

УМНОЖИТЕЛЬ С НАКОПЛЕНИЕМ: МЕТОДОЛОГИЧЕСКИЕ АСПЕКТЫ*

*И. А. Соколов¹, Ю. А. Степченко², С. Г. Бобков³, Ю. В. Рождественский⁴,
Ю. Г. Дьяченко⁵*

Аннотация: Представлены подходы к проектированию самосинхронной (СС) аппаратуры и рассмотрены условия внутрисистемной интеграции синхронных и СС-устройств в рамках супер-ЭВМ на примере разработки СС-устройства умножения-сложения, соответствующего стандарту IEEE 754 и выполняющего либо одну операцию двойной точности, либо одновременно две операции одинарной точности над тремя операндами. Устройство реализовано по технологии КМОП (комплементарная структура металл-оксид-полупроводник) с проектными нормами 65 нм с использованием в умножителе СС троичного кодирования. В зависимости от исполнения оно работает с асинхронным или синхронным окружением и обеспечивает производительность на уровне не менее 1 гигафлопса при времени задержки формирования результата относительно поступления входных операндов не более 6 нс.

Ключевые слова: самосинхронная схема; супер-ЭВМ; умножитель с накоплением; сумматор; конвейер; энергоэффективность

DOI: 10.14357/08696527140305

1 Введение

Современные супер-ЭВМ петафлопсного класса используют менее 35% вычислительной мощности для непосредственных вычислений [1]. Остальные ресурсы расходуются на обеспечение достоверности результатов вычислений. При этом их наработка на отказ составляет ~ 55 ч (данные на 2013 г. соответствуют супер-ЭВМ, содержащим 100 000 ядер вычислительных устройств). Для супер-ЭВМ эксафлопсного класса, где число ядер повышается почти на три порядка, а нормы проектирования сверхбольших интегральных схем (СБИС) сокращаются в 2–3 раза, показатели надежности вычислений и наработка на отказ могут оказаться существенно хуже. Решение этой проблемы возможно либо кардинальной заменой технологического базиса вычислителей, либо использова-

* Исследование выполнено при финансовой поддержке РФФИ (проект 13-07-12068 офи_м).

¹ Институт проблем информатики Российской академии наук, ISokolov@ipiran.ru

² Институт проблем информатики Российской академии наук, YStepchenkov@ipiran.ru

³ Научно-исследовательский институт системных исследований Российской академии наук, bobkov@cs.niisi.ras.ru

⁴ Институт проблем информатики Российской академии наук, YRogdest@ipiran.ru

⁵ Институт проблем информатики Российской академии наук, diaura@mail.ru

нием аппаратных методов контроля надежности и саморемонта вычислительных средств в темпе реальных вычислений. Создание и глубокая технологическая проработка нового базиса вычислительных систем (ВС) видится в весьма отдаленной перспективе, в то время как для второго варианта решения можно уже сейчас предложить СС-схемотехнику, достаточно хорошо проработанную и отвечающую выдвигаемым требованиям.

Данная работа посвящена расширению класса СС-блоков вычислительного ядра для супер-ЭВМ. Ранее был разработан и изготовлен квазисамосинхронный блок деления/извлечения корня [3, 2] 64/32-разрядной арифметики в технологии 180 и 65 нм. Здесь излагаются основные принципы проектирования СС-устройства умножения-сложения 64/32-разрядности гигафлопсного класса для супер-ЭВМ, учитывающие особенности функционирования СС-схем.

Несмотря на теоретически доказанные и практически подтвержденные преимущества СС-изделий (в том числе разработанных, изготовленных и испытанных авторами настоящей статьи), коммерческих СС-изделий немного [4], а сведения об их успешной эксплуатации в рамках крупных ВС отсутствуют. Представляется, что одной из причин является проблема встраивания СС-изделий в глобальное синхронное окружение. В ряде работ В. И. Варшавского была предложена методология построения крупных ВС в виде GALA-систем (Global Asynchronous / Local Arbitrary). Если бы ее приняли на вооружение конструкторы супер-ЭВМ, то проблема внедрения СС-изделий была бы решена. На уровне системного интерфейса супер-ЭВМ реализуется асинхронный интерфейс обмена взаимодействующих устройств, а на уровне локального интерфейса — произвольный. Таким образом, на уровне локального интерфейса проектировщик может использовать широкую гамму решений — от реализации полной самосинхронизации до использования локального генератора. Все зависит от цели проектируемого изделия.

Но именно при такой стратегической линии построения супер-ЭВМ обеспечивается возможность постепенного внедрения СС-изделий (по мере появления их на рынке) в общую структуру для достижения требуемых параметров (надежных показателей или энергоэффективности).

Операция «умножение двух операндов и сложение произведения с третьим операндом» (Fused Multiply-Add, FMA) с 1990-х гг. стала широко внедряться в микропроцессоры общего и специального назначения. В виде одной инструкции она реализована в микропроцессорах IBM POWER1 (1990 г.) и старше, Fujitsu SPARC64 (1995 г.), HP PA-8000 (1996 г.), Intel Itanium (2001 г.), IBM Cell (2005 г.), в процессорах AMD (2012 г.), в графических процессорах NVIDIA Tesla T10 (GPGPU) [5]. В 2008 г. она вошла в стандарт IEEE 754 [6]. Эта операция обычно выполняется быстрее, чем пара последовательных инструкций умножения и сложения, и точнее за счет отсутствия промежуточного округления произведения. В большинстве публикаций рассматриваются синхронные устройства FMA [7, 8], но в последние годы все больше публикаций посвящается асинхронным FMA, например [9, 10]. Анализ последних показал, что они опира-

ются на использование так называемой спекулятивной индикации (Bounded delay) и слабых транзисторов. Использование спекулятивной индикации — окончание переходных процессов индицируется только на некотором критическом пути — позволяет резко сократить объем оборудования на индикаторную часть, но приводит к потере самопроверяемости относительно константных неисправностей. Использование слабых транзисторов позволяет уменьшить общее число транзисторов, но приводит к существенному снижению уровня помехоустойчивости аппаратуры.

Более того, использование указанных приемов уменьшает полезную зону работоспособности СС-изделий, особенно в области предельно низких питающих напряжений. Именно это преимущество СС-схем позволяет существенно снизить динамическую и статическую составляющие энергопотребления для той части аппаратуры супер-ЭВМ, которая в настоящий момент либо не используется вообще (не задействована в текущей задаче), либо может быть использована с существенно меньшей производительностью, путем динамического снижения питающего напряжения [11].

