На правах рукописи

Рыженко Игорь Николаевич

МЕТОДЫ, АЛГОРИТМЫ И ПРОГРАММНЫЕ ИНСТРУМЕНТЫ АРХИТЕКТУРНО – НЕЗАВИСИМОГО ВЫСОКОУРОВНЕВОГО СИНТЕЗА ОДНОКРИСТАЛЬНЫХ ЦИФРОВЫХ СХЕМ

Специальность: 2.3.5. Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Красноярск – 2024

Работа выполнена в Федеральном государственном автономном образовательном учреждении высшего образования «Сибирский федеральный университет»

Научный руководитель: кандидат технических наук, доцент, **Непомнящий Олег Владимирович**

Официальные оппоненты: **Зюбин Владимир Евгеньевич,** доктор технических наук, доцент, Федеральное государственное автономное образовательное учреждение высшего образования «Новосибирский национальный исследовательский государственный университет», кафедра компьютерных технологий, заведующий кафедрой

Дордопуло Алексей Игоревич, доктор технических наук, Общество с ограниченной ответственностью «НИЦ супер-ЭВМ и нейрокомпьютеров», отдел математического и алгоритмического обеспечения, начальник отдела

Ведущая организация: Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА - Российский технологический университет»

Защита состоится «04» октября 2024 г. в 13:00 часов на заседании диссертационного совета 24.2.404.05, созданного на базе Сибирского федерального университета по адресу: 660074, г. Красноярск, ул. Академика Киренского, 26, ауд. УЛК 112.

С диссертацией можно ознакомиться в библиотеке и на сайте Сибирского федерального университета по адресу http://www.sfu-kras.ru

Автореферат разослан « » 2024 г.

Ученый секретарь
диссертационного совета Кайзер Юрий Филиппович

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы. Постоянно совершенствующиеся технологии производства цифровых интегральных схем и переход к новым топологическим нормам обуславливают активное внедрение передовых достижений микроэлектроники. На рынок выходят технологии изготовления трехмерных кристаллов, активно внедряются одно- и многопроцессорные системы, развивается направление производства сверхбольших интегральных схем (СБИС) с динамически реконфигурируемой архитектурой и т.д.

При разработке СБИС решаются задачи: системного и алгоритмического проектирования, оптимизации проектных решений в условиях ограниченности вычислительных ресурсов, поиска архитектурных решений при условии временных ограничений на разработку.

Этапы маршрута проектирования СБИС можно разделить на высокоуровневые и низкоуровневые (уровень логических элементов, уровень топологии). На сегодняшний день в достаточной степени разработаны и автоматизированы заключительные, низкоуровневые этапы создания проекта СБИС. Однако большинство задач системной организации процесса проектирования, задач эффективной архитектурной организации СБИС до сих пор остаются нерешенными.

В отличие от низкоуровневых этапов при описании проекта на верхних уровнях иерархии закладываются концепции системной организации всего процесса проектирования. Таким образом, на первый план выходит развитие маршрутов и технологий, базирующихся на принципах высокоуровневого проектирования, позволяющих осуществлять формирование комплексного подхода к организации всех фаз проекта.

Степень разработанности темы. Исследованиям методов проектирования вычислительных систем на кристалле, реализующих возможность высокоуровневого подхода, посвящены работы И.А. Каляева, А.Л. Стемпковского, И.И. Левина, С.Г. Русакова, А.Н. Терехова, В.В. Топоркова, А.К. Кима, А.Е. Платунова, В.Г. Немудова, А.И. Легалова, из зарубежных специалистов в первую очередь следует отметить работы Д. Доннгара, А. Санджованни-Винсентелли, Е. Ли, А. Феррари, Г. Мартина, Г. Аха, А. Джеррайи и др.

Тем не менее известные работы не позволяют однозначно утверждать о создании эффективных методологий высокоуровневого архитектурно-независимого проектирования однокристальных систем. В большинстве случаев проектируемая модель системы разрабатывается под конкретную архитектуру на прикладном языке на основе стандартных или рекомендованных производителем библиотек, используемых для конкретного воплощения в целевую платформу. Целевой платформой является цифровая интегральная схема, имеющая определенный набор базовых логических элементов и способов их соединения.

