

Самосинхронное устройство умножения-сложения с плавающей точкой

Ю.А. Степченков, Ю.В. Рождественский, Ю.Г. Дьяченко, Н.В. Морозов, Д.Ю. Степченков,
Б.А. Степанов, Д.Ю. Дьяченко, А.В. Рождественскене

Институт проблем информатики Федерального исследовательского центра «Информатика и
управление» Российской академии наук (ФИЦ ИУ РАН)

{YStepchenkov, YRogdest, YDiachenko, NMorozov, DStepchenkov}@ipiran.ru

Аннотация — Представлено самосинхронное устройство совмещенного умножения-сложения в качестве сопроцессора, относящееся к новому подклассу независимых от задержек схем с адаптивной индикацией. Оно полностью соответствует стандарту IEEE 754 и реализует одновременно операции сложения и вычитания третьего операнда из произведения первых двух. Каждый 64-разрядный операнд содержит либо одно число двойной точности, либо два числа одинарной точности. Для минимизации аппаратных затрат и энергопотребления число ступеней конвейера сокращено до двух. Оптимальное соотношение "производительность / аппаратные затраты" достигается за счет двух параллельных каналов с общим входом и выходом. Дополнительное сокращение аппаратных затрат и повышение быстродействия обеспечивается использованием адаптивной индикации. Умножение реализовано на основе самосинхронного избыточного кода. Устройство разработано на базе объемной КМОП технологии с проектными нормами 65 нм и обеспечивает производительность на уровне 3,15 гигафлопс.

Ключевые слова — дерево Уоллеса, поразрядная индикация, избыточное кодирование, троичный сумматор.

I. ВВЕДЕНИЕ

Операция "умножение с накоплением" (multiply-accumulate) издавна используется в современных процессорах цифровой обработки сигналов и предполагает сложение (вычитание) результата умножения двух входных операндов с содержимым аккумулятора. Аппаратное совмещение умножения двух входных операндов и последующего сложения (вычитания) с третьим входным операндом в одном устройстве может выполняться с двойным округлением (типично для DSP) или однократным округлением. Версия с однократным округлением получила название fused multiply-add (FMA). Она обеспечивает более высокую точность вычислений. Поэтому FMA-операция стала де-факто стандартной операцией современных центральных процессоров.

Библиография по вопросу разработки FMA устройства весьма обширна. Львиная доля всех публикаций связана с разработкой синхронных вариантов FMA [1]-[3]. В последние годы все больше публикаций посвящается асинхронным FMA [4]-[5]. Однако асинхрон-

ные решения, претендующие на титул самосинхронных устройств, опираются, как правило, на использование "слабых" транзисторов и не отвечают требованиям создания помехоустойчивых и энергоэффективных самосинхронных устройств, правильное функционирование которых не зависит от задержек в элементах (НЗ, Speed-Independent circuits) при любых условиях эксплуатации.

Отсутствие в НЗ схемах "дерева" тактовых сигналов и работа только "по запросу" обеспечивают уменьшение энергопотребления. Сохранение работоспособности НЗ схем при сверхмалых значениях питающих напряжений открывает широкие перспективы для получения "долгоиграющих" портативных изделий с аккумуляторным питанием и создания бортовых комплексов с ограниченными энергоресурсами. Устойчивая работа в экстремальных условиях достигается за счет аппаратной избыточности и дополнительных временных затрат на индикацию и фазу «спейсера» в работе НЗ схем. Однако грамотное проектирование НЗ схем позволяет существенно снизить эту избыточность, а в ряде случаев, например, в отказоустойчивых устройствах [6], получить результаты лучше, чем в синхронных аналогах.

Ранее авторами уже предпринималась попытка разработки НЗ устройства FMA гигафлопсного класса – SIFMA [7]-[8]. Однако для достижения предельного быстродействия было решено реализовать SIFMA как самосинхронную схему со спекулятивной индикацией в виде конвейера с большим числом ступеней. В результате заявленное быстродействие было достигнуто ценой отказа от стопроцентной самопроверяемости SIFMA, что фактически вывело его за пределы класса НЗ схем.

В данной статье излагаются результаты проектирования 64-разрядного действительно НЗ сопроцессора с плавающей точкой (SIFPC), выполняющего операцию FMA в соответствии со стандартом IEEE 754 и имеющего производительность на уровне 3 Гфлопс, в сравнении с предшествующей реализацией [7]-[8].

II. ОСОБЕННОСТИ SIFPC

SIFPC разработан для потоковой обработки данных. Каждый из трех обрабатываемых 64-разрядных

операндов содержит либо одно число двойной точности, либо два числа одинарной точности в соответствии со стандартом IEEE 754. В последнем случае выполняются две независимые операции FMA над двумя тройками операндов одинарной точности. Дополнительной особенностью SIFPC является одновременное вычисление не только суммы, но и разности между третьим операндом и произведением первых двух операндов.

А. Структурная схема

Разработка современных вычислительных средств ведется в направлении обеспечения минимального энергопотребления при достаточно высокой производительности. Это определяется тенденцией к использованию сравнительно невысокой тактовой частоты и большого числа вычислительных узлов на одну СБИС для высокопроизводительных компьютеров.