Указанные виды СС-схем являются неприемлемыми для использования в составе супер-ЭВМ. В рамках сложившейся на сегодня классификации целям супер-ЭВМ отвечают только два класса СС-схем:

- (1) схем, правильная работа которых не зависит от задержек элементов и от задержек в проводах при условии, что разница задержек в проводах после разветвления меньше минимально возможной задержки базового элемента библиотеки. Такие схемы называются независимыми от задержек элементов (НЗЭ). В западной классификации им соответствуют *speed-independent* (SI), или *quasi-delay-insensitive* схемы;
- (2) схем, правильная работа которых не зависит ни от задержек элементов, ни от задержек в соединяющих их проводах. По сложившейся западной классификации такие схемы называются *delay-insensitive* (DI) схемами.

Ярким представителем DI-схем является NCL-схема (Null Convention Logic) [12]. Проведенный авторами сравнительный схемотехнический анализ НЗЭ- и NCL-схем [13] показал, что «НЗЭ схемы, разрабатываемые в соответствии с методологией, продвигаемой ИПИ РАН, обладают меньшими аппаратурными затратами (в 4,49 раза при реализации двоичного счетчика, в 1,13 раза при реализации умножителя 4×4 , до 2 раз при реализации более простых логических схем), большей производительностью и меньшим энергопотреблением по сравнению с NCL-схемами». Поэтому именно их целесообразно использовать в качестве схемотехнического базиса супер-ЭВМ эксафлопсного класса.

В отличие от синхронной, НЗЭ-схемотехника использует запрос-ответную дисциплину взаимодействия блоков, СС-кодирование информационных сигналов и развитую систему индикации развития вычислительного процесса. Поэтому появление любой константной неисправности в любой цепи такой схемы приводит к остановке вычислений на соответствующем индикаторе и немедленной локали-

зации проблемы. Это позволяет практически в темпе вычислений предоставить резервный аппаратный ресурс и продолжить безошибочную работу всего устройства (осуществляется саморемонт). В ряде случаев применение НЗЭ-схемотехники может позволить в несколько раз увеличить реальную производительность супер-ЭВМ эксафлопсного класса.

Независимые от задержек элементов схемы обеспечивают уменьшение энергетических расходов, связанных с изъятием из схемы генератора тактовых импульсов и «клокового дерева», которые определяют значительную долю динамической и статической составляющих рассеиваемой мощности — от 30% до 50% от общего энергопотребления синхронной схемы. Исполнение аппаратуры в НЗЭ-базисе обеспечивает автоматический перевод в энергосберегающий режим той части аппаратуры, которая не используется в текущем цикле обработки информации. Сохранение работоспособности НЗЭ-схем при сверхмалых значениях питающих напряжений открывает широкие перспективы для проектирования энергоэффективных изделий. Введение программно-аппаратных возможностей регулировки уровня питающих напряжений позволит существенно снизить энергопотребление микросхем в ряде режимов.

Платой за такие преимущества является аппаратная избыточность и дополнительные временные затраты на индикацию и дополнительную фазу «гашения» в работе СС-элементов. Грамотное проектирование этих схем позволяет существенно снизить эту избыточность, а в ряде случаев [14] и получить лучшие, чем в синхронных схемах, результаты.

Цель данной статьи — оценка подходов к разработке SI-устройства умножения-сложения (SIFMA) по стандарту IEEE 754 [6], обладающего расширенными возможностями и ориентированного на использование в супер-ЭВМ.

2 Особенности разрабатываемого SIFMA

Область применения данного SIFMA выдвигает в качестве основного требования минимальное энергопотребление при достаточно высокой производительности [1]. Предполагаемая производительность — 1–4 Гфлопс. Она определяется сравнительно невысокой тактовой частотой для машин эксафлопсного класса и большим числом узлов SIFMA на одну СБИС. Анализ минимаксной кривой, построенной в координатах энергопотребление–площадь в соответствии с методикой, приведенной в [15], применительно к технологии 65 нм и к производительности SIFMA, позволил определить основные характеристики прототипа, выбрать структурную схему SIFMA, и определить число ступеней в конвейерной реализации на 1 Гфлопс равным 8. Предлагаемый вариант SIFMA разрабатывался в рамках стандартной технологии 65 нм с ограничением латентности операций в 6 нс.

SIFMA предназначен для потоковой обработки троек операндов путем выполнения умножения двух операндов и сложения с третьим операндом или вычитания из третьего операнда.

Операнды двойной точности представляются как 64-разрядные числа, а одинарные — как 32-разрядные числа. Это позволяет использовать 64-разрядные операнды для передачи сразу двух 32-разрядных чисел. Тем самым создается предпосылка для опциональной обработки одним устройством не только операндов двойной точности, но и одновременной обработки двух одноптипных операций одинарной точности.

Разработанный SIFMA выполняет операции в соответствии со стандартом IEEE 754 со следующими уточнениями:

- на вход поступают нормализованные операнды;
- перемножаемые операнды не являются нулевыми;
- тройка операндов содержит либо три числа двойной точности, либо шесть чисел одинарной точности; в последнем случае одновременно выполняются две операции над тройками операндов одинарной точности.

Кроме того, SIFMA опционально выполняет сразу две операции: сложение произведения двух операндов с третьим операндом и/или вычитание произведения из третьего операнда. Рассматриваемое устройство SIFMA реализует следующую совокупность действий по обработке входных операндов [7]:

1. Вычисление экспоненты результата.
2. Умножение мантисс первых двух операндов.
3. Выравнивание мантиссы третьего операнда.
4. Инвертирование выравненной мантиссы третьего операнда в случае фактического вычитания произведения из третьего операнда или наоборот.
5. Сложение и/или вычитание мантисс произведения и третьего операнда.
6. Получение модуля результата при его отрицательном значении.
7. Нормализация модуля суммы и/или разности.
8. Округление нормализованного результата.
9. Постнормализация.

Главной особенностью описываемого SIFMA является его принадлежность к классу устройств, правильно работающих при любых задержках элементов, на которых они реализованы, — к классу устройств, не зависящих от задержек элементов [3, 2]. Время выполнения операции на нем определяется не тактовым сигналом, а условиями эксплуатации: напряжением питания, температурой окружающей среды — и характером обрабатываемых операндов.

Достигаются такие характеристики с помощью кодирования сигналов специальными СС-кодами и средств индикации завершения отдельных этапов обработки данных. В подавляющем большинстве НЗЭ-схем используется парафазный код и индикаторы на Г-триггерах.

