

на правах рукописи

Шалтырев Владимир Алексеевич

**СРЕДСТВА И МЕТОДЫ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ И
СНИЖЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЯ СИСТЕМ НА КРИСТАЛЛЕ,
РЕАЛИЗУЕМЫХ НА БАЗЕ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ
ИНТЕГРАЛЬНЫХ СХЕМ**

**Специальность 05.13.05 – “Элементы и устройства вычислительной техники и
систем управления”**

**Автореферат
диссертации на соискание ученой степени кандидата
технических наук**

Автор:

Москва, 2009

Диссертация выполнена в Национальном исследовательском ядерном университете “МИФИ”.

Научный руководитель:

доктор технических наук,
профессор

Шагурина Игорь Иванович

Официальные оппоненты:

доктор технических наук,
профессор

Сахин Юлий Хананович

кандидат технических наук

Осипенко Павел Николаевич

Ведущая организация: Федеральное государственное унитарное предприятие
“Научно-исследовательский институт “КВАНТ”

Защита диссертации состоится 23 ноября 2009 года в 15-00 на заседании диссертационного совета Д 212.130.02 в конференц-зале Национального исследовательского ядерного университета “МИФИ” по адресу: 115409, Москва, Каширское шоссе, 31. Телефоны: 324-84-98, 323-91-67.

С диссертацией можно ознакомиться в библиотеке Национального исследовательского ядерного университета “МИФИ”.

Автореферат разослан 20 октября 2009 года.

Ученый секретарь
диссертационного совета,
доктор технических наук,
профессор

Скоробогатов П.К.

Общая характеристика работы

Актуальность диссертации

Развитие современных электронных систем управления, сбора и обработки данных, их постоянно растущие сложность и ресурсоемкость, а также необходимость сокращения сроков разработки требуют создания новых классов функционально сложных изделий микроэлектроники и внедрения принципиально новых технологических принципов разработки микроэлектронных устройств. При этом сокращение временных затрат на изготовление, верификацию и вывод изделия на рынок не должно влиять на качество разрабатываемых интегральных схем (ИС) и их надежность.

До недавнего времени основным средством решения данной проблемы являлось применение специализированных заказных ИС (Application Specific Integrated Circuits, ASIC). К недостаткам проектов, реализованных на ASIC, следует отнести длительное время их разработки и высокую стоимость. Подготовка к производству каждого нового варианта ASIC влечет за собой расходы в десятки и сотни тысяч долларов, что делает малосерийный выпуск ASIC экономически невыгодным и существенно повышает цену ошибки проектирования.

Принципиально новой стала методология проектирования “систем на кристалле” (СнК), допускающая многократное использование готовых, предварительно протестированных сложнофункциональных блоков (СФ-блоки, используются также термины IP-ядро, IP-блок – от “Intellectual Property”). Эта методология обеспечила значительное повышение производительности проектирования.

В качестве технологической платформы для реализации цифровых СнК могут использоваться программируемые логические интегральные схемы (ПЛИС) – их применение позволяет значительно сократить время разработки и обеспечивает возможности оперативной модификации СнК. Такая технологическая платформа позволяет в полной мере использовать как характерные особенности самих ПЛИС (например, технику динамической реконфигурации), так и преимущества IP-ядер класса “soft-core”, то есть ядер, не привязанных к какой-либо технологической платформе и описанных на высокоуровневых языках описания аппаратуры (HDL). На базе современных ПЛИС могут быть реализованы все цифровые функциональные блоки системы, то есть может быть получена полная структура системы за исключением аналоговых устройств. Разработка СнК на основе ПЛИС (System-on-Programmable-Chip, SoPC, СнПК) предполагает использование программируемых кристаллов, конфигурирование которых позволяет создавать достаточно сложные цифровые устройства. При этом экономическая эффективность реализации малых серий и единичных опытных образцов СнПК резко возрастает по сравнению с изготовлением заказных микросхем. Однако данная технологическая платформа имеет и свои недостатки: высокое энергопотребление, неполное использование ресурсов базового кристалла ПЛИС. Таким образом, становится актуальной задача повышения эффективности использования ПЛИС в качестве технологической платформы для создания СнПК.

Целью диссертации является разработка средств и методов повышения производительности и снижения энергопотребления СнК, реализуемых на базе ПЛИС.

В диссертации решаются следующие основные задачи:

1. Анализ влияния сокращения набора поддерживаемых команд на основные параметры синтезируемого процессорного ядра при сохранении совместимости с существующими средствами разработки программного обеспечения.

2. Разработка методики комплексной оценки параметров синтезируемого процессорного ядра (ресурсоемкость, производительность, энергопотребление) для сравнительного анализа различных ядер.

3. Анализ эффективности использования параллельного включения сопроцессоров в структуру проектируемой СнК для повышения скорости обработки данных, разработка методики определения эффективности параллельного включения специализированных сопроцессоров.

4. Разработка структуры и RTL-модели блока потоковой обработки данных, перспективного для использования в качестве сопроцессора в составе СнПК.

5. Разработка методики оценки эффективности динамической реконфигурации СнПК.

6. Разработка алгоритма компрессии конфигурационных файлов, допускающего аппаратную реализацию декомпрессии при минимальном использовании ресурсов ПЛИС, который обеспечит эффективную динамическую реконфигурацию структуры СнПК.

7. Разработка с использованием предложенных в диссертационной работе методов и средств СнПК, предназначенных для использования в составе систем управления и сбора и обработки данных.

Научная новизна диссертации

1. Предложена методика модификации синтезируемых процессорных ядер, основанная на исключении из набора команд инструкций, не используемых при реализации конкретного приложения. В подавляющем большинстве случаев данная методика позволяет сократить количество ресурсов ПЛИС, необходимых для реализации процессорного ядра, повысить рабочую тактовую частоту ядра, а также снизить энергопотребление системы.

2. Предложена методика оценки эффективности модификации синтезируемых процессорных ядер:

А) введена статистическая обработка экспериментальных данных для исключения влияния конкретного размещения на кристалле;

Б) доказано наличие корреляции между параметрами синтезируемых структур, реализованных на кристаллах ПЛИС с одинаковой архитектурой – данный факт позволяет проводить экстраполяцию результатов, полученных для одного кристалла ПЛИС, на другие кристаллы этого или аналогичных семейств;

В) введен набор оценочных функций, позволяющих проводить сравнительный анализ процессорных ядер по различным параметрам.

3. Предложена структура многоканальной системы обработки информации, обеспечивающая повышение производительности системы за счет использования естественного параллелизма вычислений. В отличие от существующих способов организации параллельной обработки данных, предложенный сопроцессор обладает единым интерфейсом при любом количестве параллельно включенных обработчиков. Разработаны упрощенная математическая и RTL-модели системы, позволяющие на этапе проектирования оценить эффективность системы при известном времени обработки пакета данных; изучено влияние дисперсии времени обработки пакета данных на эффективность системы.

4. Предложена методика оценки эффективности динамической реконфигурации по сравнению с классической СнПК, позволяющая на этапе разработки оценить потенциальный выигрыш в удельной производительности при использовании динамической реконфигурации и определить ресурсоемкость сопроцессора, для которого реконфигурация будет эффективной.

5. На основе проведенного анализа структуры конфигурационных данных ПЛИС предложен модифицированный алгоритм иерархического сжатия конфигурационных файлов, допускающий аппаратную реализацию без использования блочной памяти кристалла.

Практическая значимость диссертации

1. Разработан и реализован маршрут проектирования, обеспечивающий эффективное применение предложенной методики модификации процессорных ядер. На примере набора из 40 процессорных ядер и 20 тестовых приложений показано, что предложенная методика модификации способна обеспечивать сокращение ресурсоемкости синтезируемого процессорного ядра на величину до 45%, увеличение рабочей тактовой частоты на величину до 13%, сокращение динамической потребляемой мощности на величину до 8%, увеличение “вычислительной плотности” синтезируемых структур на величину до 100%.

