

# Метод оптимизации вычислительного процесса на реконфигурируемых вычислительных средах

Д.С. Артамонов, М.Г. Путря

**Аннотация.** Рассматривается метод повышения эффективности вычислений в реконфигурируемых вычислительных средах – технология альтернативных вычислений. Процессорный элемент вычислительной системы может выполнять прямую или альтернативную функцию из существующего набора функций, в зависимости от сигнала условия. Данное свойство позволяет в масштабах массива процессорных элементов значительно повысить удельную производительность всей системы в целом за счет экономии аппаратных ресурсов.

**Ключевые слова:** реконфигурируемые вычислительные системы, однородные вычислительные среды, системная вычислительная архитектура, альтернативные вычисления, процессорный элемент.

## Введение

С начала 80-ых годов в мире вычислительной техники развивается направление систолических волновых вычислительных структур или однородных вычислительных сред, предназначенных для потоковой обработки данных, в том числе в режиме реального времени [1,2]. В настоящее время это направление получило дополнительные возможности развития, благодаря совершенствованию технологий производства микрочипов, позволяющих значительно повысить степень интеграции – до сотен миллионов логических вентилей на одном чипе. Возникает новый класс вычислительных систем – реконфигурируемые вычислительные системы на однородных вычислительных средах.

Реконфигурируемая вычислительная система, или РВС, представляет собой масштабируемую матрицу однотипных вычислительных элементов (процессорных элементов, ПЭ), выполненную на системе-на-кристалле (System on Chip, SoC). Путем перепрограммирования ПЭ матрицы возможно создавать специализи-

рованные вычислительные устройства. В основе парадигмы данной вычислительной архитектуры лежит принцип управления вычислениями потоками данных, а не потоком команд – каждый вычислительный элемент системы предварительно конфигурируется на выполнение конкретной функции в составе заданного алгоритма, затем, при условии наличия данных на входах процессорного элемента, осуществляется обработка данных в соответствии с заданным функциональным оператором. Процесс вычислений для данных систем организуется как во времени, так и в пространстве.

В отличие от традиционных вычислительных систем, которые управляются программным счетчиком, функционирование вычислительных систем, управляемых потоками данных, основано на том, что функциональные операторы, у которых определены все аргументы, могут вычисляться независимо друг от друга, поэтому возможности параллельного вычисления функциональных операторов, заложенные в алгоритме решения задачи, могут быть реализованы естественным образом.

Реконфигурируемая вычислительная система включает в себя:

- масштабируемый двумерный массив процессорных элементов;
- буферы входных и выходных данных;
- устройство управления.

Вычислительным ядром РВС является двумерный массив идентичных процессорных элементов, в котором каждый процессорный элемент соединен с каждым из своих четырех соседних процессорных элементов четырьмя однонаправленными связями – двумя входными и двумя выходными. Таким образом, каждый процессорный элемент имеет восемь входных связей и восемь выходных. Конфигурация процессорного элемента включает в себя определение функциональных операторов, которые должны вычисляться процессорным элементом в соответствии с алгоритмом решения задачи, и коммутацию соединений с соседними процессорными элементами.

Периферийные процессорные элементы имеют связи с буферами данных, которые предназначены для загрузки данных из внешней памяти и их синхронной передачи в процессорные элементы матрицы, а также для приема результатов обработки данных из процессорных элементов матрицы и их выгрузки во внешнюю память.

Устройство управления осуществляет конфигурацию двумерного массива процессорных элементов и буферов входных/выходных данных, загрузку исходных данных из внешней памяти в буферы данных и выгрузку результатов обработки из буферов во внешнюю память.

РВС может функционировать в следующих режимах:

- конфигурация двумерного массива процессорных элементов и буферов данных;
- выполнение массивом процессорных элементов потоковой обработки данных;
- приостановка выполнения массивом процессорных элементов потоковой обработки данных.

Набор уникальных свойств РВС определяет высокую эффективность их использования в областях и приложениях, связанных с высокопараллельными критическими вычислениями.

Высокая производительность данных систем достигается за счет снижения избыточности при организации управления, а также за счет эффективной реализации алгоритма в аппаратуре с учетом его естественной степени распараллеливания. Свойство высокой регулярности и масштабируемости обеспечивает быстрое и простое наращивание вычислительной мощности в зависимости от приложения, вплоть до уровня суперкомпьютера. Свойство высокой регулярности определяет относительную простоту и низкую стоимость разработки, а также низкую себестоимость изготовления вычислительной системы. Свойство реконфигурируемости аппаратуры обеспечивает её динамическую адаптацию относительно расширяемого диапазона классов обрабатываемых алгоритмов. В основе парадигмы данной вычислительной архитектуры лежит принцип управления вычислениями потоками данных.