Фундаментальной проблемой проектирования СС-схем для супер-ЭВМ является необходимость индикации завершения каждого этапа вычислений для

всех разрядов машинного слова перед началом последующего этапа обработки. Такие индикаторы требуют существенных аппаратных затрат и привносят дополнительную задержку в работу процессора ЭВМ вследствие большой длины машинного слова и избыточности СС-кодирования.

Существует целый ряд возможных решений этой проблемы:

- использование специальных многовходовых индикаторных элементов;
- сокращение числа индицируемых разрядов;
- максимально возможное сокращение числа вычислительных элементов, требующих аппаратной индикации завершения процесса вычислений.

Попытки создания эффективных многовходовых индикаторных элементов предпринимались многократно [3, 6, 7]. В ряде случаев удавалось получить некоторое сокращение временных затрат, как правило за счет снижения помехоустойчивости вследствие использования слаботочных активных элементов.

Сокращение числа индицируемых разрядов возможно либо за счет отказа от строгой самосинхронности и индикации завершения процесса по неполному набору разрядов машинного слова [1, 5], либо за счет перехода к двуполярному питанию, либо за счет использования n -значной арифметики (при $n > 2$) во внутреннем представлении чисел [16].

Первое решение существенно снижает затраты на индикацию, но в той же мере ликвидирует все преимущества СС-схем. Второй вариант уменьшает в 2 раза число индицируемых сигналов, но ведет к усложнению аппаратуры. Третий вариант приводит к усложненным алгоритмам непосредственной математической обработки одного разряда. Положительного эффекта в этом случае можно достичь лишь за счет существенного сокращения общей длительности вычислительного процесса с минимизацией числа разрядов, требующих индикации.

Применительно к SIFMA это требование означает минимизацию и, если это возможно, отказ от индикации любых промежуточных стадий процесса умножения двух чисел большой разрядности до его завершения.

2.1 Особенности реализации SIFMA большой разрядности

Наиболее эффективные алгоритмы умножения, применяемые практически во всех самых мощных и современных процессорах, базируются на алгоритмах кодирования Бута и сложения на базе дерева Уоллеса и их модификациях. При этом основные временные и аппаратные затраты приходятся на процедуру сложения частичных произведений, формируемых алгоритмом Бута, и реализуемую с помощью дерева Уоллеса. Борьба за повышение быстродействия привела к наращиванию конвейеризации в реализации алгоритма Уоллеса.

Временные затраты на операцию умножения в конвейерной реализации $t_{\text{умн.к}}$ можно ориентировочно оценить как

$$t_{\text{умн.к}} = t_{\text{конв.н}} + \frac{t_{\text{умн}}}{n},$$

где $t_{\text{конв.н}}$ — дополнительные временные затраты на организацию конвейерной обработки данных на каждом этапе конвейера; $t_{\text{умн}}$ — чистое время выполнения операции умножения без использования конвейеризации; n — число ступеней конвейера для реализации умножения.

В синхронных схемах, не требующих средств индикации завершения процессов вычислений на каждом этапе конвейера, удавалось реализовать условие

$$t_{\text{умн}} \gg t_{\text{конв.н}}.$$

Это позволяло использовать конвейеризацию в умножителе с $n = 5-7$ и выше. Однако при этом существенно росли аппаратные затраты и энергопотребление.

Переход в область субмикронных технологий (ограничение плотности тока на кристалле) и появление многоядерных процессоров внесли свои коррективы в этот процесс. В современных многоядерных процессорах стараются либо снизить до минимума число ступеней конвейера в блоке реализации дерева Уоллеса, либо отказаться от них вообще. Эта тенденция вполне соответствует и SIFMA-реализации. Вследствие наличия индикаторных цепей для них

$$t_{\text{конв.н}} \sim 0,5t_{\text{умн}}. \quad (1)$$

Это снижает эффективность конвейеризации или заставляет от нее отказаться.

Вторая проблема СС-реализации умножителя — наличие фазы гашения в двухфазной СС-дисциплине. В случае SIFMA эта проблема осложняется тем, что схема является полностью комбинаторной. Переход схемы в спейсерное состояние проходит последовательно. Все каскады дерева Уоллеса переключаются аналогично отработке рабочего состояния. Это делает длительность переключения и энергетические затраты фазы гашения практически равными соответствующим параметрам рабочей фазы. Решение этой проблемы зависит от жесткости требований к быстродействию умножителя и предполагает два варианта:

- (1) сокращение длительности фазы гашения за счет дополнительных цепей быстрой инициализации информационных структур умножителя;
- (2) использование двух параллельно и противофазно работающих умножителей.

Введение дополнительных цепей быстрой инициализации существенно сокращает временные затраты на переход в спейсерное состояние информационных цепей, и длительность фазы гашения в наибольшей степени будет определяться работой индикаторных схем. Однако соотношение (1) указывает на ограниченность такого сокращения. Кроме того, дополнительные цепи инициализации увеличивают энергопотребление умножителя и привносят дополнительную задержку в основной рабочий цикл. Эффективность сокращения временных затрат в этом случае оказывается не более 20%–30%.

Использование двух параллельно работающих умножителей в 2 раза увеличивает производительность схемы при двукратном увеличении аппаратных

затрат. Однако такое решение более предпочтительно, так как обеспечивает наименьшее энергопотребление на одну операцию из-за отсутствия дополнительных цепей инициализации.

2.2 Сокращение числа вычислительных этапов в работе умножителя

Этот подход является наиболее эффективным с точки зрения критерия быстроедействие–энергопотребление, но одновременно и наиболее сложным в реализации. Он требует большого объема исследований различных алгоритмов умножения применительно именно к СС-решению с его дополнительными требованиями. Поскольку блок умножения практически является основным блоком всех процессоров, его разработке было уделено самое серьезное внимание с момента создания вычислительных систем и предложить что-то принципиально новое не представляется возможным. В качестве прототипа было выбрано алгоритмическое решение блока, реализующего дерево Уоллеса для синхронных схем, изложенное в работе [2]. Оно позволило создать самый быстрый блок умножения к началу текущего столетия и определило основные архитектурные направления в выпускаемых процессорах большой вычислительной мощности всеми ведущими мировыми фирмами. Суть предложенного авторами метода заключается в применении избыточного кодирования операндов дерева Уоллеса. Каждый двоичный разряд операнда представляется в виде двух двоичных разрядов согласно приведенной таблице.

Такая избыточность позволяет описывать сумму двух однобитных чисел в прямом и обратном кодах одним кодовым числом и производить операцию сложения в одном кодовом разряде с сохранением переноса. Эта схема сложения обеспечивает коэффициент сжатия в дереве Уоллеса 2 : 1, реализуя наиболее быстрое сложение практически без аппаратной избыточности.

