

## МОДУЛЯРНО-ЛОГАРИФМИЧЕСКИЙ СОПРОЦЕССОР ДЛЯ МАССОВЫХ АРИФМЕТИЧЕСКИХ ВЫЧИСЛЕНИЙ\*

© 2017 г. И.П. Осинин

*Саровская лаборатория имитационного моделирования*

*(607190 Саров, ул. Маяковского, д. 42)*

*E-mail: iposinin@mail.ru*

Поступила в редакцию: 01.05.2017

Предлагаемый сопроцессор представляет собой самостоятельный сложнофункциональный (intellectual property — IP) блок системы-на-кристалле, позволяющий проводить математические вычисления над вещественными числами в уникальной модулярно-логарифмической системе счисления. Обеспечены два уровня преобразования исходных чисел: в модулярную систему счисления вместо традиционной позиционной и в логарифмическую систему счисления вместо плавающей точки. Благодаря этому сопроцессор обладает более высоким быстродействием, точностью и надежностью вычислений по сравнению с известными аналогами. Он состоит из набора одинаковых вычислительных ядер, каждое из которых выполняет одноктактовые скалярные или векторные операции. В результате проведенных исследований и разработок предложены новые научные и технические решения, реализующие предложенные способы вычислений и кодирования данных. При этом преобразование кодов в модулярно-логарифмическую систему счисления и обратно не вносит значительных временных задержек при большом потоке входных данных за счет предложенных аппаратных решений, конвейеризирующих процесс интерполяции функции логарифма и преобразования кодов системы остаточных классов. Реализован прототип устройства на базе программируемой логической интегральной схемы в виде IP-блока. Целевой рынок решения — компании разработчики универсальных процессоров.

*Ключевые слова: сопроцессор, реконфигурируемая архитектура, система остаточных классов, логарифмическая система счисления, высоконадежные вычисления.*

### ОБРАЗЕЦ ЦИТИРОВАНИЯ

Осинин И.П. Модулярно-логарифмический сопроцессор для массовых арифметических вычислений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 2. С. 22–36. DOI: 10.14529/cmse170202.

### Введение

В течение последних десятилетий наблюдается экспоненциальный рост вычислительных возможностей ЭВМ, благодаря чему высокопроизводительные вычисления стали важной технологией при проведении исследований и разработок развитых государств.

При этом помимо скорости вычислений, все большее внимание уделяется точности расчетов и надежности супер-ЭВМ, что связано с постоянным увеличением количества вычислительных ядер. Например, у TaihuLight — лидера списка Top500 [1] за июнь 2016 года их число превысило 10 млн. штук.

Традиционно для решения подобных задач используются вычисления с плавающей точкой по стандарту IEEE-754 [2], который обладает рядом существенных недостатков,

---

\* Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии (ПаВТ) 2017».

например, критические ошибки округления зачастую приводят к неверным результатам расчетов.

С другой стороны, в распространенных на рынке процессорах никак не задействовано распараллеливание самих арифметических операций на уровне системы счисления. Раскрытие этого потенциала несет в себе дополнительный потенциал повышения скорости вычислений.

Современные отечественные микропроцессоры не лишены упомянутых особенностей, при этом у них отсутствует блок векторной обработки вещественных чисел. В связи с этим актуально создание IP-блока сопроцессора, дополняющего основную систему-на кристалле и расширяющего набор команд математическими скалярными и векторными инструкциями. Применение такого сопроцессора обеспечит дополнительные конкурентные преимущества компаниям на рынке супер-ЭВМ.

В данной работе рассмотрена концепция самостоятельного IP-блока системы-на кристалле, позволяющего проводить математические расчеты с изменением традиционных принципов вычислений. Продукт не имеет прямых аналогов за счет применения модулярно-логарифмической системе счисления. Сопроцессор является конкретной реализацией научных и технических решений, приведенных в общем виде в [3].

Статья структурирована следующим образом. Первый раздел посвящен обобщению результатов исследования актуальности данной разработки. Второй содержит описание базовой технологии. В третьем приведены конкурентные преимущества предложенного сопроцессора и сценарии его внедрения. В заключении обобщены результаты, полученные в ходе данного проекта.

## 1. Исследование актуальности модулярно-логарифмических вычислений

Высокопроизводительные вычисления стали одной из важнейших технологий проведения исследований и разработок в области моделирования космоса, климата, макромира, микромира и многих других. Проведенный анализ выявил [4], что все они используют формат машинной арифметики IEEE-754. Его достоинства и недостатки, а также основные способы повышения скорости и надежности вычислений рассмотрены в данном пункте.

### 1.1. Способы повышения точности вычислений

IEEE 754 — широко распространенный стандарт, описывающий формат чисел с плавающей точкой, в котором вещественные числа представлены в виде знака, показателя степени и мантиссы.

Преимущество данного формата состоит в существенно большем диапазоне для 64-битных чисел, который составляет порядка  $10^{308}$  значений по сравнению с фиксированной точкой ( $10^{19}$  значений). Однако ему присущи и серьезные недостатки, среди которых, например, наличие округлений, возникающих в ходе вычислений, что приводит к нарушению алгебраических свойств коммутативности и дистрибутивности. Это приводит к тому, что традиционные процессоры зачастую выдают ошибочный результат вычислений на следующих типах задач [4]:

- плохо обусловленные;
- с разномасштабными коэффициентами;

- чувствительные к классу эквивалентных преобразований.

Для их верного решения требуются высокоточные вычисления, основанные на:

- программных подходах;
- повышении разрядности обрабатываемых вещественных чисел (512 бит и более);
- вычислениях в логарифмической системе счисления (ЛСС).