Реконфигурируемые вычислительные структуры предназначены для обработки потоков данных по реализованному в аппаратуре прикладному алгоритму. При этом ПЭ могут выполнять фиксированную обработку, что приводит к реализации в аппаратуре СБИС конкретного алгоритма, а могут быть реконфигурируемыми. В последнем случае возникает возможность реализации набора различных прикладных алгоритмов путем настройки ПЭ на выполнение одного из функциональных операторов из реализованного набора. Таким образом, в реконфигурируемых однородных вычислительных средах реализация конкретного прикладного алгоритма выполняется путем настройки ПЭ матрицы на выполнение необходимых функциональных операторов.

Наиболее эффективно потоковые вычислительные системы могут использоваться для обработки данных в режиме реального времени – обработки сигналов и изображений, потоков информации с датчиков, вычисления наборов математических функций, решения систем уравнений и т.д. Относительно данных приложений подобные вычислительные системы позволяют достигать производительности уровня суперкомпьютеров при значительно меньших размерах, потребляемой мощности и стоимости.

В статье рассматривается метод повышения удельной производительности (показателя производительности на единицу площади) параллельной систолической вычислительной реконфигурируемой архитектуры – введение поддержки альтернативных вычислений.

В числе реализованных в мире систем можно отметить «пионерские» проекты iWarp [3] и GAPP [4], в ходе реализации которых были подтверждены высокие характеристики данной вычислительной архитектуры и перспективность ее дальнейшего развития.

В СССР и России работы по развитию данного направления выполнялись в ряде организаций – в НПО "Астрофизика"; «Суперкомпьютерные системы», Москва; ИМаш УрО РАН и НПО автоматики, Екатеринбург, Москва; в Физико-механическом институте АН СССР, Львов; в ПО "Альфа", Рига, [5-9].

Распределение аппаратных ресурсов матрицы процессорных элементов перечисленных выше систем относительно обработки алгоритмов, включающих ветвление (большинства алгоритмов высокопроизводительных вычислений), является неэффективным, что, в общем случае, снижает показатель удельной производительности. Задача предлагаемой работы заключается в повышении эффективности реконфигурируемых вычислительных систем относительно обработки алгоритмов, включающих ветвления.

## Оптимизация аппаратных ресурсов РВС

В общем случае реконфигурируемая однородная вычислительная среда представляет собой матрицу процессорных элементов (ячеек) с межпроцессорными связями между ближайшими ячейками.

При этом каждая ячейка матрицы может выполнять заданный набор функций из заданного множества возможных функций над операндами, поступающими на заданные входы из множества всех входов ячейки и выдавать выходные данные на заданные выходы из множества всех выходов ячейки. Функции ячейки могут включать, в том числе, функции обработки данных – логические и арифметические функциональные операторы над одним, двумя, тре-

мя или более входными операндами, функции хранения входных, внутренних и выходных данных и функции передачи входных данных с заданных входов ячейки на заданные выходы – функции транзита данных.

Для реализации конкретного вычислительного алгоритма в необходимом для этого подмножестве ячеек матрицы однородной вычислительной среды настраиваются межпроцессорные связи и необходимые функции используемых ячеек. Заданный набор функций из множества возможных, используемый в конкретной ячейке для реализации прикладного алгоритма, определяется ее настройкой, битовые поля которой управляют конкретным аппаратным ресурсом ячейки – входным коммутатором выбора входов ячейки для приема входных операндов, блоком обработки данных для задания функционального оператора, блоком хранения данных, выходным коммутатором для выбора выходов ячейки, на которые передаются результаты вычислений, и т.д. Для настройки ячейки на выполнение заданного набора функций используется регистр настройки, в который в процессе конфигурирования матрицы однородной вычислительной среды загружается соответствующий код настройки ячейки. Таким образом, посредством загрузки соответствующих настроек в соответствующее подмножество (множество) ячеек достигается конфигурирование однородной вычислительной среды на выполнение заданного прикладного алгоритма. Для реализации вычислений по другому прикладному алгоритму должны быть загружены другие настройки, в общем случае, в другое подмножество ячеек матрицы, что обеспечивает реконфигурирование вычислительной системы.