Наличие двух фаз в работе любой НЗ схемы (активной – рабочей и паузы – спейсерной) наталкивает на идею использовать два параллельных канала, фазы работы которых чередуются. В реализации аналогичного устройства SIFMA [8] параллельными были только умножители и блоки обработки экспонент (УОЭ), как показано на рис. 1. Сложение-вычитание, нормализация и округление выполнялись на общих для двух каналов ресурсах.

Параллельные блоки УОЭ составляли первую ступень конвейера ядра SIFMA. Общий тракт последующих вычислений был разбит на 4 ступени конвейера. Это позволило достичь среднестатистической производительности на уровне 2,82 Гфлопс по результатам моделирования работы устройства в режимах двойной и одинарной точности. Переход от спекулятивной индикации к полной при такой же организации конвейера приводил к ухудшению производительности до уровня 2,31 Гфлопс. Анализ полученных результатов показал, что причиной снижения быстродействия в варианте с полной индикацией являлась индикаторная подсхема. Необходимость запоминания большого количества сигналов на границах ступеней конвейера в регистрах ступеней, во-первых, существенно увеличивала аппаратные затраты на реализацию как самих регистров, так и их индикаторной подсхемы, а во-вторых, выливалась в дополнительную задержку формирования общего индикатора ступени конвейера.

В связи с этим была предложена новая структурная схема устройства, выполняющего операцию FMA, – НЗ сопроцессора с плавающей точкой (SIFPC), показанная на рис. 2. В ней дублируется весь тракт обработки входных данных и используется полная индикация, а число ступеней конвейера сокращено до минимума. Это позволило сократить аппаратные затраты на реализацию промежуточных регистров и их индикаторных подсхем и уменьшить количество "бутылочных горлышек" в критическом пути вычисления результата операции. В итоге среднестатистическая производительность достигла уровня 3,15 Гфлопс при работе с синхронным окружением и 3,9 Гфлопс при отсутствии непроизводительного ожидания отклика от

синхронного окружения об успешном считывании результата с выхода SIFPC.

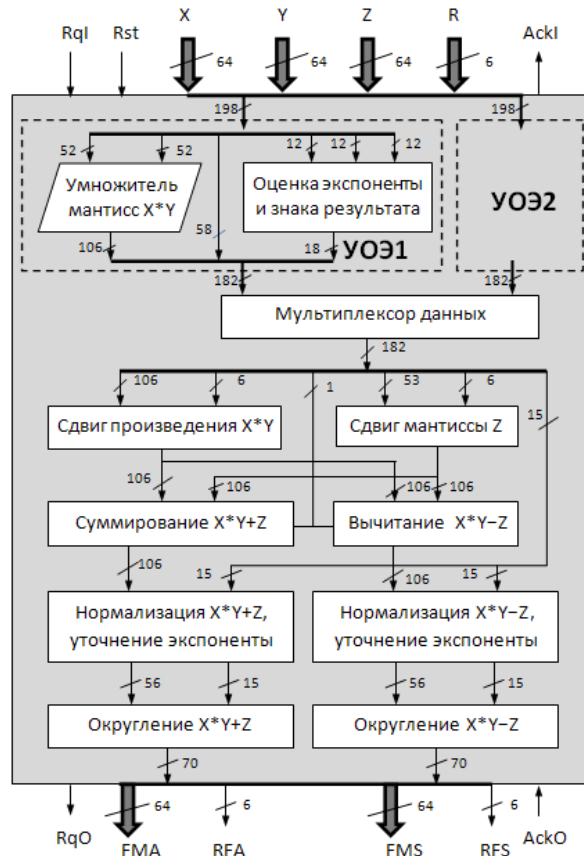


Рис. 1. Структурная схема ядра SIFMA

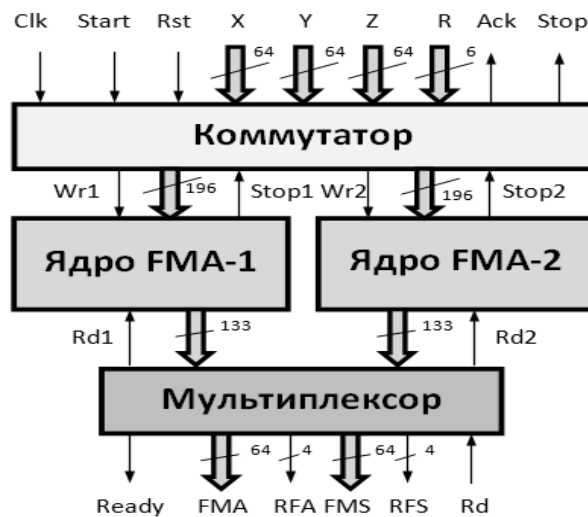


Рис. 2. Структурная схема SIFPC

Анализ минимаксной кривой, построенной в координатах "энергопотребление - площадь" в соответствии с методикой, приведённой в [9], применительно к технологии 65-нм и к предполагаемой производительности SIFPC, позволил определить основные характеристики прототипа ядра FMA и выбрать структурную схему (рис. 3) для его реализации.