Первые два подхода являются традиционными. Также возможно выполнение арифметических операций с дробями без округления результата (пакеты Wolfram Mathematica, MatLab и др.) [4]. Их реализация в современных процессорах сопровождается увеличением временных затрат из-за их алгоритмической сложности и отсутствия аппаратной поддержки в процессорах, что является существенным недостатком.

Третий вариант является перспективной альтернативой формату IEEE-754, так как в ЛСС нет выравниваний порядков. Вещественное число  $a$  представлено в ней двоичным логарифмом  $L(a)$  его абсолютной величины и знаком  $S$ .

Преобразование числа из формата IEEE-754 в ЛСС происходит следующим образом (рис.1):

- знак  $S$  переписывается без изменений;
- характеристика числа  $E$  становится целой частью числа в ЛСС, т.к.  $\log_2(2^E) = E$ ;
- вычисляется логарифм мантиисы  $M$ , который является дробной частью числа в ЛСС.

Обратное преобразование выполняется аналогично, но на последнем этапе вычисляется антилогарифм дробной части логарифмического числа.



Рис. 1. Преобразование числа двойной точности формата IEEE-754 в ЛСС

Важной особенностью ЛСС является упрощение выполнения следующих операций:

- умножение заменяется сложением;
- деление заменяется вычитанием;
- возведение в степень заменяется умножением;
- извлечение корня заменяется делением.

Данное преимущество, продиктованное свойствами логарифмов, позволит повысить быстродействие предложенного сопроцессора на классе задач, где преобладают упомянутые операции. Однако за это приходится расплачиваться более сложным выполнением операций сложение и вычитание. Например, сложение можно определить следующей формулой:

$$\log_2(a + b) = \log_2 a + \log_2(1 + 2^{\log_2 b - \log_2 a}), \quad (1)$$

где  $a$  — наименьший операнд,  $b$  — наибольший операнд. Из формулы видно, что сложение состоит из последовательности следующих преобразований:

- вычисление антилогарифма разности чисел в ЛСС;
- прибавление числа 1 к данному антилогарифму;
- вычисление логарифма от найденного числа на предыдущем шаге;

– суммирование логарифма с наименьшим из операндов.

Поскольку точка, отделяющая целую часть числа в ЛСС от дробной, является фиксированной, то выравнивание порядков не требуется, что является ключевым преимуществом по сравнению с плавающей точкой.

Основополагающий вклад в изучение ЛСС внесли ученые Дж. Коулман (J. Coleman), М. Арнолд (M. Arnold), Е. Честер (E. Chester), Д. Льюис (D. Lewis) [3, 4]. Однако, несмотря на достаточно обширный объем проведенных исследований, законченных технических реализаций на сегодняшний день несколько. Одной из них является логарифмический процессор European Logarithmic Microprocessor (ELM) [5], который имеет 32-разрядную RISC архитектуру. Однако остается малоизученным вопрос использования разрядности данных в ЛСС свыше 32 бит. Это связано с экспоненциально возрастающими аппаратными затратами на преобразование кодов из формата IEEE-754 в ЛСС и обратно, а также реализацию операции суммирования, что является недостатком данной системы счисления.

Описанные в п. 2.1 технические решения, основанные на проведенном автором исследовании, позволяют сгладить данный недостаток за счет использования более высокого порядка интерполяции, применяемого при преобразовании кодов, в чем заключается отличие от рассмотренного логарифмического процессора ELM. Кроме того, отличием является дополнительный уровень кодирования данных в системе остаточных классов. Ее особенности приведены далее.

## 1.2. Способы повышения скорости вычислений

На сегодняшний день наиболее распространены следующие пути повышения быстродействия:

- увеличение числа вычислительных ядер в составе супер-ЭВМ, а также повышение частоты процессора (например, за счет уменьшения размеров транзистора при создании СБИС);
- увеличение количества команд, выполняемых за такт, например, с помощью широкого командного слова в процессорах Эльбрус АО «МЦСТ» (до 26 операций за такт) или векторных операций (до 8 команд двойной точности за такт в сопроцессоре Intel AVX);
- совершенствование алгоритмов программ, например, за счет распараллеливания вычислений.

При этом перспективным резервом повышения производительности ядра процессора является распараллеливание выполнения самих арифметических операций на уровне разрядов системы счисления. В позиционной системе счисления (ПСС), на базе которой функционирует подавляющее большинство известных процессоров, этому препятствует последовательное распространение единиц переноса. Традиционное решение кроется в использовании вычислительного конвейера, однако, его аппаратная сложность значительно возрастает по мере увеличения числа ступеней, а положительный эффект достигим лишь при непрерывном потоке исходных данных.

Перспективной альтернативой является модулярная система счисления, или, по-другому, система остаточных классов (СОК), где вычисления над каждой цифрой числа из базиса оснований  $\{p_1, p_2, \dots, p_n\}$  осуществляются независимо друг от друга, где  $n$  — количество остатков.

Пусть СОК состоит из двух оснований  $\{p_1=5, p_2=7\}$ . Диапазон чисел ограничен произведением модулей. Преобразуем числа — найдем остатки от деления на  $p_1$  и  $p_2$ , например, кортеж  $\{2,3\}$  является уникальным представлением числа 17. Остатки, обладая существенно меньшей разрядностью, могут суммироваться, либо умножаться параллельно:

$$\begin{aligned}
 & \begin{cases} 17_{10} = \{2_{\text{mod}5}, 3_{\text{mod}7}\} \\ 10_{10} = \{0_{\text{mod}5}, 3_{\text{mod}7}\} \end{cases} \\
 + & \\
 & 27_{10} = \{2_{\text{mod}5}, 6_{\text{mod}7}\}
 \end{aligned} \tag{2}$$

Предложенные технические решения на базе СОК используются в основном для цифровой обработки сигналов, так как применение плавающей точки затруднено ввиду сложной реализации деления на степени числа 2. Научные работы Н.И. Червякова, И.Я. Акушского, А. Сасаки (A. Sasaki), Г. Гарнер (H. Garner), А. Омонди (A. Omondi) [5, 6] посвящены данному направлению.

