

$$3. b := 2b + P \cdot (k_2 = 0) = 6P + 0 = 6P.$$

$$4. b := 2b + P \cdot (k_1 = 0) = 12P + 0 = 12P.$$

$$5. b := 2b + P \cdot (k_0 = 1) = 24P + P = 25P.$$

Выход: $kP = 25P$.

Как видно, для получения точки $25P$ необходимо 5 итераций вместо 7 итераций метода аддитивных цепочек. Преимуществом предложенного алгоритма является требование всего одного ОЗУ для хранения промежуточных значений b .

Итак, разработанный метод показывает, что для того, чтобы эффективно использовать модифицированную систему координат Якобиана, необходимо выбирать ключи k , которые в двоичной системе имеют меньшее число единиц. При таком выборе ключа для выполнения скалярного умножения точек количество операций сложения уменьшается за счет увеличения числа удвоений,

которые в модифицированной системе Якобиана выполняются с минимальным временем.

Литература

1. Cohen H. Efficient elliptic curve exponentiation // Advances in Cryptology-proceeding of ICICS'97. Springer-Verlag, 1997. – P. 282-290.
2. Chudnovsky D.V., Chudnovsky G.V. Sequences of numbers generated by addition in formal groups and new primary and factorization tests // Advances in Applied Math. №7, 1986. – P. 385-434.
3. Червяков Н.И., Головкин А.Н. Модифицированный алгоритм Монтгомери скалярного умножения точки на эллиптической кривой // Компьютерные науки и технологии. Белгород: Изд. ГиК, 2009. – С. 104-108.

УДК 681.3

МАТЕМАТИЧЕСКИЕ МОДЕЛИ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ БОЛЬШОЙ РАЗМЕРНОСТИ С ВЫСОКОЙ СТЕПЕНЬЮ ДОСТОВЕРНОСТИ

Мезенцева О.С., Алексеев А.И.

В статье представлены математические модели распределенного хранения и обработки данных большой размерности, базирующиеся на аппарате пороговых схем и обладающие высокой степенью достоверности и отказоустойчивости. Для их построения используется аппарат модулярной арифметики, что позволяет добиться существенного повышения коэффициента ускорения и отказоустойчивости предложенных алгоритмов, открывает перспективы использования разработанных моделей в системах реального времени. Предложено эффективное отображение алгоритмических решений для реализации действий с данными большой размерности на аппаратную базу нейропроцессора NM6403.

Введение

Задачи факторизации, тестирования и определения простоты чисел специального вида, системы спутниковой навигации, и многие другие ресурсоемкие задачи требуют обработки больших объемов информации в широком диапазоне с высокой достоверностью и точностью вычислений. Как правило, подобные задачи сводятся к вычислительным проблемам, оперирующим целочисленными переменными, значения которых на несколько порядков превышают максимум машинного диапазона.

Для решения многих прикладных и теоретических проблем необходимо проведение вычис-

лений уже в сверхбольших диапазонах, которые, с одной стороны, являются обобщением больших диапазонов, но в то же время принципиально отличаются от последних. Так как вычисления в них должны проводиться над потенциально бесконечными математическими объектами, их явные числовые записи невозможно или нецелесообразно хранить в памяти по причине принципиальной недостаточности вычислительных ресурсов, в результате чего в один момент времени возможны хранение и обработка лишь одного элемента представления. При таком подходе к распределенному хранению и обработке данных высокой размерности неизбежно повышается вероятность потери части представления числа. Выход из строя даже незначительного количества узлов в совокупности с высокой трудоемкостью алгоритмов обработки чисел из большого и сверхбольшого диапазонов может привести к серьезным и сложно решаемым проблемам восстановления.

Параллельные вычислительные структуры являются идеальной основой для построения устойчивых к отказам вычислительных средств. Ключевую роль в процессе функционирования таких вычислительных устройств играет способность сохранения работоспособного состояния за счет снижения в допустимых пределах каких-либо показателей качества при возникновении сбоев и отказов в системе. Достоинство данного

подхода к выполнению процедур обеспечения отказоустойчивости реализуется в полной мере при перераспределении исходных данных между сохранившимися вычислительными ресурсами при деградации системы. Однако, существующие методы обеспечения достоверности передачи и обработки данных высокой размерности не лишены недостатков, среди которых можно отметить излишнюю избыточность, повышенные требования к среде передачи, низкую эффективность средств локализации и устранения сбоев, особенно в случае пакетирования ошибок.

В то же время, на фоне развития сетевых технологий и увеличения объемов обрабатываемых данных, все большее значение приобретает разработка технологий высоконадежного хранения данных и удаленного высокоскоростного доступа к ним, средств и методов управления нагрузкой серверов и каналов связи. Размеры многих локальных сетей достигают значений, при которых подобные сети по техническим и топологическим характеристикам мало отличаются от глобальных. Проблемы хранения информации и обеспечения доступа к ней в глобальных и локальных сетях и вычислительных системах приобретают одинаковый характер и требуют единого подхода к их решению.