Попытка использовать парафазное кодирование для каждого из двоичных разрядов этого представления, как это обычно делается при переходе от синхронных схем к СС-решениям, приводит к заметному снижению эффективности алгоритма и большой аппаратной избыточности.

Предложенная схема кодирования очень похожа на парафазное СС-кодирование двоичного разряда и вызывает желание задействовать третье неиспользуемое состояние парафазного кода для описания «-1». Однако эта кажущаяся по-

Сравнение методов кодирования

Синхронное двоичное кодирование			Самосинхронное троичное кодирование			
Кодируемое состояние	Двоичный код		Кодируемое состояние	Троичный код		
	А	Б		Аp	Am	An
+1	1	0	+1	1	0	0
0	0	0	0	0	0	1
-1	0	1	-1	0	1	0
Не используется	1	1	Спейсер	0	0	0

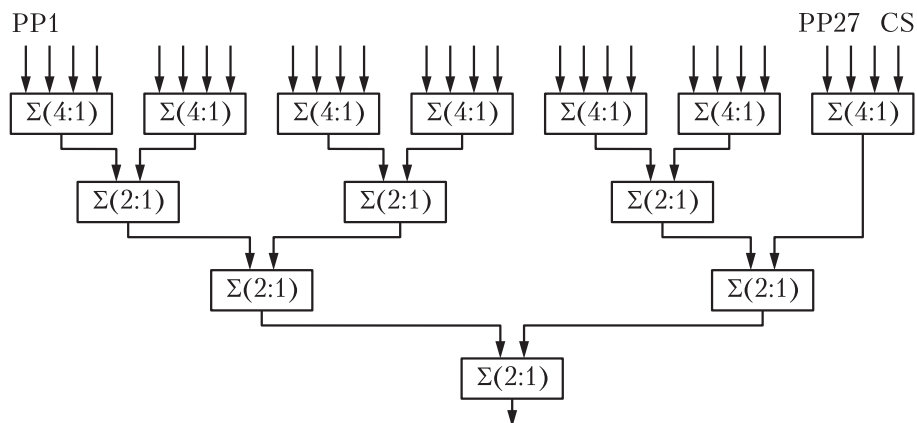


Рис. 1 Схема СС реализации дерева Уоллеса для 53-разрядных чисел для троичного кодирования двоичных разрядов

хожесть приводит к несамосинхронному коду представления двоичного разряда. Согласно формуле построения кодов двухфазной СС-дисциплины операций для описания трех состояний и спейсера минимально необходимы три бита [4]. После проведенного анализа возможных СС-кодов авторами была предложена схема кодирования, представленная в таблице. Выбранный код оказался очень удачным.

Полученная СС-реализация дерева Уоллеса для сложения операндов, кодированных алгоритмом Бута, для 53-разрядных чисел представлена на рис. 1. Здесь PP1–PP27 — частичные произведения в парафазном коде, CS — корректирующее слагаемое в парафазном коде, а остальные внутренние и выходные сигналы представлены в СС троичном коде. За счет СС троичного кодирования внутренних сигналов, предложенной в [13] методики преобразования каждого второго частичного произведения и учета дополнительного корректирующего частичного произведения CS первый каскад дерева Уоллеса обеспечивает сжатие входных операндов с 27 частичных произведений до 7 троичных сигналов.

Для сравнения на рис. 2 показана соответствующая СС-реализация дерева Уоллеса в парафазном коде. Здесь CI1–CI24 — входные переносы из предыдущего разряда дерева, CO1–CO24 — выходные переносы в следующий разряд. Все сигналы парафазные. Троичная СС-реализация обеспечивает сокращение временных затрат по сравнению с классическим СС алгоритмом более чем на 20%. Дополнительное снижение аппаратных затрат также составило около 20%, а число этапов сжатия снизилось с 7 до 4.

3 Проблемы индикации SIFMA

Корректная работа любой SI-схемы обеспечивается индицированием окончания всех переключений схемы. Схемы, в которых индицируются выходы всех

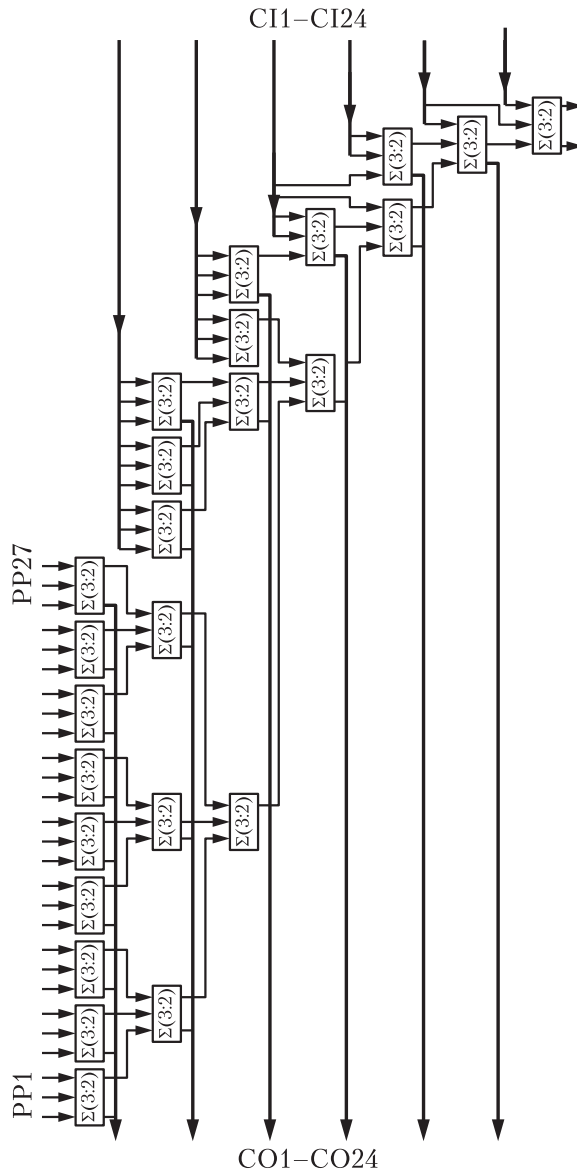


Рис. 2 Дерево Уоллеса в параформальном коде

без исключения элементов в обеих фазах работы (спейсерной и рабочей), называются полумодулярными [17]. Однако индикаторная подсхема, как правило, существенно замедляет работу полумодулярной многоурядной схемы. «Узким горлом» многоурядных СС-устройств является общий индикатор устройства, объединяющий все поразрядные индикаторные сигналы в один общий индикаторный сигнал.