Известный модулярный процессор общего назначения, функционирующий на базе искусственных нейросетей, был разработан в Ставрополе в 2005 г. [7]. Однако недостаточное внимание в научных работах уделено возможности совместного использования преимуществ СОК и ЛСС, в чем состоит ключевое отличие предложенного сопроцессора.

### 1.3. Способы повышения надежности вычислений

Одновременно с тенденцией экспоненциального роста производительности супер-ЭВМ, все большую важность приобретает безошибочность вычислений, так как даже одиночная ошибка может привести к неверному результату многодневных расчетов. Вероятность возникновения таких ошибок увеличивается пропорционально числу вычислительных ядер. Их появление при сотнях тысяч ядер становится достаточно частым явлением.

Кроме того, существует множество задач, где помимо скорости вычислений, необходима отказоустойчивость аппаратуры. Среди них системы, работающие в реальном времени. Все их объединяет то, что процесс счета не должен прерываться ни при каких обстоятельствах. На сегодняшний день это достигается с помощью следующих средств:

- резервирование на уровне компонент системы;
- дублирование или троирование вычислений;
- программная обработка исключительных ситуаций и т.п.

Недостаток перечисленных подходов состоит в том, что часть оборудования простаивает, либо дублирует работу, вплоть до момента отказа одного из компонентов системы. Кроме того, применяются корректирующие коды, например, контроль четности регистров процессора, но они могут обнаружить лишь ошибку, возникшую при чтении/записи памяти и не приспособлены для выявления ошибок, появившихся в арифметическом устройстве.

Иначе обстоит дело при использовании системы остаточных классов, которая, помимо потенциала естественного параллелизма вычислений, обладает средствами повышения надежности вычислений на уровне системы счисления. При этом используется единый помехоустойчивый код для обнаружения ошибок, возникающих при обработке информации в арифметическом устройстве и передаче ее по каналам связи.

В этом случае полный диапазон представления чисел  $P$  делится на рабочий диапазон  $P_n$  из  $n$  модулей и контрольный  $P_k$  из  $k$  модулей. Данные названия условны, так как модули совершенно равноправны. При выполнении любой арифметической операции над двумя числами из рабочего диапазона результат также принадлежит этому диапазону  $P_n$ , в противном случае произошла ошибка. Подробнее алгоритм описан в [5, 7] отечественными учеными И.Я. Акушским, И.А. Калмыковым, Н.И. Червяковым, которые внесли значительный вклад в развитие данного направления.

## 2. Описание базовой технологии

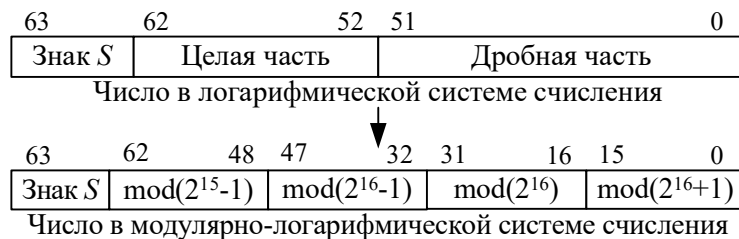
В данном пункте приведено описание предложенной модулярно-логарифмической системы счисления и технических решений сопроцессора, реализующих арифметические операции на ее базе.

Фиксированная точка в ЛСС позволяет работать с вещественным операндом, как с целым числом, что позволяет ввести следующий уровень преобразования — в СОК, образуя при этом новую модулярно-логарифмическую систему счисления (МЛСС), объединяющую преимущества обеих систем. По количеству разрядов формат числа в МЛСС может совпадать с форматом IEEE-754, обеспечивая эквивалентный диапазон представления вещественных чисел, например, двойную точность при использовании 64 разрядов.

Предлагаемый набор  $P = \{p_1 = 2^{15} - 1, p_2 = 2^{16} - 1, p_3 = 2^{16}, p_4 = 2^{16} + 1\}$  обеспечивает минимальное снижение (порядка 0,003%) диапазона представления 63 разрядного числа (64-й отведен под знак). Выбранные основания выполняют условие взаимной простоты и позволяют упростить преобразование кодов из ПСС в СОК и обратно, благодаря модулям вида  $2^n \pm 1$ , где  $n$  — разрядность модуля [7].

Прямое преобразование из IEEE-754 в МЛСС происходит в два этапа (рис. 2):

- характеристика исходного числа с плавающей точкой становится целой частью логарифма, одновременно с этим вычисляется логарифм по основанию два от мантиссы исходного числа, разряд знака не изменяется; точка, отделяющая целую часть вычисленного логарифма от дробной, отбрасывается;
- вычисляются остатки от деления полученного целого числа на основания СОК.



**Рис. 2.** Преобразование числа двойной точности из ЛСС в МЛСС

Преобразование из МЛСС в формат IEEE-754 проходит в обратной последовательности:

- по известным остаткам вычисляется позиционное представление логарифма; точка, отделяющая целую часть от дробной, восстанавливается на прежней позиции;
- вычисляется антилогарифм дробной части (мантисса), целая часть становится порядком искомого числа с плавающей точкой, разряд знака не изменяется.

Следует учитывать, что при одинаковом числе двоичных разрядов ПСС и СОК, диапазон представления чисел  $P$  последней будет меньше, так как в общем случае  $n$ -

разрядный модуль числа меньше чем  $2^n$  чисел, кодируемых тем же количеством разрядов ПСС. В связи с этим выбор набора оснований СОК является первостепенной задачей.

Перевод чисел из IEEE-754 в МЛСС и обратно осуществляется в предлагаемом сопроцессоре с помощью преобразователя кодов, рассмотренного далее.