Анализ российских и зарубежных литературных источников позволил сделать вывод об отсутствии единого подхода к проектированию систем распределенного хранения и обработки данных. Большинство традиционных решений ориентировано на конкретный и зачастую довольно узкий спектр задач и характеризуется низкой масштабируемостью и рядом других существенных ограничений. Лишенные многих из этих недостатков системы с масштабируемой избыточностью также не имеют широкого распространения по причине сложности эксплуатации и недостаточной надежности. Успешное развитие сетевой инфраструктуры, средств и методов сетевого взаимодействия, повышение производительности вычислительных систем в целом непосредственно связано с разработкой математических моделей достоверного распределенного хранения и обработки данных большой размерности, высокоскоростных методов доступа к ним, а также алгоритмов их реализации.

Математическая модель хранения данных большой размерности

Алгоритмы решения фундаментальных задач, формулируемых в сверхбольших диапазонах,

имеют близкий вид и сводятся к вычислению соотношения [1]:

$$C = A^{f(B)} - \left[A^{f(B)} / B \right] \cdot B, \quad (1)$$

где $A, B, f(B)$ – большие числовые величины; $A^{f(B)}$ – сверхбольшая числовая величина; к форме сравнения по модулю $P > C$:

$$C(\text{mod } P) \equiv (A^{f(B)}(\text{mod } P) - [A^{f(B)} / B] \times (\text{mod } P) \cdot B(\text{mod } P))(\text{mod } P), \quad (2)$$

где P – диапазон модулярной арифметики $C(\text{mod } P)$.

Система остаточных классов (СОК), благодаря внутреннему параллелизму, модульности, возможности арифметической коррекции ошибок, является наиболее подходящей базой для реализации высокоскоростных алгоритмов достоверной обработки данных большой размерности в распределенных компьютерных системах.

Известный метод разбиения и сборки данных, основанный на аппарате пороговых схем состоит из следующей последовательности шагов [1]:

1. *Построение $GF(N)$* . Существует два основных метода построения конечных полей. Выбор того или иного варианта основывается на требуемом числе элементов поля. Для поля, содержащего p элементов, где p – простое, кольцо вычетов $Z_n (n = p)$ является полем, и элементы Z_p есть $0, 1, \dots, p-1$. Для более реального в рассматриваемой ситуации случая $GF(q)$, $q = p^n$, где p – простое: $K = F_p[x] / \langle f(x) \rangle$ является полем тогда и только тогда, когда $f(x)$ есть многочлен, неприводимый над полем F_p , $|K| = p^m$, $m = \deg(f)$.

2. *Представление исходных данных в виде элементов $GF(N)$* . Рассматривая k -мерное пространство векторов L над $GF(N)$, исходная последовательность данных может быть представлена в виде последовательности векторов из L : $d = (l_1, l_2, \dots, l_m)$.

3. *Построение набора векторов E* . Необходимо построить набор векторов E из L , состоящий из n векторов так, что любые k из E есть базис в L :

$$\forall p, p \in GF(N), p \neq 0: (p^0; p^1 \dots p^k),$$

$$\begin{vmatrix} 1 & 1 & \dots & 1 \\ p_1 & p_2 & \dots & p_n \\ \dots & \dots & \dots & \dots \\ p_1^{n-1} & p_2^{n-1} & \dots & p_n^{n-1} \end{vmatrix} = \prod_{1 \leq j \leq i \leq n} (p_i - p_j), \quad (3)$$

$$\prod_{k=1}^n p_k = p_1 p_2 \dots p_n -$$

определитель Вандермонда.

4. Формирование матрицы $A_{k \times k}$ и нахождение обратной ей A^{-1} :

$$\forall e_i \in E: (l_1, e_i), (l_2, e_i), \dots, (l_m, e_i). \quad (4)$$

где e_i – вектора построенного набора E ; l_i – элементы исходной последовательности данных.

По условию формирования E построенная матрица имеет обратную A^{-1} .

5. Сборка последовательности. Выражение для сборки исходной последовательности имеет вид

$$D = \bigcup_i l_i, \quad (5)$$

где $l_i = A^{-1}S_i$, S_i – столбец A , содержащий i -ые компоненты всех частей.

Для построения математической модели хранения данных большой размерности используется аппарат системы остаточных классов, позволяющий распараллелить реализацию арифметических операций, что обеспечивает более высокие скорости выполнения операций, чем при использовании позиционных систем счисления и существенно повышает надежность вычислений, благодаря высокой способности к арифметической самокоррекции [2].

