КИБЕРНЕТИКА

Я.Н. НИКОЛАЙЧУК, М.Н. КАСЯНЧУК, И.З. ЯКИМЕНКО

УДК 519.7

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АНАЛИТИЧЕСКОГО ВЫЧИСЛЕНИЯ КОЭФФИЦИЕНТОВ БАЗИСНЫХ ЧИСЕЛ ПРЕОБРАЗОВАНИЯ КРЕСТЕНСОНА

Аннотация. Представлены теоретические основы аналитического вычисления коэффициентов базисных чисел преобразования Крестенсона, что существенно уменьшает количество операций, необходимых для перевода чисел из системы остаточных классов в десятичную систему исчисления. При этом соответствующий подбор модулей позволяет достичь эффективного использования всех регистров разрядной сетки.

Ключевые слова: система остаточных классов, система модулей, базисные числа, преобразование Крестенсона, теоретико-числовые базисы.

ВВЕДЕНИЕ

В настоящее время одной из главных тенденций в развитии средств вычислительной техники является создание высокопроизводительных вычислительных устройств [1]. Это обусловлено необходимостью решения весьма важных для теории и практики математики задач, требующих вычислений с целыми многоразрядными числами или величинами, изменяющимися в сравнительно больших диапазонах [2].

В связи с этим интенсивно развивается прикладной и вычислительный аспекты теории чисел, которые применяются в технологических системах для надежности передачи, хранения и обработки цифровой информации. Это приводит к необходимости решения большого количества задач, когда возникают вычисления, при которых целочисленные переменные могут намного превышать разрядную сетку существующих универсальных компьютерных средств, что особенно актуально с развитием криптографических методов и средств защиты информации [3, 4].

АНАЛИЗ ПУБЛИКАЦИЙ

Любая вычислительная структура тесно связана с теоретико-числовыми базисами (ТЧБ), в которых задан способ кодирования (представления) элементов некоторой конечной модели действительных чисел элементами одного или нескольких алфавитов [5].

Арифметические свойства каждой системы счисления, которые порождаются соответствующими ТЧБ, прежде всего определяются характером межразрядных связей, возникающих в ходе выполнения соответствующих арифметико-логических операций [6]. Исследования показывают [5, 7], что в пределах обычных (десятичной, двоичной) позиционных систем счисления скачкообразного ускорения выполнения арифметических операций сложения, вычитания и умножения практически достичь невозможно. Это объясняется тем, что значение каждого разряда любого числа, кроме младшего, зависит от значения не только одноименных операндов, но и от всех младших разрядов, т.е. позиционная система счисления обладает строго последовательной структурой и необходимостью выполнения сквозных переносов, число которых пропорционально разрядности процессора. Таким

© Я.Н. Николайчук, М.Н. Касянчук, И.З. Якименко, 2014

образом, использование позиционных систем счисления приводит к существенному уменьшению быстродействия, увеличению вычислительной и временной сложности использующихся алгоритмов. Особенно это актуально для повышения эффективности обработки многоразрядных чисел [2].

Таким образом, для решения многих научных, технических и прикладных задач мощности современных компьютеров может быть недостаточно. Несмотря на то, что ресурсы сверхновой вычислительной техники, которая функционирует в позиционной системе счисления, постоянно совершенствуются и увеличиваются, они в принципе не могут быть безграничными. Это означает, что обширные классы существующих методов и алгоритмов в пределах быстродействия, обеспечивающегося современными вычислительными системами, не могут быть реализованы практически, т.е. позиционные системы счисления в настоящее время исчерпывают свои возможности для построения высокопроизводительных вычислительных систем. Фундаментальная стратегия теоретических и практических исследований заключается в использовании подходов, основанных на интенсивном применении в вычислительных системах различных форм параллелизма. Данной особенностью обладают непозиционные коды с параллельной структурой. Наиболее перспективная из них — система остаточных классов (СОК), которая позволяет реализовать идею распараллеливания на уровне выполнения элементарных арифметических операций сложения, вычитания и умножения [7, 8]. Представление операндов в виде остатков от деления на сравнительно небольшие взаимно простые модули позволяет избежать межразрядных переносов и намного уменьшить числа, над которыми выполняются операции. Кроме того, СОК, благодаря своему природному внутреннему параллелизму, в последние годы выдвигается как наиболее приоритетная базовая основа для передовых высокопроизводительных компьютерных технологий, в частности, таких как мультипроцессорная [9], суперкомпьютерная, нейросетевая [10] и т.д.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОК

