

УДК 004.021

ТАБЛИЧНЫЙ МЕТОД ПРЯМОГО ПРЕОБРАЗОВАНИЯ ДВОИЧНЫХ ЧИСЕЛ В СИСТЕМУ ОСТАТОЧНЫХ КЛАССОВ С МОДУЛЯМИ $\{2^f - 1\}$

Исупов К.С., Князьков В.С.

ФГБОУ ВПО «Вятский государственный университет», Киров, e-mail: isupov.k@gmail.com

Проведен анализ классического и табличного методов прямого преобразования двоичных чисел в систему остаточных классов (СОК). Учитывалась временная и ёмкостная сложность параллельных алгоритмов, основанных на исследуемых методах. Выявлены основные недостатки исследуемых методов: классический метод имеет высокую временную сложность ввиду экспоненциальной формы зависимости количества выполняемых операций вычитания от числа оснований (модулей) СОК p_i , $i = 1, 2, \dots, n$; табличный метод прямого преобразования для оснований произвольного вида требует хранения подстановочных таблиц большого объема, в результате чего они не могут быть размещены в блоках КЭШ памяти первого уровня, а обращение к КЭШ памяти второго и последующего уровней приводит к существенному снижению скорости прямого преобразования. Предложен новый табличный метод преобразования двоичных чисел в систему остаточных классов с основаниями $\{p_i = 2^{f_i} - 1\}$, отличающийся тем, что в подстановочной таблице хранится не само значение остатка по основанию p_i , а номер соответствующего бита унарного кода, который следует установить значением логической единицы для получения значения остатка по основанию p_i . Это позволяет ускорить в 2,43 раза выполнение преобразования двоичных чисел в СОК относительно известного табличного метода для оснований произвольного вида, сократив при этом ёмкостную сложность алгоритма (размер подстановочной таблицы) приблизительно в $k/\log_2 k$ раз, где k – разрядность каждого основания p_i , а $\log_2 k$ – округленный в большую сторону логарифм $\log_2 k$.

Ключевые слова: система остаточных классов, прямое преобразование, подстановочные таблицы

LOOK-UP TABLE METHOD OF FORWARD BINARY TO RESIDUE NUMBER SYSTEM CONVERSION FOR MODULI $\{2^f - 1\}$

Isupov K.S., Knyazkov V.S.

Vyatka State University, Kirov, e-mail: isupov.k@gmail.com

Fast forward conversion of binary numbers to residue number system (RNS) is a very important issue. We have done the analysis of the classical and look-up table methods of forward binary to RNS conversion. Take into account the time and space complexity of parallel algorithms based on these methods. The main disadvantages these methods identified: classical method has a high time complexity, due to the exponential depending number of operations of subtraction on the number of RNS bases (modules) p_i , $i = 1, 2, \dots, n$. Look-up table method of forward conversion for arbitrary bases requires storing a large size look-up tables, and these tables cannot be placed in the cache L1 blocks, and an appeal to the cache memory of the second (L2) and subsequent levels leads to a significant reduction of forward conversion rate. A new look-up table method for binary to residue number system with bases $\{p_i = 2^{f_i} - 1\}$ proposed. Its main difference is that in the look-up table is not stored residue for base p_i and stored number of the corresponding bits of the unary code, which should be set the value of the logical unit to obtain the residue for base p_i . This speeds up to 2,43 times the performance of converting binary numbers in the RNS with respect to the known look-up table method for the arbitrary modules and reduce space complexity (the size of look-up table) approximately $k/\log_2 k$ times, where k is capacity of each base p_i , $\log_2 k$ – rounded up $\log_2 k$.

Keywords: residue number system, forward conversion, look-up tables

В высокоточных разрядно-параллельных вычислениях система остаточных классов (СОК) обладает рядом бесспорных преимуществ перед позиционными системами счисления. Она позволяет обрабатывать отдельные знакопозиции многоразрядных чисел без необходимости учета переносов между ними, поэтому хорошо подходит для многоядерной архитектуры современных процессоров универсального назначения. Тем не менее эти преимущества омрачены сложностью преобразования чисел между позиционным и модулярным представлениями.

Получение модулярного числа на основе известного позиционного называется прямым преобразованием. Получение позиционного числа на основе известного модулярного называется, в свою очередь, обратным преобразованием. Несмотря на то,

что обратное преобразование является более сложной операцией, прямое преобразование также далеко не простая процедура. В общем случае разрядность преобразуемого позиционного (двоичного) числа и число оснований СОК могут оказаться большими, вследствие чего операция прямого преобразования может оказаться сильно растянутой во времени и поэтому требует проведения отдельного исследования.

1. Формальная постановка задачи прямого преобразования

Требуется преобразовать двоичное число $M' = d_t d_{t-1} \dots d_2 d_1$ в СОК с попарно взаимно простыми основаниями $p_1, p_2, \dots, p_n$, получив модулярное число

$$\tilde{M} = \langle m_1, m_2, \dots, m_n \rangle \text{ такое, что}$$

$$\tilde{M} = \langle |M'|_{p_1}, |M'|_{p_2}, \dots, |M'|_{p_n} \rangle,$$

где $m_i \in [0, p_i - 1]$ – i -я знакопозиция (модулярный разряд) модулярного числа $\tilde{M}$; $|M'|_{p_i}$ – операция получения остатка от деления M' на i -е основание p_i . Разрядная сетка вычислительных устройств, на которых выполняется преобразование, определяется значением k , а преобразуемое t -разрядное число $M' = d_t d_{t-1} \dots d_2 d_1$ в общем случае выходит за ее пределы, т.е. $t > k$, при этом, если $t = k \cdot v$, то t -разрядные арифметические операции представляются в виде v последовательно выполняемых k -разрядных операций.

