквивалентных аппаратных затрат и имеют эквивалентный комбинационный путь, что говорит о сопоставимой задержке распространения сигналов.

Таким образом, на основе полученных аналитических выражениях и рекуррентной формуле возможно строить вычислители любых наборов оснований более простым способом, чем в случае классического метода Гарнера, автоматизировав синтез схем.

Преобразование кода системы остаточных классов в традиционный позиционный код путем расширения базиса СОК

Предлагается использовать метод расширения базиса СОК в качестве преобразователя кодов СОК в коды традиционной ПСС. Для этого в качестве целевого модулярного сумматора достаточно применить сумматор по модулю 2^n, где n – разрядность целевой машины, т. е. применить традиционный двоичный сумматор. Данный преобразователь кодов имеет асимптотику времени вычислений $O(n)$, эквивалентную асимптотике преобразователя на основе метода ортогонального базиса. Достоинствами данного метода являются:

– выполнение всех операций на модулярных вычислительных устройствах;

– отсутствие необходимости хранить константы больших размеров;

– отсутствие вычислений с длинной арифметикой;

– высокая скорость преобразования.

В табл. 3 приведена аналитическая оценка аппаратных затрат программируемой логической интегральной схемы (ПЛИС) при реализации устройства обратного преобразования кодов СОК в коды ПСС методом Гарнера для расширения базиса и методом ортогональных базисов.

Таблица 3

Потребление ячеек ПЛИС при преобразовании из СОК в традиционную систему счисления

Набор модулей СОК	Метод Гарнера				Метод ортогональных базисов	
	Аналитическая форма		Классический вариант			
	Количество ячеек	Длина	Количество ячеек	Длина	Количество ячеек	Длина
(2)	0	0	0	0	0	0
(2, 3)	4	3	5	4	10	4
(2, 3, 5)	16	5	18	6	32	5
(2, 3, 5, 7)	36	7	38	8	78	6
(2, 3, 5, 7, 11)	57	9	60	10	135	7

Как видно из табл. 3, метод Гарнера требует меньше ячеек ПЛИС для своей реализации (разница более чем в 2 раза), причем с увеличением диапазона СОК достигается большее преимущество, зависящее от конкретного набора оснований. Основные затраты при реализации метода ортогональных базисов приходятся на вычисление ранга числа при нормировании результата. Следует отметить, что методы имеют эквивалентную длину комбинационного пути. При аналитической оценке аппаратных затрат предполагалось использование табличных вычислений (без встроенных умножителей), оптимизация трассировщиком не учитывалась. Длина комбинационного пути при использовании метода Гарнера несколько выше, число сумматоров в пути при распространении сигнала одинаково, увеличение длины комбинационного пути связано с размещением таблиц, время срабатывания которых минимально (т. к. требуется один дешифратор для срабатывания мультиплексора, реализующего таблицу поиска LUT). Вышеизложенное позволяет сделать вывод об эквивалентности времени вычислений и уменьшении аппаратных затрат.

Заключение

Таким образом, в ходе исследований были получены следующие результаты.

1. На основе классического варианта метода Гарнера получена аналитическая форма метода для расширения базиса СОК.

2. Для вычисления нового остатка СОК по существующему набору оснований получена рекуррентная формула.

3. Данные выражения позволяют организовать алгоритмический синтез аппаратных реализаций расширения базиса более простым путем.

4. Приведены аппаратные реализации для некоторых оснований.

5. Оценка времени вычислений и аппаратных затрат некоторых методов расширения базиса СОК показала, что асимптотика эквивалентна для всех методов, отличие состоит в сложности более низкого порядка и константе.

6. Предложено нестандартное использование метода расширения базиса СОК в качестве преобразователя кодов СОК в коды ПСС с целью уменьшения потребления ресурсов.

СПИСОК ЛИТЕРАТУРЫ

1. *Omondi A., Premkumar B.* Residue number systems. Theory and Implementation. London, Imperial College Press. 2007. 310 p.

2. *Гранкин В. В., Мезенцева О. С.* К вопросу о реализации модулярных вычислителей элементарных функций в среде LabVIEW и их реализации на ПЛИС // Материалы VI Междунар. науч.-техн. конф. «Инфокоммуникационные технологии в науке, производстве и образовании». Ставрополь: СевКавГТИ, 2014. С. 248–254.

3. *Banerji D. K., Brzozowski J. A.* On Translation Algorithms in Residue Number Systems // IEEE Transactions on Computers. 1972. Vol. C-21, No. 12, pp. 1281–1285.

4. *Garner H. L.* The residue number system // IRE Trans. Electronic Computer. 1959. Vol. EC-8. P. 140–147.

5. *Червяков Н. И., Мезенцева О. С., Лавриненко И. Н., Сивоплясов Д. В.* Метод расширения динамического диапазона модулярного нейрокомпьютера // Нейрокомпьютеры: разработка, применение. 2005. № 7. С. 64–69.

6. *Червяков Н. И., Лавриненко И. Н., Лавриненко С. В., Мезенцева О. С.* Методы и алгоритмы округления, масштабирования и деления чисел в модулярной арифметике // 50 лет модулярной арифметике. Материалы Междунар. науч.-техн. конф. (Зеленоград, 2005). М.: МИЭТ, 2005. С. 291–310.