Особенностью предлагаемого подхода к реализации SIFMA являются следующие принципы организации индикации:

- необходимая и достаточная, но упрощенная индикация рабочей фазы каждой ступени конвейера;
- использование спекулятивной индикации для ускорения взаимодействия ступеней конвейера.

С учетом того, что результат обработки данных комбинационной схемой (КС) в каждой ступени конвейера сохраняется в выходном регистре ступени и там индицируется в обеих фазах работы, нет необходимости индицировать КС в рабочей фазе. Используемое парафазное и специальное троичное СС-кодирование данных на всех этапах обработки гарантирует однократное переключение выходов КС из спейсера в рабочее состояние при переключении всей ступени конвейера из спейсера в рабочую фазу. Поэтому появление рабочего состояния после спейсера на информационных входах всех разрядов выходного регистра ступени гарантирует готовность результата.

Напротив, при переключении КС в спейсер индицируются все элементы, так как перед переходом в следующее рабочее состояние должна быть уверенность в том, что все элементы схемы переключились в спейсер. В противном случае возможны «гонки» и неоднократные срабатывания выходов элементов, что является нарушением принципов СС-реализации.

Поскольку КС в каждой ступени конвейера SIFMA является сложным функциональным многоурядным устройством, такая упрощенная индикация существенно сокращает аппаратные затраты и ускоряет формирование индикаторных сигналов, участвующих в запрос-ответном взаимодействии ступеней конвейера.

Спекулятивный индикатор отслеживает окончание переключения только элементов, стоящих на критическом пути обработки данных ступенью конвейера. Фактически это индикатор одного разряда тракта обработки многоурядных данных. В многоурядной схеме он формируется с гораздо меньшей задержкой, чем общий индикатор. Поскольку взаимодействие ступеней конвейера SI-схемы основано на двухфазной дисциплине работы [17], спекулятивный индикатор текущей ступени может использоваться в качестве сигнала разрешения переключения в противоположную фазу работы предшествующей ступени. Это позволяет предыдущей ступени начать переключение в очередную фазу, не дожидаясь подтверждения окончания переключения всех разрядов текущей ступени,

что существенно (на 30%–50% в зависимости от сложности схемы и разрядности данных) ускоряет работу конвейера.

Логическим обоснованием такого подхода к индикации многоразрядных SI-схем служит следующее. Спекулятивный индикатор контролирует один из аналогичных разрядов тракта обработки данных, в том числе разряд выходного регистра. При правильном выборе соответствующего разряда в подавляющем большинстве случаев он будет переключаться не раньше (а если и раньше, то не намного) остальных разрядов схемы. Поэтому спекулятивный индикатор как бы отражает работу всех разрядов ступени. Общий индикатор оказывается более задержанным из-за того, что он «собирает» индикаторные сигналы со всех разрядов схемы с помощью «дерева» индикаторных элементов. Чем больше разрядов в схеме, тем более задержанным окажется общий индикаторный сигнал в сравнении со спекулятивным индикатором.

Стопроцентная самопроверяемость конвейера SIFMA обеспечивается использованием полного индикаторного выхода каждой ступени в качестве дополнительного разрешения переключения этой ступени конвейера в противоположную фазу работы. Такая схема по-прежнему индицирует константные неисправности в полном объеме, хотя уже и не обеспечивает бесперебойной работы при любых задержках элементов. С практической точки зрения она обладает всеми преимуществами СС-схем перед синхронными аналогами. С точки зрения полноты индикации рассматриваемый SIFMA не является полумодулярным, но при правильном выборе источника спекулятивного индикатора будет работоспособным в широком диапазоне условий эксплуатации (напряжения питания, температуры), технологических и электрических параметров компонентов и обязательно остановится при возникновении константной неисправности.

4 Структурная схема SIFMA

Как основа для реализации супер-ЭВМ блок SIFMA должен обладать достаточным быстродействием. В синхронных схемах проблема быстродействия сложного вычислительного тракта наиболее эффективно решается с помощью конвейеризации вычислений. При этом разрядность схемы, если она достаточно регулярная, не играет роли.

В НЗЭ-схемах конвейеризация также используется. Но здесь уже число разрядов в схеме напрямую влияет на ее быстродействие, так как приводит к усложнению и замедлению подсхемы индикации. Кроме того, конвейеризация умножителя требует использования регистров для хранения промежуточных результатов, разрядность которых намного превышает разрядность входных операндов. Это дополнительно усложняет как саму схему, так и индикаторную подсхему и обеспечивает лишь небольшое увеличение быстродействия. Другими словами, повышение быстродействия самой сложной части SIFMA — умножителя — с помощью его конвейеризации оказывается «нерентабельным». Поэтому было принято решение для увеличения быстродействия SIFMA использовать

