$$\frac{P_i(R_i - r_0) + P_j r_0}{R_i} = P_z - q_i \frac{\mu \ln(R_i / r_c)}{2\pi k_{ij} H},$$

откуда

$$P_j = \frac{P_z R_i - P_i(R_i - r_0)}{r_0} - q_i \frac{R_i \mu \ln(R_i / r_c)}{2 r_0 \pi k_{ij} H}.$$

Литература

1. Вахитов Г. Г. Эффективные способы решения задач разработки неоднородных нефтеводоносных пластов. М.: Гостоптехиздат, 1963.

2. Чекалин А. Н. Численное решение задач фильтрации в водонефтяных пластах. Казань: Изво Казанского университета, 1982.

3. Peaceman D. W. Interpretation of Well-Block Pressures in Numerical Reservoir Simulation // SPE Journal. 1978. Vol. 18. P. 183–194.

4. Peaceman D. W. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability // SPE Journal. 1983. Vol. 23, N 3. P. 531–543.

5. Ding Yu, Renard G. A New Representation of Wells in Numerical Reservoir Simulation // SPE Reservoir Engineering. May 1994. P. 140–144.

6. Азиз Х., Сеттари Э. Математическое моделирование пластовых систем. М.: Недра, 1982.

7. Forsyth P. A. A Control-Volume, Finite-Element Method for Local Mesh Refinement in Thermal Reservoir Simulation // SPE Reservoir Engineering. November 1990. P. 561–566.

8. Губайдуллин Д. А., Никифоров А. И., Цепаев А. В. Алгоритмы распараллеливания на сгущающихся сетках в задаче трехфазной фильтрации // Вычислительные методы и программирование. 2007, № 8. С. 360–367.

МОДУЛЯРНО-ЛОГАРИФМИЧЕСКАЯ СТРУКТУРА ДЛЯ МАССОВЫХ АРИФМЕТИЧЕСКИХ ВЫЧИСЛЕНИЙ

И. П. Осинин, В. С. Князьков, Т. В. Князькова

Вятский государственный университет, г. Киров

1. Введение

Крупные задачи, критичные к скорости вычислений, возникают в самых разнообразных областях: в различных сферах промышленности, моделирования климата, в физике, космологии и многих других. Их решение с использованием средств вычислительной техники требует постоянного повышения скорости вычислений, что особенно актуально при создании систем рекордной производительности. На сегодняшний день большинство этих задач решается с использованием кластерных решений.

При этом известны такие традиционные пути повышения быстродействия, как совершенствование техпроцесса, например, создание трехмерных транзисторов, совершенствование алгоритмов выполнения программ, например, распараллеливание вычислений. Однако резерв повышения производительности заложен в ускорении выполнения самих вычислительных операций, например, при применении модулярной арифметики на базе системы остаточных классов (СОК). Естествен-

ный параллелизм устройств, функционирующих на основе СОК, позволяет распараллелить процесс вычислений, как на программном, так и на аппаратном уровне, а модульность и однородность обеспечивает эффективное проектирование структур в сверхбольшом интегральном исполнении (СБИС). Научные работы Червякова Н. И., Акушского И. Я., Akio Sasaki, Garner H., Omondi A. посвящены этому направлению [1, 2].

При этом у СОК имеются и недостатки, среди которых медленное выполнение операций округления и деления, что существенно затрудняет применение плавающей точки. Решение, описанное в данной работе, состоит в применении логарифмической системы счисления (ЛСС), в которой отсутствует операция выравнивания порядков, а деление исходных чисел заменяется вычитанием их логарифмов. Основополагающий вклад внесли ученые Coleman J., Arnold M., Chester E., Lewis D. [3, 4].

С другой стороны, конвейерным параллелизмом обладают однородные вычислительные среды (ОВС), то есть среды, аппаратура которых может реконфигурироваться, меняя свои функции, в зависимости от решаемых, вычислительной системой задач. Это позволяет эффективно адаптировать архитектуру системы под структуру решаемой задачи, обеспечивая тем самым высокий уровень скорости вычислений. Данной тематике посвящен ряд работ отечественных и зарубежных ученых Варшавского В. И., Каляева И. А., Князькова В. С., Flynn M., Moore G., MacSorley L. [5]. В общем случае ОВС представляет собой массив вычислительных ячеек структуры, которые объединены регулярными связями. Ее применение в вычислительном ядре СБИС-процессора позволило практически пропорционально повышать производительность с увеличением числа ячеек в силу естественного параллелизма их работы.

Объединение в единой процессорной архитектуре рассмотренных форм естественного параллелизма приведено далее в данной статье:

– в пункте 1 (введение) приведена проблематика исследования и обзор близких по тематике работ;

– в пункте 2 рассмотрена общая организация архитектуры процессора, включающая арифметическое устройство, вычислительные ядра и преобразователи кодов;

– в пункте 3 рассмотрены средства повышения надежности вычислений на уровне архитектуры модулярно-логарифмического процессора;

– в пункте 4 описан прототип предлагаемого процессора, рассчитана его производительность и проведено сравнение с аналогами;

– в пункте 5 (заключение) приведено обобщение результатов, полученных в ходе данного исследования.

2. Организация архитектуры процессора

Параллелизм на уровне разрядов чисел в случае использования позиционной системы счисления (ПСС) существенно ограничен фактом распространения межразрядных переносов, что негативно сказывается на скорости его работы. Данный недостаток отсутствует в системе остаточных классов (СОК), где вычисления по каждому модулю (основанию) из базиса $\{p_1, p_2, ..., p_n\}$ осуществляются независимо друг от друга.