2. Разработаны RTL-модели синхронной и локально-синхронной многоканальных систем обработки данных. Показано, что при равномерном, нормальном и пуссоновском распределениях времени обработки пакета данных эффективность системы может быть повышена за счет увеличения размера буфера тэгов до $2N$ слов. Проведен сравнительный анализ эффективности синхронной и локально-синхронной схем, на его основе сформулированы рекомендации по выбору одного из вариантов многоканальной схемы при различных временах обработки пакета данных.

3. На примере набора из 20 различных IP-блоков показано, что применение предложенного алгоритма компрессии конфигурационных данных ПЛИС обеспечивает сокращение объема конфигурационных данных в 1,8..10,9 раз. Определен диапазон параметров алгоритма, обеспечивающих максимальную степень компрессии конфигурационных данных.

4. Разработано программное обеспечение для ПК, реализующее предложенный алгоритм иерархической компрессии.

5. Разработана RTL-модель блока аппаратного декомпрессора, определены параметры данного блока при его реализации на ПЛИС семейства Virtex4. Показано, что предложенный алгоритм декомпрессии допускает аппаратную реализацию без использования блочной памяти в отличие от других алгоритмов, используемых для компрессии конфигурационных данных ПЛИС.

Внедрение результатов диссертации

1. В Малой ускорительной лаборатории МИФИ (г. Москва) разработан, испытан и внедрен в научную и учебную деятельность лаборатории аппаратно-программный комплекс СПЕКТР, построенный на базе ПЛИС Spartan3AN. Использование предложенной методики модификации процессорных ядер в ходе проектирования СнПК, лежащей в основе аппаратной части комплекса, позволило повысить точность проводимых измерений в среднем на 12% и снизить себестоимость аппаратной части комплекса на 20%.

2. При разработке вычислительных блоков 5П4ИИ01, 674ИИ01 с унифицированными ячейками обработки сигналов 5П2ХЛ002, 672ХЛ002 на базе ПЛИС для радиолокационных станций 5П-27, 67Н6Е, 1РЛ-123Е (ВНИИРТ, г. Москва) использованы предложенные в диссертации методика оценки эффективности динамической реконфигурации и IP-ядро декомпрессора конфигурационных файлов, включенное в состав блоков. Применение алгоритма компрессии позволило сократить в 2..8 раз объем памяти, необходимой для хранения конфигурационных данных ПЛИС.

Положения, выносимые на защиту

1. Методика модификации синтезируемых процессорных ядер, основанная на исключении из набора команд инструкций, не используемых при реализации конкретного приложения.

2. Методика оценки эффективности модификации синтезируемых процессорных ядер, позволяющая проводить сравнительный анализ синтезируемых процессорных ядер с учетом таких параметров, как производительность, ресурсоемкость, энергопотребление.

3. Структура многоканальной системы обработки информации, обеспечивающая повышение производительности системы за счет использования естественного параллелизма вычислений.

4. Методика оценки эффективности динамической реконфигурации по сравнению с классической СнПК, позволяющая на этапе проектирования системы оценить выигрыш в удельной производительности, достигаемый за счет применения динамической реконфигурации структуры СнПК, а также определить ресурсоемкость сопроцессора, для которого реконфигурация будет эффективной.

5. Алгоритм иерархической компрессии конфигурационных данных ПЛИС, допускающий аппаратную реализацию без использования блочной памяти кристалла.

Апробация результатов диссертации

1. Всероссийская научно-техническая конференция “Проблемы разработки перспективных микроэлектронных систем - 2006”.
2. Международная конференция “Актуальные проблемы твердотельной электроники и микроэлектроники - 2006”.
3. Всероссийская научно-техническая конференция “Проблемы разработки перспективных микро- и наноэлектронных систем - 2008”.
4. Научные сессии МИФИ 2006, 2007, 2008, 2009.

Публикации

По результатам диссертации опубликовано 3 статьи, 12 тезисов докладов и 1 научно-технический отчет.

Структура и объем диссертации

Диссертация состоит из введения, 4 глав, заключения, списка литературы, включающего 148 наименований и приложения, содержащего акты о внедрении результатов диссертации. Содержание диссертации изложено на 147 страницах машинописного текста, включая 54 рисунка и 27 таблиц.

Основное содержание диссертации

Постановка задачи

Классическая структура СнК представлена на рис. 1. Основным блоком такой СнК обычно является процессор, выполняющий программную обработку данных. Специализированные блоки обработки обеспечивают аппаратное выполнение функций, специфических для данной системы. Это могут быть, например, блоки цифровой обработки сигналов, преобразователи потоков данных и другие устройства. Таймеры, АЦП и ЦАП, широтно-импульсные модуляторы и другие блоки могут интегрироваться в состав системы в качестве периферийных устройств. Интерфейс с внешними устройствами обеспечивается с помощью параллельных и последовательных портов, различных шинных и коммуникационных контроллеров и других интерфейсных блоков.

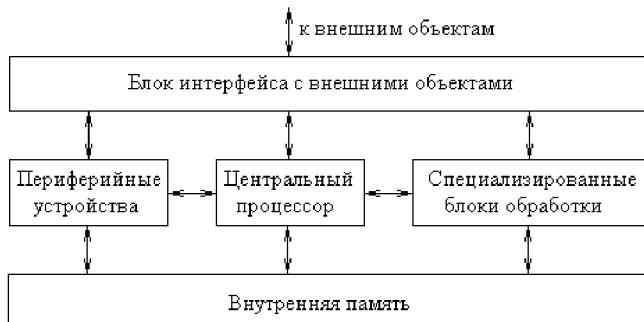


Рис. 1. Структура классической СнК.

Для повышения производительности СнПК и сокращения ее энергопотребления могут быть использованы следующие методы:

- 1) применение в составе СнПК специализированных процессорных ядер, оптимизированных для выполнения конкретных алгоритмов;
- 2) реализация параллельных вычислений путем увеличения количества специализированных блоков обработки данных или организации многопроцессорной системы;
- 3) использование динамической реконфигурации системы, которая позволяет изменять функции специализированных блоков обработки данных в соответствии с текущими потребностями системы.

Применение специализированных процессорных ядер

Если обработка данных в системе осуществляется с использованием процессорного ядра, становится возможной оптимизация данного ядра с целью повышения его производительности при реализации конкретного алгоритма обработки данных. Данная концепция проектирования специализированных процессорных ядер успешно используется рядом синтезируемых процессоров, позволяющих вводить специальные “пользовательские” инструкции (Tensilica Xtensa, ASIP Meister, Xilinx MicroBlaze и др). Аналогичные решения предлагают и производители систем проектирования, использующих языки описания архитектуры (ADL). Такие решения имеют один существенный недостаток – при изменении набора инструкций процессора требуется также модификация средств разработки программного обеспечения (ПО), что существенно снижает эффективность такого маршрута проектирования. При этом подобные средства проектирования являются дорогостоящими коммерческими продуктами, требующими довольно большого времени для освоения. Параметризуемые синтезируемые ядра, как правило, не обладают достаточной гибкостью для модификации набора команд.

В то же время для модификации процессорного ядра может быть применен альтернативный подход: исключение из архитектуры ядра поддержки инструкций и блоков, не используемых данным конкретным приложением. Такой подход позволяет избежать модификации средств разработки программного обеспечения и потенциально способен обеспечивать как сокращение ресурсоемкости процессорного ядра, так и повышение его рабочей тактовой частоты за счет сокращения комбинационных цепей. Для оценки эффективности такой модификации требуется разработать критерии, позволяющие количественно оценить влияние модификации на производительность и энергопотребление процессорного ядра, а также на количество ресурсов кристалла, необходимых для его реализации.

В диссертации предложен маршрут проектирования, обеспечивающий исключение поддержки неиспользуемых инструкций из состава синтезируемого процессорного ядра, описана методика оценки эффективности такой модификации. На примере тестового набора процессорных ядер и приложений проведена оценка влияния предложенной методики модификации на ресурсоемкость, рабочую тактовую частоту и энергопотребление ядер.

Организация многоканальной обработки данных

На рис.2 приведена структура классической микропроцессорной системы, в составе которой процессорное ядро соединяется с периферийными (ПУ) и интерфейсными (ИУ) устройствами с помощью системной шины. В ситуации, когда специализированные блоки обработки соединяются с процессорным ядром посредством общей шины, именно эта связь становится “бутылочным горлом” системы, ограничивающим ее производительность.