На Рис. 1 приведена обобщенная структурная схема исходной ячейки реконфигурируемой однородной вычислительной среды, содержащая, в том числе, входной коммутатор, блок обработки данных, выходной коммутатор и регистр кода настройки.

Поля управления кода настройки представляют собой двоичные коды требуемой размерности для управления соответствующими аппаратными ресурсами ячейки, включая представленные блоки ячейки, а также любые другие – например, блоки памяти, каналы транзи-

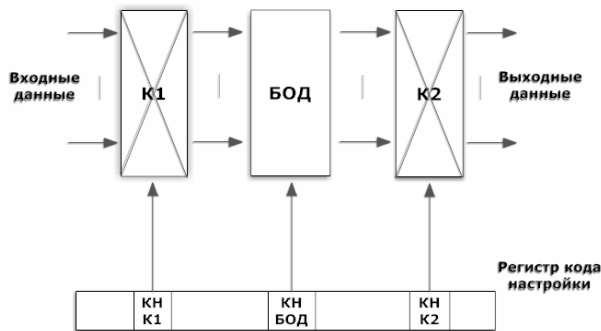


Рис. 1. Обобщенная структурная схема исходной ячейки реконфигурируемой однородной вычислительной среды

та, устройство управления ячейкой и т.д. Каждый бит кода настройки хранится в отдельном триггере регистра кода настройки.

Блок обработки данных ячейки (БОД) может быть как простым устройством, например, арифметико-логическим (АЛУ), так и составным и иерархическим и включать множество блоков обработки данных – АЛУ, умножитель, делитель и т.д., каждый из которых может быть сконфигурирован на выполнение одного функционального оператора из заданного множества определенных для данного блока возможных функциональных операторов.

Для простоты дальнейшего изложения будем считать, что БОД является простым блоком, выполняющим один функциональный оператор из множества определенных для него операторов, в зависимости от соответствующего поля кода настройки ячейки.

Каждая ячейка однородной вычислительной среды выполняет функцию преобразования данных – операндов, последовательно поступающих на внешние входы по одному разряду за один такт сигнала синхронизации в течение определенного количества тактов сигнала синхронизации. Число тактов обработки определяется выполняемым функциональным оператором и форматом обрабатываемых данных, который является переменным (от 1 до  $N$  разрядов). Между поступающими на входы операндами могут быть паузы в  $P$  тактов сигналов синхронизации.

Поле кода настройки КН К1, управляющее входным коммутатором К1, задает адреса источников входных операндов для функционального оператора, заданного в БОД соответ-

ствующим полем кода настройки КН БОД. Поле кода настройки КН К2, управляющее выходным коммутатором К2, задает адреса внешних выходов ячейки, на которые будут выдаваться результаты вычислений из БОД последовательно, по одному разряду за один такт сигнала синхронизации.

Таким образом, на данной ячейке выполняется обработка потока данных, поступающих на ее входы до тех пор, пока не закончится массив входных данных, или пока выполняемый алгоритм не достигнет заданного состояния, или пока ячейка не будет перенастроена на выполнение другого функционального оператора над данными путем загрузки нового кода настройки в регистр кода настройки.

Известно [10], что для реализации любого прикладного вычислительного алгоритма достаточно использовать ограниченный набор вычислительных структур, составленных из простейших функциональных операторов и включающий «конкатенацию», «выбор» и «повторение».

«Конкатенация» представляет собой последовательное выполнение в заданном порядке заданного набора функциональных операторов над входными операндами.

«Выбор» представляет собой вычислительную структуру, в которой только один из множества возможных операторов выполняется в зависимости от состояния программы или какого-то результата промежуточных вычислений – условия. В частности, к «выбору» относится вычислительная структура типа IF <условие> THEN S1 ELSE S2. «Условие» в данном случае есть однобитовая булева переменная, принимающая значения TRUE и FALSE. При этом если ее значение TRUE, то вычислительная структура выполняет вычисления в соответствии с функциональным оператором S1, в противном случае – S2. Вычисления S1, S2 могут быть как простыми функциональными операторами, так и сложными композитными или иерархическими вычислительными структурами, включающими структуры «конкатенации», «выбора» или «повторения».

