ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ТЕХНОЛОГИЙ ПЕРЕДАЧИ И ОБРАБОТКИ ИНФОРМАЦИИ И СИГНАЛОВ

УДК 681.3

МЕТОДЫ, АЛГОРИТМЫ И ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ ОСНОВНЫХ ПРОБЛЕМНЫХ ОПЕРАЦИЙ, ВЫПОЛНЯЕМЫХ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ

Червяков Н.И.

В статье рассмотрены методы, алгоритмы и аппаратная реализация ускоренного определения знаков, сравнения модулярных чисел, основанные на принципе использования относительных значений анализируемых чисел, определяемых произведением модулей (оснований) системы остаточных классов.

Ключевые слова: система остаточных классов, алгоритм, приближенный метод

Введение

Развитие высокопроизводительных и надежных вычислительных систем, обладающих свойством отказоустойчивости, базируется на идеях создания вычислительных средств с параллельной структурой, использующих параллельное представление и обработку данных. К их числу относятся непозиционные коды – коды, основанные на модулярной арифметике, то есть коды, в которых данные представляются в системе остаточных классов (СОК) [1-2].

Если фиксированный ряд положительных чисел $p_1, p_2, ..., p_n$ назвать основаниями (модулями) СОК, то под системой остаточных классов понимается такая непозиционная система счисления, в которой любое целое положительное число A представляется в виде набора остатков (вычетов) от деления представляемого числа на выбранные основания системы $A = (\alpha_1, \alpha_2, ..., \alpha_n)$, где α_i – наименьшие неотрицательные вычеты (остатки) числа по модулям $p_1, p_2, ..., p_n$. Цифры α_i данного представления по выбранным модулям образуются следующим образом

$$a_i = res + A(\mathrm{mod}\, p_i) = A - \left[\frac{A}{p_i}\right] p_i, (\forall i \in [1, n]), \quad (1)$$

где $\left[\frac{A}{p_i}\right]$ – целочисленное частное, p_i – основания (модули) – взаимнопростые числа. В теории чисел доказано, что если $\forall i \neq j (p_i, p_j) = 1$, то представление (1) является единственным, при условии $0 \le A < P$, где $P = p_1 p_2 ... p_n = \prod_{i=1}^{n} p_i$ – диапазон представления чисел, то есть существует число A, для которого:

$$\begin{aligned} A &\equiv \alpha_1 (\mathrm{mod}\, p_1); \\ A &\equiv \alpha_2 (\mathrm{mod}\, p_2); \\ &\dots \\ A &\equiv \alpha_n (\mathrm{mod}\, p_n). \end{aligned} \quad (2)$$

Для чисел диапазона $[0, P)$, представленных в виде $A = (\alpha_1, \alpha_2, ..., \alpha_n)$, арифметические операции сложения, вычитания и умножения выполняются с остатками α_i независимо друг от друга по простым правилам. К достоинствам такого представления и обработки чисел относится также малоразрядность остатков, что позволяет эффективно применять табличные методы обработки. Вычислительные системы, построенные на основе СОК, обладают высокой производительностью и надежностью. Однако возникают серьезные трудности при реализации непозиционных процедур, к которым относятся: нахождение вычета (остатка) числа; определение знака числа (в СОК знак числа представлен в неявном виде); сравнение модулярных чисел; определение переполнения; операции деления, масштабирования, расширения, исправления ошибок и другие. Выполнение этих операций является довольно проблематичным. Большинство приложений СОК не требуют использования этих операций. Фундаментальной операцией здесь является сравнение величины модулярных чисел, которое может быть использовано при обнаружении переполнения динамического диапазона, определения знака чисел, исправления ошибок и других, время выполнения которых может быть уменьшено до времени выполнения модульного деления вместе со сложением, вычитанием и умножением, а также масштабирования вместе с расширением.

Необходимо отметить, что даже в тех случаях, когда СОК ограничена приложениями, в которых преобладающими операциями являются сложение и вычитание, не дает возможности полностью исключить проблематичные операции. Так,

в вычислениях особенно важно масштабирование, так как во многих приложениях, для которых СОК особенно хороша, могут встретиться операции, приводящие к росту чисел, которые, в свою очередь, могут привести к переполнению. Поэтому, чтобы гарантировать, что все результаты лежат в пределах допустимого диапазона, необходимо проводить контроль в процессе производимых вычислений. В связи с этим возникает необходимость быстрого выполнения вышеуказанных операций.

Кроме того, решение практически любой задачи управления требует сравнения в необходимый момент времени состояния управляемых объектов с заданными состояниями, соответствующими алгоритму функционирования систем. Целью сравнения является обнаружение факта совпадения или несовпадения значений величин, равенства или неравенства чисел, больших или меньших некоторых значений. Конкретное выполнение операции сравнения может определяться аппаратным или программным способом.

Сравнение модулярных чисел по их числовому значению может быть выполнено точно или приближенно. Прямой метод сравнения двух чисел путем перехода от остаточного представления к взвешенному представлению и затем выполнения обычного сравнения является дорогостоящим [3-4].