Существуют другие варианты интерфейса – так, например, в ядре MicroBlaze предусмотрено наличие до 16 интерфейсов FSL (Fast Simplex Link), предназначенных для быстрой передачи данных между процессорным ядром и сопроцессорами без участия общей системной шины. Каждый из таких интерфейсов обеспечивает одностороннюю передачу 32-разрядных слов между регистрами процессора и периферийными устройствами, причем на каждую передачу требуется всего два такта. При организации связей между процессорным ядром и сопроцессорами использование таких интерфейсов является предпочтительным, так как позволяет существенно снизить загрузку общей шины. Количество таких выделенных интерфейсов всегда ограничено, что накладывает соответствующие ограничения на количество используемых сопроцессоров и общую вычислительную мощность системы.

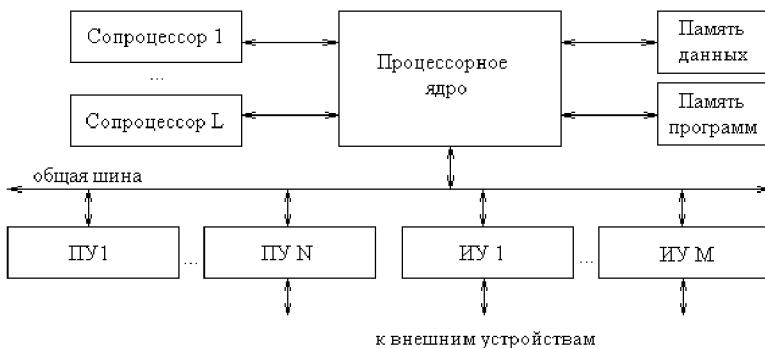


Рис.2. Структура СнК на основе синтезируемого ядра MicroBlaze.

Одним из основных методов повышения производительности отдельных сопроцессоров в настоящее время является организация конвейерной обработки. Однако существуют алгоритмы, для которых конвейеризация невозможна или сильно затруднена – так, например, для алгоритма хэширования MD5 первые конвейерные решения появились только в 2008-2009 годах.

Для повышения производительности вычислительных систем и подсистем также широко используется естественный параллелизм вычислений (векторные, многопроцессорные, SPMD-системы и др). В случае СнПК, построенных на базе синтезируемых процессорных ядер, применение таких решений затруднено уже упомянутыми ограничениями на количество выделенных интерфейсов процессорного ядра, а также тем фактом, что наращивание количества одновременно используемых сопроцессоров в

синхронных системах неизбежно приводит к разрастанию цепей синхронизации и снижению рабочей тактовой частоты. Вопрос об изменении тактовой частоты СнПК при увеличении количества сопроцессоров, в литературе практически не затрагивается, хотя в некоторых работах отмечается нелинейность роста производительности системы при увеличении количества однотипных блоков в векторных системах. Таким образом, возникает необходимость в анализе влияния увеличения количества сопроцессоров на производительность системы.

В диссертации предложена структура потокового сопроцессора, реализующего параллельную обработку данных: в отличие от существующих решений, предложенная структура обладает единым интерфейсом при любом количестве параллельно включенных блоков обработки данных. Проводятся анализ эффективности предложенной системы, оценка влияния дисперсии времени обработки данных и оценка вероятности потерь в случае неконтролируемого потока данных.

Применение динамической реконфигурации структуры СнПК

Современные ПЛИС обладают возможностью изменения конфигурации части кристалла без какого-либо воздействия на остальные ресурсы кристалла. Это означает, что в ходе работы СнПК возможна замена одного или нескольких сопроцессоров в соответствии с текущими потребностями. В силу того, что одновременно в составе системы находится только один (или несколько) из общего числа необходимых сопроцессоров (рис. 1.2), реализующих альтернативные алгоритмы обработки данных, такое решение позволяет:

- 1) существенно снизить энергопотребление системы;
- 2) снизить требования к количеству логических ресурсов кристалла, необходимых для реализации системы.

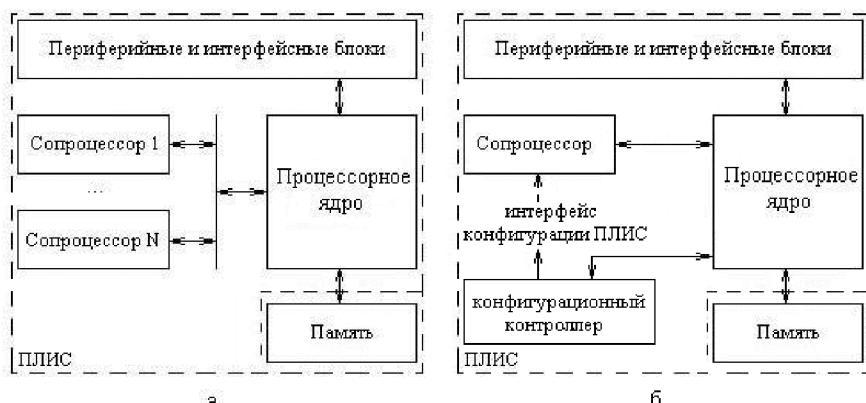


Рис. 3. а) структура системы с несколькими сопроцессорами;
б) структура системы с одним реконфигурируемым сопроцессором.

Для проведения реконфигурации необходимо включение в систему специального контроллера, способного взаимодействовать с конфигурационным интерфейсом ПЛИС. Время, необходимое для выполнения

реконфигурации, которое влияет на производительность системы, зависит как от пропускной способности конфигурационного интерфейса, так и от объема конфигурационных данных. Оценка эффективности использования динамической реконфигурации, учитывая данные факторы, в известных источниках не проведена.

При использовании динамической реконфигурации возникает необходимость в хранении в составе системы данных, необходимых для конфигурации ПЛИС. Так как объем энергонезависимой памяти, реализуемой в современных встраиваемых системах на базе flash-памяти, ограничен, становится актуальной задача сжатия конфигурационных данных. Существует ряд методик, позволяющих осуществить такое сжатие, но, как правило, все они реализуются программно – такая реализация повышает нагрузку на процессорное ядро системы, требует выделения существенного количества оперативной памяти (это верно и в случае аппаратной реализации словарных методов сжатия) и затрудняет модификацию процессорного ядра, описанную ранее. Аппаратно реализуемые словарные методы и методы семейства Лемпела-Зива требуют использования блочной памяти, в большинстве случаев являющейся наиболее дефицитным ресурсом ПЛИС. Сравнительный анализ эффективности сжатия конфигурационных данных с использованием различных алгоритмов в литературе не проводится.

В диссертации предложена методика оценки эффективности применения динамической реконфигурации, позволяющая оценить потенциальный выигрыш в удельной производительности на этапе проектирования системы. Для сокращения объема конфигурационных данных предложен модифицированный алгоритм иерархического сжатия, допускающий аппаратную реализацию без использования блочной памяти кристалла.

Методика оценки эффективности модификации синтезируемых процессорных ядер

В качестве параметров синтезируемых процессорных ядер предлагается использовать следующие величины:

1) A – ресурсоемкость процессорного ядра, т.е. количество ресурсов ПЛИС, необходимое для его реализации (оценивается как

$$A = \sum_{i=1}^n A_i \cdot N_i,$$

где A_i есть площадь блока i -го типа, а N_i – количество блоков данного типа, используемых при реализации данной структуры на кристалле);

2) F – максимальная рабочая тактовая частота процессорного ядра;

3) C – время выполнения тестового приложения (или набора приложений) в тактах;

4) P - потребляемая процессорным ядром мощность.

В качестве производных параметров при этом используются общее время выполнения тестовых приложений, вычисляемое как $T=(C/F)$, и энергопотребление, которое может быть вычислено как $E=(P*T)$.