Пример. Пусть СОК состоит из двух оснований $\{p_1 = 5,\ p_2 = 7\}$. Диапазон чисел ограничен произведением модулей $P = \prod_{i=1}^{n} p_i = 5 \cdot 7 = 35$. Преобразуем число – найдем остатки от деления на p_1 и p_2, например, картеж $\{2, 3\}$ является уникальным представлением числа 17. Остатки, обладая существенно меньшей разрядностью, могут суммироваться, либо умножаться параллельно:

$$+\begin{cases} 17_{10} = \{2_{\mathrm{mod}\,5}, 3_{\mathrm{mod}\,7}\} \\ 12_{10} = \{2_{\mathrm{mod}\,5}, 5_{\mathrm{mod}\,7}\} \end{cases}$$

$$29_{10} = \overline{\{4_{\mathrm{mod}\,5}, 1_{\mathrm{mod}\,7}\}}$$

2.1. Арифметическое устройство

В разработанном и запатентованном арифметическом устройстве [6] именно эта особенность стала первой плоскостью распараллеливания арифметических операций. Например, система модулей из рассмотренного примера ложится в два независимых вычислительных канала. В общем случае каналов может быть n – по количеству остатков СОК (рис. 1).

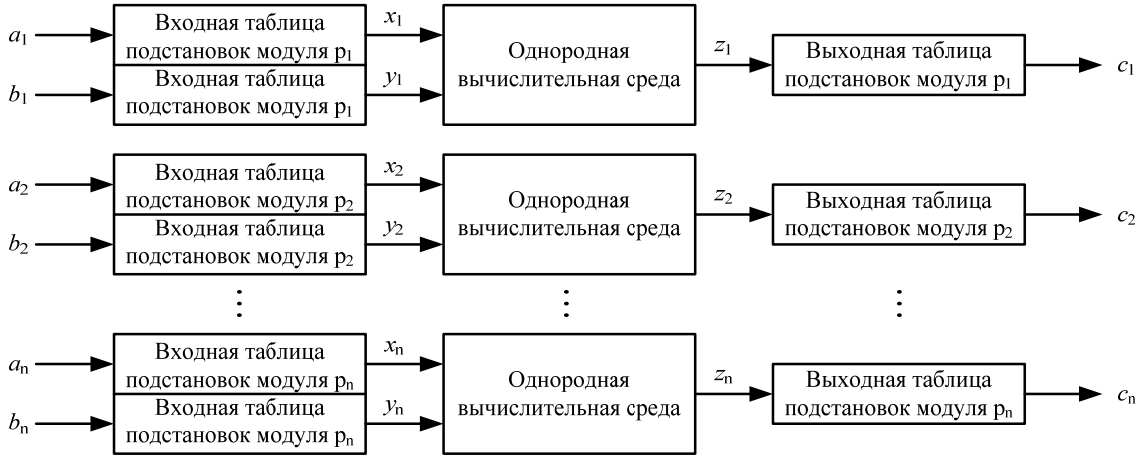

Рис. 1. Организация арифметического устройства

Важным преимуществом СОК является также то, что все основные арифметические операции могут выполняться также быстро, как обычное суммирование. Для операций умножение и деление нацело это становится возможным при использовании входных и выходных таблиц подстановок (Look up table – LUT), осуществляющих однотактные преобразования операндов. Для i-ых остатков a_i и b_i первого и второго операнда соответственно подставляются дискретные логарифмы $x_i = \left|\log_g a_i\right|_{p_i}$, $y_i = \left|\log_g b_i\right|_{p_i}$, затем вычисляется их сумма $z_i = x_i + y_i$ и подставляется ее антилогарифм $c_i = \left|g^{z_i}\right|_{p_i}$, где g – порождающий элемент [1] поля GF (p_i), p_i – модуль СОК, $i \in [1, n]$.

Так как операция сложение остатков является основополагающей в разработанном арифметическом устройстве (к ней сводятся операции вычитание, умножение и деление нацело), ее выполнение распараллелено во второй плоскости с помощью ОВС – квадратной матрицы базовых элементов (БЭ) размерностью $m + 1$, где m – разрядность модуля i-го вычислительного канала. В такой систолической структуре (рис. 2,*а*) базовые элементы (рис. 2,*б*) соединены регулярными связями, образуя вычислительный конвейер. На информационные входы v_j и w_j ОВС подаются соответствующие разряды двоичных m-разрядных остатков x_i и y_i, с информационных выходов u_i снимаются разряды искомой суммы этих остатков, где $i \in [1, n]$, $j \in [1, m]$.

В состав БЭ входят: информационные входы S_1 и S_2, вход синхронизация c, вход сброса r, информационные выходы Q_1 и Q_2, логический элемент «исключающее ИЛИ» (=1), логический элемент «И» и два информационных D-триггера (на рис. 2,*б* обозначены «Т»).

Время заполнения вычислительного конвейера ОВС составляет $T_{\text{ОВС}} = t_{\text{БЭ}}(m+1)$, где m – разрядность двоичного модуля i-го вычислительного канала, $t_{\text{БЭ}} = \max\{t_{\text{XOR}}, t_{\text{AND}}\} + t_{\text{DFF}}$ – время срабатывания БЭ, которое складывается из времени работы логического элемента и информационного DFF-триггера.