После нахождения обратной матрицы предлагается выполнить перевод коэффициентов матрицы A^{-1} в коды системы остаточных классов. Таким образом, математическая модель хранения данных большой разрядности примет вид:

$$A_{\tilde{N} \tilde{E}}^{-1} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \dots & \varepsilon_{1k} \\ \varepsilon_{21} & \varepsilon_{22} & \dots & \varepsilon_{2k} \\ \dots & \dots & \dots & \dots \\ \varepsilon_{k1} & \varepsilon_{k2} & \dots & \varepsilon_{kk} \end{pmatrix}^{-1}, \quad (6)$$

где ε_{ij} – модулярные представления элементов векторов $e_i \in E$, соответствующих скалярным произведениям $E_i = (l_j, e_i)$. Обобщенный алгоритм, реализующий разработанную модель хранения числовой последовательности высокой разрядности, представлен на рис. 1.

Одним из наиболее трудоемких шагов стадии разборки числовой последовательности высокой разрядности является нахождение матрицы, обратной A . При малых значениях k предлагается использовать алгоритм вычисления обратной матрицы, основанный на рекурсивной блочной схеме умножения

матриц в QT-формате. Указанная схема особенно эффективна в случае, когда коэффициенты матрицы являются элементами конечного поля, характеристика которого не превосходит $2 \cdot 10^8$ [3]. Для значений $k > 1286$ повышение производительности может быть достигнуто применением алгоритма Штрассена для нахождения обратной матрицы [4]. Трудоемкость алгоритма умножения Штрассена, на котором базируется одноименный метод отыскания обратной матрицы, составляет $O(n^{\log_2 7})$ против $O(n^3)$ для традиционного метода [5].

Используя модификации метода Штрассена для параллельных вычислений (алгоритм Штрассена-Ньютона), можно добиться существенного снижения порога значений k , при котором становится целесообразным применение этого алгоритма (до $k > 64$).

В зависимости от размерности матрицы, обращение производится одним из двух способов:

1. Для $k \leq 64$: четыре квадратных блока, полученные в процессе разбиения исходной матрицы (дополненной в случае необходимости до порядка 2^n введением нулевых или единичных элементов) подвергаются повторному разбиению до тех пор, пока порядок блока не достигнет единицы. Результирующая структура является кватернарным деревом.

$$A = \begin{pmatrix} A & C \\ B & D \end{pmatrix}, \begin{pmatrix} I & -A^{-1}\tilde{N} \\ 0 & I \end{pmatrix}, \begin{pmatrix} I & 0 \\ 0 & (D - BA^{-1}C)^{-1} \end{pmatrix}, \begin{pmatrix} I & 0 \\ -B & I \end{pmatrix} \begin{pmatrix} A^{-1} & 0 \\ 0 & I \end{pmatrix}. \quad (7)$$

2. Для $k > 64$: используется параллельная модификация алгоритма Штрассена для обращения матриц (как и в первом случае, при необходимости матрица дополняется нулевыми или единичными элементами):

$$\begin{aligned} P_1 &= A_{11}^{-1}, \quad P_2 = A_{21} \times P_1, \quad P_3 = P_1 \times A_{12}, \\ P_4 &= A_{21} \times P_3, \quad P_5 = P_4 - A_{22}, \quad P_6 = P_5^{-1}, \\ \tilde{N}_{11} &= P_1 - P_3 \times C_{21}, \quad \tilde{N}_{12} = P_3 \times P_6, \\ \tilde{N}_{21} &= P_6 \times P_2, \quad C_{22} = -P_6, \end{aligned} \quad (8)$$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}.$$