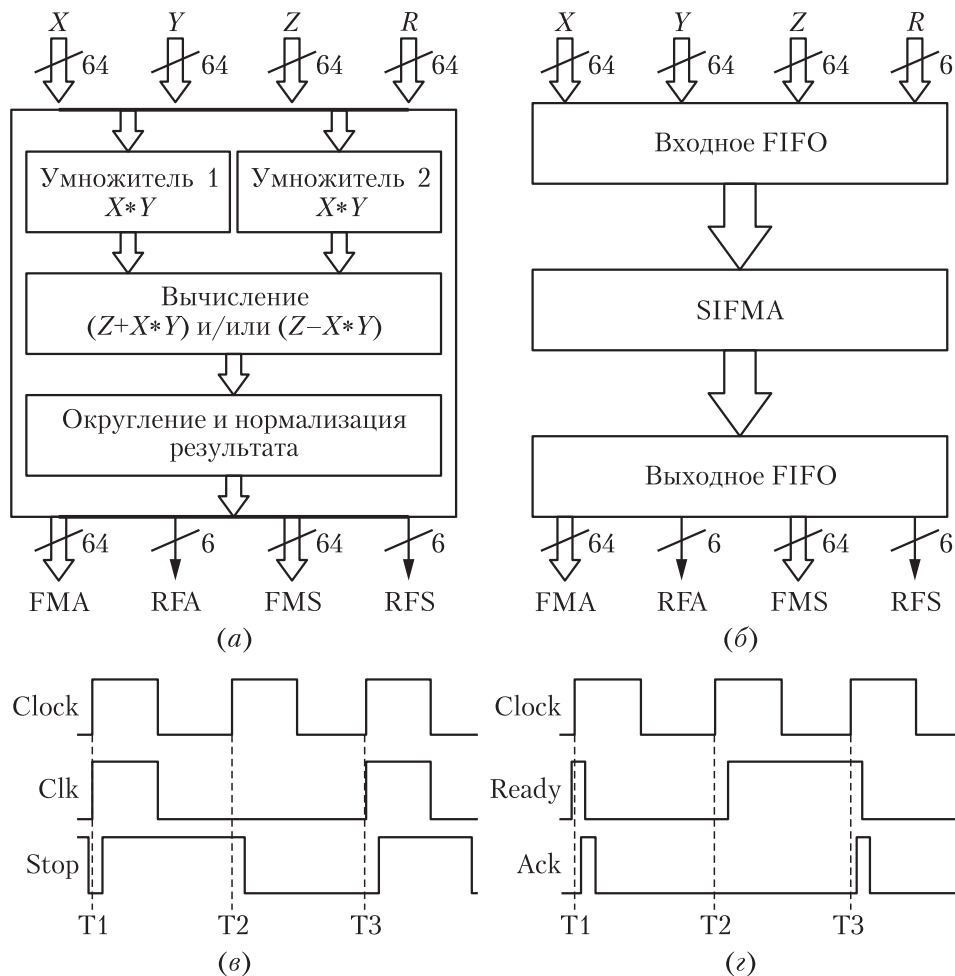


Рис. 3 Умножитель с накоплением: (а) структурная схема; (б) интерфейс с окружением; (в) проблема входного интерфейса; (г) проблемы выходного интерфейса

два блока умножителя, работающих параллельно, как показано на структурной схеме (рис. 3, а).

Здесь входные операнды — 64-разрядные обрабатываемые числа X , Y и Z и 6-разрядные признаки операции R (тип округления, точность и характер операции). Тип округления — двухразрядный признак, определяющий способ округления результата в соответствии с требованиями стандарта IEEE 754. Характер операции — двухразрядный признак, задающий количество и тип операций, выполняемых над операндами. Использование признака характера

операции позволяет минимизировать энергопотребление SIFMA в случаях, когда требуется выполнение только одной операции (сложения или вычитания) над произведением и третьим операндом.

Отметим, что параллельно вычисляются только произведения операндов X и Y из двух последовательных троек операндов, поступающих на вход SIFMA.

Дальнейшие вычисления выполняются поочередно для каждой тройки входных операндов в одном общем тракте обработки данных. Такое решение обеспечивает наилучшее соотношение «быстродействие / аппаратные затраты» при реализации SIFMA.

Время выполнения операции блоком SIFMA, как и любым НЗЭ-устройством, зависит от многих факторов: условий эксплуатации (напряжения питания, температуры окружающей среды), типа операции, вида операндов. Функциональные блоки SIFMA работают под управлением сигналов запрос-ответного взаимодействия и не требуют сигналов синхронизации.

Однако при работе с синхронным окружением возникает необходимость приема входных операндов по фронту сигнала системной частоты. В этом случае отсутствие потерь на приеме обеспечивается двумя способами:

- (1) подбором системной частоты исходя из наихудшего случая, при которой SIFMA заведомо успеет обработать любую тройку операндов при любых предусмотренных условиях эксплуатации,
- (2) использованием устройств сопряжения синхронных и асинхронных устройств, предотвращающих потерю операндов за счет приостановки «накачки» данных со стороны синхронного окружения.

Второй способ является более эффективным и быстродействующим, поскольку в его рамках легко реализовать буферизацию данных на входе/выходе SIFMA.

Для обеспечения максимального быстродействия SIFMA при работе с синхронным окружением схему на рис. 3, *а* предлагается дополнить входным и выходным НЗЭ-FIFO (first in, first out), как показано на рис. 3, *б*. Самосинхронный FIFO накапливает поступающие тройки операндов и признаки операции, обеспечивая формирование предупреждения о заполнении FIFO. Сигнал предупреждения Stop устанавливается ($Stop = 1$) в том случае, если занята входная ячейка FIFO. Это происходит при приеме каждой тройки операндов, как показано на рис. 3, *в*, где Clock — сигнал глобальной синхронизации, а Clk — тактовый вход устройства SIFMA. С продвижением операндов вглубь FIFO по мере обработки предыдущих операндов входная ячейка освобождается и сигнал Stop снимается ($Stop = 0$). Это дает синхронному окружению возможность вовремя приостановить подачу данных на вход SIFMA.

На рис. 3, *в* в моменты времени $T1$ и $T3$ SIFMA оказывается готовым к приему очередных данных и тактовый сигнал Clk формируется, а в момент времени $T2$ снятие сигнала Stop опаздывает по отношению к активному фронту тактового

сигнала Clock и заставляет синхронное окружение приостановить запись данных в SIFMA.

Последующие блоки SIFMA последовательно обрабатывают поступающие данные в СС-режиме: по мере готовности данные передаются из текущего блока в следующий, а текущий блок переходит в режим ожидания данных из предыдущего блока.

Блок FIFO на выходе SIFMA также предотвращает потерю производительности SIFMA, когда время формирования очередного результата превысит период системной частоты, из-за чего один из циклов чтения результата синхронным окружением может быть пропущен, как показано на рис. 3, з.

Сигнал Ready индицирует готовность результата на выходе SIFMA. Синхронное окружение фиксирует его по активному фронту тактового сигнала Clock и сигналом Ack = 1 подтверждает прием данных. В моменты времени T1 и T3 результат оказывается готовым перед появлением очередного активного фронта Clock, а в момент времени T2 опаздывает по отношению к активному фронту тактового сигнала. Из-за этого возникает «холостой» такт: результат выполнения очередной операции с выхода SIFMA считывается с опозданием на один такт.

Использование входного и выходного НЗЭ-FIFO позволяет решить проблемы максимально быстрой «накачки» SIFMA входными операндами и в полной мере использовать способность SIFMA выполнять операцию умножения-сложения за разное время в зависимости от типа операции, вида операндов и условий эксплуатации.

В соответствии с описанными принципами были разработаны два варианта SIFMA [18] — для синхронного и асинхронного окружения. Они подтвердили требуемый уровень производительности и энергопотребления.

Изложенные результаты получены для версии кодирования алгоритма Бута Radix2, снижающей вдвое количество суммируемых операндов в дереве Уоллеса. Исследования других вариантов алгоритма Бута для Radix3 и Radix4 в синхронном исполнении [19] указывают на возможность получения дополнительного выигрыша в быстродействии и аппаратных затратах до 10%. Однако эти алгоритмы требуют предварительной подготовки операндов, равных трем значениям множимого, т. е. требуют изначально дополнительной операции сложения. Вопрос целесообразности использования таких алгоритмов в СС-схемах требует дополнительных исследований.