7. *Bajard J. C., Didier L. S., Kornerup P.* Modular multiplication and base extensions in residue number systems // Proceedings of the 15th IEEE Symposium on Computer Arithmetic. 2001. Vol. 2. P. 59-65.

8. *Shenoy A. P., Kumaresan R.* Fast base extension using a redundant modulus in RNS // IEEE Transactions on Computer. 1989. 38 (2). P. 292–296.

9. *Червяков Н. И., Рядов С. А., Сахнюк П. А., Шапошников А. В.* Модулярные параллельные вычислительные структуры нейропроцессорных систем. М.: Физматлит, 2003. 288 с.

Статья поступила в редакцию 1.06.2016

ИНФОРМАЦИЯ ОБ АВТОРЕ

Гранкин Виталий Владимирович – Россия, 355012, Ставрополь; Северо-Кавказский федеральный университет; аспирант кафедры инфокоммуникаций; vta0@yandex.ru.

V. V. Grankin

ANALYTICAL FORM OF GARNER'S METHOD OF BASIS EXTENSION OF THE RESIDUE NUMBER SYSTEM

Abstract. The basic methods of extension of the basis of the residue number system are analyzed, the analytical assessments of the apparatus expenses are received. The conclusion on higher efficiency of the classical variant of Garner's method while apparatus operating in terms of apparatus and time expenses is made. In accordance with the existing algorithmic description the analytical form of Garner's method is designed; it helps to simplify the synthesis of the numerators of the basis extension of the residue number system, in particular apparatus operations, due to the fact, that the algorithmic method is replaced with its recurrent formula. The comparison of the apparatus operations of the modules of the basis extension of the residue number system based on the pro-

posed analytical form and Garner's classical method showed the equivalence. It is proposed to use the analytical form of Garner's method to carry out the transformation of the codes of the residue number system into the positional codes of the traditional number system. The effectiveness of this solution while apparatus operating is shown.

Key words: residue number system, Garner method, basis extension, backward conversion, programmable logic device.

REFERENCES

1. Omondi A., Premkumar B. *Residue number systems. Theory and Implementation*. London, Imperial College Press, 2007. 310 p.

2. Grankin V. V., Mezentseva O. S. K voprosu o realizatsii moduliarnykh vychislitelei elementarnykh funktsii v srede LabVIEW i ikh realizatsii na PLIS [To the question of actualization of modular calculators of elementary functions in LabVIEW environment and their implementation on PLIS]. *Materialy VI Mezhdunarodnoi nauchno-tekhnicheskoi konferentsii «Infokommunikatsionnye tekhnologii v nauke, proizvodstve i obrazovanii».* Stavropol, SevKavGTI, 2014. P. 248–254.

3. Banerji D. K., Brzozowski Ja. A. On Translation Algorithms in Residue Number Systems. *IEEE Transactions on Computers*, 1972, vol. C-21, no. 12, pp. 1281–1285.

4. Garner H. L. The residue number system. *IRE Trans. Electronic Computer*, 1959, vol. EC-8, pp. 140–147.

5. Cherviakov N. I., Mezentseva O. S., Lavrinenko I. N., Sivopliasov D. V. Metod rasshireniia dinamicheskogo diapazona moduliarnogo neirokomp'iutera [Method of extension of the dynamic range of the modular neurocomputer]. *Neirokomp'iutery: razrabotka, primenenie*, 2005, no. 7, pp. 64–69.

6. Cherviakov N. I., Lavrinenko I. N., Lavrinenko S. V., Mezentseva O. S. Metody i algoritmy okrugleniia, masshtabirovaniia i deleniia chisel v moduliarnoi arifmetike [Methods and algorithms of approximation, scaling and division of numbers in modular arithmetic]. *50 let moduliarnoi arifmetike, Materialy Mezhdunarodnoi nauchno-tekhnicheskoi konferentsii (Zelenograd, 2005).* Moscow, MIET, 2005. P. 291–310.

7. Bajard J. C., Didier L. S., Kornerup P. Modular multiplication and base extensions in residue number systems. *Proceedings of the 15th IEEE Symposium on Computer Arithmetic*, 2001, vol. 2, pp. 59–65.

8. Shenoy A. P., Kumaresan R. Fast base extension using a redundant modulus in RNS. *IEEE Transactions on Computer*, 1989, 38 (2), pp. 292–296.

9. Cherviakov N. I., Riadnov S. A., Sakhniuk P. A., Shaposhnikov A. V. *Moduliarnye parallel'nye vychislitel'nye struktury neiroprotsessornykh sistem* [Modular parallel computational structure of neuro-processor systems]. Moscow, Fizmatlit Publ, 2003. 288 p.

The article submitted to the editors 1.06.2016

INFORMATION ABOUT THE AUTHOR

Grankin Vitaliy Vladimirovich – Russia, 355012; Stavropol; North Caucasus Federal University; Postgraduate Student of the Department of Infocommunications; vta0@yandex.ru.