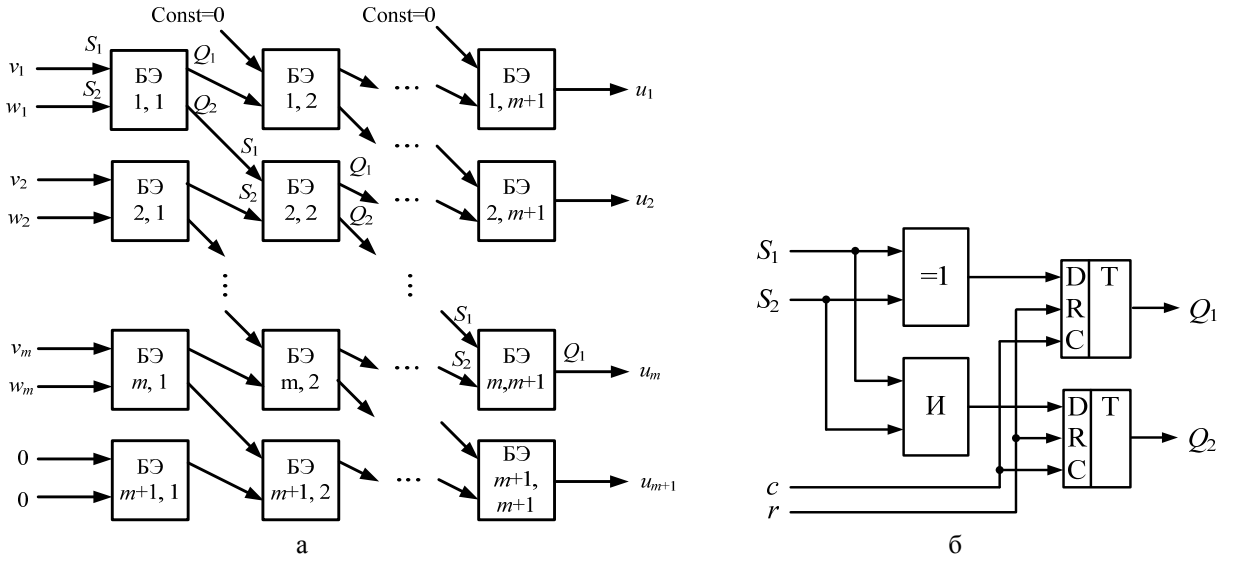

Рис. 2. Организация: а – однородной вычислительной среды, б – базового элемента

После заполнения конвейера время выполнения арифметической операции равно $t_{БЭ}$. Таким образом, первый уровень распараллеливания обеспечен независимым счетом по каждому модулю, второй уровень – конвейерным выполнением операций в ОВС. Ускорение в общем случае достигает k раз для операций сложение и вычитание и k^2 раз для операций умножение и деление нацело по сравнению с последовательными алгоритмами ПСС, где k – разрядность позиционных операндов.

2.2. Вычислительное ядро

Рассмотренное арифметическое устройство является частью вычислительного ядра (рис. 3), где для снижения задержек доступа к памяти используется иерархическая структура памяти с многократным расслоением доступа для каждого модуля СОК. Помимо трехпортовой кэш-памяти (два порта – чтение, один – запись), имеются буферные очереди операндов и результата. Благодаря линейной структуре FIFO, они обладают высоким быстродействием, обеспечивая своевременную подкачку данных в арифметическое устройство.

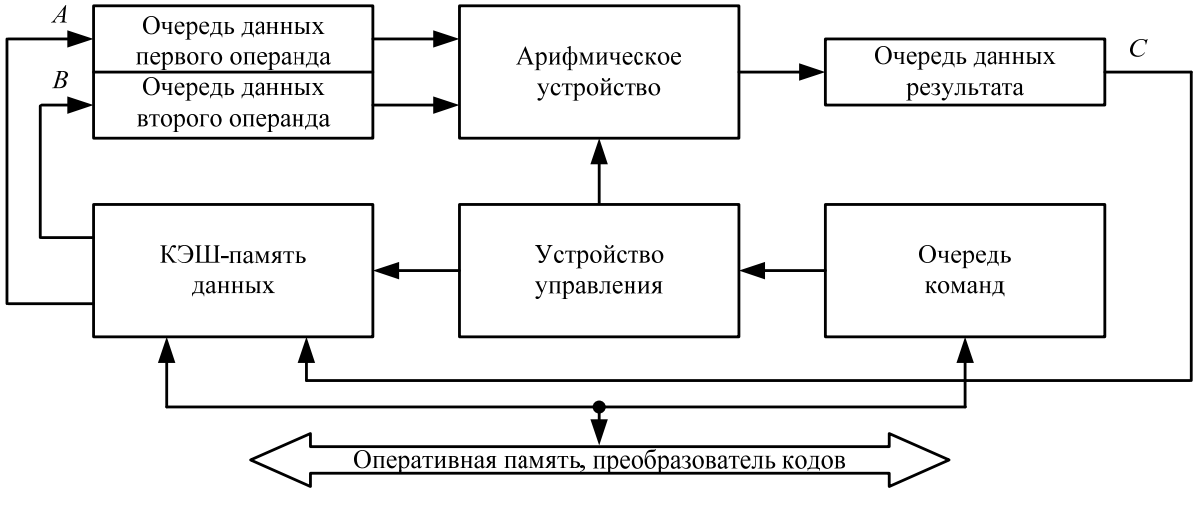

Рис. 3. Организация вычислительного ядра

В разработанном модулярно-логарифмическом MIMD-процессоре таких вычислительных ядер может быть несколько. Рассмотрим вариант, когда их четыре, как представлено на рис. 4. На исходные коды четырехядерного 32-разрядного процессора получено свидетельство о регистрации программы на языке программирования аппаратуры [7].

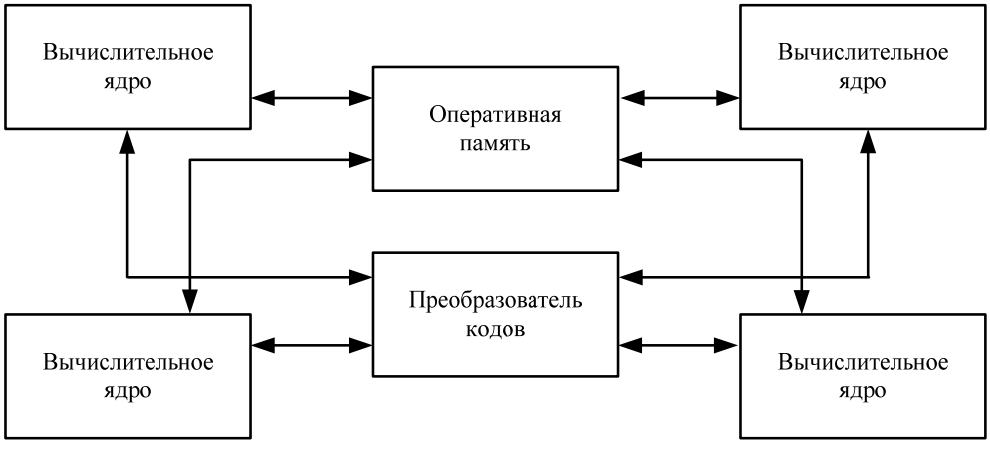

Рис. 4. Организация модулярно-логарифмического процессора

В системе команд процессора используется универсальный 32-разрядный формат (рис. 5), в котором присутствуют следующие поля:

– смещение в сегменте данных исходных операндов и результата;
– вид адресации;
– код операции;
– флаг векторной команды (команда будет применена к строке данных из 8 операндов).