Основы многих алгоритмов, которые могут быть использованы для точного сравнения, базируются на следующих методах: ортогональных базисов; оценки интервалов чисел, с использованием функции Эйлера и универсальной позиционной характеристики, которая представляется коэффициентами обобщенной позиционной системы счисления (ОПСС), а также с использованием функций ранга и ядра чисел и другие. Указанные методы исследованы в работах [1-2], где показано, что в любом случае добиваются того, чтобы необходимая информация о величине числа извлекалась из представления остатков, что влечет за собой сложность как временную, так и аппаратную.

Приближенный метод вычисления основных проблемных операций в системе остаточных классов

С целью упрощения процесса сравнения модулярных чисел рассмотрим приближенный метод, который позволяет абсолютно правильно реализовать основные классы процедур принятия решений: анализ наличия определенного значения в конкретном разряде; проверка равенства (неравенства) двух значений; сравнение двух значений (больше, меньше), которые обеспечивают решение основного круга задач, возникающих при аппаратной или программной реализации реальных процессов.

Суть приближенного метода сравнения модулярных чисел основана на использовании относительных величин анализируемых чисел к полному диапазону Китайской теоремы об остатках [4], которая связывает позиционное число A с его представлением в остатках $(\alpha_1, \alpha_2, ..., \alpha_n)$, где α_i – наименьшие неотрицательные вычеты числа относительно модулей системы остаточных классов $p_1, p_2, ..., p_n$, следующим выражением

$$A = \left| \sum_{i=1}^{n} \frac{P}{p_i} \left| P_i^{-1} \right|_{p_i} \alpha_i \right|_P , \qquad (3)$$

где $P = \prod_{i=1}^{n} p_i$, p_i – модули СОК, $\left| P_i^{-1} \right|$ – мультипликативная инверсия P_i относительно p_i, и $P_i = \frac{P}{p_i} = p_1 p_2 ... p_{i-1} p_{i+1} ... p_n$.

Если (3) разделить на константу P, соответствующую диапазону чисел, то получим приближенное значение

$$\frac{A}{P} = \left| \sum_{i=1}^{n} \frac{\left| P_i^{-1} \right|_{p_i}}{p_i} \alpha_i \right|_1 \approx \left| \sum_{i=1}^{n} k_i \alpha_i \right|_1 , \qquad (4)$$

где $k_i = \frac{\left| P_i^{-1} \right|_{p_i}}{p_i}$ – константы выбранной системы, а α_i – разряды числа, представленного в СОК, при этом значение каждой суммы будет в интервале $[0, 1)$. Конечный результат суммы определяется после суммирования и отбрасывания целой части числа с сохранением дробной части суммы. Целая часть числа представляет собой ранг числа, то есть такую непозиционную характеристику, которая показывает, сколько раз диапазон системы P был превзойден при переходе от представления чисел в системе остаточных классов к его позиционному представлению. Определение ранга будет производиться непосредственно в процессе выполнения операции суммирования констант k_i . Дробная часть может быть записана также как $A \bmod 1$, потому что $A = \lfloor A \rfloor + A \bmod 1$ [5]. Количество разрядов дробной части числа определяется максимально возможной разностью между соседними числами. При необходимости точного сравнения необходимо вычислить значение (4), которое является эквивалентом

преобразования из СОК в позиционную систему счисления. Для решения задач основных процедур принятия решения достаточно знать приблизительно значения чисел A и B по отношению к динамическому диапазону P, которое выполняется достаточно просто, но при этом верно определяется соотношение $A = B$, $A > B$ или $A < B$.

Пример 1. Пусть дана система оснований $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$, объем диапазона $P = 2 \cdot 3 \cdot 5 \cdot 7 = 210$. Допустим, что в заданной СОК будут представлены только положительные числа. Величины $P_1 = \frac{P}{p_1} = 105$,

$P_2 = \frac{P}{p_2} = 70$, $P_3 = \frac{P}{p_3} = 42$, $P_4 = \frac{P}{p_4} = 30$.

Сравним два числа $A_1 = 25$ и $A_2 = 30$, представленные в СОК по основаниям p_1, p_2, p_3, p_4, так $A_1 = (1,1,0,4)$, $A_2 = (0,0,0,2)$. Для этого найдем константы $k_i = \frac{\left|P_i^{-1}\right|_{p_i}}{p_i}$:

$$k_1 = \frac{\left|\frac{1}{105}\right|_2}{2} = \frac{1}{2} = 0{,}5;$$

$$k_2 = \frac{\left|\frac{1}{70}\right|_3}{3} = \frac{1}{3} \approx 0{,}3333; \quad k_3 = \frac{\left|\frac{1}{42}\right|_5}{5} = \frac{3}{5} = 0{,}6;$$

$$k_4 = \frac{\left|\frac{1}{30}\right|_7}{7} = \frac{4}{7} \approx 0{,}5714.$$

По (4) получим

$$\frac{A_1}{P} \approx \left|1 \cdot 0{,}5 + 1 \cdot 0{,}3333 + 0 \cdot 0{,}6 + 4 \cdot 0{,}5714\right|_1 \approx$$
$$\approx 0{,}1189;$$

$$\frac{A_2}{P} \approx \left|0 \cdot 0{,}5 + 0 \cdot 0{,}3333 + 0 \cdot 0{,}6 + 2 \cdot 0{,}5714\right|_1 \approx$$
$$\approx 0{,}1428.$$

Так как $\frac{A_2}{P} > \frac{A_1}{P}$, то есть $0{,}1428 > 0{,}1189$, то $A_2 > A_1$, и действительно $30 > 25$.