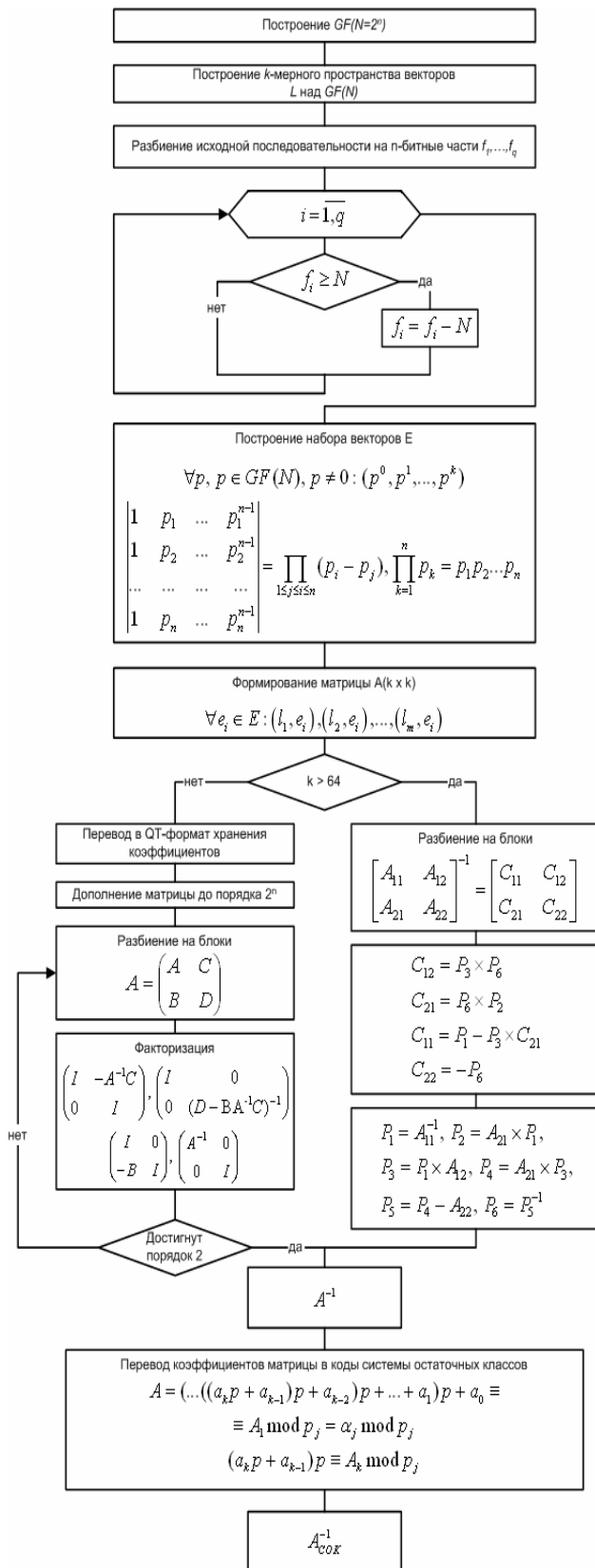


Рис. 1. Обобщенный алгоритм, реализующий модель хранения числовой последовательности высокой разрядности

В качестве метода преобразования позиционных кодов в коды СОК используется модифицированный метод Горнера, или метод последова-

тельного умножения и суммирования по модулю, основанный на представлении числа A , записанного в позиционной системе счисления с основанием p :

$$A = a_k p^k + a_{k-1} p^{k-1} + \dots + a_1 p^1 + a_0 p^0, \text{ или} \\ A = \sum_{i=0}^k a_i p^i, \quad \forall i: 0 \leq a_i \leq p-1, \quad (9)$$

в виде:

$$A = (\dots((a_k p + a_{k-1})p + a_{k-2})p + \dots + a_1)p + a_0 \equiv \\ \equiv A_1 \bmod p_j = \alpha_j \bmod p_j, \quad (10)$$

где $(a_k p + a_{k-1})p \equiv A_k \bmod p_j$.

Операция сборки базового метода (5) состоит из k^2 операций умножения и сложения в поле Галуа, что в свою очередь представляет собой k^2 операций обычного сложения и обычного умножения и $2k^2$ операций вычисления остатка от деления.

Учитывая, что число тактов на выполнение операции вычисления остатка от деления для современных SISD-процессоров превышает число тактов на выполнение операции умножения в среднем в четыре раза, обосновано, что значительное увеличение общей трудоемкости алгоритма связано именно с этим участком вычислений [6]. Использование математического аппарата системы остаточных классов позволяет существенно повысить быстродействие рассматриваемого участка.

Математическая модель обработки данных большой размерности

Построенная математическая модель обработки числовой последовательности высокой разрядности имеет вид:

$$l_i = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \dots & \varepsilon_{1k} \\ \varepsilon_{21} & \varepsilon_{22} & \dots & \varepsilon_{2k} \\ \dots & \dots & \dots & \dots \\ \varepsilon_{k1} & \varepsilon_{k2} & \dots & \varepsilon_{kk} \end{pmatrix}^{-1} S_i, \quad (11)$$

$$D_1 = \bigcup_i l_i, \quad D_2 = \bigcup_j l_j, \quad (12)$$

$$D = (|\alpha_1 \oplus \beta_1|_{p_1}^+, \dots, |\alpha_n \oplus \beta_n|_{p_n}^+),$$

$$D_i = p_{i+1} \left| \delta |\alpha_{i+1} - D_{i-1}|_{p_i}^+ \prod_{i=1}^i p_i \right|_{p_i}^+ + \alpha_{i+1}, \quad (13)$$

где S_i – столбец A , содержащий i -е компоненты всех частей; $D_1 = (\alpha_1 \dots \alpha_n)$, $D_2 = (\beta_1 \dots \beta_n)$ – собранные части исходной последовательности; $p_1 \dots p_n$ – система оснований СОК; $\oplus$ – модульная

операция; $\delta_i = \left| \left(\prod_{i=1}^i p_i - p_{i+1} \right)^{\varphi\left(\prod_{i=1}^i p_i\right)-1} \right|_{\prod_{i=1}^i p_i}^+$;

$\varphi(x)$ – функция Эйлера.