Фундаментальной основой СОК является теория чисел, в частности китайская теорема об остатках [11]. Любое целое положительное число N в десятичной системе счисления представляется в СОК в виде остатков $(b_1, b_2, \ldots, b_k)_{p_1, p_2, \ldots, p_k}$ от деления N на каждый из попарно взаимно простых модулей: $N_{10} = (b_1, b_2, \ldots, b_k)_{p_1, p_2, \ldots, p_k}$, где $b_i = N \bmod p_i$, k — количество модулей. При этом должно выполняться условие $N \le P - 1$ $\left(P = \prod_{i=1}^{k} p_i\right)$.

Обратное преобразование из базиса Крестенсона в десятичную систему счисления достаточно громоздкое и основывается на использовании китайской теоремы об остатках [12]:

$$N = \left(\sum_{i=1}^{k} b_i B_i \right) \bmod P, \qquad (1)$$

где $B_i = M_i m_i$, $M_i = P / p_i$, m_i находится из выражения $(M_i m_i) \bmod p_i = 1$ и должно выполняться условие $\left(\sum_{i=1}^{k} B_i \right) \bmod P = 1$.

В настоящее время известно три способа поиска обратного элемента: 1) последовательным перебором m_i, пока не будет выполняться условие $M_i m_i \bmod p_i = 1$; 2) с помощью функции Эйлера: $m_i = M_i^{-1} \bmod p_i = M_i^{\varphi(p_i)-1} \bmod p_i$; 3) с помощью расширенного алгоритма Евклида.

Все они достаточно громоздки, требуют больших вычислительных затрат и временных ресурсов при выполнении делений с остатком, возведении в степень, нахождении функции Эйлера. Причем все операции должны выполняться над очень большими числами, что может привести к переполнению разрядной

сетки. К недостаткам следует отнести также отсутствие возможности аналитического определения обратного элемента.

Я.Н. Николайчук предложил совершенную форму СОК (СФ СОК), в которой подбор модулей такой, что $M_i \bmod p_i = 1$, т.е. $m_i = 1$ [13]. Дальнейшее развитие данная теория получила в работах [14, 15]. Показан метод выбора системы модулей для СФ СОК и разработана ее модификация, в которой $m_i = \pm 1$. Однако в этих случаях не совсем рационально используются регистры разрядной сетки, количество которых, как правило, равно степени двойки.

Исходя из сказанного выше, цель данной публикации — разработка методов подбора модулей для эффективного использования регистров разрядной сетки, а также выведение аналитической формулы для поиска обратного элемента при соответствующем подборе модулей.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АНАЛИТИЧЕСКОГО НАХОЖДЕНИЯ КОЭФФИЦИЕНТОВ БАЗИСНЫХ ЧИСЕЛ СОК

Рассмотрим набор модулей:

$$\begin{cases} p_1 = 2^n - 1, \\ p_2 = 2^n + 1, \\ p_3 = 2^{2n} + 1, \\ p_4 = 2^{4n} + 1, \\ \dots \\ p_i = 2^{n \cdot 2^{i-2}} + 1, \\ \dots \\ p_{k-1} = 2^{n \cdot 2^{k-3}} + 1, \\ p_k = 2^{n \cdot 2^{k-2}} + 1. \end{cases} \tag{2}$$

Из системы (2) нетрудно видеть, что каждый следующий модуль на две единицы больше от произведения всех предыдущих. Этим определяется взаимная простота модулей, поскольку все они нечетны. Кроме того, диапазон рассматриваемых десятичных чисел для возможных расчетов ограничивается выражением $P = 2^{n \cdot 2^{k-1}} - 1$, где n — степень двойки в модуле p_1.