Рассмотрим случай, когда рабочий диапазон разбит на два интервала $\left[0, \frac{P}{2}\right)$ – положительные числа, и $\left[-\frac{P}{2}, \frac{P}{2} - 1\right)$ – отрицательные числа. В традиционных ЭВМ определение абсолютных величин двух чисел A_1 и A_2 производится путем вычисления $A_1 - A_2$ и определения знака разности. В системе остаточных классов недостаточно определить знак путем $\left|A_1 - A_2\right|_P$, так как $A_1 - A_2$ могут выходить за диапазон $\left[-\frac{P}{2}, \frac{P}{2} - 1\right)$ и это приведет к неправильному результату.

Пример 2. Вариант неправильного определения сравнения чисел на основе определения знака.

Пусть $A_1 = \frac{P}{3}$, $A_2 = -\frac{P}{3}$, очевидно, что $A_1 > A_2$. Согласно (4) определим $\frac{A_1}{P}$ и $\frac{A_2}{P}$. Основания СОК выберем такими же, как и в первом примере. Тогда, используя дополнительный код, получим $A_1 = (0,1,0,0)$, и $A_2 = (0,2,0,0)$.

Находим $\frac{A_1}{P} \approx 1 \cdot 0{,}3333 = 0{,}3333$;

$\frac{A_2}{P} \approx 2 \cdot 0{,}3333 = 0{,}6666$. Откуда $\frac{A_1}{P} > \frac{A_2}{P}$, что неверно, так как число A_2 входит в отрицательный интервал, то есть $\left[-\frac{P}{2}, \frac{P}{2} - 1\right)$. Следовательно сравнение приведет к неверному результату $A_1 < A_2$.

Для правильного определения сравнения чисел необходимо проверить знаки A_1 и A_2, и тогда алгоритм сравнения будет иметь вид:

1. Определить знаки A_1 и A_2.
2. Если A_1 и A_2 без знаков, то положительный знак разности относительных величин означает большее число.
3. Если A_1 и A_2 имеют один и тот же знак, то проверяется $\left|\frac{A_1}{P} - \frac{A_2}{P}\right|_1$.
4. Если A_1 и A_2 имеют разные знаки, то $0 \le \left|\frac{A_1}{P} - \frac{A_2}{P}\right|_1 < 0{,}5$; при $A_1 < A_2$ и $0{,}5 \le \left|\frac{A_1}{P} - \frac{A_2}{P}\right|_1 < 1$ при $A_1 > A_2$.

Таким образом, сравнение чисел со знаком требует предварительного анализа знаков сравниваемых чисел.

Известно [2], что при кодировании дополнительным кодом отрицательная часть динамического диапазона находится у верхнего предела полного диапазона. Положительные числа из дополнительного диапазона отображаются на область $\left[0, \frac{P+1}{2}\right)$ при нечетных P и на область $\left[0, \frac{P}{2}\right)$ при четных P. Отображение динамичес-

кого диапазона на соответствующую область для избыточного кода СОК показано на рис. 1.

Рис. 1. Схема расположения положительных и отрицательных чисел на диапазоне избыточного кода СОК

Это обстоятельство может привести к ошибке сравнения, как в вышерассмотренном примере, так как отрицательные числа попадают в верхнюю часть полного диапазона, и все отрицательные числа будут давать ошибки, что не соответствует действительности в силу разнесения динамического диапазона.

Для преодоления этой трудности необходимо произвести сдвиг отрицательной области путем вращения остаточного кольца в положение, указанное на рис. 2. Пунктиром показана область, которая перенесена в начало диапазона.

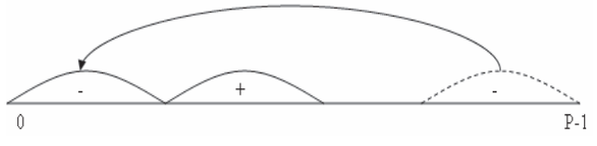

Рис. 2. Схема выполнения операции сдвига полярности СОК

В результате отрицательные числа будут отображены в начальной части динамического диапазона.

Показанное на рис. 2 вращение называется сдвигом полярности, и его можно осуществить путем прибавления перед сравнением модулярных чисел константы $C = \frac{P-1}{2}$ при нечетном P или $C = \frac{P}{2}$ при четных P к каждому $A \in [0, P)$.

Если $C_i \equiv |C|_{p_i}^{+}$, то сдвиг полярности в пределах СОК оказывается простым остатком, определяемым по формуле $\alpha_{ic} = |\alpha_i + c_i|_{p_i}^{+}$, в которой α_{ic} обозначает остаточные цифры после сдвига полярности.

Пример 3. Сравнить модулярные числа разных знаков A_1 и $-A_2$. Система оснований СОК такая же, как и в примере 1: $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$.

Пусть число $A_1 = 17 = (1,2,2,3)$, $A_2 = -19 = (1,1,4,5)$. Тогда дополнительный код $A_2 = (p_1 - 1, p_2 - 1, p_3 - 4. p_4 - 5) = (1,2,1,2)$. Требуется сравнить числа A_1 и A_2.