5 Заключение

В супер-ЭВМ эксафлопсного класса, когда число ядер достигает сотен миллионов, практически необходимо существенно усиливать аппаратную составляющую средств обеспечения надежности и достоверности результатов вычислений. Предложенная СС-схемотехника для реализации основных вычислительных узлов обеспечивает высокую эффективность в решении этой задачи.

Впервые в отечественной и зарубежной практике предпринята попытка разработки действительно 64/32-разрядного SIFMA-устройства в виде схемы, поведение которой не зависит от задержек элементов и задержек в проводах до точки разветвления.

При разработке СС сумматоров и блоков умножения целесообразно использовать наряду с парафазным троичный СС-код.

Из всего многообразия алгоритмов следует выбирать аппаратно оправданные решения с минимальным числом этапов, требующих индикации завершения операций обработки.

Алгоритм суммирования по дереву Уоллеса рекомендуется по возможности целиком выполнять в одной ступени конвейера, применяя два параллельных вычислительных блока в противофазных режимах.

Литература

1. Семенов Ю. А. Суперкомпьютеры и Watson. <http://book.itep.ru/10/supercomp.htm>.
2. Дьяченко Ю. Г., Рождественский Ю. В., Морозов Н. В., Степченков Д. Ю. Квазисамосинхронный вычислитель: практическая реализация // Проблемы разработки перспективных микро- и наноэлектронных систем: III Всеросс. науч.-технич. конф. (МЭС-2008): Сб. научных тр. — М.: ИПИМ РАН, 2008. С. 435–440.
3. Степченков Ю. А., Дьяченко Ю. Г., Бобков С. Г. Квазисамосинхронный вычислитель: методологические и алгоритмические аспекты // Проблемы разработки перспективных микро- и наноэлектронных систем: III Всеросс. науч.-технич. конф. (МЭС-2008): Сб. научных тр. — М.: ИПИМ РАН, 2008. С. 441–446.
4. Bink A., York R. ARM996HS: The first licensable, Clockless 32-bit processor core // IEEE Micro, 2007. Vol. 27. No. 2. P. 58–68.
5. Bing H. Acceleration of spiking neural network on general purpose graphics processor. Ph.D. Dissertation. — University of Dayton, 2010. 43 p.
6. IEEE Standard for Floating-Point Arithmetic 754-2008. — IEEE, 2008. doi:10.1109/IEEE STD. 2008.4610935.
7. Pillai R. V. K., Shah S. Y. A., Al-Khalili A. J., Al-Khalili D. Low power floating point MAFs — a comparative study // 6th Symposium (International) on Signal Processing and Its Applications, 2001. Vol. 1. P. 284–287.
8. Seidel P.-M. Multiple path IEEE floating-point fused multiply-add // 46th IEEE Midwest Symposium (International) on Circuits and Systems Proceedings, 2003. P. 1359–1362.
9. Noche J. R., Araneta J. C. An asynchronous IEEE floating-point arithmetic unit // Proc. Sci. Diliman, 2007. Vol. 19. No. 2. P. 12–22.
10. Manohar R., Sheikh B. R. Operand-optimized asynchronous floating-point units and method of use therefore. U.S. Patent No. 20130124592, May 2013.
11. Руткевич А., Бумагин А., Гондарь А. и др. Методы снижения энергопотребления в строго самосинхронных микропроцессорных схемах // Компоненты и технологии, 2009. № 9. С. 109–114.

12. *Smith S. C., Di J.* Designing asynchronous circuits using NULL Convention Logic (NCL) // Synthesis Lectures on Digital Circuits and Systems, 2009. Vol. 4. No. 1. P. 61–73.
13. *Соколов И. А., Степченко Ю. А., Бобков С. Г., Захаров В. Н., Дьяченко Ю. Г., Рождественский Ю. В., Сурков А. В.* Базис реализации супер-ЭВМ эксафлопсного класса // Информатика и её применения, 2014. Т. 8. Вып. 1. С. 45–70.
14. *Степченко Ю. А., Петрухин В. С., Дьяченко Ю. Г.* Опыт разработки самосинхронного ядра микроконтроллера на базовом матричном кристалле // Проблемы разработки перспективных микроэлектронных систем — 2005: Всеросс. науч.-технич. конф. (МЭС-2005): Сб. научных тр. — М.: ИПИМ РАН, 2005. С. 235–242.
15. *Galal S., Horowitz M.* Energy-efficient floating-point unit design // IEEE Trans. Comput., 2011. Vol. 60. No. 7. P. 913–922.
16. *Бриллюэн Л.* Наука и теория информации / Пер с англ. — М.: Физматлит, 1960. 391 с. (*Brillouin L.* Science and information theory. New York: Academic Press Publ., 1956. 350 p.)
17. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах / Под ред. В. И. Варшавского. — М.: Наука, 1986. 400 с.
18. *Степченко Ю. А., Рождественский Ю. В., Дьяченко Ю. Г., Морозов Н. В., Степченко Д. Ю., Рождественскене А. В., Сурков А. В.* Самосинхронный умножитель с накоплением: практическая реализация // Системы и средства информатики, 2014. Т. 24. № 3. С. 44–58.
19. *Bewick G. W.* Fast multiplication: Algorithms and Implementation. Ph.D. Dissertation. — Stanford University, 1994. 155 p.

Поступила в редакцию 20.08.14

FUSED MULTIPLY-ADD: METHODOLOGICAL ASPECTS

I. Sokolov¹, Y. Stepchenkov¹, S. Bobkov², Y. Rogdestvenski¹, and Y. Diachenko¹

¹Institute of Informatics Problems, Russian Academy of Sciences, Moscow 119333, 44-2 Vavilova Str., Russian Federation

²Scientific Research Institute for System Studies, Russian Academy of Sciences, 36 bld. 1 Nakhimovsky Prosp., Moscow 117218, Russian Federation

Abstract: The paper presents approaches to designing self-timed (ST) equipment and analyzes conditions of in-system integration of synchronous and ST units in a supercomputer network taking the ST Fused Multiply-Add (FMA) unit as an example. Self-timed FMA complies with the IEEE 754 Standard and performs either one double precision FMA operation or one or two single precision operations simultaneously under three operands. It utilizes the ST-ternary encoding and the 65-nanometer CMOS (complementary metal-oxide-semiconductor) technology as the implementation basis. Depending on realization, it works with asynchronous or synchronous environment and provides not less than 1 GFlops performance with latency not more than 6 ns with respect to input data arrival.