Рис. 3. Структурная схема ядра FMA

Сигналы Wr и Ack обеспечивают асинхронный входной интерфейс: первый из них отражает готовность данных на входе SIFPC, а второй подтверждает успешное завершение приема входных данных SIFPC. Аналогично сигналы $Ready$ и Rd обеспечивают выходной асинхронный интерфейс: сигнал $Ready$ индицирует готовность результата; сигнал Rd подтверждает окончание чтения результата асинхронным окружением.

Ядро FMA содержит входное и выходное FIFO. Они повышают быстродействие FMA при работе с синхронным окружением [7] за счет буферизации потока данных. Входное и выходное FIFO реализованы как полуплотные СС регистры сдвига [10, рис. 11.9] емкостью 3 слова.

Вход Clk подключается к источнику тактовых сигналов синхронного окружения или к источнику сигнала подтверждения готовности входных операндов и признаков операции в асинхронном окружении. Выход Ack используется только асинхронным окружением в рамках запрос-ответного взаимодействия окружения и SIFPC.

Поскольку все функциональные блоки в SIFPC являются НЗ схемами, обновление результата на выходе SIFPC произойдет только после подтверждения окружением успешного чтения результата с помощью сиг-

нала $Rd=1$. В случае с асинхронным окружением это реализуется естественным образом в рамках запрос-ответного взаимодействия SIFPC и окружения. Синхронное окружение может использовать выход $Ready$ для записи результата в свои регистры, а в качестве сигнала Rd использовать либо сигнал, производный от сигнала $Ready$, либо системный тактовый сигнал. Однако в последнем случае быстродействие SIFPC будет занижено из-за непроизводительного ожидания активного фронта тактового сигнала после выдачи сигнала готовности результата.

Традиционно алгоритмы умножения базируются на модификациях алгоритма кодирования Бута и конвейеризованном дереве Уоллеса (ДУ). Однако разработчики современных многоядерных процессоров стараются снизить до минимума число ступеней конвейера, реализующего ДУ. Этой тенденции соответствует и SIFPC: умножение мантисс, вычисление экспоненты результата и выравнивание третьего операнда реализуются одной ступенью конвейера.

Обычно НЗ схемы, к числу которых относится и SIFPC, реализуются с использованием парафазного (dual-rail) кодирования данных. Однако исследования показали [7], что в НЗ устройстве FMA наилучшие параметры обеспечивает избыточное (троичное) самосинхронное кодирование.

В. Избыточное СС кодирование

Умножитель является наиболее сложным блоком в составе SIFPC. В качестве его прототипа была выбрана реализация для синхронных схем [11]. В ней используется избыточное кодирование операндов, обеспечивающее коэффициенты сжатия в ДУ 4:1 на первом каскаде и 2:1 на всех последующих каскадах. Анализ возможных СС кодов привел к выбору избыточного СС кода [12], представленного в табл. 1.

Таблица 1

Избыточное СС кодирование

Кодируемое состояние	Троичный код		
	AP	AM	A0
+1	1	0	0
0	0	0	1
-1	0	1	0
спейсер	0	0	0

НЗ схема одного разряда троичного сумматора с полной индикацией внутренних сигналов и выходов показана на рис. 4. Пунктирным овалом выделена индикаторная подсхема. Входные операнды и выход суммы представлены в избыточном СС коде (табл. 1). Остальные сигналы имеют парафазное кодирование.

Каждый сумматор первого каскада ДУ дополняются двумя преобразователями разности двух парафазных операндов в один троичный операнд (рис. 5).

Для сравнения на рис. 6 приведена схема одноразрядного НЗ сумматора с парафазным кодированием сигналов. Она обладает меньшей сложностью реализа-

ции, что сказывается и на суммарной сложности ДУ: 1 разряд троичного ДУ реализуется на 2530 КМОП транзисторах, в то время как аналогичный парафазный вариант имеет 2260 транзисторов. Но из-за меньшего коэффициента сжатия парафазного ДУ его быстродействие оказывается хуже быстродействия ДУ на троичных сумматорах на 16%.