Проверка знака числа A_1. Для определения знака числа A_1 сравним его с константой $K = \frac{P}{2}$. Тогда относительная величина числа A_1 по отношению к величине числа K определяется как $A_1 \approx |k_1 \cdot 1 + k_2 \cdot 2 + k_3 \cdot 2 + k_4 \cdot 3|_1 = |0{,}5 \cdot 1 + 0{,}3333 \cdot 2 + 0{,}6 \cdot 2 + 0{,}5714 \cdot 3|_1 \approx 0{,}0808$.

Представление относительной величины константы $K = \frac{P}{2} = (1,0,0,0)$. Далее находим $\frac{K}{P} \approx |0{,}5 \cdot 1 + 0{,}3333 \cdot 0 + 0{,}6 \cdot 0 + 0{,}5714 \cdot 0|_1 = 0{,}5$. Отсюда $\frac{K}{P} - \frac{A_1}{P} = 0{,}5 - 0{,}0808 = 0{,}4192$. Разница положительная, то есть число $\frac{A_1}{P} < \frac{K}{P}$, поэтому число A_1 входит в первый интервал и является положительным.

Проверка знака числа A_2 проходит аналогично: $\frac{A_2}{P} \approx |0{,}5 \cdot 1 + 0{,}3333\,2 + 0{,}6 \cdot 1 + 0{,}5714\,2|_1 = 0{,}9094$. ; $\frac{K}{P} - \frac{A_2}{P} = 0{,}5 - 0{,}9094 = -0{,}4094$. Разность отрицательная, и число A_2 входит во второй интервал, очевидно, что оно является отрицательным. Для правильного сравнения чисел A_1 и A_2 необходимо провести сдвиг полярности чисел A_1 и A_2, так как число A_2 является отрицательным. После сдвига получаем $A_1^{'} = (0,2,2,3)$ и $A_2^{'} = (0,2,1,2)$.

Определим относительные величины $A_1^{'}$ и $A_2^{'}$.

$\frac{A_1^{'}}{P} \approx |0{,}5 \cdot 0 + 0{,}3333 \cdot 2 + 0{,}6 \cdot 2 + 0{,}5714 \cdot 3|_1 = 0{,}5818$;

$\frac{A_2^{'}}{P} \approx |0{,}5 \cdot 0 + 0{,}3333 \cdot 2 + 0{,}6 \cdot 1 + 0{,}5714 \cdot 2|_1 = 0{,}4094$.

Найдем разность относительных величин, тогда $\frac{A_1^{'}}{P} - \frac{A_2^{'}}{P} = 0{,}5818 - 0{,}4094 = 0{,}1724$ – разность положительная. Следовательно $A_1 > A_2$.

Рассмотренные приближенные методы определения таких позиционных характеристик модулярного кода, как определение знака числа и сравнение чисел, показали, в отличие от известных, простоту их вычисления, так как для их реализации не используется вычисление коэффициентов ОПСС, которое требует больших аппаратных и временных затрат. По этой причине данный метод представляет собой особую важность и является одним из лучших решений, по мнению автора, на настоящее время. Пере-

численные операции являются важнейшими для машинной модулярной арифметики, и их применение может дать значительные преимущества не только в таких приложениях, в которых основная доля вычислений приходится на точное умножение, возведение в степень больших чисел в сочетании со сложением и вычитанием, но и в которых довольно часто появляется необходимость сравнения и определения знака числа. Известно [6], что теорема кодирования Сабо гласит, что нет лучших методов определения позиционных характеристик, при которых не используется их однозначность, чем перевод чисел из СОК в ОПСС, поскольку величины числа в модулярном представлении существенным образом зависят от всех остатков числа. Назовем этот метод определения позиционных характеристик точным методом. Однако проведенные исследования по использованию приближенных методов показали простоту определения позиционных характеристик по сравнению с методами на основе ОПСС. Таким образом, можно сделать вывод о том, что теорема Сабо справедлива только при точных методах.

Для повышения эффективности обработки данных в модулярной арифметике рассмотрим приложения приблизительных методов для определения знака числа и сравнения модулярных чисел.

Определение знака модулярных чисел

Известно [1-2; 4; 7], что для определения знака числа используются номера интервалов, в которых расположено число, что позволяет получить оценку исследуемого числа по его величине с точностью до величины интервала. Числовой диапазон P может быть разбит на p_i интервалов величиной

$$\left[j\frac{P}{p_i}, (j+1)\frac{P}{p_i} \right), \; j = 1,2,...,p_i . \qquad (5)$$

В качестве второго машинного нуля выбирается точка числового диапазона $\frac{p_{n+1}}{2}\frac{P}{p_n}$. Числа, расположенные в поддиапазонах $\left[0, \frac{p_{n+1}}{2}\frac{P}{2}\right)$ и $\left[\frac{p+1}{2}\frac{P}{p_n}, P\right)$, считаются числами разных знаков.

Если дано представление $(\alpha_1, \alpha_2, ..., \alpha_n)$, то для того чтобы установить знак числа, которое оно представляет, достаточно решить задачу о принадлежности этого числа к определенному интервалу. В случае если $p_i = 2$, достаточно решить задачу о принадлежности этого числа к первой $\left[0, \frac{P}{2}\right)$ или второй $\left[\frac{P}{2}, P\right)$ половине диапазона $[0, P)$. Эта задача решается сравнением данного представления с представлением $\frac{P}{2}$, при условии, что $p_1 = 2$. Все известные методы реализуют данный алгоритм на основе использования абсолютных величин, здесь же мы предлагаем использовать относительные величины, что существенно упрощает преобразование, сохраняя при этом основные функциональные возможности.