2. Классический метод прямого преобразования

Классический метод преобразования позиционного числа в систему остаточных классов сводится к нахождению значения каждой i -й знакопозиции (модулярного разряда) $m_i = |M'|_{p_i}$, $i = 1, 2, \dots, n$, по формуле

$$m_i = |M'|_{p_i} = M' - \left\lfloor \frac{M'}{p_i} \right\rfloor \cdot p_i, \quad (1)$$

где $\left\lfloor \frac{M'}{p_i} \right\rfloor$ – наибольшее целое, не превышающее $\frac{M'}{p_i}$.

Другими словами, каждый модулярный разряд m_i представляет собой наименьший неотрицательный целочисленный остаток от деления исходного позиционного целого числа M' на соответствующее целочисленное основание p_i .

Оценка временных затрат. Для общности рассмотрения будем считать, что операция (1) нахождения остатка от деления M' на p_i реализуется посредством выполнения группы операций вычитания.

Пусть t – разрядность преобразуемого двоичного числа $M' = d_t d_{t-1} \dots d_2 d_1$, k – разрядность основания p_i . Условимся, что $p_i \approx 2^k$, а $M' = 2^t - 1$.

Диапазон изменения чисел $\tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$ в системе остаточных классов с основаниями $p_1, p_2, \dots, p_n$ определяется интервалом

$$\left[0, \prod_{i=1}^n (p_i) - 1 \right],$$

а, с учетом условия $p_i \approx 2^k$, $i = 1, 2, \dots, n$,

$$M' = 2^{t-1} \cdot d_t + 2^{t-2} \cdot d_{t-1} + \dots + 2^2 \cdot d_3 + 2^1 \cdot d_2 + 2^0 \cdot d_1 = \sum_{j=1}^t (2^{j-1} \cdot d_j), \quad (2)$$

$$\left[0, \prod_{i=1}^n (2^k) - 1 \right].$$

Тогда $M' = 2^t - 1 \approx \prod_{i=1}^n (2^k) - 1$ – максимальное число, представимое в системе остаточных классов с основаниями $p_1, p_2, \dots, p_n$. Следовательно,

$$t \approx \log_2 \left(\prod_{i=1}^n 2^k \right) \approx \log_2 (2^{k \cdot n}) \approx k \cdot n.$$

Для вычисления каждой знакопозиции m_i по формуле (1) потребуется $(2^{k \cdot n} - 1)$ t -разрядных операций вычитания. Пусть $T_{sub(k)}$ – время выполнения операции вычитания двоичных k -разрядных чисел. Операция t -разрядного вычитания, с учетом условия $t \approx k \cdot n$, сопоставима по времени с n последовательными операциями k -разрядного вычитания, т.е. $T_{sub(t)} \approx n \cdot T_{sub(k)}$. Следовательно, для параллельного вычисления всех знакопозиций $m_1, m_2, \dots, m_n$ потребуется время, приближенно равное

$$(2^{k \cdot n} - 1) \cdot n \cdot T_{sub(k)}.$$

Недостатки. При большом количестве оснований $p_1, p_2, \dots, p_n$ допустимая разрядность t позиционного числа $M' = d_t d_{t-1} \dots d_2 d_1$ может быть много больше разрядности k оснований p_i , $i = 1, 2, \dots, n$. Поэтому операция определения модулярного числа $\tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$ путем вычисления выражения (1) для каждой знакопозиции m_i , $i = 1, 2, \dots, n$, может оказаться существенно растянутой во времени ввиду экспоненциальной формы зависимости количества выполняемых операций вычитания от числа оснований $p_1, p_2, \dots, p_n$.

3. Табличный метод прямого преобразования для оснований произвольного вида

Получение модулярного числа $\tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$ можно осуществить посредством использования подстановочной таблицы (Look-up Table) [6, с. 267]. Основная идея данного метода преобразования заключается в следующем. Пусть имеется $M' = d_t d_{t-1} \dots d_2 d_1$ – t -разрядное целое двоичное число, изменяющееся в пределах интервала $[0, 2^t - 1]$. Величину данного числа можно выразить посредством взвешенной суммы

где $j = 1, 2, \dots, t$ – позиции двоичных разрядов $d_j \in \{0, 1\}$, в соответствии с которыми однозначно определяются веса этих разрядов в двоичной системе счисления (в данном случае номером младшего разряда является

единица, а не ноль, вследствие чего истинный вес j -го разряда равен $j - 1$).