Обобщенный алгоритм, реализующий разработанную модель обработки числовой последовательности высокой разрядности, представлен на рис. 2.

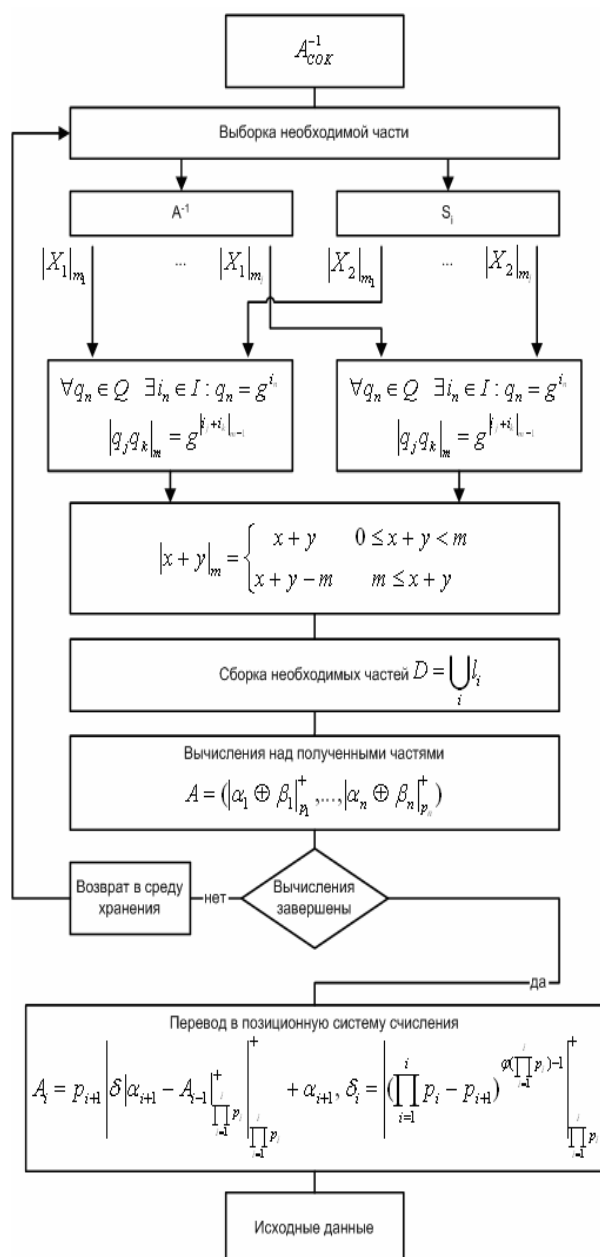


Рис. 2. Обобщенный алгоритм, реализующий разработанную модель обработки числовой последовательности высокой разрядности

Помимо упомянутых свойств распределенных вычислительных систем, основанных на аппарате пороговых схем, одной из отличительных особенностей такого подхода является то, что хранимые данные могут собираться не полностью, а лишь в том объеме, который необходим на данный момент, причем сборка каждой части может производиться независимо от других.

В процессе сборки над частями могут производиться арифметические операции, что позволяет сократить количество необходимых переводов в позиционную систему счисления и выполнять трудоемкую операцию преобразования кода из системы остаточных классов в позиционный код только на финальной стадии вычислений. Для такого преобразования применяется метод, использующий функции Эйлера, основанный на представлении числа A в виде

$$A = p_1 l_1 + \alpha_1, \quad (14)$$

где l_1 показывает, сколько раз p_1 укладывается в числе A , и выражении

$$A_i = p_{i+1} \left| \delta_i \left| \alpha_{i+1} - A_{i-1} \right|_{\prod_{j=1}^i p_j} \right|_{\prod_{j=1}^i p_j}^+ + \alpha_{i+1}, \quad (15)$$

$$\text{где } \delta_i = \left| \left(\prod_{i=1}^i p_i - p_{i+1} \right)^{\varphi\left(\prod_{i=1}^i p_i\right)-1} \right|_{\prod_{i=1}^i p_i}^+;$$

$\varphi(x)$ – функция Эйлера.

Использование векторноматричных процессоров для реализации разработанных алгоритмов

Разработанные алгоритмы могут быть реализованы на любой аппаратной платформе, поддерживающей параллелизм операций, однако, наиболее эффективное отображение алгоритмических решений на аппаратную базу может быть получено при использовании разработки НТЦ «Модуль» – процессора Л1879ВМ1 (NM6403), являющегося одной из наиболее высокопроизводительных моделей, реализующих SIMD-архитектуру. Процессор NM6403 работает с машинными командами 32-х и 64-х разрядного формата и в этом смысле представляет собой суперскалярный микропроцессор со статической VLIW-архитектурой. Одной из важнейших особенностей процессора NM6403 является возможность работы с операндами произвольной длины в диапазоне 1-64 Бит. Эта характеристика процессора особен-

но важна при рассмотрении ее в свете специфики используемого математического аппарата: оба уровня обеспечения параллелизма разработанного комплекса – вычисления в остаточных классах и механизм пороговых схем – позволяют динамически управлять нагрузкой узлов.