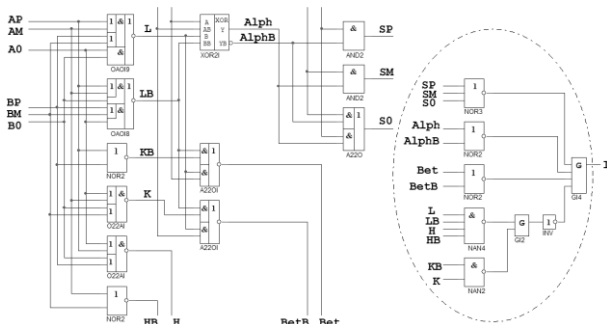


Рис. 4. Одноразрядный троичный НЗ сумматор

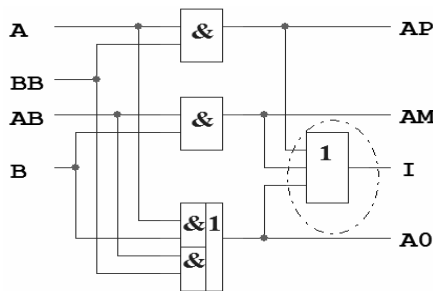


Рис. 5. Преобразователь разности парафазных операндов в троичный операнд

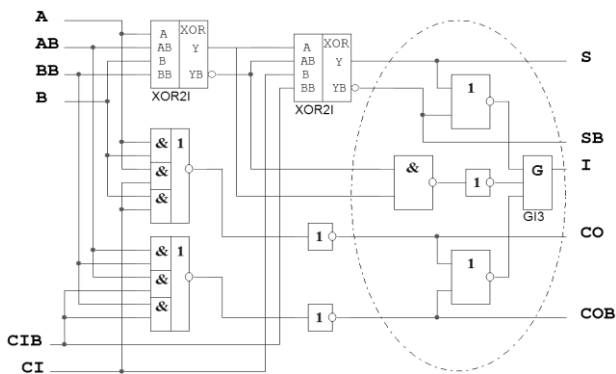


Рис. 6. Одноразрядный НЗ сумматор с парафазным кодированием

Таким образом, реализация ДУ с избыточным СС кодированием обеспечивает существенное повышение быстродействия умножителя по сравнению с классическим СС алгоритмом. Это достигается за счет сокращения количества этапов сжатия с 7 до 4, хотя аппаратные затраты троичного дерева Уоллеса при этом оказываются на 12% больше, чем в парафазном варианте.

Производительность SIFPC в значительной степени зависит от реализации его индикаторной подсхемы.

С. Индикация SIFPC

SIFPC является устройством, обрабатывающим многоразрядные данные. Классическая индикация такого рода устройства, построенная на формировании общего индикатора для каждого функционального блока, приводит к существенному замедлению его работы. "Узким горлом" многоразрядных НЗ устройств является общий индикатор устройства, объединяющий все индикаторные сигналы в один общий индикаторный сигнал, который используется для организации запрос-ответного взаимодействия между функциональными блоками НЗ устройства.

С точки зрения теории [10], схема не является НЗ схемой, если хотя бы один элемент, в котором было инициировано переключение в инверсное состояние в процессе перехода из одной фазы работы схемы в противоположную фазу, не успел завершить это переключение до изменения состояния его входов. Однако анализ практических НЗ схем показывает, что почти в любой НЗ схеме не всегда в формировании очередного рабочего состояния информационных выходов принимают участие абсолютно все элементы, в которых было инициировано переключение в рабочую фазу. Это создает предпосылку к упрощению индикации многоразрядных НЗ схем, в том числе и SIFPC.

В основе подхода к упрощению индикации на уровне одного разряда лежат два свойства, характеризующие схемы на КМОП транзисторах и НЗ схемы с правильно организованной дисциплиной сигналов:

1. Переключение элемента на КМОП транзисторах прекращается, если состояние его выхода соответствует состоянию его входов.

2. В процессе переключения НЗ схемы из спейсера в рабочую фазу каждый элемент схемы может переключиться в рабочее состояние только один раз.

Это позволяет использовать необходимую и достаточную, но упрощенную индикацию рабочей фазы и стопроцентно полную индикацию спейсерной фазы в любом НЗ функциональном блоке. Действительно, полная индикация спейсерной фазы гарантирует, что переключение схемы в очередное рабочее состояние будет всегда начинаться из состояния, в котором все элементы схемы находятся в спейсере. Переключение всех выходов в рабочую фазу говорит о том, что с точки зрения окружения переключение схемы в новое рабочее состояние завершилось.

Упрощенная (адаптивная) индикация реализуется путем формирования индикаторных выходов первого каскада индикаторной подсхемы с помощью логических функции "ИЛИ" для информационных сигналов с нулевым спейсером или функций "И" для информационных сигналов с единичным спейсером. При этом на вход каждого элемента первого каскада индикаторной подсхемы должны поступать составляющие хотя бы одного парафазного сигнала.

На рис. 7а показана схема простейшего мультиплексора 2:1 с парафазными информационными входа-

ми и входами выборки с нулевым спейсером и индикаторной подсхемой, являющейся стопроцентно полной в обеих фазах работы схемы, реализованная в КМОП базе с ограничением (не более трех) на число последовательно включенных транзисторов с каналом р-типа. Рис. 7б демонстрирует схему такого же мультиплексора с упрощенной индикацией. Анализ этих схем показывает, что упрощенная индикаторная подсхема в 1,1 раза проще и на 10-15% быстрее своего аналога.