С учетом (2), выражение для получения i -й знакопозиции $m_i = |M'|_{p_i}$ будет иметь вид

$$m_i = \left| 2^{t-1} \cdot d_t + 2^{t-2} \cdot d_{t-1} + \dots + 2^2 \cdot d_3 + 2^1 \cdot d_2 + 2^0 \cdot d_1 \right|_{p_i} = \left| \sum_{j=1}^t (2^{j-1} \cdot d_j) \right|_{p_i}. \quad (3)$$

Придерживаясь правил выполнения модулярной операции сложения, выражение (3) правомерно записать в виде

$$m_i = \left| 2^{t-1} \cdot d_t \right|_{p_i} + \left| 2^{t-2} \cdot d_{t-1} \right|_{p_i} + \dots + \left| 2^2 \cdot d_3 \right|_{p_i} + \left| 2^1 \cdot d_2 \right|_{p_i} + \left| 2^0 \cdot d_1 \right|_{p_i}.$$

Так как $d_j \in \{0, 1\}, j = 1, 2, \dots, t$, то для нахождения модулярного числа $\tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$ можно воспользоваться подстановочной таблицей (табл. 1), состоящей из n столбцов и t строк. Тогда процесс преобразования

$M' = d_t d_{t-1} \dots d_2 d_1 \rightarrow \tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$ сведется к выбору из подстановочной таблицы значений $|2^{j-1}|_{p_i}, i = 1, 2, \dots, n$ (для разрядов $d_j = 1$), и их суммированию по модулю p_i .

Таблица 1

Подстановочная таблица для преобразования t -разрядных двоичных чисел в систему остаточных классов с n основаниями

j	$ 2^{j-1} _{p_1}$	$ 2^{j-1} _{p_2}$	...	$ 2^{j-1} _{p_n}$
1	1	1	...	1
2	$ 2 _{p_1}$	$ 2 _{p_2}$	...	$ 2 _{p_n}$
3	$ 2^2 _{p_1}$	$ 2^2 _{p_2}$	...	$ 2^2 _{p_n}$
...	...	...	...	...
t	$ 2^{t-1} _{p_1}$	$ 2^{t-1} _{p_2}$	...	$ 2^{t-1} _{p_n}$

Например, в табл. 2 приведены все значения $|2^{j-1}|_{p_i}, i = 1, \dots, 4, j = 1, \dots, 8$, для ос-

нований $p_1 = 3, p_2 = 5, p_3 = 7, p_4 = 11$ и разрядности двоичных слов до 8 бит.

Таблица 2

Значения $|2^{j-1}|_{p_i}$ для преобразования двоичных чисел разрядности вплоть до 11 бит в СОК с основаниями 3, 5, 7, 11

j	2^{j-1}	$ 2^{j-1} _3$	$ 2^{j-1} _5$	$ 2^{j-1} _7$	$ 2^{j-1} _{11}$
1	1	1	1	1	1
2	2	2	2	2	2
3	4	1	4	4	4
4	8	2	3	1	8
5	16	1	1	2	5
6	32	2	2	4	10
7	64	1	4	1	9
8	128	2	3	2	7

Пример. Пусть имеется восьмибитное целое число $M' = 10001001_2 = 137_{10}$. Необходимо определить модулярное число в системе остаточных классов с основаниями $p_1 = 3, p_2 = 5, p_3 = 7, p_4 = 11$.

1. Определяем позиции ненулевых бит в двоичном числе $M' = 10001001_2; j = 1, 4, 8$, поэтому

$$M' = 2^{1-1} + 2^{4-1} + 2^{8-1}.$$

2. Определяем знакопозиции m_i , $i = 1, 2, 3, 4$, модулярного числа $\tilde{M} = \langle m_1, m_2, m_3, m_4 \rangle$, выполняя для каждого основания p_i выборку из подстановочной таблицы (табл. 2) и суммирование значений:

$$m_1 = |1 + 2 + 2|_3 = 2; \quad m_2 = |1 + 3 + 3|_5 = 2;$$

$$m_3 = |1 + 1 + 2|_7 = 4; \quad m_4 = |1 + 8 + 7|_{11} = 5.$$

Получили $\tilde{M} = \langle 2, 2, 4, 5 \rangle$.

Таким образом, использование подстановочных таблиц позволяет получать модулярные представления чисел без необходимости вычисления выражения (1).

Оценка временных затрат. Пусть $T_{cmp(t)}$ – время выполнения битового анализа t -разрядного двоичного слова, $T_{L2(k)}$ – время выборки из памяти ЭВМ k -разрядного двоичного слова (имеется ввиду время доступа к КЭШ памяти второго уровня), а $T_{add(k)}$ – время выполнения операции сложения двух двоичных k -разрядных чисел. Пусть

$$p_i \approx 2^k, \quad M' = 2^t - 1.$$

Для получения каждой знакопозиции m_i на основании табличного метода необходимо выполнить одну операцию битового анализа t -разрядного двоичного числа для определения весов ненулевых разрядов, t обращений к подстановочной таблице за

$$T_{cmp(t)} + t \cdot T_{L2(k)} + (t-1) \cdot T_{add(k)} + (t-1) \cdot (T_{sub(k)} + T_{sub(\lceil \log_2 t \rceil)}),$$

а так как $t \approx k \cdot n$, то с учетом несложных преобразований получаем время

$$n \cdot (T_{cmp(k)} + k \cdot T_{L2(k)}) + (n \cdot k - 1) \cdot (T_{add(k)} + T_{sub(k)} + T_{sub(\lceil \log_2(k \cdot n) \rceil)}).$$

Недостатки. При большом количестве оснований подстановочные таблицы становятся большими, в результате чего не могут быть полностью записаны в модули памяти, располагающиеся непосредственно на вычислительном ядре (т.е. в КЭШ первого уровня L1), а доступ к КЭШ памяти второго уровня осуществляется значительно медленнее. Например, при использовании десяти 64-битных оснований для хранения общей подстановочной таблицы, содержащей значения $|2^{j-1}|_{p_i}$ для двоичных слов вплоть до $2^{64 \cdot 10}$, потребуется $10 \cdot 640 \cdot 64/8 = 51200$ байт памяти, в то время, как объем КЭШ памяти L1 современных универсальных процессоров на каждое ядро составляет в среднем 32–48 КБ [3–5]. Размещение подстановочных таблиц в КЭШ памяти второго и последующего уровней неизбежно приводит к тому, что время $T_{L2(k)}$