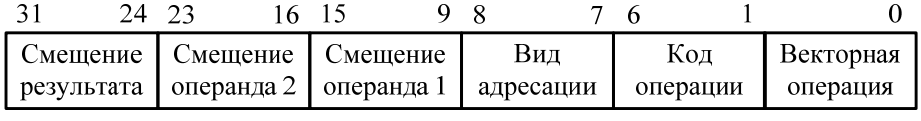

Рис. 5. Формат команды

Используются следующие виды адресации:

– непосредственная (операнд в команде);
– регистровая прямая (первый операнд в регистре константы);
– относительная прямая (смещение указано в команде);
– индексная автоинкрементная (смещение в адресном счетчике).

Задействовано порядка 30 команд, среди которых:

– арифметические операции;
– управление процессом вычислений;
– передача управления прерываниям.

Ключевой особенностью процессора является динамическая реконфигурация числа ядер за счет уникальности набора оснований каждого вычислительного ядра. Ядра могут работать либо параллельно и независимо (32 бита), либо объединяться парами для удвоения точности (64 бита), либо объединяться в одно ядро для учетверения точности вычислений (128 бит). Это бывает необходимо для устранения ошибки переполнения разрядной сетки в процессе счета с помощью увеличения диапазона представления чисел.

Рассмотрим 34-разрядный формат данных (рис. 6). Принадлежность числа в СОК к одному из четырех наборов оснований определяется двухразрядным полем «группа модулей», остальные 32 разряда с помощью набора модулей $p_1 - p_4$ кодируют позиционное число [1]. Для преставления

64-х и 128-ми разрядных чисел модулярное представление выглядит аналогично, отличается лишь количество используемых оснований. Для 64-х битных чисел используются восемь восьмиразрядных остатков, а для 128-ми битных шестнадцать. Группы оснований при этом должны быть различны.

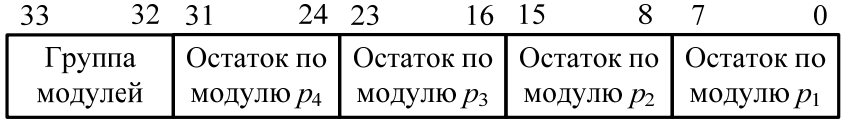

Рис. 6. Формат данных

Для преобразования из одной группы модулей в другую (при передаче данных для вычисления на другом ядре) применяется преобразователь кодов, рассмотренный далее.

2.3. Преобразователь кодов

Наиболее значительным фактором, сдерживающим широкое распространение СОК, является медленное выполнение операций преобразования из позиционной системы и обратно, вычисление позиционной характеристики числа и некоторых других. Поэтому особую важность имеет повышение скорости их выполнения. Рассмотрим, за счет чего в процессоре происходит их ускорение на примере операции преобразования кодов СОК в ПСС. Одним из методов преобразования кода системы остаточных классов в позиционный код является способ преобразования через интервальную характеристику числа, позволяющую оценить величину числа без полного преобразования в ПСС [1]. Для этого используется выражение:

$$A = \left| \sum_{i=1}^{n} l_i \, a_i \right|_{p_i} \; p_{\max} + a_{\max},$$

где A – позиционное представление числа, $p_{\max}$ и $a_{\max}$ – наибольший модуль СОК и остаток по этому модулю соответственно, $l_i = \dfrac{p_i^{\phi(p_i)} - 1}{p_i}$, a_i – i-ый остаток числа в СОК, n – количество модулей СОК, $\varphi(p_i)$ – функция Эйлера [1] модуля p_i, $i \in [1; n]$. Данный способ удобен тем, что позволяет вычислять интервальную характеристику числа. Она необходима при выполнении немодульных операций, требующих позиционного представления числа. Имея такую характеристику числа в СОК, преобразование в ПСС осуществляется за одну операцию умножение с суммированием (рис. 7). Как и коррекция по модулю, эта операция является частным случаем мультиоперандного суммирования, то есть суммирования нескольких слагаемых одновременно.

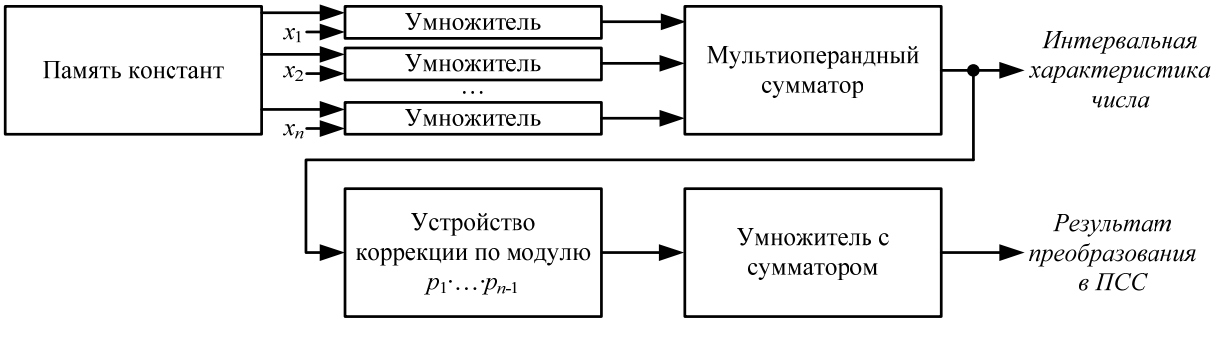

Рис. 7. Организация преобразователя кодов из СОК в ПСС

Разработанный мультиоперандный сумматор представляет собой систолическую структуру, функционирующую на базе арифметики разрядных срезов, которая также обладает естественным параллелизмом.

Одномерный массив слагаемых M представим приведенной ниже двумерной битовой матрицей V размерностью m строк и n столбцов, где m – число слагаемых, n – разрядность модулей чисел.

$$V = \begin{pmatrix} v_{1,n},\ldots,v_{1,2},v_{1,1} \\ v_{2,n},\ldots,v_{2,2},v_{2,1} \\ \ldots \\ v_{m,n},\ldots,v_{m,2},v_{m,1} \end{pmatrix}.$$

Можно выполнять обработку матрицы V не по строкам – скалярным значением слагаемых в двоичном представлении, как это реализуется в известных способах суммирования, а по столбцам – разрядным срезам SR_i исходных чисел, $i \in [1; n]$. Каждый разрядный срез является одномерным двоичным вектором из m элементов.