## 2.1. Преобразователь кодов

Первый этап кодирования числа в МЛСС состоит в вычислении логарифма мантиссы, которая принимает значения на интервале от 1 до 2. В известных технических решениях [3, 4] использована линейная интерполяция с дополнительной коррекцией ошибки, что позволило обойтись относительно малыми аппаратными затратами, обеспечивая погрешность преобразования порядка  $E=10^{-7}$ , достаточную при вычислениях с одинарной точностью, но совершенно не достаточную для преобразования чисел двойной точности.

Уникальность предложенного решения состоит в применении квадратичной интерполяции с линейной интерполяцией ошибки каждого подинтервала, что позволяет обеспечить погрешность преобразования порядка  $E=10^{-16}$ , что достаточно при вычислениях с двойной точностью. Таким образом, логарифм  $y$  мантиссы  $x$  вычисляется по следующей формуле:

$$y = a \cdot x^2 + b \cdot x + c + (al \cdot xl + bl) \cdot maxerr; \quad (3)$$

где  $a, b, c$  — константы интерполяции интервала, которому принадлежит  $x$ ;

$xl$  — младшие 32 разряда мантиссы, определяющие подинтервал;

$al, bl, maxerr$  — константы интерполяции ошибки подинтервала.

Техническое решение, реализующее рассмотренное преобразование в аппаратуре, обладает параллелизмом конвейерного типа и состоит из сумматоров и умножителей. Его применение позволит уменьшить ошибку интерполяции в миллиард раз (с  $10^{-7}$  до  $10^{-16}$ ) по сравнению с известным решением, оперирующем одинарной точностью при повышении аппаратных затрат в 8 раз.

Второй этап кодирования числа в МЛСС состоит в вычислении остатков от деления логарифмического представления числа на основания СОК. Благодаря специально выбранным основаниям, находящимся вблизи числа  $2^n$ , вычисление остатков происходит с помощью сумматоров, разрядность которых не превышает 16 бит, что значительно сократит аппаратные затраты данного блока по сравнению с известными решениями [5].

Помимо основной функции преобразования кодов из формата IEEE-754 в МЛСС и обратно, аппаратный преобразователь кодов выполняет следующие операции:

- вычисление позиционной характеристики модулярного числа, что необходимо, например, при сравнении чисел, обнаружении переполнения разрядной сетки и машинного нуля;
- модулярно-логарифмическое сложение и вычитание вещественных чисел.

Остальные арифметические операции производятся в вычислительном ядре.

## 2.2. Вычислительное ядро

Каждое вычислительное ядро сопроцессора ведет независимую обработку по определенному модулю СОК. При этом операции умножение, деление и возведение в степень выполняются в векторном или скалярном виде.

В общем случае длина вектора ограничена лишь аппаратными затратами и скоростью выборки данных. Например, в рассмотренном варианте сопроцессора вектор имеет разрядность 128 бит, что составляет 8 упакованных в строку остатков, принадлежащих разным числам в МЛСС. Данный выбор обусловлен соотношением с векторами, обрабатываемыми расширением Intel AVX2, в котором за один такт работы выполняется операция сразу над 8 числами.

Для обработки векторов в вычислительном ядре предусмотрено восемь 16-разрядных сумматоров и столько же умножителей. Устройство работает следующим образом. Каждый такт из кэш-памяти на обработку поступают два исходных вектора. В зависимости от команды над остаточными представлениями логарифмов выполняется одна из следующих операций:

- сложение (соответствует умножению исходных вещественных чисел);
- вычитание (соответствует делению)
- умножение (соответствует возведению в степень).

Результирующий вектор записывается в кэш-память. Причем, для устранения задержек, связанных с извлечением и записью данных, целесообразно применить трехпортовую кэш-память, расположенную на кристалле. За один такт такая память может выполнить чтение двух строк и запись одной строки.

Таким образом, в отличие от арифметико-логических устройств известных универсальных процессоров, малая разрядность обрабатываемых остатков и регулярность структуры вычислительного ядра позволяет распараллелить вычисления на макро-уровне операндов, составляющих вектор, и микро-уровне независимого счета остатков каждого операнда. При этом арифметические операции в ядре выполняются за один такт без использования конвейеризации.

### 2.3. Реконфигурируемая архитектура сопроцессора

В предложенной СОК количество оснований равно четырем, соответственно, сопроцессор состоит из четырех независимых ядер, которые подключены к преобразователю кодов и оперативной памяти (рис. 3). Кэш-память каждого ядра хранит информацию не обо всем числе, а лишь об остатке, по которому ведется обработка. Это позволяет организовать независимые вычисления в каждом ядре, а также некогерентную кэш-память.

Ключевой особенностью сопроцессора является динамическая реконфигурация, позволяющая управлять разрядностью вычислений, что бывает актуально для задач, не требующих высокой точности, но обладающих высокой степенью векторного параллелизма, например, к ним относится глубокое обучение нейросетей [2].

Возможны следующие варианты реконфигурации сопроцессора:

- 32 векторные операции половинной точности;
- 16 векторных операций одинарной точности;
- 8 векторных операций двойной точности.

В случае, когда на одном кристалле размещено  $n$  сопроцессоров, максимальная разрядность обрабатываемых целых чисел равна  $64 \cdot n$ . Например, при использовании восьми сопроцессоров в одной системе, диапазон обработки целых чисел достигает 512 бит с сохранением параллельной обработки по каждому остатку.