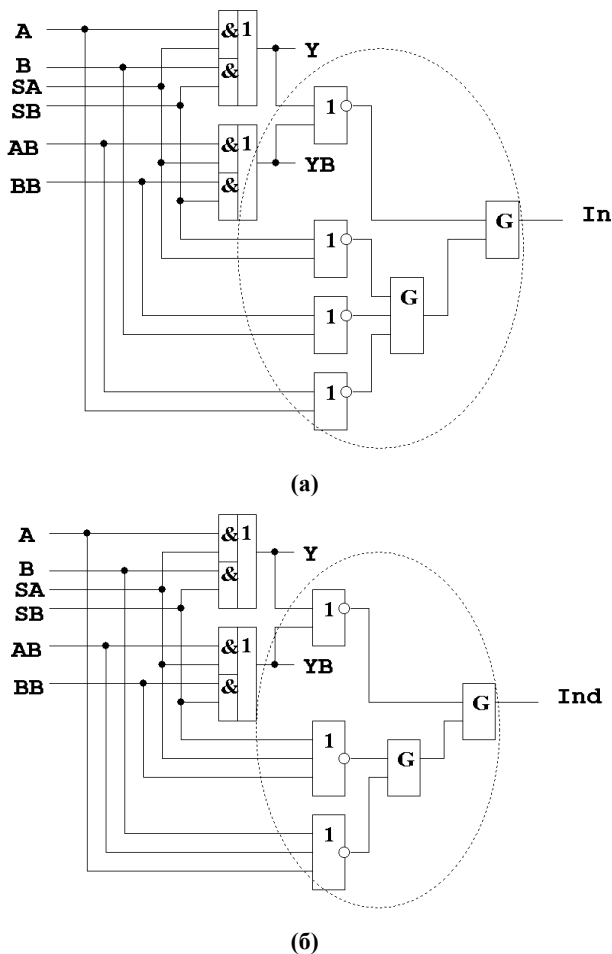


Рис. 7. Мультиплексор 2:1 с парафазными сигналами с полной (а) и упрощенной (б) индикацией

Следует отметить, что формирование общих индикаторных сигналов функциональных блоков (ступеней конвейера), участвующих в протоколе запрос-ответного взаимодействия, должно выполняться схемой полной индикации в обеих фазах работы НЗ устройства. В случае использования конвейерной структуры НЗ устройства полную индикацию в обеих фазах работы достаточно делать только для регистров на входе или выходе ступени конвейера.

Моделирование различных цифровых устройств показывает, что рассмотренный способ адаптивной индикации обеспечивает сокращение сложности индикаторной подсхемы в 1,5 – 1,7 раза (в зависимости от используемой библиотеки элементов нижнего уровня)

и повышение быстродействия индикаторной части схемы на 5 – 25%.

Применение упрощенной индикации в рабочей фазе приводит к появлению нового подкласса НЗ схем – схем с адаптивной индикацией (НЗАИ). Такая реализация гарантирует начало переключения схемы в новое рабочее состояние из подтвержденного спейсерного состояния. Это обеспечивает сохранение всех практически значимых свойств НЗ схем: обнаружение константных неисправностей, широкий диапазон работоспособности по напряжению питания и температуре окружающей среды и т.д.

Следует отметить, что НЗАИ схема перестает быть полумодулярной и вызывает диагностику о нарушении самосинхронности при ее анализе имеющимися программными средствами (например, [13]). Данное нарушение является таковым только с точки зрения формальной теории, поскольку полный контроль переключения всех элементов схемы в фазу спейсера перед следующим переходом в новое рабочее состояние обеспечивает самосинхронность схемы.

Аспекты оптимизации индикаторной подсхемы НЗ схемы с точки зрения организации запрос-ответного взаимодействия в конвейере SIFPC были подробно рассмотрены в [8].

Предложенные способы оптимизации построения индикаторной подсхемы позволили сократить аппаратные затраты всей схемы SIFPC на 12% и повысить ее быстродействие на 17% в сравнении с вариантом, представленным в [8].

III. ТОПОЛОГИЧЕСКАЯ РЕАЛИЗАЦИЯ SIFPC

Разработка SIFPC изначально велась в условиях заданного ограничения площади топологической реализации – не более 1 мм².

НЗ схемы используют избыточное кодирование информационных сигналов и индикацию переходных процессов. Поэтому сложность комбинационной НЗ схемы возрастает в 2,3 – 2,5 раза в сравнении с синхронным аналогом. В такой же пропорции увеличивается и количество сигналов в схеме.

Троичное кодирование операндов в НЗ умножителя смягчает данную проблему. Во-первых, дерево Уоллеса является совокупностью сумматоров с сохранением переноса и при троичном СС кодировании операндов каждый сумматор в умножителе имеет 3 выхода, в то время как при парафазном СС кодировании 4 выхода. Во-вторых, межразрядные сигналы переноса в троичном НЗ сумматоре передаются в соседний разряд в этом же каскаде дерева Уоллеса и не занимают вертикальные трассы.

Анализ схем кодера Бута, генератора частичных произведений (PP0...PP26, CS) и количества формируемых ими сигналов, поступающих на входы дерева Уоллеса, показал, что общее количество входов дерева Уоллеса (более 3000) превышает количество свободных трасс в слоях вертикальной разводки при задан-

ном ограничении на горизонтальный размер проектируемого умножителя. Поэтому было принято решение интегрировать кодер Бута и генератор частичных произведений в структуру дерева Уоллеса, как показано на рис. 8. Блок "Кодер Бута & Σ " реализует генератор и сумматор указанных в скобках частичных произведений. Блок " Σ " суммирует указанные в скобках частичные произведения.

Для получения двух произведений одинарной точности достаточно просуммировать частичные произведения PP0 – PP12 и корректирующий операнд CS. Сумматоры частичных произведений PP13 – PP26 используются только для умножения операндов двойной точности, имеющих разрядность 53.

Топология умножителя имеет размеры 404×490 мкм². Размер топологической реализации всей схемы SIFPC (рис. 9) составил 990×980 мкм².