выборкой k -разрядных значений $|2^{j-1}|_{p_i}$, соответствующих весам j ненулевых бит

d_j , $(t-1)$ операций суммирования, выбранных из таблицы k -разрядных двоичных слов $|2^{j-1}|_{p_i}$, и одну операцию получения

остатка от деления суммы $\sum_{j=1}^t (2^{j-1} \cdot d_j)$ на i -е основание p_i . В предельном случае сум-

ма $\sum_{j=1}^t (2^{j-1} \cdot d_j)$ состоит из t чисел, разрядность которых не превышает k , поэтому операцию получения остатка от деления $\sum_{j=1}^t (2^{j-1} \cdot d_j)$ на i -е основание p_i можно представить в виде $(t-1)$ операции вычитания k -разрядных чисел из $(k + \lceil \log_2 t \rceil)$ -разрядного числа, где $\lceil \log_2 t \rceil$ – округленный до целого в большую сторону логарифм $\log_2 t$. Тогда, считая что

$$T_{sub(k + \lceil \log_2 t \rceil)} = T_{sub(k)} + T_{sub(\lceil \log_2 t \rceil)},$$

время параллельного преобразования двоичного t -разрядного числа в СОК табличным методом будет приближенно определяться выражением

$$(T_{sub(k)} + T_{sub(\lceil \log_2 t \rceil)}),$$

выборки k -разрядного двоичного слова становится весьма большим относительно времени выполнения арифметических операций сложения, вычитания и умножения, которые работают либо с регистрами общего назначения, либо с КЭШ L1. В результате этого значительным образом снижается скорость преобразования

$$M' = d_t d_{t-1} \dots d_2 d_1 \rightarrow \tilde{M} = \langle m_1, m_2, \dots, m_n \rangle.$$

4. Табличный метод прямого преобразования для оснований 2^f-1

Предлагается метод прямого преобразования двоичных чисел в СОК, позволяющий ценой введения ограничений на допустимую форму оснований $p_1, p_2, \dots, p_n$ избавиться от недостатков рассмотренных методов.

Согласно формуле (3), модулярное число M можно выразить следующим образом:

$$\tilde{M} = \left\langle \left| \sum_{j=1}^t (2^{j-1} \cdot d_j) \right|_{p_1}, \left| \sum_{j=1}^t (2^{j-1} \cdot d_j) \right|_{p_2}, \dots, \left| \sum_{j=1}^t (2^{j-1} \cdot d_j) \right|_{p_n} \right\rangle, \quad (4)$$

где $d_j \in \{0,1\}$ – j -й бит двоичного числа

$$M' = d_t, d_{t-1}, \dots, d_j, \dots, d_2, d_1$$

Выражение (4) можно модифицировать, используя известные результаты теории чисел, в частности, свойства чисел, имеющих форму $2^f - 1$, где $f \geq 0$ – натуральное число. Для чисел вида $2^f - 1$ существует тождество [1, с. 307]

$$|2^j - 1|_{2^f - 1} = 2^{|j|_f} - 1, \quad (5)$$

которое позволяет сформулировать утверждение о получении значения величины $|2^j|_{2^f - 1}$.

Утверждение 1. Зная позиционное значение неотрицательного целого показателя степени j , величину 2^j по положительному модулю p при условии, что $p = 2^f - 1$ и $f \geq 2$, можно определить как $2^{|j|_f}$. Другими словами, при выборе модуля p из множества чисел $\{2^f - 1 \mid f \geq 2, f \in \mathbb{N}\}$ справедливо тождественное равенство

$$\tilde{M} = \left\langle \left| \sum_{j=1}^t (2^{|j-1|_{f_1}} \cdot d_j) \right|_{p_1}, \left| \sum_{j=1}^t (2^{|j-1|_{f_2}} \cdot d_j) \right|_{p_2}, \dots, \left| \sum_{j=1}^t (2^{|j-1|_{f_n}} \cdot d_j) \right|_{p_n} \right\rangle,$$

где

$$f_1 = \log_2(p_1 + 1); \quad f_2 = \log_2(p_2 + 1); \dots,$$

$$f_n = \log_2(p_n + 1),$$

при условии, что

$$p_1 = 2^{f_1} - 1, \quad p_2 = 2^{f_2} - 1, \quad \dots, \quad p_n = 2^{f_n} - 1.$$

$$\tilde{M} = \left\langle |M'|_{p_1}, |M'|_{p_2}, \dots, |M'|_{p_n} \right\rangle = \left\langle \left| \sum_{j=1}^t b_j^1 \right|_{p_1}, \left| \sum_{j=1}^t b_j^2 \right|_{p_2}, \dots, \left| \sum_{j=1}^t b_j^n \right|_{p_n} \right\rangle, \quad (7)$$

причем, в соответствии с Утверждением 1, для всех i и j справедливо неравенство $b_j^i < p_i$.

Для ускорения процесса преобразования, по аналогии с рассмотренным табличным методом для оснований произвольного

$$|2^j|_{p=2^f-1} = 2^{|j|_f},$$

где $f \geq 2, j \geq 0$ – натуральные числа, $f = \log_2(p + 1)$.