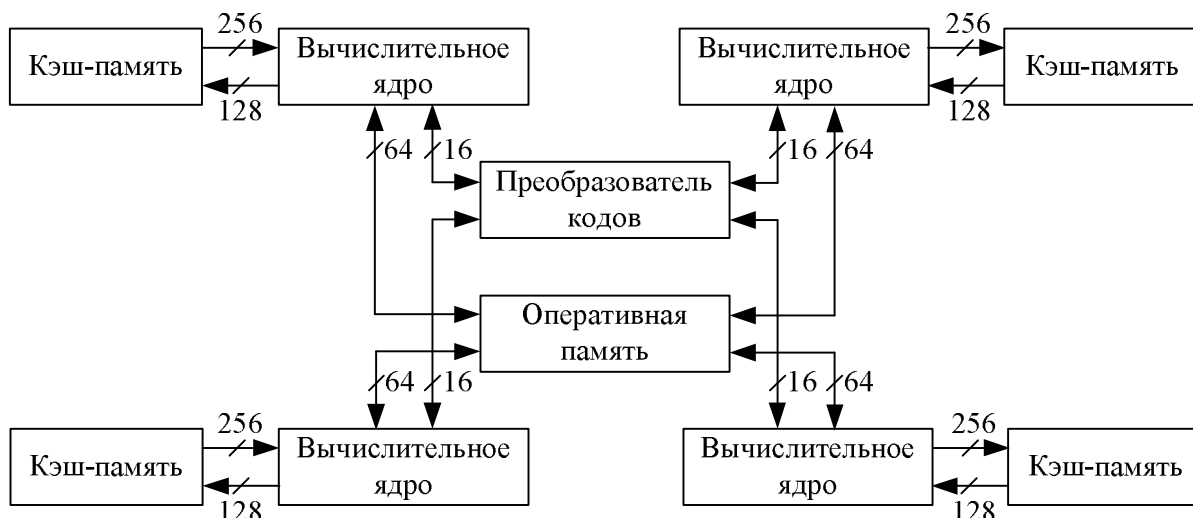


Рис. 3. Преобразование числа двойной точности формата IEEE-754 в ЛСС

Помимо изменения разрядности вычислений, реконфигурация позволяет отключить одно или несколько ядер в случае их отказа. Способ обнаружения в сопроцессоре одиночных и двойных ошибок, а также ситуаций отказа ядра базируется на свойствах СОК рассмотренных в п. 1.3. Для этого в сопроцессор необходимо добавить контрольное ядро. Диапазон представления чисел при этом делится на рабочий интервал и запрещенный интервал в отличие от универсальных процессоров, где требуется дублирование/троирование аппаратуры или процесса вычислений.

Доказано [5, 7], что при возникновении ошибки результат окажется в запрещенном интервале, причем предусмотрен механизм, локализирующий ядро, где произошел сбой. Если ошибки появляются постоянно в одном и том же ядре, возникает ситуация отказа и ядро блокируется. Сопроцессор при этом продолжает работу, сокращается лишь диапазон представления чисел. Если при этом нельзя снижать точность вычислений, то можно использовать вычислительные ядра другого сопроцессора, в том числе и контрольные. Такая реконфигурация позволяет создать процессор, устойчивый к постепенной деградации.

### 3. Описание базовой технологии

На данный момент модулярно-логарифмический процессор существует в виде прототипа — IP-блока на базе программируемой логической интегральной схемы (ПЛИС). Данный выбор обусловлен простотой реализации, так как можно использовать готовые функциональные блоки софт-процессора Altera NIOS, дополнив основную систему команд инструкциями сопроцессора и использовав 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.

Аппаратные затраты на создание предлагаемого IP-блока составляют около 2 млн. транзисторов, тогда как для реализации блока векторных расширений AVX2 процессора Xeon семейства Haswell требуется около 21 млн. транзисторов [1], что на порядок выше. Далее приведено их сравнение по таким параметрам, как быстродействие, точность и надежность вычислений.

### 3.1. Быстродействие

Сравнение быстродействия модулярно-логарифмического сопроцессора (МЛС) с сопроцессором Intel AVX2 при предполагаемой равной частоте работы устройств (2,6 ГГц) приведено на рис. 4, где арифметические операции над числами двойной точности, упакованными в вектора размерностью 8 элементов, обозначены следующим образом:

- SUM — суммирование соответствующих пар операндов векторов;
- HSUM — редукционное (мультиоперандное) суммирование элементов вектора;
- MUL/DIV — умножение/деление пар операндов векторов соответственно;
- POW — возведение операндов первого вектора в степень, хранимую во втором векторе;
- $\log_2 x$  — логарифм по основанию 2 от каждого элемента вектора;
- matmul — перемножение плотных матриц размерностью  $8 \times 8$  элементов (состоит из 64 операций умножения MUL и 64 редукционных суммирований HSUM);
- polinom — вычисление многочлена восьмой степени (состоит из одной операции POW, одного умножения MUL и одного редукционного суммирования HSUM).

Время выполнения операции на универсальном процессоре замерялось с помощью функции `clock()` библиотеки `time.h`. Для гарантированного исполнения векторных инструкций Intel применялись функции из набора арифметических операций библиотеки `immintrin.h`.

Наибольшее ускорение в предложенном сопроцессоре достигается при использовании операций, использующих свойства логарифмов. Например, для операций возведение в степень разница составляет три порядка в пользу сопроцессора, что объясняется отсутствием аппаратной реализации этих операций в сопроцессорах Intel.