Конфигурация векторного сопроцессора зависит от решаемой задачи и может изменяться динамически, в ходе выполнения программы. Для разработанной программы используется конфигурация, основанная на 8-битных операндах [7]. С целью уменьшения вклада операции загрузки в общую производительность вы-

числений, процессор NM6403 предусматривает использование теневой матрицы. Перекодировка весовых коэффициентов в специальный вид, используемый теневой матрицей, занимает 32 такта, однако, выполнение этой операции может выполняться в фоновом режиме, параллельно с другими инструкциями, причем смена рабочей и теневой матриц происходит за один такт.

На рис. 3 приведены схемы функций сложения и умножения больших чисел из разработанной библиотеки. На рис. 4 представлена схема функции локализации и исправления ошибок.

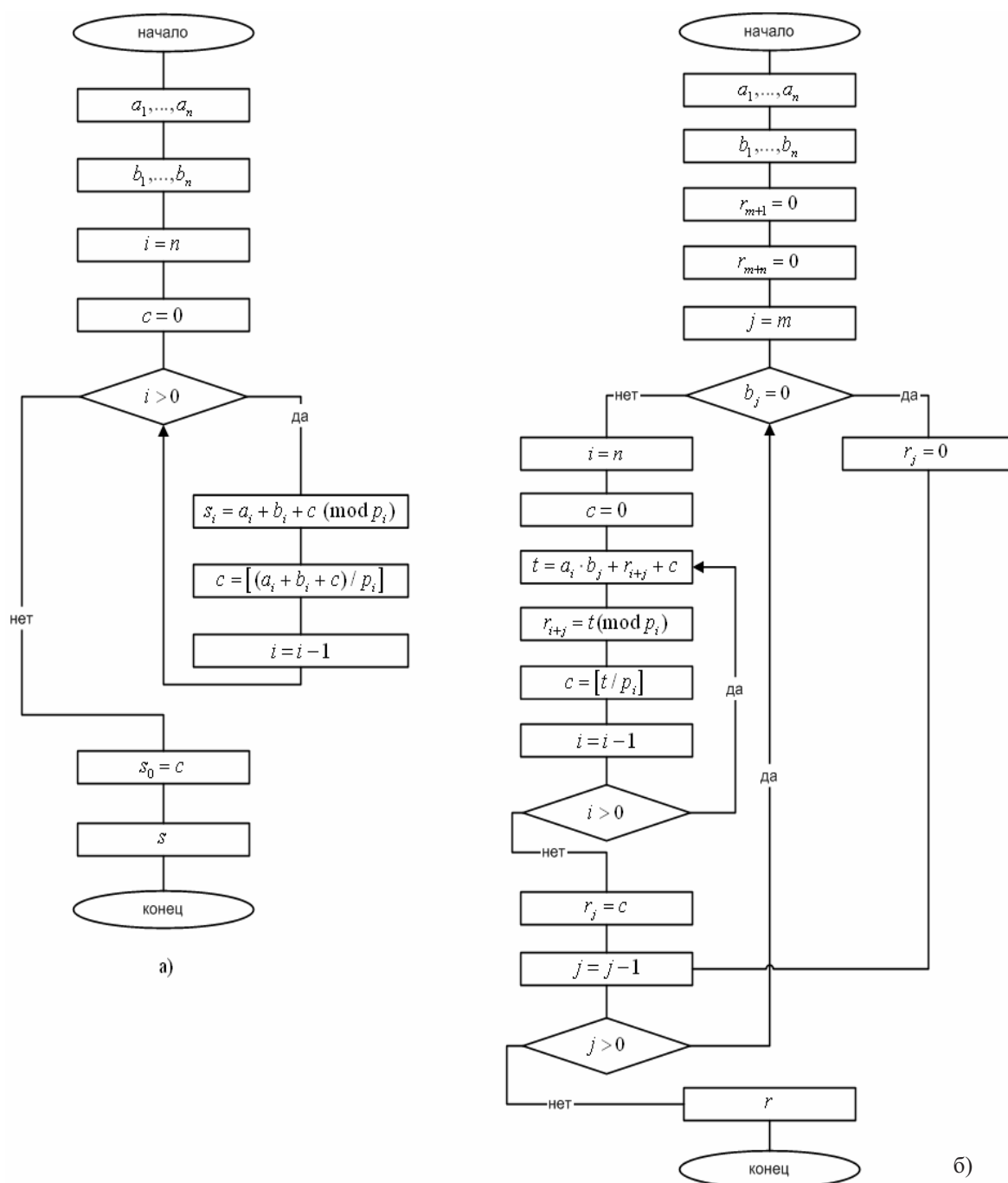


Рис. 3. Алгоритмы вычисления суммы (а) и произведения чисел (б) из большого диапазона

Код функций ориентирован на процессор NM6403, однако, благодаря совместимости коммуникационных портов процессоров NM6403 и серии TMS320C4х, библиотека допускает расширение средствами других вычислительных устройств. Аппаратно-логическая схема разработанной библиотеки приведена на рис. 5.