Доказательство. Так как для любого целого $f \geq 2$ справедливо сравнение $1|_{2^f-1} \equiv 1$, то соотношение $|2^j - 1|_{2^f-1} = 2^{|j|_f} - 1$ эквивалентно тождеству $|2^j|_{2^f-1} - 1 = 2^{|j|_f} - 1$. Отсюда следует, что для любых целых $f \geq 2$ и $j \geq 0$ справедливо тождество

$$|2^j|_{2^f-1} = 2^{|j|_f}.$$

ч.т.д.

Если все основания $p_1, p_2, \dots, p_n$, являющиеся модулями для определения модулярных чисел $\tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$, выбирать из множества чисел вида $2^f - 1$, где f – натуральное число, превышающее единицу, то, в силу Утверждения 1, выражение (4) примет вид

Введем обозначение:

$$b_j^i = \begin{cases} 0, & \text{если } d_j = 0, \\ 2^{|j-1|_{f_i}} & \text{если } d_j = 1, \end{cases} \quad (6)$$

где d_j – j -й бит целого двоичного числа $M' = d_t d_{t-1} \dots d_2 d_1$, i – номер знакопозиции m_i . Тогда

вида, предлагается использовать подстановочную таблицу (табл. 3), однако хранить в этой таблице следует не сами k -разрядные вычеты $|2^{j-1}|_{p_i}$, а вычисленные заранее $\lceil \log_2 k \rceil$ -разрядные значения $|j-1|_{f_i}$ для всех $j = 1, 2, \dots, t$.

Таблица 3

Таблица значений $|j-1|_{f_i}$ для перевода t -разрядных двоичных чисел в систему остаточных классов с n основаниями

j	1	2	3	4	...	t
$ j-1 _{f_1}$	0	1	$ 2 _{f_1}$	$ 3 _{f_1}$	...	$ t-1 _{f_1}$
$ j-1 _{f_2}$	0	1	$ 2 _{f_2}$	$ 3 _{f_2}$	...	$ t-1 _{f_2}$
...	...	...	...	...	...	...
$ j-1 _{f_n}$	0	1	$ 2 _{f_n}$	$ 3 _{f_n}$	...	$ t-1 _{f_n}$

В качестве примера, в табл. 4 приведены все значения $|j-1|_{f_i}$ для оснований $p_1 = 3 = 2^2 - 1, p_2 = 7 = 2^3 - 1, p_3 = 31 = 2^5 - 1,$

$p_4 = 127 = 2^7 - 1$ ($f_1 = 2, f_2 = 3, f_3 = 5, f_4 = 7$) и разрядности t преобразуемых двоичных чисел до 8 бит.

Таблица 4

Значения $|j-1|_{f_i}$ для перевода двоичных чисел разрядности вплоть до 8 бит в СОК с основаниями 3, 7, 31, 127

j	1	2	3	4	5	6	7	8
$ j-1 _2$	0	1	0	1	0	1	0	1
$ j-1 _3$	0	1	2	0	1	2	0	1
$ j-1 _5$	0	1	2	3	4	0	1	2
$ j-1 _7$	0	1	2	3	4	5	6	0

При использовании описанной подстановочной таблицы процесс прямого преобразования

$$M' = d_t d_{t-1} \dots d_2 d_1 \rightarrow \tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$$

будет заключаться в выборе из подстановочной таблицы значений $|j-1|_{f_i}, i = 1, 2, \dots, n$ (только для разрядов $d_j = 1$ двоичного числа $M' = d_t d_{t-1} \dots d_2 d_1$), формировании на основе выбранных значений k -разрядных унарных двоичных кодов $(0_{k-1} 0_{k-2} \dots 1_{|j-1|_{f_i}} \dots 0_1 0_0)_{2^k}$ соответствующих значениям $b_i^j = 2^{|j-1|_{f_i}} = |2^{j-1}|_{p_i}$, и суммированию полученных значений b_i^j по всем модулям p_i , в соответствии с выражением (7), для $j \in [1, t]$ при $d_j = 1$.

Таким образом, основная концепция предлагаемого метода прямого преобразования заключается в том, чтобы хранить в подстановочных таблицах не само значение остатка $|2^{j-1}|_{p_i}$, а номер $|j-1|_{f_i}$ соответствующего бита k -разрядного унарного кода, который следует установить значением логической единицы для получения значения остатка $b_i^j = 2^{|j-1|_{f_i}} = |2^{j-1}|_{p_i}$, в случае, если бит d_j в двоичном числе $M' = d_t d_{t-1} \dots d_2 d_1$ ненулевой.

Рассмотрим на конкретном примере параллельный алгоритм приведения позиционного числа $M' = d_t d_{t-1} \dots d_2 d_1$ к модулярному числу $\tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$, построенный на базе предлагаемого метода.

Пример. Пусть имеется восьмибитное целое двоичное число

$M' = 10001001_2 = 137_{10}$. Необходимо определить модулярное число $\tilde{M} = \langle m_1, m_2, m_3 \rangle$ в системе остаточных классов с основаниями $p_1 = 3 = 2^2 - 1, p_2 = 7 = 2^3 - 1, p_3 = 31 = 2^5 - 1$.

1. Определяем позиции ненулевых бит в двоичном числе $M' = 10001001_2: j = 1, 4, 8$.