На рис. 3 приведена схема для определения знака модулярного числа, представленного по модулям $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$. Схема содержит входные регистры RG_i, $\forall i = [1...4]$ для временного хранения остатков чисел по соответствующим модулям, просмотровые таблицы LUT_i, $\forall i = [1...4]$ для хранения произведений $\left|\frac{P_i^{-1}}{p_i}\right|_{p_i} \cdot \alpha_i$ и параллельный сумматор. Процесс определения интервала сводится к выявлению принадлежности данного числа к одной из двух половин диапазона $[0, P)$, к первой $\left[0, \frac{P}{2}\right)$ или второй $\left[\frac{P}{2}, P\right)$, где $\frac{P}{2}$ принимается в качестве нуля. Эта задача решается сравнением относительной величины $\frac{A}{P}$ с относительной величиной $\frac{K}{P} = \frac{P}{2P} = \frac{1}{2}$. Исходное число положительное, если $\frac{A}{P} < \frac{1}{2}$, и отрицательное, если $\frac{A}{P} \geq \frac{1}{2}$.

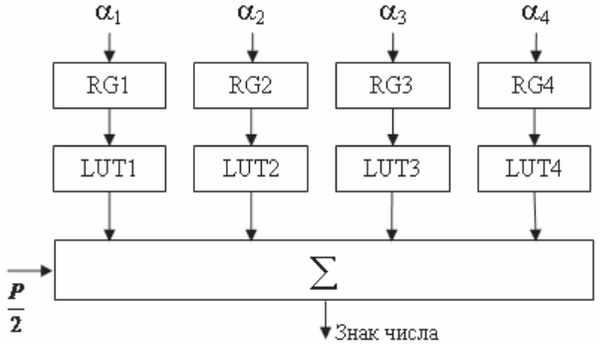

Рис. 3. Схема определения знака числа

Схема работает следующим образом.

Код числа A, для которого необходимо определить интервал, что равносильно определению знака числа, поступает на входные регистры RG_i в двоичном коде (каждый разряд СОК кодируется двоичным кодом). Сигналы с выходов

регистров поступают на входы просмотровых таблиц LUT. В просмотровых таблицах хранятся произведения констант k_i и остатков α_i, то есть $\frac{\left|P_i^{-1}\right|_{p_i}}{p_i}\alpha_i$, представленных в естественной форме двоичной дроби в дополнительном коде. Количество элементов памяти (N) просмотровых таблиц определяется выражением $N=\sum_{i=1}^{n} p_i$.

Выходные сигналы просмотровых таблиц в дополнительном двоичном коде поступают на вход сумматора, в котором уже записана константа во время начальной установки (дополнительный код используется для того, чтобы операцию вычитания заменить операцией сложения). Знак результата сложения определяет интервал: первый или второй, что соответственно определяет знак числа.

Пример 4. Пусть дана система оснований $p_1=2$, $p_2=3$, $p_3=5$, $p_4=7$. Тогда $P=210$. Константы k_i соответственно равны $k_1=0{,}5$, $k_2\approx 0{,}3333$, $k_3=0{,}6$, $k_4\approx 0{,}5714$.

Дано: число $A=(1,1,2,0)$. Требуется определить знак числа A.

Решение: в регистры поместим следующие значения $RG_1=1$, $RG_2=1$, $RG_3=2$, $RG_4=0$. Входные значения регистров являются адресными входами LUT запоминающих элементов, которые принимают следующие значения: $LUT_1=0{,}5; LUT_2=0{,}3333\cdot 1=0{,}3333;$ $LUT_3=\left|0{,}6\cdot 2\right|_1=\left|1{,}2\right|_1=0{,}2$; $LUT_4=0$, которые поступают на вход сумматоров.

После суммирования получим:

$$\frac{A}{P}=\left|0{,}5+0{,}333+0{,}2\right|_1=\left|0{,}033\right|_1.$$

Операция суммирования в данном случае может быть выполнена с помощью одноразрядного десятичного сумматора или четырехразрядного двоичного сумматора, так как младшие значения разряда на результат суммирования не имеют влияния.

Тогда $\frac{K}{P}-\frac{A}{P}=0{,}5-0{,}033=0{,}467$. Разность положительная, то есть число $A<\frac{P}{2}$, откуда видно, что число A входит в первый интервал и следовательно, является положительным. Определение знака осуществляется за n суммирований, где n – число оснований (модулей) СОК. Время суммирования определяется логической глубиной устройства (количество последовательно соединенных элементов) и временем суммирования в одном сумматоре. Для уменьшения времени суммирования схему сумматора можно реализовать по принципу дерева (рекурсивного сдваивания). Кроме того, реализация сумматора может быть выполнена на искусственных нейронных сетях.