Специализированная ОВС (рис. 8,*а*) выполняет подсчет количества единичных бит в i-ом разрядном срезе, подаваемом на входы $v_{j,i}$ в параллельно-конвейерном режиме, после чего младший разряд подсчета становится разрядом искомой суммы (выход u), а остальные являются разрядами переноса в следующий разрядный срез SR_{i+1}, $i \in [1; n]$, n – разрядность исходных чисел, $j \in [1; m]$, m – количество слагаемых. Такая организация вычислений позволяет добиться повышения скорости вычислений в m раз (время мультиоперандного суммирования в ОВС составляет n тактов при $n >> m$) по сравнению с традиционным последовательным способом, где время мультиоперандного суммирования составляет $n \cdot m$ тактов [8].

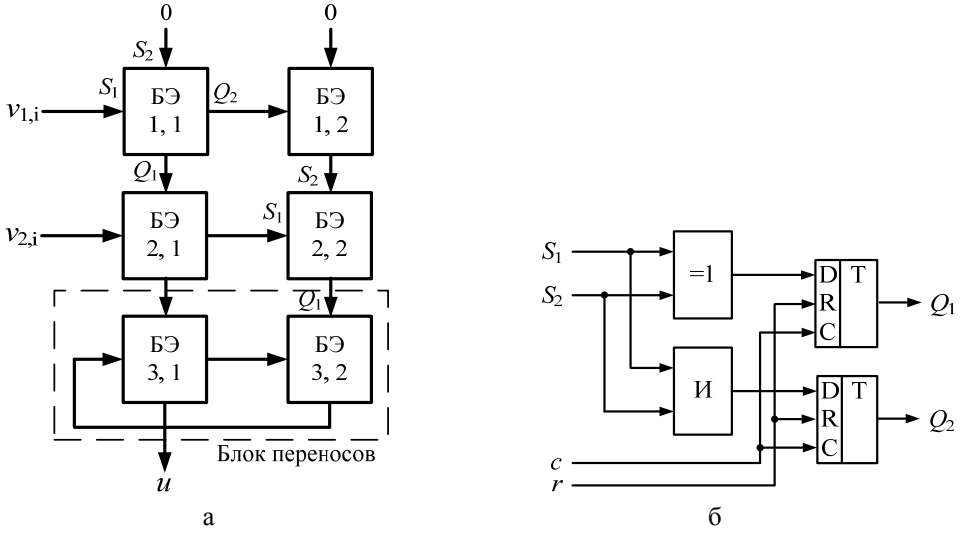

Рис. 8. Организация: а – фрагмента ОВС для мультиоперандного суммирования, б – базового элемента

Пример. Требуется вычислить сумму модулей четырех двоичных чисел: $v_1 = 00111_2$, $v_2 = 00101_2$, $v_3 = 00001_2$, $v_4 = 00111_2$. Тогда:

$$V = \begin{pmatrix} 00111 \\ 00101 \\ 00001 \\ 00111 \end{pmatrix}.$$

Битовая матрица V массива слагаемых содержит пять разрядных срезов: $SR_1=1111_2$, $SR_2=1001_2$, $SR_3=1101_2$, $SR_4=0000_2$, $SR_5=0000_2$. На первом этапе выполняется параллельное вычисление количества единиц B_i в них: $B_1 = 100_2$, $B_2 = 010_2$, $B_3 = 011_2$, $B_4 = 000_2$, $B_5 = 000_2$. Младший разряд B_1 является младшим разрядом $s_1 = 0$ итоговой суммы чисел. Затем вычисляется значение $B_2^* = (B_1 \rightarrow) + B_2 = 100_2$, где $\rightarrow$ обозначает операцию сдвига двоичного числа на один разряд вправо. Младший разряд полученной суммы B_2^* является вторым $s_2 = 0$ разрядом искомой суммы S. В итоге серии аналогичных вычислений вычисляется искомая сумма $S(s_5, s_4, s_3, s_2, s_1) = 10100_2$.

Важным моментом является использование тех же БЭ (рис. 8,*б*), что и в ОВС, входящей в состав арифметического устройства. Это позволяет значительно повысить регулярность элементов процессора, что, в конечном счете, обеспечивает высокую технологичность в случае массового производства на базе СБИС. Однако, помимо естественного параллелизма, СОК обладает свойствами, повышающими надежность процесса вычислений. Средства, которые их реализуют в разработанном процессоре, приведены в следующем разделе.

3. Надежность вычислений

Надежность является одним из основных свойств любого вычислительного устройства, во многом определяющим его пригодность для использования по назначению. На сегодняшний день в супер-ЭВМ содержатся сотни тысяч и даже миллионы процессорных ядер. Сбои и отказы в таких системах приводят к тому, что алгоритм программы становится недетерминированным и в результате расчета можно получить разный результат при одинаковых наборах входных данных. Кроме того, существуют области вычислений, где ошибка в вычислениях может привести к фатальным последствиям. Например, к ним относятся задачи наведения ракет, управления атомной электростанцией, функционирования космических аппаратов и другие.

Существуют такие традиционные пути повышения надежности, как:

– резервирование на уровне компонент системы;

– дублирование вычислений;

– дополнительная обработка исключительных ситуаций и т. п.

Их недостаток состоит в том, что часть оборудования простаивает, либо дублирует работу, вплоть до отказа одного из компонентов системы. Данного недостатка можно избежать, внедрив средства повышения надежности на уровне архитектуры. Рассмотрим более подробно, как это реализовано в предлагаемом процессоре.

3.1. Средства повышения надежности модулярно-логарифмического процессора

Сочетание корректирующих свойств системы остаточных классов и динамической реконфигурации ОВС позволили реализовать следующие средства повышения надежности.