2. Выбираем из подстановочной таблицы (табл. 4) набор значений номеров соответствующих бит k -разрядных унарных кодов, которые следует установить значением логической единицы, для каждого основания p_i :

$$p_1: |1-1|_2 = 0, |4-1|_2 = 1, |8-1|_2 = 1;$$

$$p_2: |1-1|_3 = 0, |4-1|_3 = 0, |8-1|_3 = 1;$$

$$p_3: |1-1|_5 = 0, |4-1|_5 = 3, |8-1|_5 = 2.$$

3. На основании определенных позиций ненулевых бит формируем 5-разрядные унарные двоичные коды, соответствующие значениям слагаемых $b_i^j = 2^{|j-1|_{f_i}} = |2^{j-1}|_{p_i}$ для каждого основания p_i (нулевое значение позиции соответствует унарному коду с единицей в младшем разряде, первое значение позиции соответствует коду с единицей в разряде, следующем за младшим и т.д.):

$$p_1: b_1^1 = 00001_2, b_4^1 = 00010_2, b_8^1 = 00010_2;$$

$$p_2: b_1^2 = 00001_2, b_4^2 = 00001_2, b_8^2 = 00010_2;$$

$$p_3: b_1^3 = 00001_2, b_4^3 = 01000_2, b_8^3 = 00100_2.$$

4. Находим модулярное число $\tilde{M} = \langle m_1, m_2, m_3 \rangle$ в соответствии с выражением (7), определяя каждый модулярный разряд m_i как сумму полученных значений b_i^j :

$$\tilde{M} = \left\langle \left| \sum_{j=1}^8 b_j^1 \right|_3, \left| \sum_{j=1}^8 b_j^2 \right|_7, \left| \sum_{j=1}^8 b_j^3 \right|_{31} \right\rangle =$$

$$\left\langle |00001_2 + 00010_2 + 00010_2|_3, |00001_2 + 00001_2 + 00010_2|_7, |00001_2 + 01000_2 + 00100_2|_{31} \right\rangle =$$

$$\langle |1+2+2|_3, |1+1+2|_7, |1+8+4|_{31} \rangle = \langle 2, 4, 13 \rangle.$$

Получили $\tilde{M} = \langle 2, 4, 13 \rangle$.

Подобный метод прямого преобразования был представлен учеными А. Omondi

и В. Premkumar в работе [7, с. 50]. Согласно этому методу, вычисление каждого остатка $|2^{j-1}|_{2^f-1}$ выполняется следующим образом:

$$|2^{j-1}|_{2^f-1} = |2^{q \cdot f + h}|_{2^f-1} = |2^{q \cdot f}|_{2^f-1} \cdot |2^h|_{2^f-1} = 1 \cdot |2^h|_{2^f-1},$$

т.е. выполняется разложение величины 2^{j-1} на сомножители: $2^{q \cdot f}$, остаток от деления на модуль $2^f - 1$ которого дает единицу, и 2^h , остаток от деления на модуль которого дает искомое значение $|2^{j-1}|_{2^f-1}$, т.е. показатель r , по существу, определяется выражением $r = |j-1|_f$. Принципиальным отличием предлагаемого метода от метода, представленного в работе [7, с. 50], является использование подстановочной таблицы.

Предлагаемый метод преобразования позволяет, с одной стороны, избавиться от необходимости выполнения трудоемких операций получения остатка от деления исходного позиционного t -разрядного числа M' на модули p_i , присущих классическому методу, а с другой – сократить размер подстановочных таблиц, тем самым обеспечивая возможность их распределения по модулям КЭШ памяти первого уровня (L1), в результате чего сокращается время доступа к ним. В предельном случае, при работе с k -разрядными основаниями $p_i = 2^k - 1, i = 1, 2, \dots, n$, размер подстановочной таблицы для хранения $\lceil \log_2 k \rceil$ -разрядных значений $|j-1|_{f_i}$ для всех $j = 1, 2, \dots, t$ будет определяться выражением $n \cdot t \cdot \lceil \log_2 k \rceil$, а не $n \cdot t \cdot k$, как при использовании табличного метода преобразования для оснований произвольного вида, предполагающего хранение в таблицах k -разрядных значений $|2^{j-1}|_{p_i}$, т.е. ёмкостная сложность (размер подстановочной таблицы) сокращается приблизительно в $\frac{k}{\lceil \log_2 k \rceil}$ раз, где $\lceil \log_2 k \rceil$ – округленный до целого в большую сторону $\log_2 k$. Например, при использовании десяти 64-битных оснований для хранения общей подстановочной таблицы (табл. 3), содержащей зна-

чения $|j-1|_{f_i}$ для двоичных слов вплоть до 2^{64-10} потребуется

$$10 \cdot 640 \cdot \lceil \log_2 64 \rceil / 8 = 4800 \text{ байт памяти,}$$

что в 10,67 раза меньше, чем при использовании табличного метода преобразования для модулей произвольного вида.

Оценка временных затрат. Пусть $T_{L1}(\lceil \log_2 k \rceil)$ – время выборки из модуля КЭШ памяти первого уровня $\lceil \log_2 k \rceil$ -разрядного двоичного числа, $T_{unary(k)}$ – время генерации k -разрядного унарного кода при известной позиции единичного бита, $T_{add(k)} = T_{sub(k)}$ – время выполнения k -разрядной операции сложения (вычитания) пары двоичных чисел. Пусть $p_i = 2^k - 1, i = 1, 2, \dots, n$, а $M' = 2^t - 1$.