«Повторение» обозначает выполнение вычислительной структуры или, в пределе, одного заданного функционального оператора, пока не

будет достигнуто определенное состояние программы – прекращение вычислений по условию - или пока он не будет применен ко всем элементам массива входных операндов.

Известные вычислительные системы, построенные с использованием архитектуры реконфигурируемых однородных вычислительных сред [5, 7], обладают существенным недостатком, а именно – высокими аппаратными затратами, в случае реализации на них прикладных алгоритмов, в которых широко используется вычислительная структура с ветвлением по условию – структура типа «выбор» (практически полный набор типовых алгоритмов высокопроизводительных вычислений – поиск слов в последовательности, поиск кода в битовом потоке, поиск минимального элемента в массиве, ограничение массива, произведение матриц, LU-декомпозиция (Алгоритм Краута), транспонирование матриц, скалярное произведение векторов, обработка сигналов и изображений, свертка, интерполяция, дискретное косинусное преобразование, линейная фильтрация, нелинейная фильтрация (медианный фильтр), быстрое преобразование Фурье (БПФ), полиномиальная алгебра, вычисление полинома по схеме Горнера, помехоустойчивое кодирование, кодер Рида-Соломона и др.).

При реализации в матрице однородной вычислительной среды на основе рассмотренной ячейки вычислительной структуры типа «выбор» -  $IF\ C\ THEN\ S1\ ELSE\ S2$ , где  $C$  – однобитовая булева переменная условия ветвления,  $S1$  и  $S2$  – альтернативные функциональные операторы, должно быть использовано как минимум 4 ячейки в соответствии с Рис. 2 (на рисунке показаны только используемые межпроцессорные связи).

Допустим, что каждая ячейка имеет по два входа и по два выхода на каждую сторону для связи со смежными ячейками и имеет как минимум один транзитный канал. Тогда для выполнения указанной вычислительной структуры ячейка Я2 должна быть настроена для обработки потоков данных  $A$  и  $B$  с использованием функционального оператора  $S1$ , а ячейка Я3 должна быть настроена для выполнения обработки потоков данных  $E$  и  $D$  с использованием функционального оператора  $S2$ . Ячейка Я1,

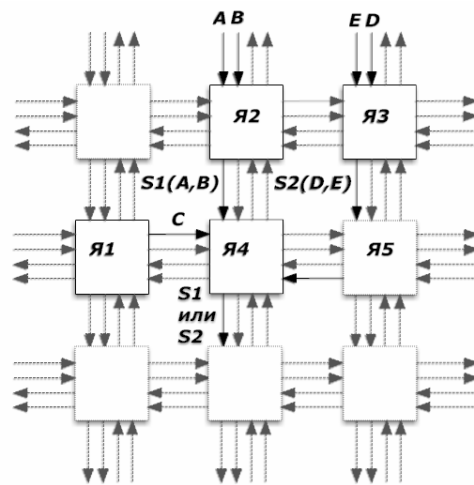


Рис. 2. Конфигурация однородной вычислительной среды для реализации вычислительной структуры  $IF..THEN..ELSE$  на основе исходной ячейки

Активные ячейки и входы-выходы обозначены жирными линиями, неактивные – пунктиром

исходя из определяемого в вычислительном алгоритме условия, формирует сигнал булевой переменной условия  $C$ , который определяет выдачу окончательного результата на выходе ячейки Я4 из двух входных результатов  $S1(A,B)$  и  $S2(E,D)$ . В ячейке Я5 используется только канал транзита для передачи результата  $S2(E,D)$  на вход ячейки Я4.

Таким образом, при реализации прикладных вычислительных алгоритмов, в которых широко используется вычислительная структура  $IF..THEN..ELSE$ , большое количество вычислительных ресурсов матрицы – ячеек, будет использовано для реализации альтернативных вычислительных структур и альтернативных функциональных операторов, что в общем случае снижает удельную производительность вычислительной матрицы и, соответственно, эффективность архитектуры в целом.

Эффективность вычислений в реконфигурируемых однородных вычислительных средах определяется, с одной стороны, возможностями распараллеливания прикладных вычислительных алгоритмов, с другой стороны – функциональными возможностями ячеек среды и размерностью матрицы, реализуемой в отдельном микрочипе заданной площади.

Таким образом, для отдельно взятого алгоритма существует возможность максимального

распараллеливания вычислений, ограничиваемая размерностью матрицы и функциональными возможностями ячеек, что выражается в количестве ячеек, требуемых для реализации заданного алгоритма.