Сравнение модулярных чисел

При сравнении двух чисел: A и B требуется определить, какое из них больше, меньше или они равные. Для определения номеров интервалов, в которых находятся эти числа, используются разные методы [1-2], но все они очень сложны. Пусть число A расположено в интервале j_1, а число B в интервале j_2. Тогда в случае $j_1\neq j_2$ операция сравнения может быть реализована просто сравнением номеров интервалов, а именно, если $j_1>j_2$, то $A>B$, если $j_1<j_2$, то $A<B$. Исключение составляет случай $j_1=j_2$. Здесь для определения большего числа требуется определить номер j_3 интервала, в котором расположена разность $A-B$. Если $0\le j_3<\frac{p_{n+1}}{2}$, то разность отрицательна, отсюда $A<B$. Если $\frac{p_{n+1}}{2}\le j_3<p_n$, то разность положительна и $A>B$.

В случае $A-B=0$ числа одинаковы по величине и по знаку. Методы определения интервалов исследованы в [1-2; 7]. Все они основаны на точных вычислениях и представляют большие трудности как по аппаратным, так и по временным затратам.

Рассмотрим метод приближенного сравнения, который основан на использовании относительных значений сравниваемых чисел. Хотя предлагаемый метод является приблизительным, он позволяет достаточно просто реализовать процесс сравнения модулярных чисел с выполнением верного сравнения.

Схема модулярного сравнения чисел (рис. 4) содержит входные регистры RGA и RGB для хранения сравниваемых чисел A и B, схемы определения знаков чисел A и B ($СОЗЧ_A$ и $СОЗЧ_B$), логического элемента «исключающее или», схемы сдвига полярности $ССПЧ_A$ и $ССПЧ_B$, просмотровые таблицы (память) $LUT_{p_i}A$ и $LUT_{p_i}B$, $i=[1...n]$, сумматор для сложения выходных значений LUT-таблиц и схемы сравнения знака разности для формирования сигналов $A=B$, $A<B$ и $A>B$.

На входные регистры RGA и RGB поступают исходные числа, представленные в СОК по модулям $p_1, p_2, ..., p_n$. С выходов регистров сигнал поступает на вход схем определения знаков числа, $ССПЧ_A$ и $ССПЧ_B$. Выходные сигналы схем определения знаков чисел A и B поступают на вход элемента «исключающее или». При

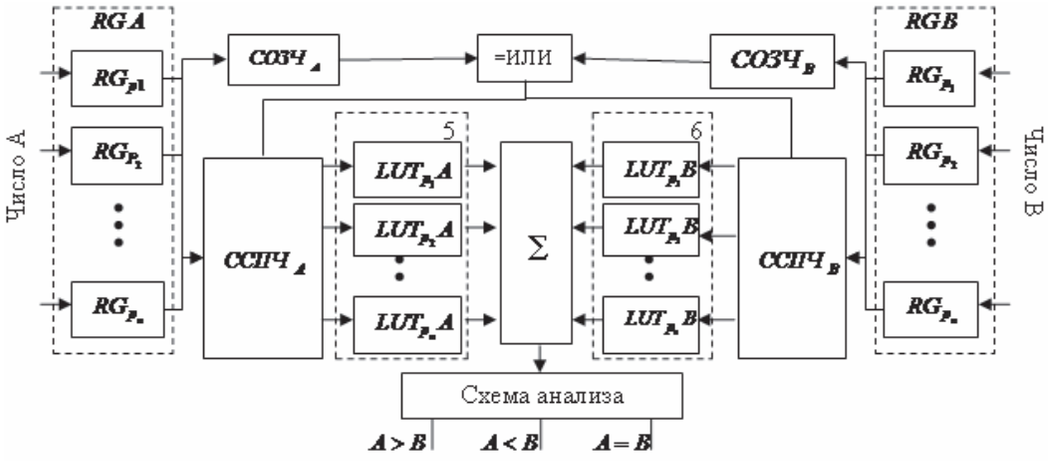

Рис. 4. Схема сравнения модулярных чисел

разных знаках чисел A и B выходной сигнал элемента «исключающее или», которому приписано значение c_i, поступает на входы схем сдвига полярности $ССПЧ_A$ и $ССПЧ_B$. Выходные сигналы $ССПЧ_A$ и $ССПЧ_B$ являются адресными входами просмотровых таблиц $LUT_{p_i}A$ и $LUT_{p_i}B$. Элементы памяти LUT хранят константы $k_i = \frac{\left|P_i^{-1}\right|_{p_i}}{p_i}\alpha_i$, $k_i = \frac{\left|P_i^{-1}\right|_{p_i}}{p_i}\beta_i$, где $\alpha_i \equiv A \bmod p_i$, $\beta_i \equiv B \bmod p_i$, $\forall i \in [1,4]$.

Сигналы с выхода просмотровых таблиц $LUT_{p_i}A$ и $LUT_{p_i}B$ поступают на вход сумматора, где производится взвешенное суммирование $\frac{A}{P}$ и $\frac{B}{P}$ с формированием знака полученной разности чисел A и B, который анализируется в схеме анализа результата сравнения ($A = B$, $A < B$ и $A > B$). Рассмотрим пример сравнения модулярных чисел.

Пример 5. Пусть дана система оснований $p_1 = 2$; $p_2 = 3$; $p_3 = 5$; $p_4 = 7$. Сравнить модулярные числа $A = (1,2,2,3)$ и $B = (1,2,1,2)$. Исходные числа находятся в регистрах RGA и RGB, которые поступают на входы схем определения знака числа $СОЗЧ_A$ и $СОЗЧ_B$. В этих схемах происходит сравнение исходных чисел с константой $\frac{P}{2}$.