При разработке систем параллельной обработки данных на СБИС существует ряд проблем:

- отсутствуют методы эффективной выработки архитектурных решений для однокристальных систем параллельной обработки информационных потоков, не зависящие от конечной формы реализации;

- отсутствуют инструментальные средства, обеспечивающие эффективный перенос архитектурно-независимого высокоуровневого описания решаемых прикладных задач на целевую платформу;

- за редким исключением языки программирования, применяемые на современном этапе для описания однокристальных систем параллельной обработки данных,

либо предназначены для схемотехнического описания, либо ориентированы на традиционное последовательное программирование.

Известные попытки решения этих проблем в основном направлены на устранение семантического разрыва между высокоуровневым и низкоуровневым описанием систем с использованием универсальных языков программирования или разработкой собственных языковых средств от специализированного ассемблера до высокоуровневого языка параллельного представления (COLAMO). Такие подходы реализуют решение задачи высокопроизводительных вычислений на реконфигурируемых платформах, таких как ПЛИС (программируемая логическая интегральная схема). Кроме того, известные инструментальные средства, за редким исключением, не нашли широкого применения и, как правило, не выходят за рамки академических проектов. Следовательно, разработка методов и инструментальных средств, обеспечивающих архитектурно-независимое описание однокристальных вычислительных систем, обладающих массовым параллелизмом, является актуальной задачей.

При этом требуется создание новых подходов к описанию СБИС на архитектурном уровне, позволяющих обеспечить максимальную абстракцию алгоритмов функционирования от архитектуры целевого кристалла и их представление на языках описания аппаратуры для адекватного отображения на нижние уровни в иерархии проектирования.

Цель и задачи исследования. Целью работы является повышение надежности и скорости разработки цифровых СБИС посредством разработки метода и инструментальных средств для высокоуровневого синтеза цифровых однокристальных систем.

Для достижения указанной цели в работе решаются следующие **задачи**:

1. Исследование методов, алгоритмов, языков и программных инструментов высокоуровневого проектирования сверхбольших интегральных схем.

2. Выбор и адаптация модели вычислений, языка параллельного программирования и разработка метода архитектурно-независимого описания параллельных алгоритмов и программ для цифровой обработки данных на СБИС.

3. Разработка алгоритмов и программных инструментов для процесса архитектурно-независимого проектирования и трансляции высокоуровневого представления СБИС в языки описания аппаратуры

4. Разработка с использованием предложенного метода и программных инструментов проектов цифровых интегральных схем и оценка эффективности предложенных решений на основе результатов практической реализации.

Объект исследования. Методы, алгоритмы и языки высокоуровневого проектирования СБИС.

Методы исследований. Поставленные задачи решены с использованием теории графов, имитационного моделирования, анализа и синтеза цифровых логических схем. При разработке основных положений диссертации применялись методы системного анализа вычислительных систем, функционально-потокового и объектно-ориентированного проектирования и программирования. Прикладное программное обеспечение реализовано на языке C++.

Научная новизна:

1. Впервые разработан метод высокоуровневого синтеза цифровых однокристальных интегральных систем на основе функционально-потокового параллельного (ФПП) языка, являющийся архитектурно-независимым. В отличии от существующих, разработанный метод позволяет реализовать переносимость исходного высокоуров-

невого описания алгоритма функционирования СБИС на различные целевые платформы за счет свертки параллелизма.

2. Разработана модифицированная модель вычислений ФПП языка для описания функционального состава и алгоритмов управления вычислениями, позволяющая выполнять описания цифровых СБИС.

3. Впервые разработаны алгоритмы преобразования высокоуровневого архитектурно-независимого описания цифровых систем на основе ФПП языка, позволяющие выполнять его автоматическое преобразование в низкоуровневое архитектурно-зависимое представление.

Теоретическая значимость работы. Результаты работы могут быть использованы для повышения эффективности процесса разработки цифровых СБИС. Результаты диссертационных исследований включают развитие теории параллельного программирования, создания и сопровождения программных средств поддержки проектирования цифровых интегральных схем.

Практическая значимость работы. На основе предложенных методов и алгоритмов разработано программное обеспечение в виде комплекта прикладных программ для поддержки процесса высокоуровневого проектирования цифровых схем на базе функционально-потоковой парадигмы, реализующее архитектурно-независимый метод синтеза описания СБИС. Получены свидетельства о регистрации программ для ЭВМ.

Результаты исследования использованы при разработке и изготовлении на действующем производстве СБИС блока цифровой обработки сигналов ГНСС (глобальная навигационная спутниковая система) приемника, СБИС блока цифровой обработки сигнала из состава системы спутниковой связи, СБИС бортового комплекса управления для малых космических аппаратов. Результаты практического применения диссертационного исследования подтверждены актами о внедрении.