Показано, что использование принципов векторной обработки и соответствующей аппаратной базы позволяет повысить производительность разработанных алгоритмов до уровня применимости в системах реального времени.

Проведенный анализ отказоустойчивости, опирающийся на метод расчета надежности параллельных структур на основе аппарата функций случайных аргументов с использованием DN-распределения, показал, что введение дополнительной единицы избыточности повышает вероятность безотказной работы системы за расчетный период в среднем на 20%. Таким образом, дополнительная избыточность хранимой информации оправдывается повышенной безотказностью системы и представляется целесообразной.

На основе анализа производительности наиболее трудоемкой части общего алгоритма (вычисление $A^{-1}S_i$) для вычислительной системы из 32 узлов были получены следующие значения показателя ускорения: 19,63 для 16-разрядных операндов, 9,91 для 32-разрядных операндов, что превосходит аналогичные показатели для традиционно используемых алгоритмов в среднем на 32% и 14% соответственно. Анализ производительности функций обработки больших чисел (на примере функции вычисления произведения) показал повышение быстродействия в среднем на 47% для чисел до 10240 десятичных разрядов, и на 16% для чисел большей размерности.

Таким образом, авторами обосновано введение регулируемой избыточности системы хранения данных большой размерности, позволяющей осуществлять эффективное их восстановление в случае выхода из строя одного и более узлов. Разработаны математические модели распределенного хранения и обработки данных большой размерности, базирующиеся на аппарате пороговых схем и обеспечивающие высокую степень достоверности. Для построения математических моделей используется аппарат модулярной арифметики, что позволяет добиться существенного повышения коэффициента ускорения и отказоустойчивости предложенных алгоритмов, открывает перспективы их использования в системах реального времени.

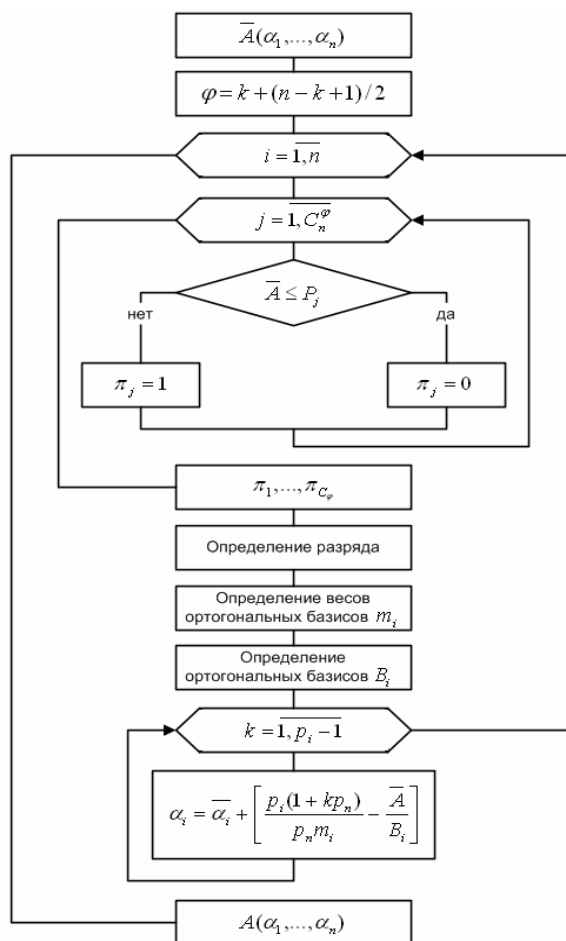


Рис. 4. Схема функции локализации и устранения ошибок

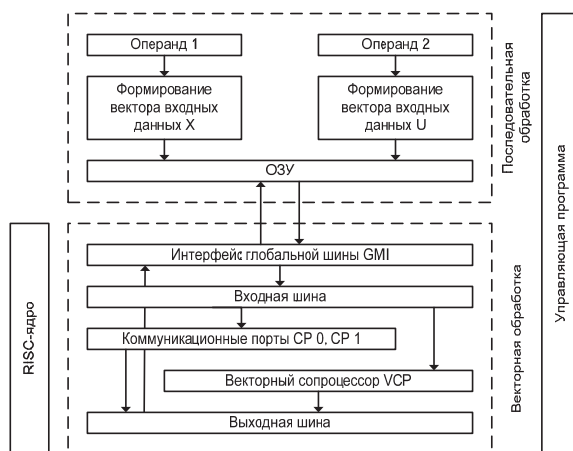


Рис. 5. Аппаратно-логическая схема библиотеки базовых арифметических функций для проведения вычислений в большом диапазоне