Уменьшение количества ячеек, требуемых для реализации прикладного алгоритма при заданной степени распараллеливания, открывает возможности для дополнительного распараллеливания вычислений по алгоритму в рамках матрицы заданной размерности, что обеспечивает повышение производительности вычислительной системы относительно вычислений по данному алгоритму. Это, в свою очередь, определяет повышение параллелизма на уровне выполнения одной вычислительной задачи.

Уменьшение количества ячеек для реализации различных алгоритмов открывает возможности использования одной матрицы однородной вычислительной среды для одновременного выполнения нескольких прикладных алгоритмов, что также повышает производительность вычислительной системы относительно заданного множества вычисляемых прикладных алгоритмов. Это обеспечивает повышение параллелизма на уровне вычислительных задач, выполняемых в вычислительной системе.

Для уменьшения количества ячеек, требуемых для реализации прикладного алгоритма, могут использоваться различные способы – увеличение функциональных возможностей ячейки, снижение количества ячеек, требуемых для реализации различных вычислительных структур типа «повторение», «выбор», «конкатенация», развитие коммутационных возможностей матрицы путем увеличения количества подсоединенных к ячейке соседних ячеек, введения глобальных связей и т.д.

Для повышения эффективности вычислений в реконфигурируемых однородных вычислительных средах предлагается, в отличие от рассмотренной выше обобщенной ячейки однородной вычислительной среды, которая обеспечивает обработку потоков входных операндов в соответствии с единственным настроенным функциональным оператором, реализовать в каждой ячейке реконфигурируемой однородной вычислительной среды альтернативные функциональные операторы. Это по-

зволяет уменьшить количество ячеек, используемых для реализации вычислительных структур типа «выбор», поскольку в зависимости от состояния булевой переменной возможно выполнение вычисления либо одного, либо другого функционального оператора в одной ячейке вычислительной среды, а не в двух.

Для реализации ветвей альтернативных вычислений с использованием более сложных вычислительных структур, кроме альтернативных функциональных операторов, предложено также использовать альтернативные внешние входы для приема альтернативных входных операндов для оператора  $S1$  и оператора  $S2$  и альтернативные внешние выходы, задающие альтернативные направления выдачи результатов вычислений по оператору  $S1$  или по оператору  $S2$ .

С целью повышения эффективности вычислений, во-первых, расширяется код настройки ячейки и регистр кода настройки ячейки для ее загрузки и хранения на время выполнения конкретного прикладного вычислительного алгоритма путем введения дополнительных альтернативных полей управления соответствующими аппаратными ресурсами ячейки – входным коммутатором, блока обработки данных и выходным коммутатором.

Во-вторых, вводятся коммутаторы альтернативных кодов настройки для управления соответствующими аппаратными ресурсами ячейки, управляемые сигналом булевой переменной условия.

В-третьих, вводится коммутатор выбора сигнала булевой переменной условия из множества возможных, включая сигналы, поступающие на все внешние входы ячейки и возможные внутренние сигналы ячейки и константы «0» и «1».

В-четвертых, для управления коммутатором выбора сигнала булевой переменной условия в код настройки ячейки вводится еще одно дополнительное поле.

Как видно из рисунка, мультиплексор МУ, на входы которого поступают внешние входы ячейки и, возможно, внутренние сигналы, управляется полем кода настройки КН МУ, который задает адрес источника сигнала переменной условия. При этом значение «1» сигнала переменной условия обеспечивает подачу

полей кода настройки ячейки – КН К11, КН БОД1, КН К21 на соответствующие блоки ячейки: входной коммутатор, блок обработки данных и выходной коммутатор, а значение «0» - подачу альтернативных полей КН К12, КН БОД2, КН К22.

Дополнительно могут быть использованы несколько сигналов условия, которые обеспечивают независимое управление мультиплексорами М1-М3, увеличивая гибкость использования данного способа для повышения эффективности вычислений. При этом в код настройки должны быть введены дополнительные поля для управления мультиплексорами задания источников сигналов условия.

Обобщенная структурная схема модифицированной ячейки приведена на Рис. 3.

При использовании модифицированной ячейки вычислительная структура типа «выбор» может быть реализована согласно Рис. 4 (на рисунке изображены только используемые межпроцессорные связи).