С учетом того, что структурные блоки в различных семействах ПЛИС обладают различной логической емкостью и структурой, для усреднения

экспериментальных данных необходима нормировка: для каждого из кристаллов за “единичную” принимается ресурсоемкость одной и той же структуры, после чего ресурсоемкости всех прочих структур вычисляются относительно данной:

$$A_j^* = A_j / A_k,$$

где j – номер процессорного ядра в наборе ($\forall j \in [1; N_{cores}]$), k – номер ядра с “единичной” ресурсоемкостью ($k \in [1; N_{cores}]$ фиксировано), N_{cores} – количество ядер в рассматриваемом наборе, A_j^* - нормированная ресурсоемкость ядра с номером j , A_j / A_k - ненормированные ресурсоемкости ядер с номерами j/k .

С целью определения характера распределения оценок тактовых частот проведено исследование по определению тактовой частоты при многократном размещении ряда синтезированных структур на одних и тех же кристаллах. С использованием критерия Смирнова-Колмогорова можно показать, что данное распределение является нормальным. Это означает, что для статистической обработки параметров синтезируемых структур могут использоваться оценки выборочного среднего, среднеквадратичного отклонения и стандартной ошибки среднего:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \quad s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}, \quad c.o.\bar{x} = \frac{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}{n}$$

С ростом количества размещений оценка сходится, причем для достижения точности порядка 2-3% достаточно обычно провести 15-20 различных размещений.

Поскольку смысл модификации процессорного ядра состоит в снижении ресурсоемкости, уменьшении потребляемой мощности, а также повышении вычислительной производительности, соответствующем сокращению времени выполнения задачи, такие оценочные функции могут быть представлены в виде

$$U = \frac{1}{P^k A^l T^m},$$

где показатели степени $k, l, m = \overline{0,1}$ играют роль “весовых коэффициентов”. При введении подобных функций все процессорные ядра из рассматриваемого набора могут быть отсортированы по значению той или иной функции, представляющей интерес в каждом конкретном случае. В дальнейшем рассматривается набор оценочных функций, позволяющих определить минимумы занимаемой площади, времени выполнения задачи, потребляемой мощности и общего энергопотребления, а также определить максимум “вычислительной плотности”:

$$U_1 = \frac{1}{A}; U_2 = \frac{1}{T}; U_3 = \frac{1}{P}; U_4 = \frac{1}{E} = \frac{1}{TP}; U_5 = \frac{1}{AT}.$$

Зная погрешности экспериментально определенных величин, можно оценить погрешности оценочных функций согласно формуле

$$\delta U_i = \sum_j k_j \delta x_j,$$

где δU_i - относительная погрешность оценочной функции, δx_j - относительные погрешности экспериментально определенных величин, входящих в выражение для U_i , k_{ij} - весовые коэффициенты, определяемые степенной зависимостью U_i от x_j .

Методика модификации синтезируемых процессорных ядер путем сокращения набора поддерживаемых команд

Общий алгоритм предлагаемой методики модификации представлен на рис. 4. Исходными данными являются HDL-код процессорного ядра и код приложения. Следует отметить, что HDL-код ядра может быть как ориентированным на определенную элементную базу (то есть использовать логические примитивы реальных блоков ПЛИС), так и независящим от нее - так, например, описание регистрового файла может включать в себя примитивы блочной памяти, а блок умножения может быть реализован с использованием аппаратных умножителей ПЛИС. Исключение составляют случаи, когда логические функции описываются с использованием примитивов "мелкой логики" (отдельных LUT и триггеров) – в таких случаях модификация будет затруднена.

Предлагаемая методика модификации реализуется в виде следующей последовательности этапов.

1) **Компиляция и линковка приложения** позволяют получить исполняемый бинарный код. В ходе данного этапа могут быть использованы как стандартные средства разработки ПО, так и самостоятельно разработанные - последние необходимы в тех случаях, когда исходная архитектура процессорного ядра не обеспечивает полной совместимости с архитектурой-прототипом или не имеет такого прототипа вообще.

2) **Синтаксический анализ полученного бинарного или ассемблерного кода** позволяет выделить подмножество команд, используемых при выполнении исходного приложения ("используемое подмножество"). Подмножество команд исходной архитектуры, не вошедших в используемое подмножество, подлежит исключению и далее называется "исключаемым".

3) **Модификация ядра на основе полученной информации об используемом и исключаемом подмножествах команд** проводится в два этапа:

а) **модификация блока дешифрации команд:** HDL-код, реализующий формирование выходных сигналов блока при поступлении команд из исключаемого подмножества, удаляется;

б) **модификация блоков с постоянными входными сигналами:** в случае, если в результате упрощения блока дешифрации команд какие-либо его выходные сигналы имеют постоянное значение для всех команд используемого подмножества, данные сигналы заменяются константами для всех блоков, использующих их в качестве входных сигналов. Дальнейшая оптимизация логических функций будет проведена средствами САПР в ходе синтеза автоматически.

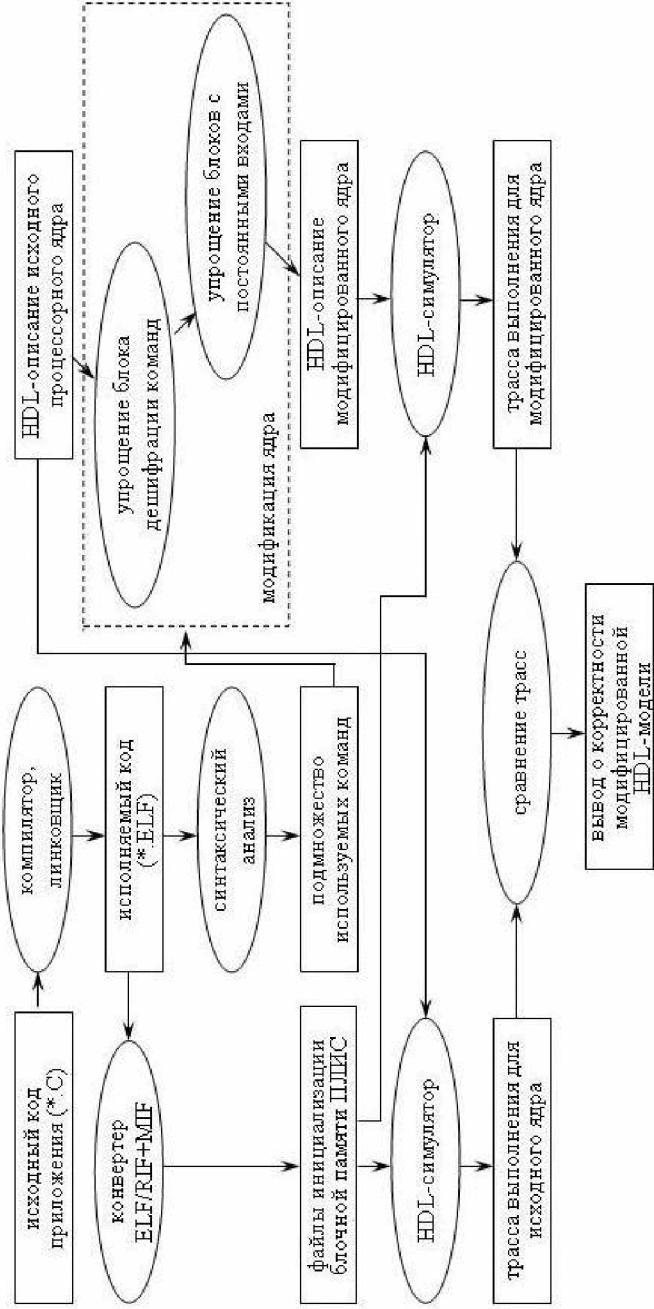


Рис. 4. Методика модификации синтезируемого ядра путем сокращения набора поддерживаемых команд

4) **HDL-моделирование выполнения приложения на исходном и модифицированном ядрах** позволяет получить две трассы, на основе сравнения которых может быть сделан вывод о корректности модифицированного HDL-описания: в случае идентичности трасс модификация признается корректной.

Влияние описанной модификации на ресурсоемкость синтезируемой структуры однозначно – в силу упрощения отдельных блоков количество ресурсов ПЛИС, необходимых для реализации структуры, может только уменьшаться. Поскольку критические комбинационные цепи могут как укорачиваться, так и удлиняться, изменение рабочей тактовой частоты в результате проведения модификации предсказать заранее невозможно.