Для получения каждой знакопозиции m_i модулярного числа $\tilde{M} = \langle m_1, m_2, \dots, m_n \rangle$ на основании предлагаемого метода в предельном случае необходимо выполнить одну операцию битового анализа t -разрядного двоичного числа для определения весов ненулевых разрядов (для того, чтобы не выбирать из таблицы избыточные значения $|j-1|_{f_i}$ когда $d_j = 0$ и не вычислять в явном виде выражение (6)), t обращений к подстановочной таблице за выборкой $\lceil \log_2 k \rceil$ -разрядных значений $|j-1|_{f_i}$, t операций генерации k -разрядного унарного кода, $(t-1)$ операций суммирования k -разрядных двоичных унарных кодов, соответствующих значениям b_i^j , и одну операцию получения остатка от деления $\sum_{j=1}^t b_j^i$ на модуль p_i , которую можно представить в виде $(t-1)$ операции вычитания k -разрядных чисел из $(k + \lceil \log_2 t \rceil)$ -разрядного числа.

Следовательно, время параллельного вычисления всех знакопозиций $m_1, m_2, \dots, m_n$ составит

$$T_{cmp(t)} + t \cdot T_{L1(\lceil \log_2 k \rceil)} + t \cdot T_{unary(k)} + (t-1) \cdot T_{add(k)} + (t-1) \cdot (T_{sub(k)} + T_{sub(\lceil \log_2 t \rceil)}).$$

Так как по условиям $t \approx k \cdot n$, то после несложных преобразований полу-

чаем выражение времени преобразования:

$$n \cdot [T_{cmp(k)} + k \cdot (T_{L1(\lceil \log_2 k \rceil)} + T_{unary(k)})] + (n \cdot k - 1) \cdot (T_{add(k)} + T_{sub(k)} + T_{sub(\lceil \log_2(k \cdot n) \rceil)}).$$

5. Сопоставление оценок временной сложности методов прямого преобразования

В качестве базовой единицы времени для сопоставления полученных оценок временной сложности рассмотренных методов будем использовать время T такта вычислителя (далее просто – такт вычислителя), считая, что разрядная сетка имеет размер k , поэтому операции сложения, вычитания, битового анализа и формирования унарного кода в пределах k разрядов выполняются за 1 такт, т.е. $T_{add(k)} = T_{sub(k)} = T_{cmp(k)} = T_{unary(k)} = T$, а операции над числами, разрядность ко-

торых больше k , реализуются посредством выполнения набора k -разрядных операций. Условимся, что время $T_{L1(k)}$ выборки не более k двоичных разрядов из КЭШ памяти первого уровня (L1) составляет 3 такта [2], а время $T_{L2(k)}$ выборки из КЭШ памяти второго уровня (L2) – приблизительно 14 тактов (результаты тестирования показали, что на разных процессорах эта оценка изменяется от 8 тактов до 21 такта [2]). В табл. 5 представлены выраженные в тактах оценки временной сложности параллельных алгоритмов, основанных на рассматриваемых методах преобразования.

Таблица 5

Оценки временной сложности параллельных алгоритмов преобразования

Базовый метод	Полученные оценки временных затрат	Временная сложность, выраженная во времени T выполнения одного такта
Классический метод	$(2^{k \cdot (n-1)} - 1) \cdot n \cdot T_{sub(k)}$	$(2^{k \cdot (n-1)} - 1) \cdot n \cdot T$
Табличный метод для оснований произвольного вида	$n \cdot (T_{cmp(k)} + k \cdot T_{L2(k)}) + (n \cdot k - 1) \cdot (T_{add(k)} + T_{sub(k)} + T_{sub(\lceil \log_2(k \cdot n) \rceil)})$	$T \cdot [n \cdot (1 + 17 \cdot k) - 3]$
Табличный метод для оснований вида	$n \cdot [T_{cmp(k)} + k \cdot (T_{L1(\lceil \log_2 k \rceil)} + T_{unary(k)})] + (n \cdot k - 1) \cdot (T_{add(k)} + T_{sub(k)} + T_{sub(\lceil \log_2(k \cdot n) \rceil)})$	$T \cdot [n \cdot (1 + 7 \cdot k) - 3]$

Таким образом, ценой ограничения допустимой формы оснований $P_1, P_2, \dots, P_n$, удалось снизить временные затраты на преобразование двоичных чисел в систему остаточных классов приблизительно в

$$\frac{T \cdot [n \cdot (1 + 17 \cdot k) - 3]}{T \cdot [n \cdot (1 + 7 \cdot k) - 3]} \approx 2,43 \text{ раза}$$

относительно табличного метода для оснований произвольного вида. Классический метод преобразования не рассчитан на большое число оснований и большую разрядность преобразуемых двоичных чисел, поэтому его временная сложность при больших значениях n и k во много раз превосходит временную сложность табличных методов.

Заключение

Таким образом, предложен новый табличный метод параллельного преобразо-

вания позиционных двоичных чисел в систему остаточных классов с основаниями $P_1, P_2, \dots, P_n$, отличающийся тем, что в качестве оснований выбираются попарно взаимно-простые числа вида $p_i = 2^{f_i} - 1$, $i = 1, 2, \dots, n$, а в подстановочной таблице хранится не само значение остатка $|2^{j-1}|_{p_i}$, а номер $|j-1|_{f_i}$ соответствующего бита k -разрядного унарного кода, который следует установить значением логической единицы для получения значения остатка

$$b_i^j = 2^{|j-1|_{f_i}} = |2^{j-1}|_{p_i},$$

что позволяет ускорить приблизительно в 2,43 раза выполнение процедуры преобразования двоичных чисел в систему остаточных классов относительно известного табличного метода для оснований произвольного вида, сократив при этом емкостную сложность алгоритма (размер под-

становочной таблицы) приблизительно в $\frac{k}{\lceil \log_2 k \rceil}$ раз, где k – разрядность каждого основания p_i .