Результаты работы внедрены в процесс подготовки кадров высшей квалификации ФГАОУ ВО “Сибирский федеральный университет”.

Положения, выносимые на защиту:

1. Разработанный метод высокоуровневого синтеза цифровых СБИС на основе функционально-потоковой парадигмы(ФПП) позволяет реализовать архитектурную независимость и переносимость исходного высокоуровневого описания СБИС на различные целевые платформы.
2. Разработанные модели и алгоритмы высокоуровневого синтеза позволяют автоматически выполнять преобразование высокоуровневого представления на функционально-потоковом языке параллельного программирования в низкоуровневое представление на языке описания аппаратуры и осуществлять свертку параллелизма исходного высокоуровневого описания СБИС.
3. Разработанное инструментальное средство синтеза описания СБИС позволяют реализовать автоматическое преобразование высокоуровневого архитектурно-независимого представления цифровых схем в низкоуровневые архитектурно-зависимые представления для разных целевых платформ.

Степень достоверности результатов подтверждается корректным использованием концепций и подходов в области теории параллельного программирования, методов обработки данных, теории графов, теории объектно-ориентированного программирования. Теоретические результаты подтверждены практической реализацией зарегистрированного в Федеральной службе по интеллектуальной собственности

комплекта прикладного программного обеспечения, внедрением на действующее производство сложно-функциональных блоков и СБИС.

Апробация работы. Основные положения и результаты исследований докладывались и обсуждались на открытых межкафедральных семинарах ИКИТ СФУ, открытых лекциях и семинарах НУЛ «Микропроцессорные системы СФУ», а так же на отраслевых, Всероссийских и Международных конференциях и семинарах: Международной конференции «Решетневские чтения» (Красноярск, 2015 г.); II и III Всероссийских научно-технических конференциях «Системы связи и радионавигации» (Красноярск, 2015, 2016 г.г.); 5-й Всероссийской научно-технической конференции «Суперкомпьютерные технологии» (Геленджик 2018 г.); Международной IEEE-сибирской конференции по управлению и связи «SIBCON-2019». (Томск, 2019 г.), IX Международной научно-практической конференции “Фундаментальные основы инновационного развития науки и образования”, г. Пенза, 2020.

Соответствие специальности. Диссертация соответствует паспорту специальности 2.3.5. «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»: пункту №1 (Модели, методы и алгоритмы проектирования, анализа, трансформации, верификации и тестирования программ и программных систем), пункту № 2 (Языки программирования и системы программирования, семантика программ) и пункту № 8 (Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования).

Личный вклад автора. В диссертации представлены результаты работы, в которых автору принадлежит определяющая роль. Постановка задачи и разработка концепции метода высокоуровневого синтеза проводилась с научным руководителем.

Публикации. По результатам диссертационных исследований и их практического применения опубликовано 17 печатных работ, из них 5 в ведущих рецензируемых журналах, включенных в список ВАК РФ, 3 работы в рецензируемых изданиях, индексируемых в ведущих зарубежных БД WoS/Scopus. Получены регистрационные свидетельства на 8 программ для ЭВМ.

Структура и объем работы. Диссертационная работа изложена на 122 страницах и состоит из введения, четырех глав, заключения, списка литературы из 103 источников и 3 приложений.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы, формулируются цель и задачи исследования, описываются методы исследования, излагаются основные положения научной новизны, значение работы для теории и практики. Приведен список положений, выносимых на защиту. Изложена структура диссертации и краткое содержание работы по главам.

В первой главе изложены результаты анализа маршрутов и способов проектирования СБИС. Определены основные тенденции развития при проектировании однокристальных систем. Выделены недостатки и достоинства существующих методов и определены перспективные подходы к проектированию. Рассматриваются известные в данной области научные направления. Определены роль и место языковых средств описания аппаратуры в современной иерархии системного проектирования. Рассмотрены модели вычислений, парадигмы и языки программирования с точки зрения их применимости к области разработки СБИС. На основе проведенного анали-

за определены основные задачи диссертационного исследования и требования к разрабатываемому методу высокоуровневого синтеза СБИС.

Среди рассмотренных широко распространённых методов описания архитектуры СБИС выделены: описания схемы и соединений (netlist), функциональное описание схемы СБИС на уровне регистровых передач и поведенческое описание.