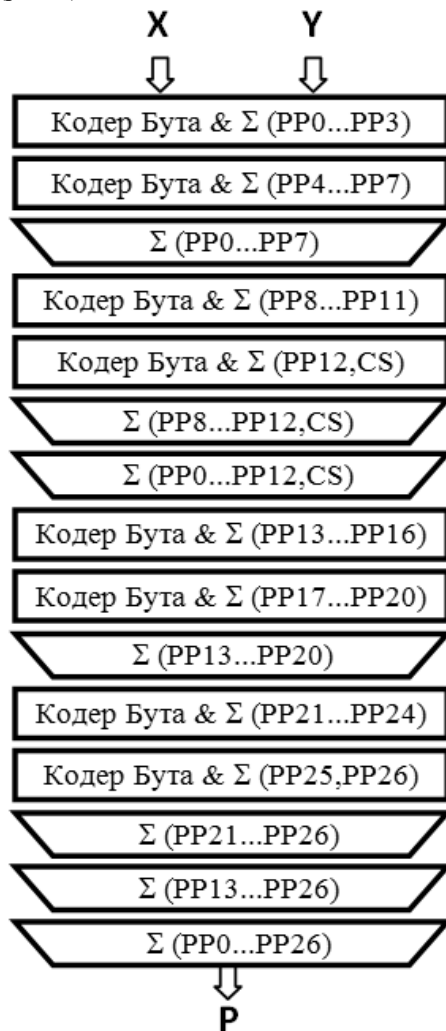


Рис. 8. Организация вычислений в умножителе

IV. ПАРАМЕТРЫ SIFPC

SIFPC был спроектирован в стандартной 65 нм КМОП объемной технологии с 6 слоями металлизации. Параметры SIFPC в сравнении с синхронным аналогом близкой производительности [9] приведены в табл. 2.

Временные и энергетические параметры получены на основе моделирования с учетом паразитных емкостей и резисторов топологической реализации для статистически достоверного набора комбинаций входных операндов двойной и одинарной точности.

Быстродействие определялось для типовых условий эксплуатации (1,0 В напряжения питания, 25⁰С), так как производительность НЗ схем всегда соответствует текущим условиям эксплуатации, а сами НЗ схемы не требуют учета наихудшего случая.

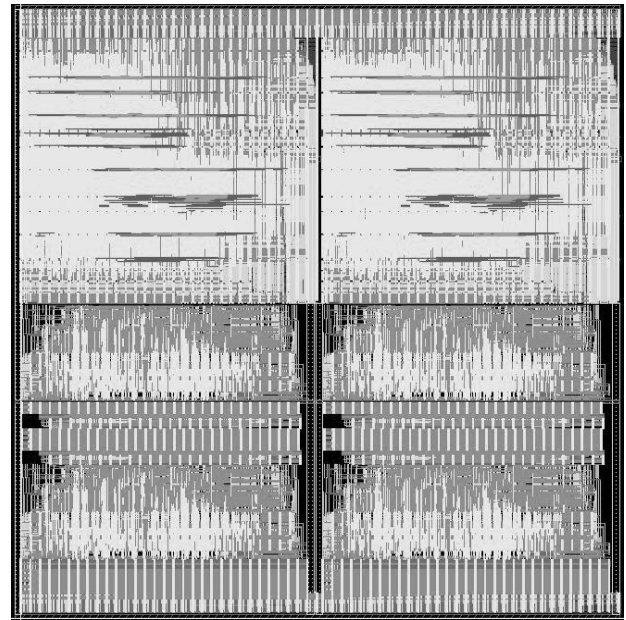


Рис. 9. Топологическая реализация SIFPC

Таблица 2

Параметры SIFPC

Наименование параметра	Аналог	SIFPC
Частота работы, ГГц	1,03	1,05
Площадь топологии, мм ²	0,312	0,94
Латентность, нс	10,8	1,84
Производительность, Гфлопс	2,06	3,15
Эффективность площади, мм ² /Гфлопс	0,151	0,298
Диапазон работоспособности по напряжению питания $V_{пит}$	$V_{пит} \pm 10\%$	$V_{пор} \dots V_{проб}$
Обнаружение константных неисправностей	–	+

Следует отметить, что SIFPC обладает большей функциональностью по сравнению с аналогом: за один цикл он способен обработать одну тройку операндов двойной точности или две тройки операндов одинарной точности, вычисляя при этом не только сумму, но и разность произведения первых двух операндов и третьего операнда. Кроме того, он имеет намного более широкий диапазон работоспособности, ограниченный лишь пороговыми напряжениями КМОП транзисторов ($V_{пор}$) и напряжением пробоя полупроводниковых структур ($V_{проб}$), и прекращает работу при обнаружении константных неисправностей [10]. Платой за

эти преимущества является большая сложность реализации и, в связи с этим, большее энергопотребление. Энергопотребление может быть снижено до требуемой величины за счет уменьшения питающего напряжения при соответствующем снижении производительности. За счет меньшего числа ступеней конвейера латентность SIFPC в 5,9 раз меньше, чем у синхронного аналога.