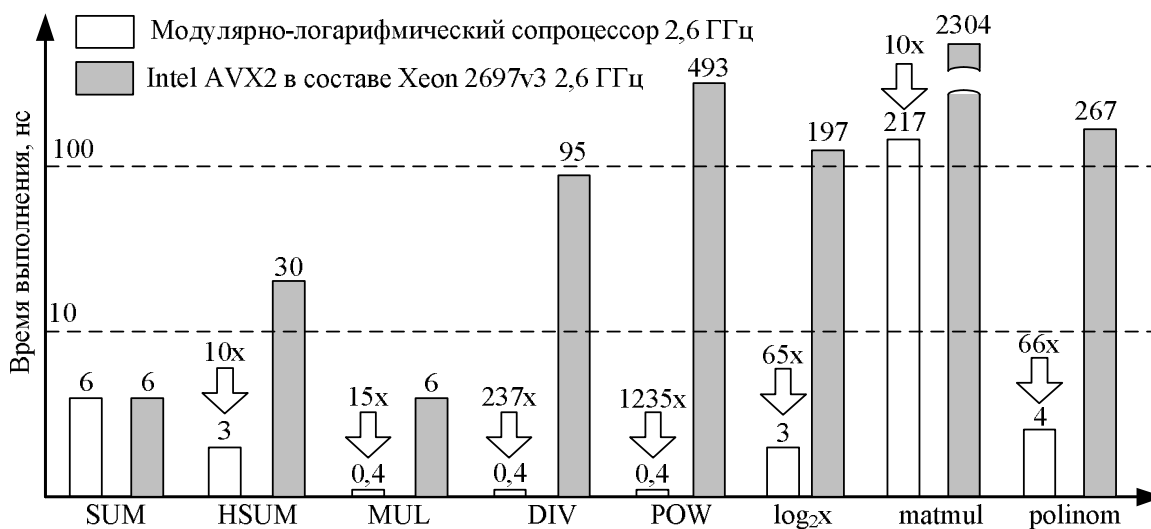


Рис. 4. Преобразование числа двойной точности формата IEEE-754 в ЛСС

Однотактовое выполнение векторных операций умножение, деление и возведение в степень позволяет получить значительное преимущество по скорости выполнения определенного класса задач, где таких операций большинство. К ним относится вычисление многочленов  $n$ -ой степени, например, разложение в ряд Тейлора, перемножение матриц и ряд других.

В данном тестировании принято, что преобразование в МЛСС и обратно аппаратно конвейеризировано и не вносит значительных временных задержек в вычислительный процесс.

### 3.2. Точность вычислений

Одним из вариантов тестирования точности вычислений является решение задачи Коши методом Эйлера. Рассмотрим дифференциальное уравнение  $x'(t)=t$ , где  $x_0=0$ ,  $t_0=0$ . Аналитическое решение этого уравнения известно, а значит можно вычислить относительную погрешность решения, полученного численным путем в арифметике с плавающей точкой. Шаг интегрирования равен  $1/2^i$ , где  $i$  — натуральное число.

Результаты расчетов с двойной точностью в МЛС и формате IEEE-754 на рис. 5, где показана зависимость относительной погрешности решения задачи Коши от шага интегрирования.

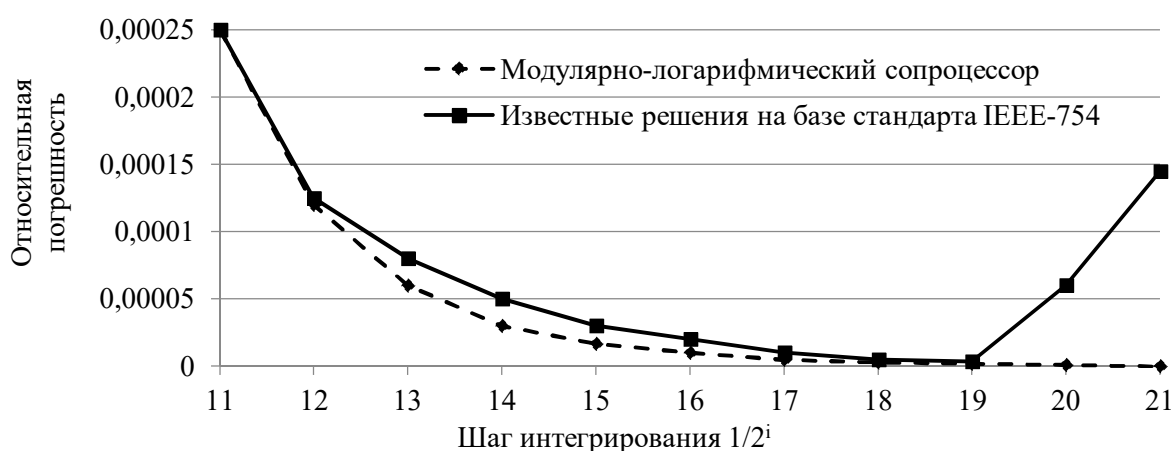


Рис. 5. Результаты тестирования точности вычислений

Из рис. 5 видно, что при уменьшении шага интегрирования относительная погрешность вычислений по стандарту IEEE-754 уменьшается, однако, после значения 19 резко возрастает, что обусловлено ошибками округления. Подобный эффект отсутствует при расчетах с помощью МЛС, что позволяет сделать вывод о более высокой точности вычислений в данном случае.

### 3.3. Оценка надежности

При построении надежностных характеристик предположим, что:

- в начальный момент времени в сопроцессоре имеется  $n=k+r$  ядер, где  $k=3$  — количество информационных ядер,  $r=1$  — количество контрольных ядер;
- минимальное количество каналов для функционирования сопроцессора  $n-(r+1)=2$ ;
- отказавшие каналы не восстанавливают работоспособность;
- отказы каналов являются статистически независимыми событиями.

Таким образом, надежностная структура МЛС соответствует модели скользящего резервирования, где резервные элементы находятся в нагруженном состоянии [3]. Вероятность его безотказной работы вычисляется следующим образом:

$$R_{MLC} = \sum_0^{3k+4r} P^{4(k+r)-i} \cdot (1-P)^i, \quad (4)$$

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

В традиционных системах высокой надежности определение и оперативная коррекция ошибок возможна лишь при условии одновременной работы нескольких устройств по принципу голосования, например, «2 из 3». Вероятность безотказной работы такой системы:

$$R_{\text{ПСС}} = 3 \cdot P_{\text{ПСС}}^2 - 2 \cdot P_{\text{ПСС}}^3, \quad (5)$$

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

Результаты расчета надежности обоих вариантов при интенсивности отказов  $\lambda = 10^{-5} \text{ ч}^{-1}$  представлены графически на рис. 6.