Определено, что развитие способов описания архитектуры СБИС идет по пути повышения уровня абстракции несколькими основными путями, среди которых следует выделить: расширение существующих языков описания аппаратуры (HDL – Hardware Description Language) – конструкциями для более высокоуровневого поведенческого описания, создание инструментов синтеза архитектурных решений СБИС посредством высокоуровневых языков, применяющихся для разработки программного обеспечения, создание новых языков высокоуровневого описания аппаратуры на основе языков программирования.

Рассмотрены основные маршруты проектирования СБИС: традиционный маршрут нисходящего проектирования и маршрут с абстракцией проекта на системном уровне (ESL – Entire System Level).

Определено что при традиционном маршруте архитектура СБИС представляется в виде трехуровневой модели. При этом представлении выделяют: функциональный, структурный и уровень топологии (геометрии) микросхемы. (Рисунок 1). Традиционный маршрут проектирования основывается на представлении проекта на одном из HDL-языков и представляет собой последовательный спуск по уровням.

Выявлены следующие основные недостатки традиционного маршрута: наличие семантического разрыва между представлениями проекта на системном уровне и уровне регистровых передач (RTL – Register Transfer Level), изначальное разделение программной и аппаратной составляющих с последовательным итерационным проектированием и раздельным моделированием каждой, частичное или полное ручное преобразование проекта при переходе от уровня к уровню с жесткой привязкой применяемых библиотек к выбранной платформе, ручная разработка программного обеспечения процессорной части, а так же ограниченные возможности анализа альтернативных вариантов разрабатываемой системы.

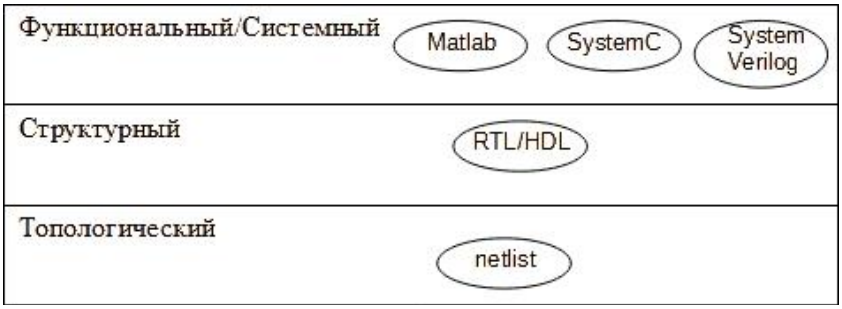

Рисунок 1 – Уровни модели в проектировании СБИС.

При ESL-проектировании разработка осуществляется «сверху вниз». При этом используется принцип имитационного моделирования, при котором в разрабатываемой модели четко выделены структура, связи между компонентами и способы обмена информацией (Рисунок 2).

При ESL проектировании верхний уровень абстракции состоит из уровня сообщений (Message Level) и уровня передач (Transaction Level). Переход на RTL-уровень осуществляется либо вручную - разработкой программного кода на языке

HDL по полученной модели, либо полуавтоматически из описания на языке C генерируется Verilog/VHDL-код. Полная автоматизация перехода на RTL невозможна, исключение составляет синтезируемый код SystemC.

HLS (High level synthesis) методы основаны использовании универсальных высокоуровневых языков программирования, изначально не предназначенных для описания СБИС. Переход к структурному уровню осуществляется полуавтоматическим преобразованием высокоуровневого описания в описание на HDL языке.

Обобщенная характеристика существующих методов приведена в таблице 1.

Таблица 1. Сравнение методов проектирования

Метод	**Недостатки**
Традиционный	Семантический разрыв между системным и структурным уровнем Ручное преобразование между уровнями
ESL	Семантический разрыв между системным и структурным уровнем
HLS	Ручное либо полуавтоматическое преобразование при переходе между системным и структурным уровнем

Показано что СБИС представляет собой схему параллельно-конвейерной обработки данных. При этом эффективность реализации такой схемы зависит от метода представления и способа реализации исходного параллельного алгоритма, а также результата синтеза вентильной структуры СБИС для целевой платформы с учетом имеющихся ресурсных ограничений.