Литература

1. Тормасов А.Г., Хасин М.А., Пахомов Ю.И. Модель распределенного хранения данных с регулируемой избыточностью // Эл. журнал «Исследовано в России», 35-2001. – С. 355-364. <http://zhurnal.apelarn.ru/articles/2001/035.pdf>

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. Малашонок Г.И., Аветисян А.И., Валеев Ю.Д., Зуев М.С. Параллельные алгоритмы компьютерной алгебры // Труды Института Системного Программирования РАН. 2004. – С. 169-180.
4. D'Alberto P., Nicolau A. Adaptive Strassen's matrix multiplication // International Conference on Supercomputing, Proceedings of the 21st Annual International Conference on Supercomputing. Seattle, Washington. 2007. – P. 284-292.
5. Bailey D.H., Gerguson H.R.P. A Strassen-Newton algorithm for high-speed parallelizable matrix inversion // Conference on High Performance Networking and Computing. Proceedings of the 1988 ACM/IEEE Conference on Supercomputing. Orlando, Florida, USA, 1988. – P. 419-424.
6. Intel 64 and IA-32 Architectures Optimization Manual / Intel. 2002. – Section C.
7. НТЦ «Модуль». Архитектура процессора цифровой обработки сигналов Л1879ВМ1 (NM6403) / НТЦ «Модуль». 2006. <http://www.module.ru/files/nm6403arch-r.pdf>.

УДК 621.396.4

ОБРАБОТКА СИГНАЛОВ НА ОСНОВЕ МОДЕЛИ В ВИДЕ ОБОБЩЕННОГО НЕЛИНЕЙНОГО УРАВНЕНИЯ ШРЕДИНГЕРА

Григоров И.В.

Рассматривается способ реализации нелинейных фазовых фильтров, построенных на основе модели в виде обобщенного нелинейного уравнения Шредингера, в котором учитываются дисперсионные эффекты высших порядков.

Введение

Нелинейные фазовые фильтры (НФФ) [1-2] представляют собой сравнительно новый класс устройств цифровой обработки сигналов. Они являются достаточно универсальными и могут использоваться для решения ряда радиотехнических задач, в частности для компенсации дисперсионных искажений в различных каналах связи, в том числе и волоконно-оптических [3-4]. НФФ обычно строятся на основе модели в виде нелинейного уравнения Шредингера (НУШ) следующего вида:

$$i \frac{\partial \Psi}{\partial \eta} + \alpha \frac{\partial^2 \Psi}{\partial \tau^2} + f(\Psi) \Psi = 0, \quad (1)$$

где $f(\Psi)$ – произвольная нелинейная функция характеризующая вид нелинейности фильтра, α – постоянный параметр. Уравнение (1) используется в различных областях физики для описания нелинейных волновых процессов. В частности, в нелинейной оптике оно описывает процесс распространения оптических импульсов с комплексной огибающей $\Psi(\eta, \tau)$ по нелинейным оптическим волокнам (ОВ). В этом случае нелинейная функция $f(\Psi)$ выбирается квадратичной (такая нелинейность ОВ называется кубической)

$$f(\Psi) = \kappa |\Psi|^2, \quad (2)$$

здесь κ – параметр нелинейности ОВ. Уравнение (1) в этом случае принимает вид [5]:

$$i \frac{\partial \Psi}{\partial \eta} + \alpha \frac{\partial^2 \Psi}{\partial \tau^2} + \kappa |\Psi|^2 \Psi = 0, \quad (3)$$

здесь η и τ имеют смысл, соответственно, пространственной и временной координат, а α – параметр, влияющий на степень дисперсионного уширения оптического импульса в процессе эволюции.

Нелинейный фазовый фильтр представляет собой устройство цифровой обработки сигналов, которое можно рассматривать как цифровой аналог (1) или, в частности, (2). Он реализуется на основе метода расщепления по физическим факторам [5] и, в простейшем случае, реализуется в виде двух последовательно соединенных звеньев – нелинейного и линейного. Эти звенья имеют соответственно коэффициент преобразования мгновенных значений

$$H(\Psi) = \exp[i f(\Psi)] \quad (4)$$

и импульсную характеристику

$$g(\tau) = g_0 \exp\left(\frac{i a \tau^2}{2}\right), \quad (5)$$

или соответствующую передаточную функцию

$$G(i\omega) = \exp(-i a \Delta \eta \omega^2), \quad (6)$$