Для нахождения обратного элемента $m_i = M_i^{-1} \bmod p_i$ запишем систему уравнений:

$$\begin{cases} M_1 \bmod (2^n - 1) = (2^n + 1)(2^{2n} + 1)(2^{4n} + 1) \dots (2^{n \cdot 2^{i-2}} + 1) \dots \\ \qquad \dots (2^{n \cdot 2^{k-3}} + 1)(2^{n \cdot 2^{k-2}} + 1) \bmod (2^n - 1); \\ M_2 \bmod (2^n + 1) = (2^n - 1)(2^{2n} + 1)(2^{4n} + 1) \dots (2^{n \cdot 2^{i-2}} + 1) \dots \\ \qquad \dots (2^{n \cdot 2^{k-3}} + 1)(2^{n \cdot 2^{k-2}} + 1) \bmod (2^n + 1); \\ M_3 \bmod (2^{2n} + 1) = (2^{2n} - 1)(2^{4n} + 1) \dots (2^{n \cdot 2^{i-2}} + 1) \dots \\ \qquad \dots (2^{n \cdot 2^{k-3}} + 1)(2^{n \cdot 2^{k-2}} + 1) \bmod (2^{2n} + 1); \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ M_i \bmod (2^{n \cdot 2^{i-2}} + 1) = (2^{n \cdot 2^{i-2}} - 1) \dots (2^{n \cdot 2^{k-3}} + 1)(2^{n \cdot 2^{k-2}} + 1) \bmod (2^{n \cdot 2^{i-2}} + 1); \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ M_{k-1} \bmod (2^{n \cdot 2^{k-3}} + 1) = (2^{n \cdot 2^{k-3}} - 1)(2^{n \cdot 2^{k-2}} + 1) \bmod (2^{n \cdot 2^{k-3}} + 1); \\ M_k \bmod (2^{n \cdot 2^{k-2}} + 1) = (2^{n \cdot 2^{k-2}} - 1) \bmod (2^{n \cdot 2^{k-2}} + 1). \end{cases} \tag{3}$$

В первом уравнении (3) для каждого множителя правой части получается остаток 2, поэтому $M_1 \bmod (2^n - 1) = 2^{k-1} \bmod (2^n - 1)$. Во втором уравнении (3) остаток от первого множителя равняется –2, все остальные равняются 2, поэтому $M_2 \bmod (2^n + 1) = -2^{k-1} \bmod (2^n + 1)$.

Во всех остальных уравнениях, аналогично второму, первый остаток составляет –2, другие равняются 2, причем с увеличением номера уравнения на единицу количество множителей (соответственно, и двоек) тоже уменьшается на единицу. В результате таких вычислений получим систему:

$$\begin{cases} M_1 \bmod (2^n - 1) = 2^{k-1} \bmod (2^n - 1); \\ M_2 \bmod (2^n + 1) = -2^{k-1} \bmod (2^n + 1); \\ M_3 \bmod (2^{2n} + 1) = -2^{k-2} \bmod (2^{2n} + 1); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ M_i \bmod (2^{n \cdot 2^{i-2}} + 1) = -2^{k-(i-1)} \bmod (2^{n \cdot 2^{i-2}} + 1); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ M_{k-1} \bmod (2^{n \cdot 2^{k-3}} + 1) = -4; \\ M_k \bmod (2^{n \cdot 2^{k-2}} + 1) = -2. \end{cases} \tag{4}$$

Теперь ищем величины $m_i = M_i^{-1} \bmod p_i$. В первом уравнении степень двойки в правой части запишем так: $k - 1 = a_1 n + k_1$, где $0 \le k_1 < n$. Тогда $2^{k-1} \bmod (2^n - 1) = (2^n)^a \cdot 2^{k_1} \bmod (2^n - 1) = 2^{k_1} \bmod (2^n - 1)$. Отсюда видно, что

$$m_1 = \frac{2^n}{2^{k_1}} = 2^{n-k_1}. \tag{5}$$

Аналогично из второго уравнения имеем $k - 1 = a_2 n + k_2$. Тогда

$$-2^{k-1} \bmod (2^n + 1) = -(2^n)^{a_2} \cdot 2^{k_2} \bmod (2^n + 1) = -(-1)^{a_2} \cdot 2^{k_2} \bmod (2^n + 1) =$$

$$= (-1)^{a_2+1} \cdot 2^{k_2} \bmod (2^n + 1) = \begin{cases} 2^{k_2} \bmod (2^n + 1), & a_2 \text{ нечетное}, \\ -2^{k_2} \bmod (2^n + 1), & a_2 \text{ четное}. \end{cases}$$

Рассмотрим два возможных случая.