Определено, что при проектировании цифровых схем требуется подход, обеспечивающий максимальное абстрагирование исходных алгоритмов от архитектуры целевого кристалла и реализующий механизм перехода на RTL-уровень с поддержкой параллелизма на уровне операций. Необходим такой способ представления алгоритмов на самом верхнем уровне иерархии, который при последующем нисходящем проектировании обеспечивал бы сохранение и преобразование параллелизма, задаваемого на самом верхнем уровне.

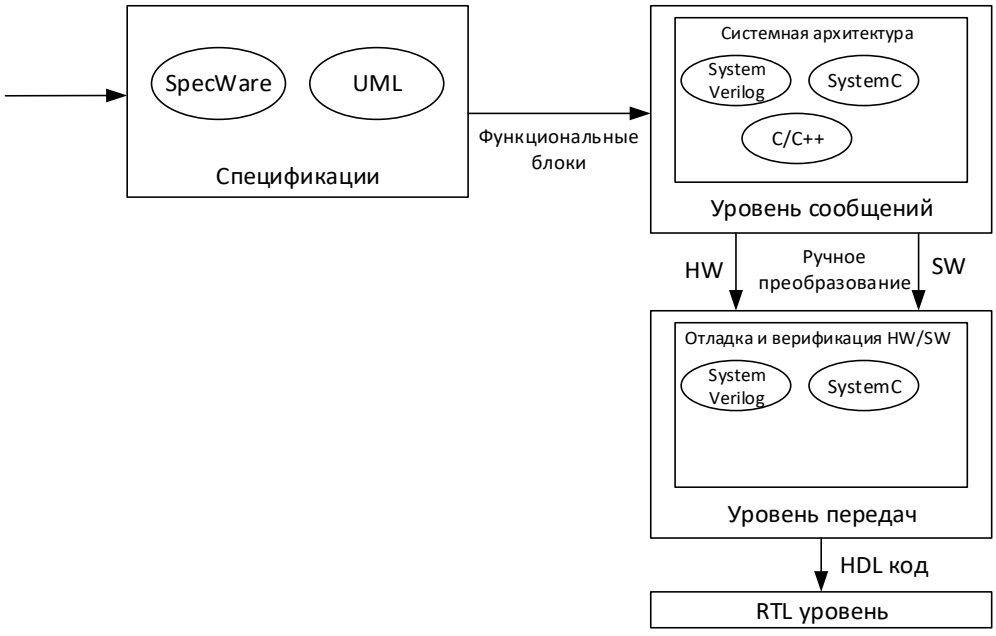

Рисунок 2 – ESL-Проектирование

Результаты анализа современных методов и маршрутов высокоуровневого синтеза СБИС показывают, что при описании исходного параллельного алгоритма последний либо изначально представляется параллельным, либо его последовательное представление поэтапно заменяется параллельным вручную или в полуавтоматическом режиме. Аналогично решаются задачи разработки программ для параллельных вычислительных систем и задачи эффективного распараллеливания и переносимости ПО на различные параллельные вычислительные архитектуры. В данном случае эффективные решения получают при использовании соответствующих парадигм параллельного программирования и специальных языковых и инструментальных средств. Следовательно, решение означенных проблем может быть найдено в области технологий параллельного программирования.

Рассмотрены языковые средства и модели вычислений, применяемые для описания СБИС и разработки переносимого параллельного программного обеспечения. Определено, что для обеспечения архитектурной независимости требуется изменения степени параллелизма без изменения исходного алгоритма. Определены требования к парадигме параллельных вычислений служащей основой для создания технологии архитектурно-независимого синтеза СБИС, а именно: отсутствие явного управления вычислениями (управление по готовности данных), модель потока данных и параллелизм на уровне операций. При этом предложено обеспечить переносимость параллельных алгоритмов за счет свертки параллелизма.

Анализ языковых средств позволил выделить следующие группы языков для синтеза СБИС: универсальные высокоуровневые языки, проблемно-ориентированные языки, функциональные языки.

Выделены основные недостатки применяемых для синтеза проектных решений языков, среди которых основными являются: семантический разрыв между высокоуровневым описанием алгоритма и низкоуровневым представлением СБИС в универсальных языках, отсутствие учета параллелизма на уровне операций, свойственного архитектуре СБИС и архитектурная зависимость для специализированных языков разработки.

Показано, что описание исходного алгоритма СБИС для высокоуровневого синтеза на основе функционально-потокового подхода позволяет уменьшить семантический разрыв и является перспективным.

Полученные результаты позволили перейти к разработке метода и алгоритмов преобразования архитектурно-независимого описания интегральных схем в архитектурно-зависимые представления на языках описания аппаратуры.