На примере тестового набора из 40 синтезируемых процессорных ядер (система команд MIPS-I, до 7 ступеней конвейера, три различных реализации сдвигателя) и 20 тестовых приложений показано, что предложенная методика модификации способна обеспечивать:

- сокращение ресурсоемкости процессорного ядра на величину до 45%;
- увеличение рабочей тактовой частоты (и соответствующее сокращение времени выполнения задач) на величину до 13%;
- снижение динамической потребляемой мощности на величину до 8%;
- сокращение общего энергопотребления при выполнении приложений на величину до 15%;
- увеличение “вычислительной плотности” синтезируемых структур на величину до 100%.

Полученные в ходе данного эксперимента результаты позволяют утверждать, что нормированные оценки ресурсоемкости, рабочей тактовой частоты и потребляемой мощности, полученные для определенного кристалла ПЛИС, могут быть распространены на кристаллы ПЛИС с аналогичной структурой (например, между семействами ПЛИС StratixI/II/III).

Структура многоканальной системы обработки данных

Предложенная структура вычислительного блока с пространственным распараллеливанием обработки потока однородных данных имеет вид, приведенный на рис. 5. Существенным отличием данной системы от существующих решений является единство интерфейса: для внешних устройств сопроцессор является единым устройством независимо от количества блоков обработки данных, содержащихся в нем.

Алгоритм работы такого сопроцессора состоит в следующем.

1) Контроллер шины осуществляет запись слова (пакета) данных во входной буфер данных, имеющий организацию FIFO. В случае контролируемого потока данных, то есть потока, скорость которого может регулироваться, контроллер шины проверяет флаг заполнения, генерируемый входным буфером, и не осуществляет запись, если буфер заполнен. В случае неконтролируемого потока запись в заполненный буфер не осуществляется – происходит потеря данных.

2) При наличии свободных блоков обработки пакет данных считывается из входного буфера блоком распределения потока и записывается в свободный

обработчик с меньшим порядковым номером. Одновременно происходит запись тэга, содержащего номер обработчика, получившего данные, в буфер тэгов, также имеющий организацию FIFO – в дальнейшем эти тэги используются для восстановления порядка результатов. Обработчик, получивший данные, после получения данных выставляет для блока распределения флаг занятости.

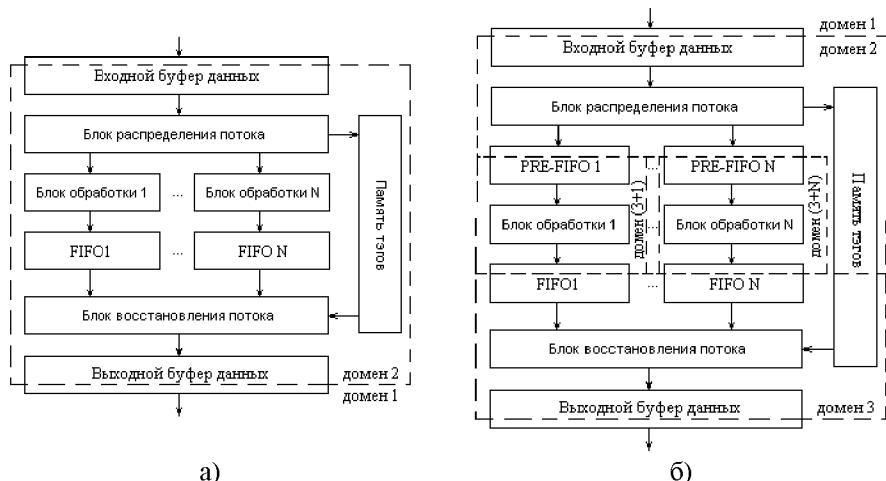


Рис. 5. Структурная схема потокового сопроцессора а) синхронная схема
б) локально-синхронная схема.

3) После выполнения алгоритма обработки пакета обработчик записывает полученный результат в индивидуальный FIFO-буфер, если тот не заполнен, и сбрасывает флаг занятости, отслеживаемый блоком распределения потока; в случае заполнения индивидуального буфера результат сохраняется в обработчике, сброса флага занятости не происходит.

4) Блок восстановления потока при наличии данных в памяти тэгов читает оттуда первый тэг и проверяет флаг пустоты у индивидуального буфера с номером, соответствующим данному тэгу. При обнаружении данных в данном индивидуальном буфере блок восстановления производит их считывание и запись в выходной буфер данных, после чего выбирает из памяти тэгов следующий тэг.

Поскольку входной и выходной буферы данных могут являться асинхронными, возможно разделение доменов синхронизации – это позволяет сделать тактовые частоты основной схемы и сопроцессора независимыми. В случае локально-синхронной схемы возможно также динамическое управление тактовыми частотами отдельных блоков обработки.

В результате анализа работы предложенной схемы была получена ее упрощенная математическая модель. В частности, время обработки M пакетов данных с использованием N блоков обработки составляет

$$T(M)|_N = \begin{cases} M \cdot T_0 + T_{\text{вып}}, & (N-1)T_0 \geq T_{\text{вып}} \\ T_0 + (T_0 + T_{\text{вып}}) \cdot \frac{M}{N}, & (N-1)T_0 \leq T_{\text{вып}} \end{cases}$$

где $T_{\text{вып}}$ - время обработки одного пакета данных одним обработчиком, а T_0 - задержка, обусловленная структурой сопроцессора. При этом эффективность системы при любом N и $M>>1$ определяется выражением

$$E(N) = \begin{cases} \frac{T_{\text{вып}}}{T_0 + T_{\text{вып}}}, & N=1 \\ \frac{T_{\text{вып}}}{NT_0}, & (N-1)T_0 \geq T_{\text{вып}} \\ \frac{T_{\text{вып}}}{T_0 + T_{\text{вып}}}, & (N-1)T_0 \leq T_{\text{вып}} \end{cases}$$

Данная зависимость была подтверждена экспериментально в ходе исследования созданной RTL-модели системы (см. рис. 6), экспериментально определенное значение T_0 составило 5 тактов в случае синхронной схемы и 8 тактов в случае локально-синхронной схемы.

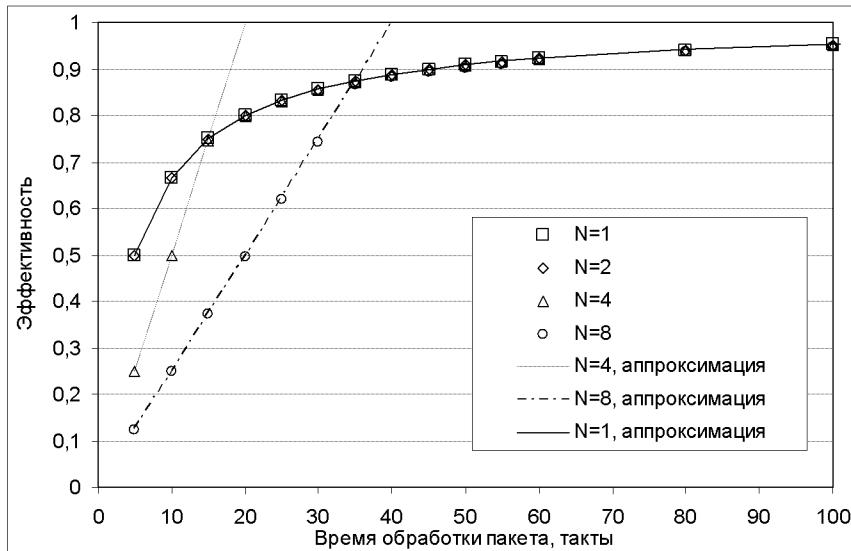


Рис. 6. Зависимость эффективности системы от времени обработки одного пакета данных и ее аппроксимация.

Для оценки влияния масштабирования на рабочую тактовую частоту синхронной системы были проведены размещение и разводка рассматриваемой системы на кристаллах семейств Virtex4 (Xilinx) и StratixIII (Altera). На основании полученных данных можно сделать вывод о том, что при масштабировании системы, т.е. при увеличении количества блоков обработки, помимо увеличения производительности за счет параллелизма вычислений

будет иметь место снижение рабочей тактовой частоты системы. Для учета данного эффекта возможно введение поправочного коэффициента

$$k_F(N) = \frac{F(N)}{F(1)} = (1 - K) + K \cdot \exp\left[-\frac{N-1}{C}\right],$$

где K и C – экспериментально определяемые константы. В случае локально-синхронной схемы экспериментально определенное изменение рабочей тактовой частоты находится в пределах погрешности эксперимента (2%), то есть соблюдается условие $k_F(N) \approx 1$.