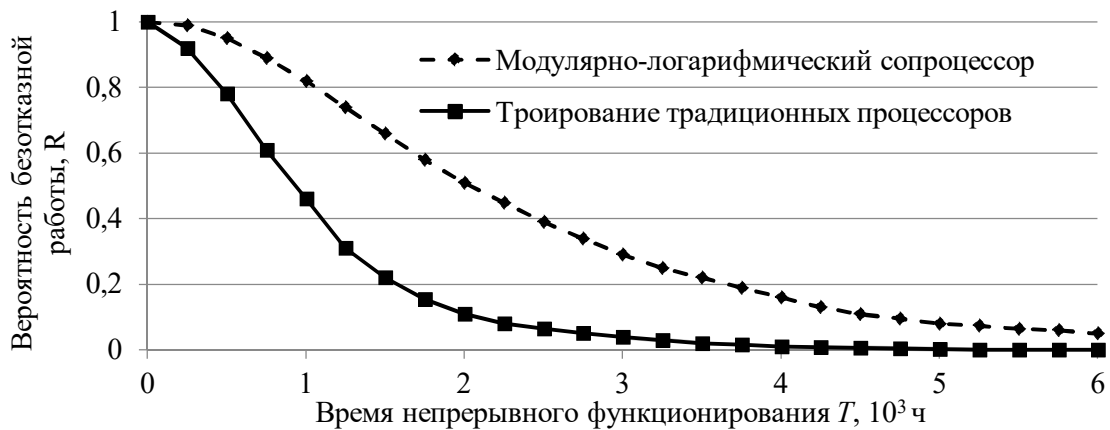


Рис. 6. Результаты тестирования точности вычислений

Анализ полученных зависимостей свидетельствует о преимуществе в надежности МЛС. Например, в течение  $t=2000$  часов функционирования он обеспечивает вероятность безотказной работы  $R=0,5$ ; тогда как позиционный процессор  $R=0,1$ . Избыточность аппаратных затрат составляет 25% и 66% соответственно.

## Заключение

Анализ приведенных в данном отчете способов повышения скорости, точности и надежности вычислений выявил затруднения на пути дальнейшего улучшения характеристик без внесения изменений на фундаментальном уровне самих принципов вычислений.

Перспективными системами счисления, призванными внести упомянутые изменения, являются модулярная система счисления как замена позиционной системы счисления и логарифмическая система счисления в качестве замены плавающей точки по стандарту IEEE-754. Таким образом, актуальной является разработка вычислительного устройства, функционирующего в базе совмещенной модулярно-логарифмической системы счисления.

В результате проведенных исследований и разработок предложены новые научные и технические решения, реализующие предложенные способы вычислений и кодирования данных, что позволило добиться следующих конкурентных преимуществ.

Высокое быстродействие по сравнению с известными аналогами достигается за счет следующих факторов:

- отсутствие переносов между модулями одного числа;
- замена умножение/деление на сложение/вычитание;
- замена возведение в степень/извлечение корня на умножение/деление.

Например, вычисление произвольного полинома восьмой степени в прототипе сопроцессора на ПЛИС выполнено втрое быстрее (90 нс против 267 нс) по сравнению с серверным процессором Intel Xeon 2697 v3 при разнице тактовой частоты в 26 раз в пользу Intel (0,1 ГГц против 2,6 ГГц).

Более высокая точность вычислений по сравнению с использованием плавающей точки достигается благодаря следующим факторам:

- отсутствие округления результата при выполнении операций умножение, деление, возведение в степень;
- отсутствие выравнивания порядков, в результате чего происходит потеря значащих разрядов.

Это позволяет верно решать ряд задач, критичных к ошибкам округления в перечисленных операциях и содержащих разномасштабные данные, в отличие от универсальных процессоров.

В отличие от известных вычислительных устройств, высокая надежность обеспечена на аппаратном уровне за счет следующих механизмов:

- выявление одиночных и двойных ошибок в вычислительном устройстве;
- реконфигурация ядер сопроцессора при отказе одного или нескольких из них с продолжением процесса счета.

Стоит отметить, что преобразование кодов в МЛСС и обратно не вносит значительных временных задержек при большом потоке входных данных за счет предложенных аппаратных решений, конвейеризирующих процесс интерполяции функции логарифма и преобразования кодов системы остаточных классов.

Одним из перспективных направлений внедрения разработки являются отечественные процессоры, например, Эльбрус-8С, Байкал-М, КОМДИВ-64. Применение сопроцессора в этих системах-на-кристалле позволило бы дополнить их конкурентные преимущества рассмотренными, что, в конечном счете, сыграло бы важную роль в импортозамещении вычислительных компонент.

## Литература

1. Top 10 Sites for November 2016 URL: [www.top500.org/lists/2016/11](http://www.top500.org/lists/2016/11) (дата обращения: 26.01.2017).
2. 754-2008 — IEEE Standard for floating-Point Arithmetic. Revision of ANSI/IEEE Std 754-1985 URL: [ieeexplore.ieee.org/document/4610935](http://ieeexplore.ieee.org/document/4610935), 2008 (дата обращения: 26.01.2017).
3. Осинин И.П. Высоконадежный модулярно-логарифмический процессор с реконфигурируемой архитектурой // Параллельные вычислительные технологии (ПаВТ'2016): труды международной научной конференции (Архангельск, 28 марта – 1 апреля 2016 г.). Челябинск: Изд-во ЮУрГУ, 2016. С. 642–654.
4. Осинин И.П., Князьков В.С. Направления развития архитектуры реконфигурируемых вычислительных платформ. Математическое моделирование развивающейся экономики, экологии и технологий (ЭКОМОД-2016): Труды IX Всероссийской