Ограничением предложенного метода является специальный вид модулей – $2^f - 1$. Однако данных оснований в большинстве случаев оказывается достаточно для обеспечения необходимого диапазона представления чисел в СОК. Кроме этого, предложенный метод с небольшими изменениями может быть адаптирован для оснований вида $2^f + 1$ [7, с. 52]. Использование в качестве оснований СОК чисел из набора

$$\{2^f - 1, 2^f + 1 \mid f \geq 2, f \in \mathbb{N}\}$$

позволит существенно расширить допустимый диапазон представления чисел.

Список литературы

1. Кнут Д. Искусство программирования для ЭВМ. Т. 2. Получисленные алгоритмы. Пер. с англ. – М.: Мир, 1977. – 728 с.
2. Обзор микроархитектур современных десктопных процессоров: Часть 3: организация кэшей данных, внешние интерфейсы процессора, эволюция и ближайшие перспективы развития. – Режим доступа: <http://www.ixbt.com/cpu/cpu-microarchitecture-part-3.shtml>. (дата доступа: 08.06.2012).
3. AMD Processors for Servers and Workstations: AMD Opteron™ Processor. Compare AMD Product Specs: AMD Phenom™, AMD Athlon™, AMD Opteron™, AMD Sempron™, AMD Turion™ 64 Processors Mobile Technology, ATI Radeon Graphics Cards, and AMD Powered Motherboards. – Режим доступа: <http://products.amd.com/en-us/OpteronCPUDetail.aspx?id=756>. (date of access: 04.06.2012).
4. Intel® Core™ i7 Processor Family for the LGA-2011 Socket: Datasheet, Volume 1 of 2. Процессоры Intel для Ultrabook™, ноутбуков, ПК и серверов. – Режим доступа: <http://www.intel.ru/content/dam/doc/datasheet/core-i7-lga-2011-datasheet-vol-1.pdf>. (date of access: 04.06.2012).
5. Intel® Xeon® Processor E7-8800/4800/2800 Product Families: Datasheet, Volume 2 of 2. Процессоры Intel для Ultrabook™, ноутбуков, ПК и серверов. – Режим доступа: <http://www.intel.ru/content/dam/www/public/us/en/documents/datasheets/xeon-e7-8800-4800-2800-families-vol-2-datasheet.pdf>. (date of access: 03.06.2012).
6. Koren I. Computer arithmetic algorithms. 2nd ed. – Natick, MA: A K Peters, 2002. – 281 p.

7. Omondi A. Residue Number Systems theory and Implementation. – London : Imperial College Press, 2007. – 312 p.

References

1. Knut D. Iskusstvo programmirovaniya dlja JeVM. T. 2. Poluchislennyye algoritmy. Per. s angl. Moscow: World, 1977.
2. Obzor mikroarhitektur sovremennyh desktopnyh processorov: Chast' 3: organizacija kjeshej dannyh, vneshnie interfejsy processora, jevoljucija i blizhajshie perspektivy razvitiya iXBT.com. Available at: <http://www.ixbt.com/cpu/cpu-microarchitecture-part-3.shtml> (accessed 8 June 2012).
3. AMD Processors for Servers and Workstations: AMD Opteron™ Processor. Compare AMD Product Specs: AMD Phenom™, AMD Athlon™, AMD Opteron™, AMD Sempron™, AMD Turion™ 64 Processors Mobile Technology, ATI Radeon Graphics Cards, and AMD Powered Motherboards. Available at: <http://products.amd.com/en-us/OpteronCPUDetail.aspx?id=756> (accessed 4 June 2012).
4. Intel® Core™ i7 Processor Family for the LGA-2011 Socket : Datasheet, Volume 1 of 2. Процессоры Intel для Ultrabook™, ноутбуков, ПК и серверов. Available at: <http://www.intel.ru/content/dam/doc/datasheet/core-i7-lga-2011-datasheet-vol-1.pdf> (accessed 4 June 2012).
5. Intel® Xeon® Processor E7-8800/4800/2800 Product Families : Datasheet, Volume 2 of 2. Процессоры Intel для Ultrabook™, ноутбуков, ПК и серверов. Available at: www.intel.ru/content/dam/www/public/us/en/documents/datasheets/xeon-e7-8800-4800-2800-families-vol-2-datasheet.pdf (accessed 3 June 2012).
6. Koren, Israel. Computer arithmetic algorithms. 2nd ed. Natick, MA: A K Peters, 2002.
7. Omondi, Amos R., and Benjamin Premkumar. Residue number systems theory and implementation. London: Imperial College Press, 2007.

Рецензенты:

Бутаев М.М., д.т.н., профессор, ученый секретарь научно-технического совета, ОАО «Научно-производственное предприятие «Рубин», г. Пенза;

Частиков А.В., д.т.н., профессор, декан факультета прикладной математики и телекоммуникаций ФГБОУ ВПО «Вятский государственный университет», г. Киров.

Антонов А.В., д.т.н., профессор, декан факультета кибернетики Обнинского института атомной энергетики Национального исследовательского ядерного университета МИФИ, г. Обнинск.

Работа поступила в редакцию 04.09.2012.