Во-первых, это выявление и исправление ошибок за счет расширения вычислительного диапазона путем назначения одного или нескольких вычислительных каналов в качестве контрольных (рис. 9). Доказано [1], что при возникновении ошибки, результат операции окажется в запрещенном контрольном интервале. Причем, в процессоре применен механизм, локализующий вычислительный канал, где произошел сбой. В случае подобного сбоя происходит исправление ошибки путем вычисления верного остатка результата и инкремент счетчика ошибок вычислительного канала.

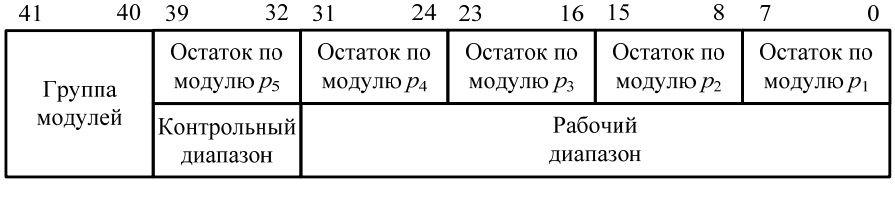

Рис. 9. Отображение областей чисел динамического диапазона

Во-вторых, при превышении заданного порога в счетчике ошибок возникает ситуация отказа и канал блокируется. Однако в отличие от повсеместно распространенных универсальных вычислительных устройств, модулярно-логарифмический процессор (МЛП) продолжает работу, сокращается лишь диапазон представления чисел. Если при этом нельзя снижать точность вычислений, то можно использовать вычислительные каналы другого ядра, в том числе и контрольные (рис. 10). Такая реконфигурация позволяет создать процессор, устойчивый к постепенной деградации. В пределе, вычисления будут продолжаться, пока хотя бы один вычислительный канал будет работоспособен. При этом разрядность обрабатываемых чисел будет равна разрядности канала.

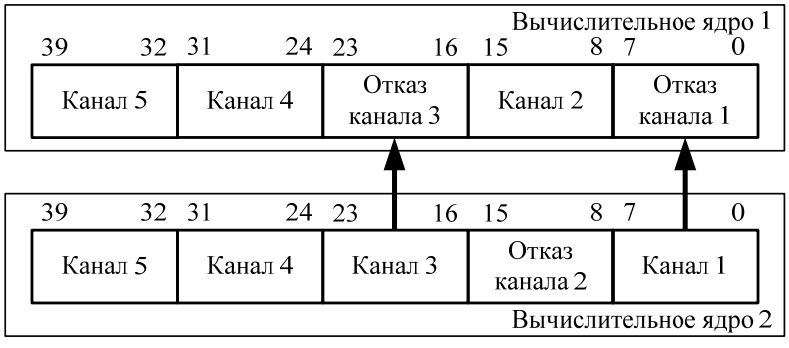

Рис. 10. Замена отказавших вычислительных каналов путем реконфигурации

В-третьих, рассмотренная в п. 2.2 динамическая реконфигурация числа ядер позволяет управлять разрядностью операндов для устранения ошибок переполнения разрядной сетки. Затем, при необходимости, разрядность чисел может быть понижена до исходной.

В-четвертых, на базе предложенного процессора имеется возможность реализовать совершенно новые подходы в области аппаратного шифрования данных, например, можно настолько быстро менять параметры остатков, чтобы период действия ключа на базе текущих параметров был меньше минимального времени его вскрытия [2].

Кроме того, в процессе счета могут возникать неявные ошибки, связанные с округлением чисел в процессе вычислений по стандарту IEEE-754. К тому же, работа с плавающей точкой связана с делением чисел, что негативно сказывается на скорости выполнения операций в СОК, так как данная операция является немодульной. Эти причины привели к необходимости применения аналога плавающей точки – логарифмической системы счисления (ЛСС), где в процессе вычислений отсутствуют фазы нормализации и денормализации, что избавляет от ошибок округления и необходимости деления на степень двойки.

3.2. Модулярно-логарифмический формат данных

Любое вещественное число a представлено в ЛСС его знаком и двоичным логарифмом его абсолютной величины:

$$L(a)=\begin{cases}\log_2|c\cdot a|, & \text{если } |a|>c^{-1},\\ 0, & \text{если } |a|\le c^{-1},\end{cases}$$

где c – константа для представления чисел диапазона (0,1). Такое представление чисел позволяет производить вычисления не с вещественными числами, а с их логарифмами, причем операции умножение и деление заменяются операциями сложение и вычитание логарифмов соответственно, а возведение в степень и извлечение корня – операциями умножение и деление логарифмов соответственно. Этот факт позволяет значительно ускорить выполнение перечисленных операций по сравнению с плавающей точкой.

Однако за это приходится расплачиваться более сложным [3] выполнением операций сложение и вычитание вещественных чисел x и y:

$$\log_2(x \pm y) = \log_2 x + \log_2\left(1 \pm 2^{\log_2 y - \log_2 x}\right),$$

которые реализованы в предложенном процессоре с помощью аппаратной реализации алгоритма квадратичной интерполяции [4].

Перевод вещественного числа в модулярно-логарифмический формат проходит следующим образом:

– порядок числа с плавающей точкой становится целой частью логарифма;

– вычисляется логарифм мантиссы;

– логарифмический код числа, состоящий из целой и дробной части логарифма, преобразуется в СОК как единое целое число;

– в старший разряд числа записывается знак без изменений.

Модулярно-логарифмический формат данных, приведенный на рис. 11, соответствует одинарной точности вещественного числа по стандарту IEEE-754.

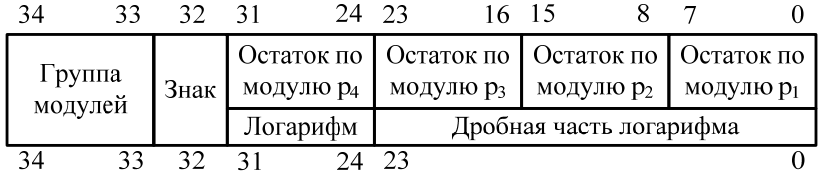

Рис. 11. Модулярно-логарифмический формат данных

3.3. Оценка надежности модулярно-логарифмического процессора

При построении надежностных характеристик предположим, что:

– в начальный момент времени в каждом из 4 вычислительных ядер имеется $n = k + r$ каналов, где $k = 4$ – количество информационных каналов, $r = 1$ – количество контрольных каналов;

– минимальное количество каналов, необходимых для функционирования процессора k;

– отказ более чем $3 \cdot k + 4 \cdot r$ каналов является отказом всего процессора;

– отказавшие каналы не восстанавливают работоспособность;

– отказы каналов являются статистически независимыми событиями.