Определим знак числа A, для этого вычислим:

$$\frac{A}{P} \approx \left|k_1 \cdot 1 + k_2 \cdot 2 + k_3 \cdot 2 + k_4 \cdot 3\right|_1 =$$

$$= \left|0{,}5 \cdot 1 + 0{,}3333 \cdot 2 + 0{,}6 \cdot 2 + 0{,}5714 \cdot 3\right|_1 \approx$$

$$\approx 0{,}0808;$$

$$\frac{K}{P} - \frac{A}{P} = 0{,}5 - 0{,}0808 = 0{,}4192.$$

Разность положительная, то есть $\frac{A}{P} < \frac{K}{P}$, поэтому число A входит в первый интервал и является положительным.

Аналогично определим знак числа B.

$$\frac{B}{P} \approx \left|k_1 \cdot 1 + k_2 \cdot 2 + k_3 \cdot 1 + k_4 \cdot 2\right|_1 =$$

$$= \left|0{,}5 \cdot 1 + 0{,}3333 \cdot 2 + 0{,}6 \cdot 1 + 0{,}5714 \cdot 2\right|_1 \approx$$

$$\approx 0{,}9094$$

$$\frac{K}{P} - \frac{B}{P/2} = 0{,}5 - 0{,}9094 = -0{,}4094.$$

Разность отрицательная, поэтому число B входит во второй интервал и является отрицательным.

Результат определения знаков чисел A и B поступает на входы элемента «исключающее или», выходной сигнал которого поступает на вход схем сдвига полярности $ССПЧ_A$ и $ССПЧ_B$. На выходах схем сдвига полярности образуются данные, соответственно, $A = (1,2,2,3) + (1,0,0,0) = (0,2,2,3)$ и $B = (1,2,1,2) + (1,0,0,0) = (0,2,1,2)$. Выходные данные схем сдвига полярности являются адресными входами просмотровых таблиц $LUT_{p_i}A$ и $LUT_{p_i}B$, согласно которым осуществляется выборка значений констант $k_i = \frac{\left|P_i^{-1}\right|_{p_i}}{p_i}\alpha_i$ в условиях примера эти значения будут равны для $LUT_{p_i}A$ $(0;0{,}3333 \cdot 2;0{,}6 \cdot 2;0{,}5714 \cdot 3)$ и для $LUT_{p_i}B$ $(0;0{,}3333 \cdot 2;0{,}6 \cdot 1;0{,}5714 \cdot 2)$. Выходные сигналы просмотровых таблиц $LUT_{p_i}A$ и $LUT_{p_i}B$ поступают на вход сумматора. В результате суммирования получим:

$$\frac{A_i}{P}-\frac{B_i}{P}\approx\left|0+0{,}3333\cdot 2+0{,}6\cdot 2+0{,}5714\cdot 3\right|_1-$$
$$-\left|0+0{,}3333\cdot 2+0{,}6\cdot 1+0{,}5714\cdot 2\right|_1=0{,}1724.$$

Разность положительная, следовательно $A>B$. Действительно, число $A=17$, $B=-19$.

Результаты сумматора анализируются в схеме анализа, при этом:

- если разность равна 0, то $A=B$,
- если разность положительная, то $A>B$,
- если разность отрицательная, то $A<B$.

В случае если сравниваются числа одной полярности, то из схемы сравнения модулярных чисел исключается схема определения знаков чисел. Тогда логическая глубина схемы (количество последовательно включенных элементов) будет $n+3$, где n – количество суммирований в сумматоре, при этом n определяется количеством модулей в системе.

Если же использовать рекурсивное сдваивание [1], тогда логическая глубина определяется выражением $\lceil\log_2 n\rceil+3$. В известных схемах [2] логическая глубина с учетом определения коэффициентов ОПСС определяется как $2n+5$. Таким образом, выигрыш в скорости предложенного метода почти равен двум.

Итак, разработанный метод и аппаратная реализация определения знака числа и сравнения модулярных чисел является одним из лучших на настоящее время по показателям сложности и скорости.

Предложенный метод может быть использован для определения переполнения динамического диапазона и определения ошибок в избыточных системах остаточных классов. Если в избыточной СОК в качестве рабочих оснований принять p_1, p_2, …, p_n, а в качестве избыточных оснований p_{n+1}, p_{n+2}, …, p_{n+r}, разрешимые значения определяются как $P=\prod_{i=1}^{n}p_i$. Факт переполнения или ошибки определяется сравнением проверяемого числа A с рабочим диапазоном P: если $A>P$, то произошло переполнение или ошибка. Одиночная ошибка может быть обнаружена так же, как и переполнение динамического диапазона, при предположении, что переполнение не происходит одновременно с ошибкой.

Выводы

1. Противоречие между вычислительной сложностью определения основных проблемных процедур в СОК и их быстродействием разрешена путем замены абсолютных величин их относительными значениями и простотой их вычисления, которая сохраняет адекватную связь числовых значений модулярных величин с их представлениям в СОК и позволяет повысить скорость выполнения немодульных операций. Благодаря этому применение системы остаточных классов может дать значительные преимущества не только в тех приложениях, в которых основная доля вычислений приходится на точное умножение, возведение в степень больших чисел в сочетании со сложением и вычитанием, но и в которых часто появляется необходимость в делении либо сравнении и определении знака числа, а также при проверке не «выходят» ли результаты за пределы допустимых значений и другие.