Выражение для определения общего коэффициента ускорения при масштабировании, т.е. увеличении количества обработчиков, имеет вид

$$S(N, F) = N \cdot k_F(N) \cdot E(N),$$

где $k_F(N)$ и $E(N)$ определяются приведенными ранее выражениями. При малых временах обработки одного пакета данных ($T_{вып} < 30$ тактов) эффект увеличения задержек в локально-синхронной схеме может сказываться на производительности схемы сильнее, чем фактор снижения тактовой частоты в синхронной схеме, однако при увеличении времени обработки ($T_{вып} > 50$ тактов) локально-синхронная схема обладает большим коэффициентом ускорения, чем синхронная (в случае, если $K=0,2$).

Если время обработки пакета не является постоянным, а зависит от поступивших данных, в рассматриваемой схеме могут возникать простой обработчиков, связанные с ожиданием результатов при восстановлении порядка потока. С увеличением дисперсии времени обработки будет увеличиваться время простоя, что приводит к снижению коэффициента ускорения. В ходе проведенного моделирования было изучено четыре типа распределений времени обработки пакета данных (равномерное, пуассоновское, нормальное, а также распределение, являющееся суперпозицией двух δ -функций). В результате было показано, что для всех рассмотренных распределений, кроме суперпозиции δ -функций, насыщение коэффициента ускорения системы происходит при размере буфера тэгов, равном $2N$ элементам (т.е. 16 тэгов в случае 8-канальной системы). Для δ -распределения необходимо значительное увеличение буфера тэгов – $4N..8N$ и более в зависимости от соотношения количеств пакетов с различными временами обработки.

Для оценки величины вероятности возникновения потерь в системе с неконтролируемым входным потоком данных проведено RTL-моделирование восьмиканальной системы в условиях максимальной загрузки: времена обработки и поступления данных имеют пуассоновское распределение, причем среднее время обработки данных в 8 раз больше, чем период поступления данных в систему. Проведенное RTL-моделирование показывает, что вероятность потерь не превышает $8 \cdot 10^{-4}$ даже в том случае, когда в системе одновременно находится только 8 пакетов данных. С увеличением входного буфера до 32 пакетов вероятность возникновения потерь снижается на порядок. В случае, когда в системе одновременно может находиться 16 пакетов, а размер входного буфера равен 16 словам, вероятность потерь не превышает $1 \cdot 10^{-6}$.

Методика оценки эффективности динамической реконфигурации

Для оценки эффективности частичной динамической реконфигурации рассматривается система, предполагающая одновременное использование одного из N алгоритмов обработки данных. Для комплексной оценки таких параметров как быстродействие и ресурсоемкость вводится “удельная производительность”, описываемая выражением

$$D = \frac{1}{A \cdot T},$$

где A – ресурсоемкость системы (т.е. количество ресурсов ПЛИС, необходимых для ее реализации), T – время выполнения ей задачи.

Реконфигурируемая система имеет преимущество в удельной производительности по сравнению с классической системой при условии

$$E_{\text{реконф}} = \frac{D_{\text{реконф}}}{D_{\text{класс}}} > 1.$$

Данное неравенство имеет решение

$$0 < \mu_{\max} < \frac{-(1+\eta) + \sqrt{(1+\eta)^2 + 4(S-\eta)/\rho}}{2}, \quad \rho > \frac{4(\eta-S)}{(1+\eta)^2},$$

где

N – общее количество альтернативных сопроцессоров;

$\frac{A_i}{A_c} = \mu_i$ – отношение ресурсоемкостей i -го сопроцессора и постоянной

(т.е. не подлежащей реконфигурации) части системы;

$\frac{A_{rc}}{A_c} = \eta$ – отношение ресурсоемкостей контроллера реконфигурации и

постоянной части системы;

$\mu_{\max} = \max\{\mu_1, \mu_2, \dots, \mu_N\}$ – ресурсоемкость наибольшего сопроцессора;

$S = \sum_{i=1}^N \mu_i - \mu_{\max}$ – суммарная ресурсоемкость всех сопроцессоров, кроме

наибольшего;

$\rho = \frac{M_{\text{кристалла}} A_c}{A_{\text{кристалла}} T_c \rho_0}$ – величина, характеризующая пропускную способность

конфигурационного интерфейса ПЛИС, объем конфигурационных данных ПЛИС, приходящийся на долю постоянной части системы, и время работы системы без реконфигурации.

На основании полученных данных можно сделать следующие выводы.

1. Эффективность динамической реконфигурации возрастает пропорционально количеству альтернативных алгоритмов обработки данных, причем при сокращении периодов использования алгоритмов эффективность реконфигурируемой системы снижается.

2. При увеличении ресурсоемкости альтернативных сопроцессоров эффективность системы с динамической реконфигурацией возрастает нелинейно, причем зависимость может быть хорошо аппроксимирована степенной функцией с показателем степени меньше 1 (см. рис. 7).

3. Ресурсоемкость максимального сопроцессора в рассмотренных случаях может существенно превышать ресурсоемкость постоянной (нереконфигурируемой) части системы.

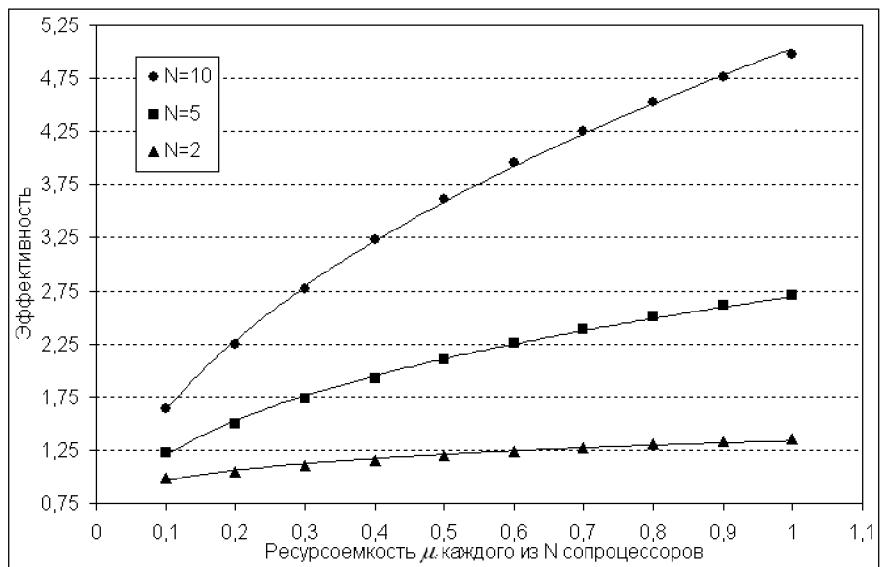


Рис. 7. Зависимость эффективности от относительного размера альтернативных сопроцессоров и их количества (при соотношении $T_c / T_e = 0,1$).

Алгоритм иерархического сжатия конфигурационных данных ПЛИС

Система с динамической реконфигурацией обладает одним существенным недостатком: для хранения конфигурационных данных в системе необходимо дополнительное количество памяти.

На основании анализа конфигурационных файлов (КФ) тестового набора из 20 различных IP-ядер были сделаны следующие выводы:

1) в рассматриваемых КФ присутствуют последовательности нулевых битов достаточно большой длины (до 7000 бит), а количество единичных битов очень мало по сравнению с количеством нулевых битов, причем доля нулевых последовательностей длины k убывает с ростом k по закону, близкому к экспоненциальному;

2) зависимость размера полученных КФ от сложности синтезируемого IP-блока хорошо аппроксимируется линейной функцией - это означает, что рассматриваемые КФ не содержат избыточной информации о неиспользуемых ресурсах кристалла.