1. a_2 нечетное; к $2^n + 2$ нужно добавить модуль $2^n + 1$ столько раз, чтобы сумма делилась на 2^{k_2} (т.е. второе слагаемое должно равняться 2^{k_2}). Итак,

$$2^n + 2 + (2^n + 1)(2^{k_2} - 2) = 2^n + 2 + 2^{n+k_2} + 2^{k_2} - 2 \cdot 2^n - 2 = 2^{n+k_2} - 2^n + 2^{k_2}.$$

Поделив на 2^{k_2}, получим обратный элемент: $m_2 = 2^n - 2^{n-k_2} + 1$.

2. a_2 четное; полученное значение m_2 нужно записать с противоположным знаком и прибавить модуль $m_2 = -(2^n - 2^{n-k_2} + 1) \bmod (2^n + 1) = 2^{n-k_2}$. Итак, окончательная формула имеет вид

$$m_2 = \begin{cases} 2^n - 2^{n-k_2} + 1, & a_2 \text{ нечетное}, \\ 2^{n-k_2}, & a_2 \text{ четное}. \end{cases} \tag{6}$$

Аналогично из третьего уравнения

$$m_3 = \begin{cases} 2^{2n} - 2^{2n-k_3} + 1, & a_3 \text{ нечетное}, \\ 2^{2n-k_3}, & a_3 \text{ четное}, \end{cases} \tag{7}$$

где k_3 и a_3 определяются из равенства $k - 2 = 2na_3 + k_3$.

Таблица 1

n	p_1	M_1	m_1	p_2	M_2	m_2	p_3	M_3	m_3	p_4	M_4	m_4	p_5	M_5	m_5	P
2	2^2-1	1	1	2^2+1	4	4	2^4+1	2^4-7	2	2^8+1	2^8-3	2^6	$2^{16}+1$,	$2^{16}-1$	2^{15}	$2^{32}-1$
3	2^{3-1}	2	4	2^3+1	2	5	2^6+1	2^6-7	2^3	$2^{12}+1$	$2^{12}-3$	2^{10}	$2^{24}+1$	$2^{24}-1$	2^{23}	$2^{48}-1$
4	2^4-1	1	1	2^4+1	1	1	2^8+1	2^8-7	2^5	$2^{16}+1$	$2^{16}-3$	2^{14}	$2^{32}+1$	$2^{32}-1$	2^{31}	$2^{64}-1$
5	2^5-1	16	2	2^5+1	2^5-15	2	$2^{10}+1$	$2^{10}-7$	2^7	$2^{20}+1$	$2^{20}-3$	2^{18}	$2^{40}+1$	$2^{40}-1$	2^{39}	$2^{80}-1$
6	2^6-1	16	2^2	2^6+1	2^6-15	2^2	$2^{12}+1$	$2^{12}-7$	2^9	$2^{24}+1$	$2^{24}-3$	2^{22}	$2^{48}+1$	$2^{48}-1$	2^{47}	$2^{96}-1$
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
i	2^i-1	16	2^{i-4}	2^i+1	$2^{i-}15$	2^{i-4}	$2^{2i}+1$	$2^{2i}-7$	2^{2i-3}	$2^{4i}+1$	$2^{4i}-3$	2^{4i-2}	$2^{8i}+1$	$2^{8i}-1$	2^{8i-1}	$2^{16i}-1$

Из i-го уравнения имеем

$$m_i = \begin{cases} 2^{n\cdot 2^{i-2}} - 2^{n\cdot 2^{i-2}-k_i} + 1, & a_i \text{ нечетное}, \\ 2^{n\cdot 2^{i-2}-k_i}, & a_i \text{ четное}, \end{cases} \tag{8}$$

где k_i и a_i определяются из равенства $k-(i-1) = 2^{i-2}na_i + k_i$.

Рассмотрим $(k-1)$-е уравнение. Нет необходимости расписывать k_{i-1}, поскольку $2^2 < 2^{n\cdot 2^{k-3}}$. Дважды добавив модуль к $2^{n\cdot 2^{k-3}}+2$ и поделив на -4, получим

$$m_{k-1} = 2^{n\cdot 2^{k-3}-2}. \tag{9}$$

Аналогично для последнего k-го уравнения $\dfrac{2^{n\cdot 2^{k-2}}+2}{-2} = -(2^{n\cdot 2^{k-2}-1}+1)$. Добавив модуль, получим

$$m_k = 2^{n\cdot 2^{k-2}-1}. \tag{10}$$

В табл. 1 приведены значения p_i, M_i, m_i, а также диапазон возможных расчетов для $k=5$ и различных значений n. Из таблицы видно, что величины M_1, m_1, M_2, m_2 при малых n могут принимать различные значения, что зависит от четности или нечетности коэффициента a_i. Другие значения M_i, m_i имеют вид, соответствующий степени двойки.