2. Решена фундаментальная проблема реализации основных проблемных операций в СОК, которые ранее определяли наибольший вклад в алгоритмическую сложность и сдерживали широкое применение СОК при разработке новых классов вычислительных систем. Внедрение полученных результатов позволяет снять это ограничение и расширить область применения модулярной арифметики.

3. Разработанные методы, алгоритмы и техническая реализация самых проблемных процедур, характеризующихся простотой и высокой скоростью выполнения операций, дополняют известный универсальный базис СОК на основе обобщенной позиционной системы счисления и позволяют разделить выполнения всех немодульных процедур на два класса: процедуры точного вычисления (вычисления остатка, округления числа, деления, масштабирования, расширения модулярной величины на дополнительные основания и коррекция ошибок) и процедуры приблизительного вычисления (сравнения модулярных чисел, вычисления ранга числа, определения знака числа и переполнения диапазона, обнаружение и локализация ошибок при кодировании помехоустойчивым кодом). Перечисленные классы процедур являются важнейшими для машинной модулярной обработки и требуют глубокого дальнейшего исследования для определения границ их эффективного применения.

4. Полученные новые результаты эффективного выполнения немодульных процедур являются развитием теории математических основ разработки и проектирования высокопроизводительных и надежных вычислительных систем, функционирующих в системе остаточных классов.

Литература

1. Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н. Нейрокомпьютеры в остаточных классах. М.: Радиотехника, 2003. – 272 с.
2. Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А. Модулярные параллельные вычислительные структуры нейропроцессорных систем. М.: Физматлит, 2003. – 288 с.
3. Червяков Н.И., Колесницкий С.В. Устройство для сравнения чисел А.с. СССР 541164, опубл. 30.12.76, бюлл. №48.
4. Omondi A., Premkumar. Residue Number Systems. Theory and Implementation. London. Imperial College Press, 2007. – 295 p.
5. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики. Пер. с англ. М.: Мир; Бином, 2006. – 703 с.
6. Кнут Д. Искусство программирования для ЭВМ. Т. 2. М.: Мир. 1980. – 840 с.
7. Червяков Н.И. Методы и принципы построения модулярных нейрокомпьютеров // 50 лет модулярной арифметике. Сб. научных трудов. М.: ОАО «Ангстрем», МИЭТ, 2005. – 775 с.

METHODS, ALGORITHMS AND TECHNICAL REALIZATION OF THE BASIC PROBLEM OPERATIONS EXECUTED IN RESIDUE NUMBER SYSTEM

Chervyakov N.I.

It's considered methods, algorithms and hardware implementation of rapid sign determination, comparison of modular numbers, which based on application of relative values of analyzing numbers defined by product of residue number system's modules (bases).

Keywords: *residue number system, algorithm, approximate method.*

Червяков Николай Иванович, доктор технических наук, профессор, Заслуженный деятель науки и техники РФ, заведующий кафедрой «Прикладная математика и информатика» Ставропольского государственного университета. Тел. (8-8652) 35-68-32 доб. 1154; 35-48-61. E-mail: k-fmf-primath@stavsu.ru

УДК 621.391.2

СОВМЕСТНАЯ ФИЛЬТРАЦИЯ ДИСКРЕТНОГО И НЕПРЕРЫВНЫХ ПАРАМЕТРОВ МНОГОМЕРНЫХ КОРРЕЛИРОВАННЫХ ИМПУЛЬСНЫХ СИГНАЛОВ

Медведева Е.В., Метелев А.П., Петров Е.П.

Рассмотрен синтез алгоритмов совместной фильтрации дискретного и непрерывных параметров многомерных двоичных коррелированных сигналов. Предполагается, что все параметры являются марковскими процессами. Совместная фильтрация параметров осуществляется в присутствии белого гауссовского шума (БГШ). Синтезированные алгоритмы требуют для своей реализации минимальных ресурсов и позволяют повысить помехоустойчивость приема импульсных сигналов.

Ключевые слова: совместная фильтрация, коррелированные сигналы, марковские процессы.

Введение. Постановка задачи

Синтез приемных устройств для обработки импульсных коррелированных сигналов ведется в большинстве случаев без учета взаимного влияния качества фильтрации дискретного параметра (манипулированной фазы, частоты и т.д.) на непрерывные параметры радиосигнала (амплитуда, задержка и т.д.). В действительности реализация статистической избыточности импульсных коррелированных сигналов приводит к увеличению вероятности их распознавания, что эквивалентно увеличению точности оценки непрерывных параметров при приеме некоррелированных импульсных сигналов. Задача синтеза в этом случае сводится к установлению механизма взаимодействия между устройствами фильтрации дискретного и непрерывных параметров импульсных коррелированных сигналов.

В различных постановках задача совместной фильтрации параметров бинарных импульсных сигналов, представленных простой цепью Маркова с двумя состояниями, рассматривалась в работах [1-4], в которых предполагалось, что импульсные сигналы некоррелированы. Это сужает возможности практического использования результатов работ [1-4] в системах передачи информации. Наиболее полные исследования совместной фильтрации параметров коррелированных импульсных сигналов проведены в работах [5-6].