Во второй главе изложены результаты формирования функционально-потоковой модели для синтеза СБИС в том числе: введена статическая система типов, исключены задержанные вычисления и предложен метод их преобразования при переходе на платформу СБИС, введено преобразование рекурсии в итеративную схему с последующим переходом к конвейерной схеме. Предложены методы преобразования параллельных списков. Для реализации метода сокращения параллелизма, при переходе на платформы с разными ресурсными ограничениями, предложен метод оценки максимальной и минимальной длины конвейера (степени параллелизма) задачи.

Модифицированная функционально-потоковая **модель** для СБИС включает в себя:

- статическую систему типов;
- преобразование задержанных вычислений;
- преобразование рекурсии;

- преобразование параллелизма;
- модифицированное представление модели на основе ациклического графа (HDL граф).

Метод архитектурно-независимого синтеза основан на следующих принципах:

- высокоуровневое архитектурно-независимое описание алгоритма функционирования СБИС;
- отсутствие управления вычислениями в высокоуровневом описании;
- переход на низкоуровневое архитектурно-зависимое описание путем преобразования параллелизма и введение конкретной стратегии управления вычислениями.

Обосновано применение и предложены модификации языка ФПП, в том числе: добавлены скалярные и векторные типы данных, а также введен механизм преобразования (приведения) типов. Введены две категории типов – базовые и составные. К базовым типам относятся:

1. целые числа со знаком (int);
2. целые числа без знака (uint);
3. логические значения (bool);
4. сигнальный тип(signal);

К составным типам относится массив (список) - набор однотипных данных.

Разработаны правила вычисления типов операций на основе типов аргументов и значения функции. Эти типы могут быть заданы программистом либо определены на этапе синтеза при контроле и автоматическом назначении типов. Для автоматического вычисления и назначения типов достаточно только задание типов входных и выходных аргументов функции верхнего уровня. При определении типа аргумента функции верхнего уровня в процессе преобразования типов для каждой операции вычисляется тип и разрядность результата. Автоматизированное вычисление типов гарантирует отсутствие ошибок потери точности.

Введено ограничение на динамическое изменение размерности списков в процессе вычислений. Для этого размерность всех списков определяется на этапе трансляции. При отсутствии определения размерности списка в исходном коде размерность вычисляется из результатов операции либо исходя из разрядности размерности списка.

Поскольку на платформе СБИС отсутствует динамическое выделение памяти, что делает невозможным реализацию рекурсивных вычислений с неизвестной на этапе трансляции глубиной рекурсии, предложено преобразовывать рекурсивные вычисления в итеративные с использованием хвостовой рекурсии и заданием глубины рекурсии на этапе трансляции. Размерность аргумента рекурсии задана непосредственно в исходном коде, либо определяется как максимальная исходя из размерности длины списка. Зная размерность можно определить глубину рекурсии. Дальнейшие преобразования рекурсии в цикл происходят путем преобразования в итеративную либо конвейерную схему в зависимости от ресурсных ограничений и глубины рекурсии.

Методы преобразования параллельных списков включают в себя раскрытие списков по данным либо по функции, раскрытие вложенных списков.

Применение функции к параллельному списку данных преобразуется в параллельный список результата. Каждый элемент списка формируется как исходный элемент списка данных, обработанный функцией:

[m1, m2,m3, …, mn] : H => [m1 : H, m2 : H, m3:H …, mn : H] .

Аналогично преобразуется параллельный список функций и элемент данных:

m : [H1, …, Hn] => [m : H1, …, m : Hn] .

В общем случае в параллельных списках данных и функций возможны вложенные параллельные списки. По правилам языка порядок их преобразований не оговаривается. Вложенность списков при преобразованиях может быть сохранена. Результат функции над вложенными данными будет следующим:

[n1, [n2, [n3], n4, n5]]:f1 => [n1:f1, [n2, [n3], n4, n5]:f1] →

[n1:f1, n2:f1, [n3, n4]:f1, n5:f1] → [n1:f1, n2:f1, n3:f1, n4:f1, n5:f1] .

Для реализации механизма отложенных вычислений при выборе альтернативных вариантов вычислений на платформе СБИС введен сигнал “валидности” данных, который через схему выбора подключен к одному из блоков альтернативных вариантов вычислений. Для поддержки реализации схем ветвлений без отложенных вычислений введено понятие signal. В результате отложенные вычисления преобразуются в схему выбора (рисунок 3). В данной схеме f1, f2 и f3 – элементы задержанного списка, преобразованные в схему выбора.