Для компрессии КФ подобной структуры (содержащих малое количество ненулевых битов) предлагается использовать алгоритм иерархического сжатия.

Параметрами алгоритма являются максимальное количество уровней компрессии L_{max} и размер блока B . Исходная последовательность битов $S[L]$

разбивается на блоки $S^*[L,i]$ длиной B бит. Для всех блоков исходной последовательности выполняется проверка содержимого блока: если все биты блока с номером i являются нулевыми, то бит с номером i в выходной последовательности битов $S[L+1]$ равен 0 (в противном случае он равен 1). Процедура повторяется рекурсивно до достижения заданного значения L_{max} . Результатом компрессии является конкатенация всех $S[L=0..L_{max}]$ с исключенными “нулевыми” блоками (это означает, что последовательность нулей длиной B в $S[L]$ описывается одним нулевым битом в $S[L+1]$). Алгоритм компрессии КФ реализуется программно на ПК при проектировании системы, далее сжатые КФ размещаются и хранятся в памяти СнПК.

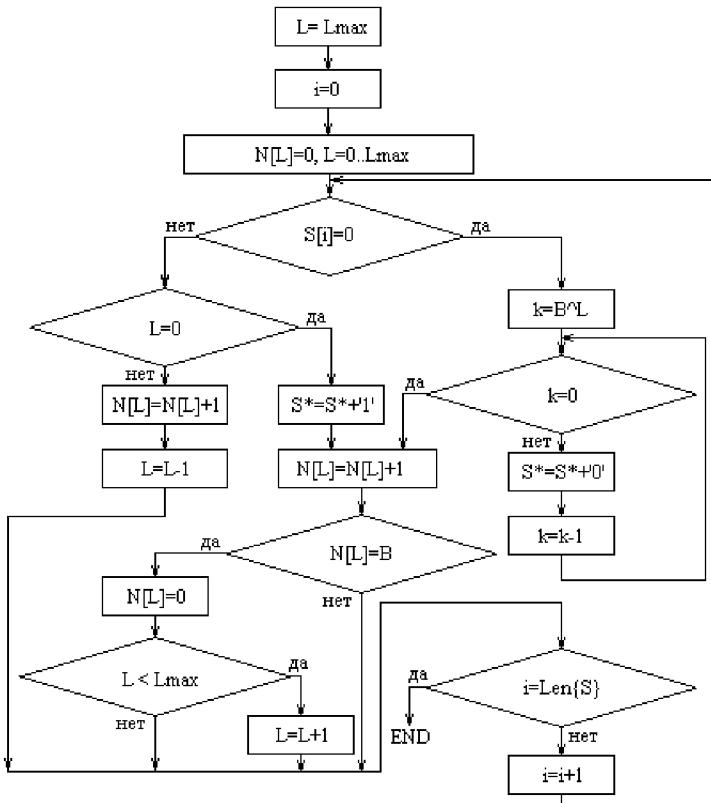


Рис.8. Блок-схема алгоритма декомпрессии.

Алгоритм декомпрессии предполагает простой синтаксический разбор полученной строки S с выделением битов или блоков для всех $L=L_{max}, 0$ и формирование выходной строки S^* (блок-схема алгоритма декомпрессии приведена на рис.8). Алгоритм допускает блочную обработку – одновременно обрабатывается только одно входное слово, выбранное из памяти. Процесс декомпрессии происходит в рабочем цикле при осуществлении

реконфигурации части СнПК: это означает, что для его аппаратной реализации в состав СнПК должен быть введен специальный блок.

RTL-модель блока декомпрессора была описана на языке Verilog и реализована на базе ПЛИС XC4VLX25 семейства Virtex4. Для аппаратной реализации декомпрессора с 32-разрядными входными словами требуется относительно небольшое количество ресурсов ПЛИС (193 LUT, 63 триггера), а его тактовая частота составляет 128 МГц (для кристалла с параметром SpeedGrade=10). Выходные данные генерируются декомпрессором с частотой, которая в случае нулевых последовательностей равна половине входной тактовой частоты, а в случае ненулевых блоков в 8 раз меньше входной тактовой частоты – это означает, что результаты декомпрессии получаются со скоростью от 16 до 64 Мбайт/с.

В сравнении с современными программными архиваторами и алгоритмами, традиционно используемыми для сжатия КФ, предложенный алгоритм выглядит вполне конкурентоспособным: экспериментальные результаты и их сравнение с теоретической оценкой, полученной в соответствии с теорией информационной энтропии Шэннона, приведены на рис. 9.

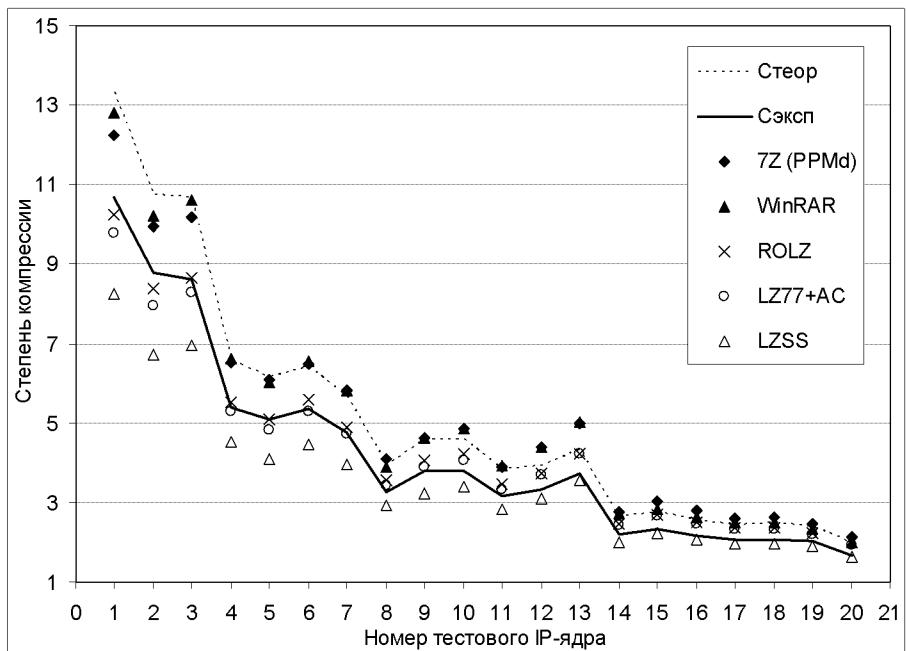


Рис. 9. Сравнение различных алгоритмов и теоретической оценки

Анализ экспериментальных данных, полученных при сжатии КФ тестового набора, позволяет сделать следующие выводы:

1) степень компрессии возрастает при увеличении количества уровней компрессии L_{max} и имеет максимум в области $L_{max}=2..4$;

- 2) степень компрессии для $L_{max}=2..5$ имеет максимум в области $B=3..4$;
- 3) в среднем по набору тестов достигаемая степень компрессии $C_{эксп}$ составляет 83% от расчетного теоретического значения, полученного в соответствии с теорией информационной энтропии Шэннона;
- 4) при малом отклонении от оптимальных параметров ($L_{max}=2..4$, $B=3..4$) происходит несущественное снижение степени компрессии – на рассмотренном наборе IP-ядер оно не превышает 2,2%.

Использованный алгоритм блочного сжатия позволяет добиться степени компрессии, которая всего на 20% меньше степеней компрессии современных программных архиваторов WinRAR и 7Z. Алгоритм LZSS уступает предложенному алгоритму в степени компрессии в среднем 14%, два более сложных алгоритма семейства Лемпела-Зива (ROLZ, LZ77+AC) выигрывают в среднем не более 7%. Максимальное различие в степенях сжатия достигается на самых больших файлах: по сравнению с LZSS иерархический алгоритм обеспечивает преимущество в 3%, но по сравнению с другими алгоритмами проигрывает до 15%. В отличие от других алгоритмов, предложенный алгоритм иерархической компрессии допускает аппаратную реализацию декомпрессора без использования блочной памяти ПЛИС.