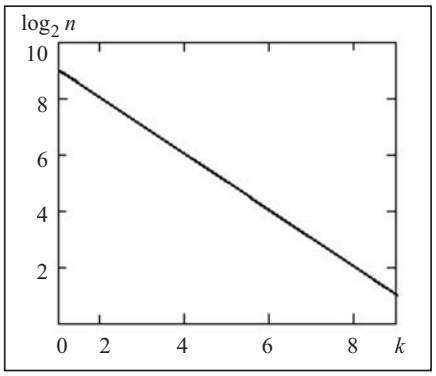

Рис. 1. График логарифмической зависимости степени n для модуля p_1 от количества модулей k

На рис. 1 показана логарифмическая зависимость степени n для модуля p_1 от количества модулей k для 512-разрядного процессора согласно выражению $n = 2^{10-k}$. Из рисунка видно, что $\log_2 n$ линейно убывает с увеличением количества модулей k.

ЗАКЛЮЧЕНИЕ

Соответствующий подбор модулей позволяет достичь эффективного использования всех регистров разрядной сетки, а также получить аналитическую формулу для поиска обратного элемента m_i, что существенно уменьшает количество операций, необходимых для перевода чисел из СОК в десятичную систему счисления.

СПИСОК ЛИТЕРАТУРЫ

1. Мельник А.А. Архитектура компьютера. — Луцк: Волынская обласная типография, 2008. — 470 с.
2. Задірака В.К., Олексюк О.С. Комп'ютерна арифметика багаторозрядних чисел. — Київ: Вища шк., 2003. — 264 с.

3. Фергюссон Н., Шнайер Б. Практическая криптография: Пер. с англ. — М.: Издательский дом «Вильямс», 2005. — 424 с.
4. Задірака В., Олексюк О. Комп'ютерна криптологія. — Київ: Вища шк., 2002. — 504 с.
5. Николайчук Я.Н. Теория источников информации. — Тернополь: ООО «Терно-граф», 2010. — 536 с.
6. Рабинович З.Л., Раманаускас В.А. Типовые операции в вычислительных машинах. — Киев: Техніка, 1980. — 264 с.
7. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. — М: Сов. радио, 1968. — 440 с.
8. Торгашев В.А. Система остаточных классов и надежность ЦВМ. — М.: Сов. радио, 1973. — 117 с.
9. Николайчук Я.Н., Волынский О.И., Кулына С.В. Теоретические основы построения и структура спецпроцессоров в базисе Крестенсона // Вестн. Хмельн. нац. ун-та. — 2007. – 1, № 3. — С. 85–90.
10. Применение искусственных нейронных сетей и систем остаточных классов в криптографии / Н.И. Червяков, А.И. Галушкин, А.А. Евдокимов, А.В. Лавриненко, И.Н. Лавриненко. — М.: Физматлит, 2012. — 270 с.
11. Бухштаб А.А. Теория чисел. — М.: Просвещение, 1966. — 384 с.
12. Бородин О.И. Теория чисел. — Киев: Высш. шк., 1970. — 275 с.
13. Николайчук Я.М. Разработка теории и комплексов технических средств формирования, передачи и обработки цифровых сообщений в низовых вычислительных сетях автоматизированных систем: Дис. … доктор техн. наук. — Киев: Академия наук УССР, Ин-т кибернетики им. В.М. Глушкова, 1991. — 573 с.
14. Касянчук М.Н. Теория и математические закономерности совершенной формы системы остаточных классов // Тр. Междунар. симпозиума «Вопросы оптимизации вычислений (ПОО–XXXV)». — Т. 1. — Киев; Кацивели. — 2009. — С. 306–310.
15. Касянчук М. Концепция теоретических положений совершенной формы преобразования Крестенсона и его практическое применение // Оптико-электронные информационно-энергетические технологии. — 2010. — № 2 (20). — С. 43–48.

Поступила 03.12.2013