- научной конференции (Киров, 4–9 июля 2016 г.). Киров: Изд-во ВятГУ, 2016. С. 486–495.
5. Coleman J.N., Chester E.I. Arithmetic on the European Logarithmic Microprocessor // IEEE Transactions on Computers. 2000. Vol. 49, No. 7. P. 702–715. DOI: 10.1109/12.863040.
  6. Ismail R.C., Hussin R., Muzad S.A. Interpolator Algorithms for approximating the LNS addition and subtraction // IEEE International Conference on Circuits and Systems (ICCAS). Kuala Lumpur, 3 – 4 Oct. 2012. P. 174–179. DOI: 10.1109/iccircuitsandsystems.2012.6408336.
  7. Червяков Н.И. Модулярные параллельные вычислительные структуры нейропроцессорных систем. Москва: Изд-во Физматлит, 2003. 288 с.
  8. Omondi A. Residue Number System: Theory and Implementation. London: Imperial College Press, 2007. 312 p.
  9. Калмыков И.А. Теоретические основы вычислений в полиномиальной системе классов вычетов, ориентированных на построение отказоустойчивых систем. Ставрополь: Изд-во СКФУ, 2006. 347 с.

Осинин Илья Петрович, к.т.н., н.с. ООО «Саровская лаборатория имитационного моделирования» (Саров, Российская Федерация)

DOI: 10.14529/cmse170202

## MODULAR-LOGARITHMIC COPROCESSOR FOR MASSIVE ARITHMETIC CALCULATIONS

© 2017 I.P. Osinin

*Sarov Laboratory of Simulation Modeling (Mayakovskogo 42, Sarov, 607190 Russia)*

*E-mail: iposinin@mail.ru*

Received: 01.05.2017

The paper presents a conceptual design of an IP module of mathematical coprocessor. It consists of a set of processing cores of the same kind which perform single-cycle scalar, or vector operations with real numbers. The processed data is represented in the modular logarithmic format that provides two levels of translating the original numbers, namely: the modular level instead of the conventional positional system and the logarithmic level instead of the floating point format. As a result of the research and development, new scientific and technical solutions are proposed that implement the proposed methods of computing and coding data. Owing to this feature a coprocessor has a higher performance, a higher accuracy and a higher level of reliability, as compared to the known analogs. Convert codes in modular-logarithmic number system and vice versa does not introduce significant time delays in a large stream of input data by offering hardware solutions pipelined process of interpolation of the logarithm function and conversion of residual classes system codes. A prototype coprocessor is an FPGA-based IP module. Companies developing general-purpose processors are the target market for this design.

*Keywords: residue number system, logarithmic number system, reconfigurable architecture, highly reliable computing.*

### FOR CITATION

Osinin I.P. Modular-logarithmic Co-processor for Massive Arithmetic Calculations. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 2. pp. 22–36. (in Russian) DOI: 10.14529/cmse170202.

## References

1. *Top 10 Sites for November 2016*. Available at: [www.top500.org/lists/2016/11](http://www.top500.org/lists/2016/11) (accessed: 26.01.2017).
2. *754-2008 — IEEE Standard for floating-Point Arithmetic*. Revision of ANSI/IEEE Std 754-1985 URL: [ieeexplore.ieee.org/document/4610935](http://ieeexplore.ieee.org/document/4610935), 2008 (accessed: 26.01.2017).
3. Osinin I.P. The Highly-modular Logarithmic Processor with a Reconfigurable Architecture. *Parallelnye vychislitelnye tekhnologii (PaVT'2016): trudy mezhdunarodnoj nauchnoj konferentsii (Arkhangel'sk, 28 marta – 1 aprelya 2016)*. [Parallel Computational Technologies (PCT'2016): Proceedings of the International Scientific Conference (Arkhangelsk, Russia, March, 28 – April, 1, 2016)]. Chelyabinsk, Publishing of the South Ural State University, 2016. pp. 642–654. (in Russian)
4. Osinin I.P., Knyaz'kov V.S. Directions of Development of Architecture of Reconfigurable Computing Platforms. *Matematicheskoe modelirovanie razvivayushcheyssya ekonomiki, ekologii i tekhnologii (EKOMOD-2016): Trudy IX Vserossiyskoy nauchnoy konferentsii (Kirov, 4–9 iyulya 2016)*. [Mathematical modeling of a developing economy, ecology and technology (ECOMOD-2016): Proceedings of the Scientific Conference (Kirov, Russia, July 4–9, 2016)]. Kirov: Publishing of the Vyatka State University, 2016. pp. 486–495. (in Russian)
5. Coleman J.N., Chester E.I. Arithmetic on the European Logarithmic Microprocessor // *IEEE Transactions on Computers*. 2000. vol. 49, no. 7. pp. 702–715. DOI: 10.1109/12.863040
6. 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. pp. 174–179. DOI: 10.1109/iccircuitsandsystems.2012.6408336
7. Chervyakov N.I. *Modulyarnye parallel'nye vychislitel'nye struktury neyroprotses-sornykh sistem* [Modular structure of the parallel computing systems neuroprocessor]. Moscow, Publishing Fizmatlit, 2003. 288 p.
8. Omondi A. *Residue Number System: Theory and Implementation*. London: Imperial College Press, 2007. 312 p.
9. Kalmykov I.A. *Teoreticheskie osnovy vychisleniy v polinomial'noy sisteme klassov vychetov, orientirovannykh na postroenie otkazoustoychivykh system*. [Theoretical bases of calculations in polynomial system of residue classes, focused on building a fault-tolerant systems]. Stavropol, Publishing North-Caucasian Federal University, 2006. 347 p.