Keywords: self-timed circuit; supercomputer; fused multiply-add; adder; pipeline; energy efficiency

DOI: 10.14357/08696527140305

Acknowledgments

This project was financially supported by the Russian Foundation for Basic Research (project 13-07-12068 ofi_m).

References

1. Semenov, Yu. A. *Superkomp'yutery i Watson* [Supercomputers and Watson]. Available at: <http://book.itep.ru/10/supercomp.htm> (accessed August 15, 2014).
2. Diachenko, Y. G., Y. V. Rogdestvenski, N. V. Morozov, and D. Y. Stepchenkov. 2008. Kvazisamosinhronny vychislitel': Prakticheskaya realizatsiya [Quasi self-timed coprocessor: Practical implementation]. *Tr. Mezhdunar. konf. "Problemy Razrabotki Perspektivnykh Mikro- i Nanojelektronnykh Sistem"* [Problems of the Perspective Micro- and Nanoelectronic Systems Development Conference Proceedings]. Moscow. 435–440.
3. Stepchenkov, Y. A., Y. G. Diachenko, and S. G. Bobkov. 2008. Kvazisamosinkhronny vychislitel': Metodologicheskie i algoritmicheskie aspekty [Quasi-self-timed coprocessor: The methodological aspects]. *Tr. Mezhdunar. konf. "Problemy Razrabotki Perspektivnykh Mikro- i Nanojelektronnykh Sistem"* [Problems of the Perspective Micro- and Nanoelectronic Systems Development Conference Proceedings]. Moscow. 441–446.
4. Bink, A., and R. York. 2007. ARM996HS: The first licensable, Clockless 32-bit processor core. *IEEE Micro*. 27(2):58–68.
5. Bing, H. Acceleration of spiking neural network on general purpose graphics processor. Ph.D. Dissertation. University of Dayton, 2010. 43 p.
6. IEEE Computer Society. 2008. IEEE Standard for Floating-Point Arithmetic 754-2008. doi:10.1109/IEEE STD. 2008.4610935.
7. Pillai, R. V. K., S. Y. A. Shah, A. J. Al-Khalili, and D. Al-Khalili. 2001. Low power floating point MAFs — a comparative study. *6th Symposium (International) on Signal Processing and Its Applications Proceedings*. 1:284–287.
8. Seidel, P.-M. 2003. Multiple path IEEE floating-point fused multiply-add. *46th IEEE Midwest Symposium (International) on Circuits and Systems Proceedings*. 1359–1362.
9. Noche, J. R., and J. C. Araneta. 2007. An asynchronous IEEE floating-point arithmetic unit. *Proc. Sci. Diliman* 19(2):12–22.
10. Manohar, R., and B. R. Sheikh. May 2013. Operand-optimized asynchronous floating-point units and method of use therefore. U.S. Patent No. 20130124592.
11. Rutkevich, A., A. Bumagin, A. Gondar', *et al.* 2009. Metody snizheniya energopotrebleniya v strogo samosinkhronnykh mikroprocessornykh skhemakh [Methods of decreasing energy consumption in strongly self-timed microprocessors]. *Komponenty i Tehnologii* [Components and Technologies] 9:109–114.

12. Smith, S. C., and J. Di. 2009. Designing asynchronous circuits using NULL Convention Logic (NCL). *Synthesis Lectures on Digital Circuits and Systems* 4(1):61–73.
13. Sokolov, I. A., Y. A. Stepchenkov, S. G. Bobkov, V. N. Zakharov, Y. G. Diachenko, Y. V. Rogdestvenski, and A. V. Surkov. 2014. Bazis realizatsii super-EVM eksaflop-snogo klasa [Implementation basis of ExaFlops class supercomputer]. *Informatika i ee Primeneniya — Inform. Appl.* 8(1):45–70.
14. Stepchenkov, Y. A., V. S. Petruhin, and Y. G. Diachenko. 2005. Opyt razrabotki samosinkhronnogo yadra mikrokontrollera na bazovom matrichnom kristalle [The experience in microcontroller's self-timed core design on FPGA]. *Tr. Mezhdunar. konf. "Problemy Razrabotki Perspektivnykh Mikro- i Nanojelektronnykh Sistem"* [Problems of the Perspective Micro- and Nanoelectronic Systems Development Conference Proceedings]. Moscow. 235–242.
15. Galal, S., and M. Horowitz. 2011. Energy-efficient floating-point unit design. *IEEE Trans. Comput.* 60(7):913–922.
16. Brillouin, L. 1956. *Science and information theory*. New York: Academic Press Publ. 350 p.
17. Varshavsky, V. I., ed. 1990. *Self-timed control of concurrent processes: The design of aperiodic logical circuits in computers and discrete systems*. Mathematics and its applications ser. Dordrecht, The Netherlands: Kluwer Academic Publ. 418 p.
18. Stepchenkov, Y. A., Y. V. Rogdestvenski, Y. G. Diachenko, N. V. Morozov, and D. Y. Stepchenkov. 2014. Samosinkhronny umnozhitel's nakopleniem: Prakticheskaya realizatsiya [Self-timed fused multiply-add unit: Practical implementation]. *Sistemy i Sredstva Informatiki — The Systems and Means of Informatics* 24(3):44–58.
19. Bewick, G. W. 1994. Fast multiplication: Algorithms and Implementation. Ph.D. Dissertation. Stanford University. 155 p.

Received August 20, 2014

Contributors

Sokolov Igor A. (b. 1954) — Academician of the Russian Academy of Sciences, Doctor of Science in technology, Director, Institute of Informatics Problems, Russian Academy of Sciences, Moscow 119333, Russian Federation; ISokolov@ipiran.ru

Stepchenkov Yuri A. (b. 1951) — Candidate of Sciences (PhD) in technology, Head of Department, Institute of Informatics Problems, Russian Academy of Sciences, Moscow 119333, Russian Federation; YStepchenkov@ipiran.ru

Bobkov Sergey G. (b. 1955) — Doctor of Science in technology, Head of Department, Scientific Research Institute for System Studies, Russian Academy of Sciences, Moscow 117218, Russian Federation; Bobkov@cs.niisi.ras.ru

Rogdestvenski Yuri V. (b. 1952) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences, Moscow 119333, Russian Federation; YRogdest@ipiran.ru

Diachenko Yuri G. (b. 1958) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences, Moscow

119333, Russian Federation; diaura@mail.ru