### Результаты исследований

Оценочные характеристики, полученные в процессе реализации логического синтеза схемы процессорного элемента вычислительной матрицы в программном средстве Xilinx PlanAhead 11, показали, что площадь модифицированного ПЭ превосходит на 10% площадь базовой ячейки. Это означает, что на одной и той же площади кристалла в базовом варианте можно было бы разместить вычислительную матрицу с количеством ячеек, равным, например, 1000, а в модифицированном - равным 909 ячеек.

В процессе реализации алгоритмов на базовой и модифицированной вычислительных архитектурах были получены показатели соотношения количества используемых для реализации конкретного алгоритма вычислительных ячеек на обеих архитектурах (Рис. 5).

Среднее значение показателя, определяющего отношение используемых ячеек при реализации типовых вычислительных алгоритмов в исходной и модифицированной архитектурах, составляет 66%. Это означает, что для размещения на модифицированной аппаратуре произвольного набора алгоритмов, требующего для реализации на базовой архитектуре 1000

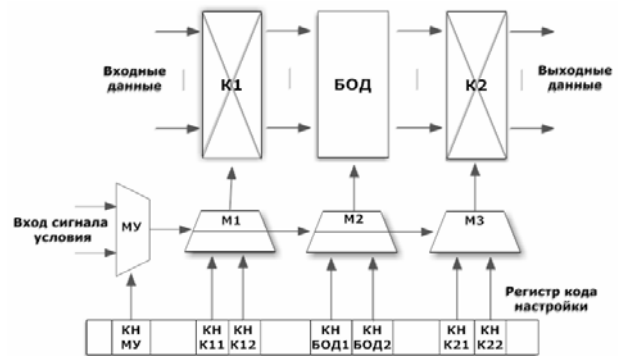


Рис. 3. Обобщенная структурная схема модифицированной ячейки реконфигурируемой однородной вычислительной среды

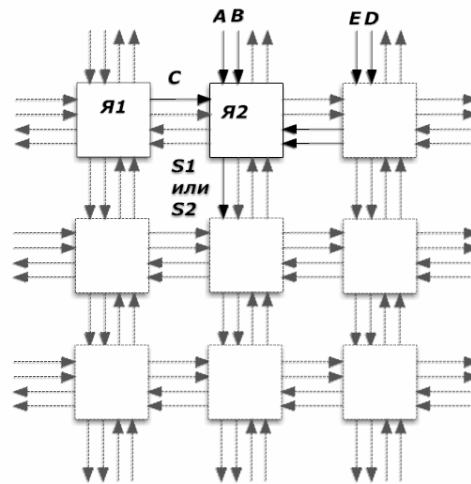


Рис. 4. Конфигурация однородной вычислительной среды для реализации вычислительной структуры IF..THEN..ELSE на основе модифицированной ячейки. Активные ячейки и входы-выходы обозначены жирными линиями, неактивные – пунктиром

ячеек, потребовалось бы 660 ячеек. Выигрыш в аппаратных затратах (900-660 = 240 ячеек) в масштабах матрицы вычислительных элементов очевиден.

Следует отметить, что значение показателя соотношения используемых аппаратных ресурсов в базовом и модифицированном исполнении является усредненным и может варьироваться в зависимости от типа реализуемых алгоритмов (фактически, от решаемой прикладной задачи) от 50 до 88%.

Таким образом, вычислительная ячейка матрицы может выполнять прямую или альтернативную функцию из существующего набора функций в зависимости от сигнала условия.