Таким образом, надежностная структура процессора соответствует модели скользящего резервирования, где резервные элементы находятся в нагруженном состоянии [1]. Тогда вероятности безотказной работы:

$$R_{\text{МЛП}} = \sum_{0}^{3k+4r} P^{4\cdot(k+r)-i}(1-P)^i,$$

где $P(t) = e^{-\lambda t}$ – вероятность безотказной работы канала, λ – интенсивность его отказа.

Определение и оперативная коррекция ошибок в ПСС возможна лишь при условии одновременной работы нескольких устройств по принципу голосования, например, «2 из 3». Вероятность безотказной работы такой системы:

$$R_{\text{ПСС}} = 3P_{\text{ПСС}}^2 - 2P_{\text{ПСС}}^3,$$

где $P_{\text{ПСС}}(t) = e^{-n\lambda_{\text{ПСС}} t}$, $\lambda_{\text{ПСС}}$ – интенсивность отказа одного разряда, n – разрядность позиционного процессора.

Результаты расчета надежности в течение $t = 2000$ часов функционирования представлены графически на рис. 12. Анализ полученных зависимостей свидетельствует о преимуществе в надежности модулярно-логарифмического процессора. Например, при интенсивности отказов $\lambda = 2 \cdot 10^{-6}$

он обеспечивает вероятность безотказной работы $R = 0{,}5$; тогда как позиционный процессор $R = 0{,}1$. Избыточность аппаратурных затрат составляет 25 % и 66 % соответственно.

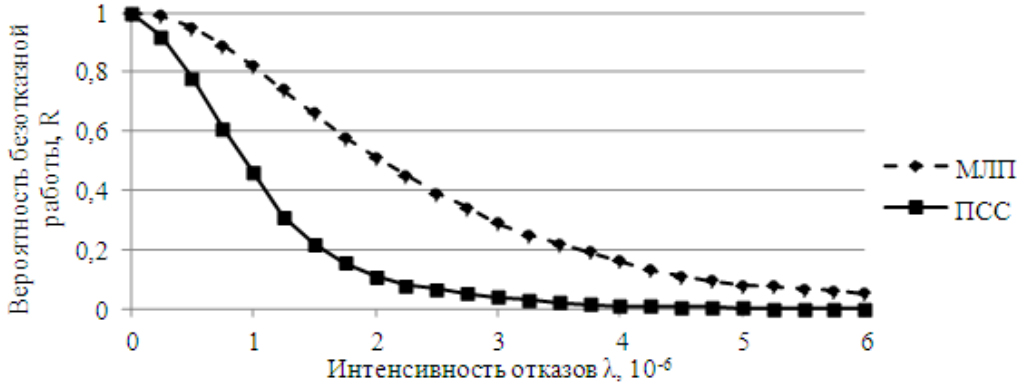

Рис. 12. Модулярно-логарифмический формат данных

4. Прототип и оценка производительности процессора

На данный момент модулярно-логарифмический процессор существует в виде прототипа – IP-блока RISC-процессора NIOS на базе программируемой логической интегральной схемы (ПЛИС). Данный выбор обусловлен простотой реализации, так как можно использовать уже готовые функциональные блоки процессора NIOS, дополняя его систему команд и используя уже готовый компилятор, для написания и отладки программ.

Проектирование функциональных схем процессора проводилось в САПР Quartus II Web Edition фирмы Altera. Отладка и тестирование выполнялось на базе платы с ПЛИС Altera Cyclone V.

Так как элементные базы ПЛИС и СБИС существенно отличаются по скоростным характеристикам, то при сравнительной оценке целесообразно использовать количество операций за такт работы устройства. Аппаратные затраты блока векторных расширений AVX процессора Xeon Phi семейства Knight Landing оценены приблизительно (порядка 86 млн. транзисторов) по изображениям планировки функциональных блоков (floorplan) выпускаемого на данный момент процессора семейства процессоров Knights Corner.

Предложенный прототип выполняет до четырех операций за такт. Аппаратные затраты составляют 9,2 млн. транзисторов. При увеличении числа вычислительных ядер до 32 штук МЛП-процессор способен догнать по числу операций за такт блок векторного расширения Intel AVX-512 (16 команд за такт). Экстраполировав количество транзисторов, необходимое для создания 32-ядерного МЛП, получим 17 % выигрыш по аппаратным затратам (73,6 млн. против 86 млн. транзисторов) по сравнению с блоком AVX-512.

Кроме того, сгруппировав ядра МЛП воедино, можно обеспечить выполнение одной 1024-разрядной операции за такт, тогда как в AVX-512 предусмотрены максимум 512-разрядные операции, выполняемые за такт. В этом случае на целочисленных операциях МЛП будет вдвое быстрее выполнять сложение (два такта AVX-512) и в 11 быстрее умножение (11 тактов AVX-512 с использованием алгоритма Карацубы [3]).

Помимо универсальных процессоров, у МЛП существуют специализированные аналоги. Например, модулярный нейропроцессор, разработанный в Ставропольском военном институте связи [1]. Ключевое отличие МЛП от него состоит в применении ОВС вместо нейросетей, а также наличии блока обработки вещественных чисел на базе логарифмической системы счисления. Другой аналог – «European Logarithmic Microprocessor», разработанный в университете Ньюкасл [3] и функционирующий на базе ЛСС. Ключевое отличие МЛП состоит в применении СОК и ОВС для распараллеливания вычислений на уровне разрядов чисел.

5. Заключение

Таким образом, в представленном модулярно-логарифмическом процессоре заложен потенциал параллельного счета и обеспечения отказоустойчивости на уровне архитектуры, что делает актуальным его применение для расчета задач, критичных не только к скорости, но и к надежности вычислений.