Таким образом, представленный вариант SIFPC обеспечивает производительность на уровне 3,15 Гфлопс за счет распараллеливания операций. Он реализует современный тренд в построении вычислительных средств высокой производительности: использование большего количества процессоров с относительно низкой производительностью, вместо повышения системной частоты каждого процессора.

Это помогает также решить проблему повышения надежности современных вычислительных систем, например, путем дублирования самопроверяющихся относительно константных неисправностей SIFPC для построения отказоустойчивых вычислителей.

V. ЗАКЛЮЧЕНИЕ

Применение НЗ схемотехники для реализации современных вычислительных средств помогает эффективно использовать аппаратные методы для контроля надёжности и достоверности результатов вычислений.

SIFPC с двумя параллельно работающими блоками FMA, разработанный по КМОП технологии с проектными нормами 65 нм, демонстрирует высокую среднюю производительность (3,15 Гфлопс при типовых условиях) и хорошую латентность (менее 2 нс).

Использование избыточного СС кодирования, упрощенной индикации и минимального числа ступеней конвейера обеспечило разработку конкурентоспособного по производительности 64-разрядного сопроцессора, реализующего FMA операцию и обладающего всеми преимуществами НЗ устройств: полной самопроверяемостью относительно константных неисправностей, сохранением работоспособности при сверхмалых значениях питающих напряжений.

ПОДДЕРЖКА

Исследование выполнено при частичной поддержке Программы фундаментальных исследований ОНИТ РАН за 2016 г. (проект 1.4) в Институте проблем информатики ФИЦ ИУ РАН.

ЛИТЕРАТУРА

- [1] R.V.K. Pillai, S.Y.A. Shah, A.J. Al-Khalili, and D. Al-Khalili, Low power floating point MAFs – A comparative study / Sixth International Symposium on Signal Processing and its Applications, Kuala Lumpur, 2001, V. 1. P. 284–287.
- [2] P.-M. Seidel, Multiple path IEEE floating-point Fused Multiply-Add / Proc. 46th IEEE International Midwest Symposium on Circuits and Systems, Cairo, Egypt, 2003. P. 1359–1362.
- [3] T. M. Bruintjes. Design of a Fused Multiply-Add Floating-Point and Integer Datapath. Master's thesis, University of Twente, Enschede, the Netherlands, 2011. 154 p.
- [4] J.R. Noche, and J.C. Araneta, An asynchronous IEEE floating-point arithmetic unit / Science Diliman, Philippines. 2007. V.19. No.2. P. 12–22.
- [5] R. Manohar, and B.R. Sheikh, Operand-optimized asynchronous floating-point units and method of use therefor, US patent, № 20130124592. May 2013.
- [6] Y. Stepchenkov, Y. Diachenko, V. Zakharov, Y. Rogdestvenski, N. Morozov, and D. Stepchenkov, Self-Timed Computing Device for High-Reliable Applications / Proc. International Workshop on power and timing modeling, optimization and simulation (PATMOS'2009), Delft, Netherlands, 2009. P. 276–285.
- [7] Соколов И.А., Степченков Ю.А., Рождественский Ю.В., Дьяченко Ю.Г. Самосинхронное устройство умножения-сложения гигафлопсного класса: методологические аспекты // Проблемы разработки перспективных микро- и нанoeлектронных систем - 2014. Сб. трудов / под общ. ред. академика РАН А.Л. Стемповского. М.: ИППМ РАН, 2014. Ч. IV. С. 51–56.
- [8] Степченков Ю.А., Рождественский Ю.В., Дьяченко Ю.Г., Морозов Н.В., Степченков Д.Ю., Сурков А.В. Самосинхронное устройство умножения-сложения гигафлопсного класса: варианты реализации // Проблемы разработки перспективных микро- и нанoeлектронных систем - 2014. Сб. трудов / под общ. ред. академика РАН А.Л. Стемповского. М.: ИППМ РАН, 2014. Ч. IV. С. 57–60.
- [9] S. Galal, and M. Horowitz, Energy-Efficient Floating-Point Unit Design // IEEE Transactions on computers. 2011. V. 60. No.7. P. 913–922.
- [10] Варшавский В.И. и др. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах. Москва: Наука, 1986. 400 с.
- [11] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Mashiko, "An 8.8-ns 54x54-bit multiplier with high speed redundant binary architecture" // IEEE Journal of Solid-State Circuits. 1996. V. 31. No. 6, pp. 773–783.
- [12] Stepchenkov Y.A., Zakharov V.N., Rogdestvenski Y.V., Diachenko Y.G., Morozov N.V., Stepchenkov D.Y. Speed-Independent Floating Point Coprocessor / IEEE East-West Design and Test Symposium, Batumi, Georgia, September 26–29, 2015. P. 111–114.
- [13] Рождественский Ю.В., Морозов Н.В., Рождественскене А.В. Подсистема событийного анализа самосинхронных схем АСПЕКТ // Проблемы разработки перспективных микро- и нанoeлектронных систем - 2010. Сб. трудов / под общ. ред. академика А.Л.Стемповского. М.:ИППМ РАН, 2010. С. 26–31.