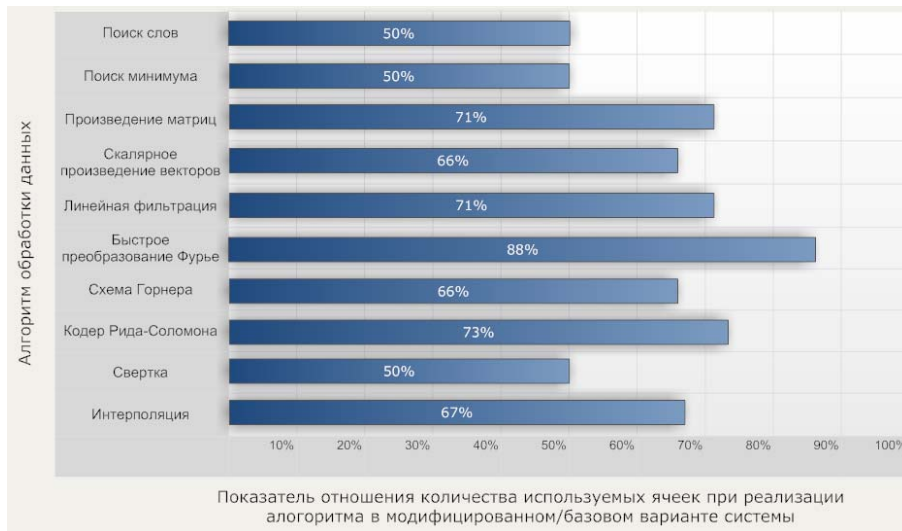


Рис. 5. Отношение количества используемых для реализации набора типовых алгоритмов вычислительных ячеек на базовой и модифицированной архитектурах

Введение поддержки альтернативных вычислений позволяет снизить аппаратные издержки относительно обрабатываемых алгоритмов, включающих ветвления, таких как поиск совпадения в битовой последовательности, скалярное произведение векторов, произведение матриц, транспонирование матрицы, дискретное косинусное преобразование, линейная фильтрация и др. (Рис. 2 и Рис. 4). В масштабах матрицы процессорных элементов такой выигрыш в аппаратных ресурсах позволяет значительно повысить удельную производительность всей системы в целом.

**Литература**

1. Kung H. T., Leiserson C. E.: Algorithms for VLSI processor arrays / Eds. in C. Mead, L. Conway. - Addison-Wesley: Introduction to VLSI Systems, 1979. - 273 p.
2. Kung S. Y. VLSI Array Processors. - Prentice-Hall, Inc., 1988. - 600 p.
3. S. An Integrated Solution to High-Speed Parallel Computing/Borkar S., Cohn R., Cox G. et al. // Proc. Supercom-

- puting '88. Kissimmee (Florida, November 14-18, 1988). - P. 330 - 339.
4. Никонов В.В., Подгурский Ю.Е. БИС ГАПП и ее применение // Зарубежная радиоэлектроника. - М., Радио и связь, 1987. №7. С 71-83
5. Шмойлов В.И., Адамацкий А.И., Кузьо М.Н., Русин Б.П., Тимченко А.В. Пульсирующие информационные решетки. Львов: Меркатор, 2004. 302 с.
6. Богачев. М.П. Архитектура вычислительной системы с однородной структурой. В кн.: Однородные вычислительные среды. - Львов: ФМИ АН УССР. 1981. - 126 с.
7. Шмойлов В.И., Русин Б.П., Кузьо М.Н. Ячейка пульсирующих информационных решёток. - Львов: Меркатор, 2001. - 34 с.
8. И.А. Каляев, И.И. Левин,Е.А. Семерников, В.И. Шмойлов. Реконфигурируемые мультиконвейерные вычислительные структуры. - Монография. Ростов-на-Дону: ЮНЦ РАН. 2008 - 393 с.
9. Палагин А.В., Опанасенко В.Н. Реконфигурируемые вычислительные системы. К.: Просвіта, 2006. 288 с.
10. Edsger W. Dijkstra. Notes on structured programming / Eds. Ole-Johan Dahl, Edsger W. Dijkstra, C. A. R. Hoare. - Structured Programming. Academic Press, 1972. - 88 p.

**Артамонов Дмитрий Сергеевич.** Аспирант кафедры Интегральной электроники и микросистем, МИЭТ. Окончил Московский институт электронной техники (Технический университет) в 2007 году. Область научных интересов: разработки в области параллельных архитектур и распределенных вычислений, исследования в области построения высокопроизводительных многопроцессорных и реконфигурируемых вычислительных архитектур. E-mail: [dmitry.artamonov@idm.ru](mailto:dmitry.artamonov@idm.ru).

**Путря Михаил Георгиевич.** Профессор кафедры Интегральной электроники и микросистем, декан факультета Электроники и компьютерных технологий Московского государственного института электронной техники, доктор технических наук, заслуженный работник высшего профессионального образования. Окончил МИЭТ в 1978 году. Автор более 80 научных работ, 15 патентов. Область научных интересов: процессы и маршруты изготовления УБИС, исследование плазменных методов формирования трехмерных структур УБИС и микросистем, методология разработки новых технологий, инновационные методы подготовки разработчиков элементной компонентной базы. E-mail: [pmg@micee.ru](mailto:pmg@micee.ru).