Основные результаты диссертации и их практическая реализация

1. Предложена методика модификации синтезируемых процессорных ядер, основанная на исключении поддержки инструкций, не используемых конкретным приложением. Разработан и реализован маршрут проектирования, обеспечивающий эффективное применение предложенной методики модификации процессорных ядер. На примере набора из 40 процессорных ядер и 20 тестовых приложений показано, что предложенная методика позволяет обеспечить сокращение ресурсоемкости процессорного ядра на величину до 45%, увеличение рабочей тактовой частоты на величину до 13%, сокращение динамической потребляемой мощности на величину до 8%, увеличение “вычислительной плотности” синтезируемых структур на величину до 100%.

2. Разработана методика оценки эффективности модификации процессорных ядер, позволяющая учесть такие параметры ядер, как количество использованных ресурсов ПЛИС, рабочая тактовая частота и энергопотребление. Использование предложенной методики дает возможность определить целесообразность модификации процессорных ядер на этапе проектирования СПК.

3. Предложено два варианта структуры сопроцессора, предназначенного для обработки массивов данных: локально-синхронная схема, в отличие от синхронной, позволяет минимизировать снижение рабочей тактовой частоты при масштабировании системы и допускает реализацию алгоритмов динамического управления энергопотреблением. В обоих вариантах при изменении количества блоков обработки не требуется изменения интерфейса с внешними объектами. Проведен сравнительный анализ эффективности синхронной и локально-синхронной схем, сформулированы рекомендации по выбору одного из вариантов многоканальной схемы при различных временах обработки пакета данных. Показано, что влияние дисперсии времени

обработки на эффективность системы в случаях равномерного, пуассоновского и нормального распределений времени обработки может быть сокращено за счет увеличения размера буфера тэгов до (2^*N) элементов.

4. Предложена методика оценки эффективности динамической реконфигурации структуры СнПК, позволяющая на этапе разработки оценить потенциальный выигрыш в удельной производительности и рассчитать максимально допустимый размер сопроцессора.

5. На основе проведенного анализа структуры конфигурационных данных ПЛИС предложен алгоритм иерархического сжатия конфигурационных файлов. На примере набора из 20 различных IP-блоков показано, что применение предложенного алгоритма обеспечивает сокращение объема конфигурационных данных в 1,8..10,9 раз. Определен диапазон параметров алгоритма, обеспечивающих максимальную степень компрессии конфигурационных данных. Показано, что предложенный алгоритм имеет преимущество (в среднем по тестовому набору IP-ядер 14%) в степени сжатия перед алгоритмом LZSS, уступает в степени сжатия более сложным алгоритмам из семейства LZx (в среднем не более 7%) и уступает около 20% в степени сжатия программным архиваторам WinRAR и 7Z. Разработано программное обеспечение для ПК, реализующее предложенный алгоритм иерархической компрессии. Разработана RTL-модель блока аппаратного декомпрессора, определены параметры данного блока при его реализации на ПЛИС семейства Virtex4. Показано, что предложенный алгоритм декомпрессии допускает аппаратную реализацию без использования блочной памяти ПЛИС.

6. В Малой ускорительной лаборатории МИФИ (г. Москва) разработан и внедрен в научную и учебную деятельность лаборатории аппаратно-программный комплекс СПЕКТР, построенный на базе СнПК. Использование предложенной методики модификации процессорных ядер позволило повысить точность проводимых измерений на 12% и снизить себестоимость аппаратной части комплекса на 20%.

7. В рамках работ по созданию радиолокационных станций 5П-27, 67Н6Е, 1РЛ-123Е (ВНИИРТ, г. Москва) разработаны вычислительные блоки 5П4ИИ01, 674ИИ01 с унифицированными ячейками обработки сигналов 5П2ХЛ002, 672ХЛ002 на базе ПЛИС. В процессе разработки были использованы методика оценки эффективности динамической реконфигурации и алгоритм компрессии конфигурационных файлов, предложенные в диссертации. Применение алгоритма компрессии позволило сократить объем памяти, необходимой для хранения конфигурационных данных, в 2..8 раз.

Получены акты об использовании результатов диссертации в следующих организациях: Всероссийский научно-исследовательский институт радиотехники (ОАО “ВНИИРТ”, г. Москва), Московский инженерно-физический институт (НИЯУ “МИФИ”, г. Москва).

Публикации по теме диссертации

1. Шагурин И., Шалтырев В. Проектирование систем на кристалле на базе FPGA компании Xilinx // ChipNews. Инженерная микроэлектроника. – 2005. – № 10. – С. 54–58.
2. Научно-технический отчет по теме 85-3-027-135: Разработка специализированного блока с ПЛИС и фрагмента тестовой задачи для реализации на СБИС. Регистрационный номер 01200507555, код ВНТИЦ 0203025050320 / Шагурин И.И., Шалтырев В.А., Волов А.С. [и др.]. – 2005. – 115 с.
3. Шалтырев В.А., Шагурин И.И. Проектирование “Систем на кристалле” с использованием среды Xilinx EDK //Научная сессия МИФИ 2006: Сборник научных трудов. – 2006. – Т. 1. – С. 126–127.
4. Шалтырев В.А. Реализация микропроцессорных систем на основе ядра MicroBlaze // Научная сессия МИФИ 2006: сборник научных трудов. – 2006. – Т. 1. – С. 124–125.
5. Волов А.С., Шалтырев В.А., Мокрецов М.О. Модуль для разработки IP-ядер на основе ПЛИС Xilinx Spartan-3 //Научная сессия МИФИ 2006: сборник научных трудов. – 2006. – Т. 1. – С. 118–11.
6. Шагурин И., Шалтырев В., Волов А. “Большие” FPGA как элементная база для реализации систем на кристалле // Электронные компоненты. – 2006. – №5. – С. 83–88.
7. Шагурин И.И., Шалтырев В.А. Создание “систем на кристалле” на основе ПЛИС с использованием синтезируемых процессорных ядер // Сборник трудов конференции “Проблемы разработки перспективных микроэлектронных систем - 2006”: сборник трудов. – 2006. – С. 382–385.
8. Шагурин И.И., Шалтырев В.А. Синтезируемые процессорные ядра как основа для систем на кристалле //Актуальные проблемы твердотельной электроники и микроэлектроники. Труды десятой международной научной конференции: сборник трудов. – 2006. – Ч. 2. – С. 141–142.
9. Гольдберг Д.Ю., Шагурин И.И., Шалтырев В.А. RTL-модель 32-разрядного микропроцессора для специальных применений // Научная сессия МИФИ 2007: сборник научных трудов. – 2007. – Т. 1. – С. 99–100.
10. Шалтырев В.А. RTL-модели блоков сжатия звуковых сигналов // Научная сессия МИФИ 2007: сборник научных трудов. – 2007. – Т.1. – С. 101–102.
11. Новицкий А.С., Шагурин И.И., Шалтырев В.А. Создание высокоуровневого описания процессорного ядра PicoBlaze // Научная сессия МИФИ 2007: сборник научных трудов. – 2007. – Т. 1. – С. 103–104.
12. Шалтырев В.А. Реализация интерфейса CLI с использованием хэш-функций и конечных автоматов // Научная сессия МИФИ 2008: сборник научных трудов. – 2008. – Т. 8. – С. 154–155.
13. Пантелеев А.Ю., Шалтырев В.А. Стековый процессор с изменяемым набором инструкций // Научная сессия МИФИ 2008: сборник научных трудов. – 2008. – Т. 8. – С.193–194.

14. Мансурская Л.А., Шалтырев В.А. IP-блок быстрого преобразования Фурье // Научная сессия МИФИ 2008: сборник научных трудов. – 2008. – Т. 8. – С. 195–196.
15. Шалтырев В.А. Методы оптимизации синтезируемых процессорных ядер при реализации “систем на кристалле” на основе ПЛИС // Проблемы разработки перспективных микро- и наноэлектронных систем – 2008: сборник трудов / под общ. ред. А.Л. Стемпковского. – М.: ИППМ РАН, 2008. – С. 458–461.
16. Шалтырев В.А., Шалтырев К.А., Шатурин И.И. Динамическая реконфигурация ПЛИС с использованием сжатых битовых потоков // Известия ВУЗов. Электроника. – 2009. – № 2 (76). – С. 43–50.