Self-Timed Floating Point Multiply-Add Unit

Y.A. Stepchenkov, Y.V. Rogdestvenski, Y.G. Diachenko, N.V. Morozov, D.Y. Stepchenkov,
B.A. Stepanov, D.Y. Diachenko, A.V. Rogdestvenskine

Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the
Russian Academy of Sciences (IPI FRC CSC RAS), IPI RAS

{YStepchenkov, YRogdest, YDiachenko, NMorozov, DStepchenkov}@ipiran.ru

Keywords — *Wallace tree, bit-wise indication, redundant coding, ternary adder.*

ABSTRACT

The subject of this paper is a Speed-Independent Floating Point Coprocessor (SIFPC) implementing Fused Multiply-Add-Subtract operation. It utilizes mixed dual-rail and redundant self-timed coding, and is compliant with IEEE 754 Standard. SIFPC processes either one operation with double precision numbers, or two simultaneous operations with single precision numbers, and calculates two results: sum and difference between product of first two operands and third operand. SIFPC consists of two identical channels with common input and output. The order of data outputting matches the order of input data. Each channel implements full data processing path and has two pipeline stages: the first is multiplier and exponent calculation, and the second is all rest parts. This reduces hardware complexity and accelerates calculations due to reducing number of intermediate registers and cutting number of "bottlenecks" in indication subcircuit of the unit. Additional speed-up of performance, compared with traditional self-timed circuit implementation, is achieved due to utilizing bit-wise and simplified (adaptive) indication. Multiplier utilizes modified Booth algorithm with Wallace tree, self-timed redundant code and ternary adders. First stage of the Wallace tree compresses four dual-rail partial products into two ternary operands. The unit is designed for standard 65-nm CMOS process. It has 0.94 mm² die size and demonstrates 3.15 Gflops performance and 1.8 ns latency.

SUPPORT

This research was partially supported by Basic Research Program of the Nanotechnologies and Informatics Technologies Section of the RAS for 2016 year (project 1.4) in the Institute of Informatics Problems of the Federal Research Center "Computer Science and Control" of the RAS.

REFERENCES

- [1] R.V.K. Pillai, S.Y.A. Shah, A.J. Al-Khalili, and D. Al-Khalili, Low power floating point MAFs – A comparative study / Sixth International Symposium on Signal Processing and its Applications, Kuala Lumpur, 2001, V. 1. P. 284-287.
- [2] P.-M. Seidel, Multiple path IEEE floating-point Fused Multiply-Add / Proc. 46th IEEE International Midwest Symposium on Circuits and Systems, Cairo, Egypt, 2003. P. 1359–1362.
- [3] T. M. Bruintjes. Design of a Fused Multiply-Add Floating-Point and Integer Datapath. Master's thesis, University of Twente, Enschede, the Netherlands, 2011. 154 p.
- [4] J.R. Noche, and J.C. Araneta, An asynchronous IEEE floating-point arithmetic unit / Science Diliman, Philippines. 2007. V.19. No.2. P. 12–22.
- [5] R. Manohar, and B.R. Sheikh, Operand-optimized asynchronous floating-point units and method of use therefor, US patent, No. 20130124592. May 2013.
- [6] Y. Stepchenkov, Y. Diachenko, V. Zakharov, Y. Rogdestvenski, N. Morozov, and D. Stepchenkov, Self-Timed Computing Device for High-Reliable Applications / Proc. International Workshop on power and timing modeling, optimization and simulation (PATMOS'2009), Delft, Netherlands, 2009. P. 276–285.
- [7] Sokolov I.A., Stepchenkov Yu.A., Rozhdestvenskij Yu.V., Diachenko Yu.G. Speed-Independent Fused Multiply-Add Unit of Gigaflops Rating: Methodological Aspects // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2014. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2014. Part IV. P. 51-56. (In Russian).
- [8] Stepchenkov Yu.A., Rozhdestvenskij Yu.V., Diachenko Yu.G., Morozov N.V., Stepchenkov D.Yu., Surkov A.V. Speed-Independent Fused Multiply-Add Unit of Gigaflops Rating: Implementation Variants // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2014. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2014. Part IV. P. 57-60. (in Russian).
- [9] S. Galal, and M. Horowitz, Energy-Efficient Floating-Point Unit Design // IEEE Transactions on computers. 2011. V. 60. No.7. P. 913–922.
- [10] Varshvskij V.I. et al. Automatic control of the asynchronic processes in the computers and discrete systems. Moscow: Nauka, 1986. 400 p. (in Russian).
- [11] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Mashiko. An 8.8-ns 54x54-bit multiplier with high speed redundant binary architecture // IEEE Journal of Solid-State Circuits. 1996. V. 31. No. 6, pp. 773-783.
- [12] Stepchenkov Y.A., Zakharov V.N., Rogdestvenski Y.V., Diachenko Y.G., Morozov N.V., Stepchenkov D.Y. Speed-Independent Floating Point Coprocessor / Proc. IEEE East-West Design and Test Symposium, Batumi, Georgia, September 26-29, 2015. P. 111- 114.
- [13] Rozhdestvenskij Yu.V., Morozov N.V., Rozhdestvenskine A.V. ASPECT – a Subsystem of Event Analysis of Self-Timed Circuits // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2010. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2010. P. 26-31. (in Russian).