Высокая производительность достигается за счет:

– распараллеливания арифметических операций на уровне остатков непозиционной системы счисления;

– параллельно-конвейерной обработки каждого модуля в однородной вычислительной среде;

– замены операций умножение и деление вещественных чисел операциями сложение и вычитание соответственно, благодаря свойствам логарифмов.

Надежность вычислений обеспечивается:

– обнаружением и коррекцией ошибок;

– устойчивостью к постепенной деградации оборудования;

– динамической реконфигурацией числа ядер для управления разрядностью вычислений;

– отсутствием ошибок округления за счет применения ЛСС.

Текущая реализация в виде IP-блока софт-процессора подтверждает реализуемость архитектурных особенностей, описанных в данной статье. В перспективе такой процессор может быть создан не только в качестве дополнения системы на кристалле, но и как самостоятельное устройство. Например, в виде арифметического ускорителя, подключаемого к компьютеру с традиционной архитектурой, либо в качестве центрального процессора на базе СБИС. В любом из этих вариантов высокая производительность, масштабируемость и реконфигурируемость достигается за счет применения однородных вычислительных сред, функционирующих в базисе системы остаточных классов и логарифмической системы счисления.

Сферами применения МЛП являются:

– высокопроизводительные вычисления, требующие работы с большой разрядностью данных − 1024 бит и более (задачи моделирования климата, исследование электромагнитного рассеяния, исследование орбитальной эволюции небесных тел и другие), где за счет распараллеливания счета на множество независимых остатков достигается прирост производительности по сравнению с известными аналогами (например, для целых чисел минимум в два раза по сравнению с Intel AVX-512 при сокращении аппаратных затрат на 17 %);

– высоконадежные вычисления, требующие постоянного контроля правильности вычислений (задачи наведения ракет, управления атомной электростанцией, функционирования космических аппаратов и подобные), где за счет применения корректирующих свойств СОК достигается уменьшение аппаратных затрат по сравнению с аналогами.

Литература

1. Червяков Н. И. Модулярные параллельные вычислительные структуры нейропроцессорных систем. М.: Физматлит, 2003.

2. Магометов Ш. Г. Передача и прием данных в вычислительных устройствах с использованием системы остаточных классов. URL [Электронный ресурс]. Режим доступа: http://www.mathnet.ru/vagtu355 (дата обращения: 24.10.2015).

3. Coleman J. N., Chester E. I. Arithmetic on the European Logarithmic Microprocessor // IEEE Transactions on Computers. 2000. Vol. 49, N 7. P. 702–715.

4. Ismail R. C., Hussin R., Muzad S. A. Interpolator Algorithms for approximating the LNS addition and substraction // IEEE International Conference on Circuits and Systems (ICCAS). Kaula Lumpur, 3–4 Oct. 2012. P. 174–179.

5. Каляев И. А. Реконфигурируемые мультиконвейерные вычислительные структуры. Ростов-на-Дону: Изд-во ЮНЦ РАН, 2008, 320 с.

6. Пат. 2477513 РФ. МПК G06F7/72. Ячейка однородной вычислительной среды, однородная вычислительная среда и устройство для конвейерных арифметических вычислений по заданному модулю / И. П. Осинин, В. С. Князьков опубл. 10.03.2013, Бюл. № 7.

7. Осинин И. П., Князьков В. С. Организация параллельно-конвейерной СБИС-структуры с реконфигурируемой микроядерной архитектурой // Известия высших учебных заведений. Серия: Технические науки. 2013, № 3. С. 74–83.

8. Пат. 2491612 РФ. Способ организации вычислений суммы n m-разрядных чисел / И. П. Осинин, В. С. Князьков опубл. 27.08.2013, Бюл. № 24.

9. Осинин И. П., Князьков В. С. Концепция разрядно-параллельного арифметико-логического устройства на базе СБИС-структур // XIV Международная конференция «Супервычисления и математическое моделирование»: сб. докл. Саров, 1–5 октября 2012. С. 449–458.

ЧИСЛЕННОЕ РЕШЕНИЕ ДИНАМИЧЕСКИХ ЗАДАЧ НЕОДНОРОДНЫХ ДЕФОРМИРУЕМЫХ СРЕД НА СУПЕР-ЭВМ

И. Б. Петров[1,2], *В. И. Голубев*[1,2], *Н. И. Хохлов*[1,2], *А. В. Фаворская*[1,2]

[1]Федеральный научный центр НИИ системных исследований РАН, г. Москва
[2]Московский физико-технический институт (государственный университет), г. Москва

1. Введение

В настоящее время сейсморазведка является наиболее распространенным методом поиска и разведки месторождений полезных ископаемых нефти и газа. Несмотря на то, что основной ее задачей является определение структуры подповерхностного пространства, важной составляющей решения данной обратной задачи является детальное описание процесса распространения сейсмических волн от сейсмического источника вглубь геологической среды и формирования сейсмических откликов от различных неоднородностей. Решение прямых задач сейсморазведки может быть использовано, например, для уточнения запасов углеводородов в районах с большим числом эксплуатируемых скважин, построения синтетических сейсмограмм для нужд региональной сейсморазведки, уточнения наличия углеводородов и их характеристик в известной геологической структуре.

В настоящей работе на основе прямых численных расчетов продемонстрировано преимущество явного выделения контактных границ по сравнению с методом сквозного счета для задач наземной сейсмической разведки. На примере Арктической задачи проведено сопоставление результатов моделирования, полученных решением полных акустических и упругих задач.

2. Математическая модель и используемый численный метод

Для математического моделирования волновых процессов в деформируемом твердом теле использовалась система динамических уравнений, объединяющая уравнения движения и реологические соотношения в виде:

$$\rho \dot{v}_i = \nabla_j \sigma_{ij},$$
$$\dot{\sigma}_{ij} = q_{ijkl}\varepsilon_{kl} + F_{ij}.$$

Здесь ρ – плотность среды, $\dot{v}_i$ – компоненты скорости смещения, $\dot{\sigma}_{ij}$, ε_{kl} – компоненты тензоров напряжения и деформаций, ∇_j – ковариантная производная по j-й координате, F_{ij} – добавочная правая часть.