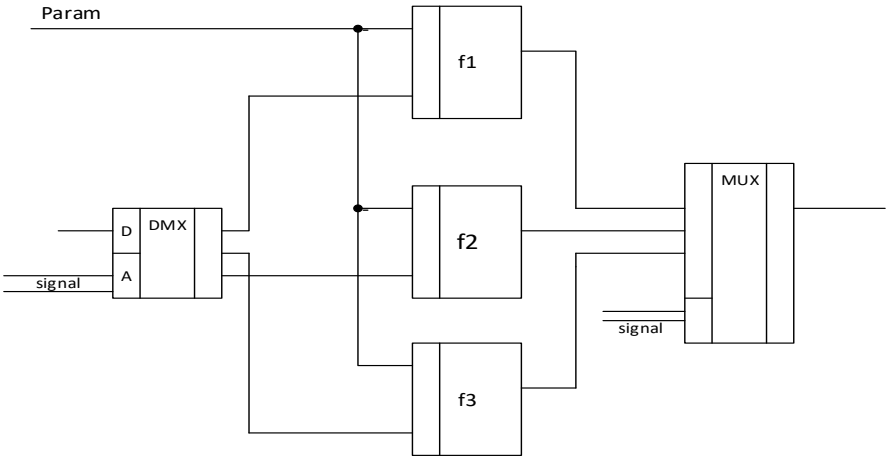

Рисунок 3 – Результат синтеза схемы выбора в отложенных вычислениях (MUX – мультиплексор, DMX – демультиплексор, f1,f2,f3 – элементы списка)

Исходный алгоритм функционирования СБИС описывается на языке ФПП. Архитектурная независимость при использовании предложенного метода обеспечивается неизменным исходным описанием алгоритма, из которого в процессе синтеза схемы СБИС, используя разные типы данных и заданные ресурсные ограничения целевых платформ, получают архитектурно-зависимые варианты реализаций с разным параллелизмом и используемыми ресурсами СБИС.

Первым этапом синтеза выполняется трансляция исходного кода на ФПП языке. При трансляции формируются промежуточные представления: информационный граф (ИГ), задающий схему обработки данных и зависимости по данным, а также управляющий граф (УГ), задающий управление вычислениями. В простейшем случае промежуточное представление включает в себя только информационный граф. В этом случае управляющий граф можно считать заданным неявно. Например, для исходного текста быстрого преобразования Фурье (БПФ), будет сформирован информационный граф (рис. 4).

Для учета всех изменений **модели** и представления архитектурно-зависимого уровня введен дополнительный промежуточный слой, названный HDL-графом. Дан-

ный слой формируется из исходного информационного графа программы путем задания типов данных и содержит все изменения, вносимые в модель. После преобразований в HDL-граф, информационный граф (рисунок 4), преобразуется в вид, приведенный на рисунке 5.

При переходе от модели с неограниченными ресурсами к реальным, выполняется преобразование максимального параллелизма, заложенного в исходный алгоритм на ФПП языке, под конкретные ограничения целевой платформы СБИС. Такие преобразования меняют степень параллелизма алгоритма – т. е. количество параллельных операций, выполняемых в различных частях алгоритма. Крайними вариантами степени параллелизма при реализации алгоритма являются полностью последовательное выполнение и выполнение алгоритма с той степенью параллелизма, которая заложена в исходное описание.

Все преобразования, меняющие порядок и стратегию вычислений, выполняются над управляющим графом и не зависят от информационного графа. Изначальное построение УГ осуществляется транслятором языка Пифагор с использованием принципа управления по готовности данных – максимальной степени параллельности алгоритма, заложенной в его описание. Управление вычислительными ресурсами в таком варианте не задается.

Согласно предложенного **метода** для перехода на платформу СБИС выполняется преобразование информационного графа к конвейерной схеме. Конвейерная схема вычислений представляет собой ярусно-параллельную форму информационного графа. Для решения задачи преобразования параллелизма под конкретные ресурсные выполняется: определение границ изменения параллелизма, алгоритм изменения параллелизма и оценка результата по ресурсным ограничениям.

При оценке параллелизма рассматриваются крайние варианты (последовательное выполнение и максимально-параллельный вариант) при этом выполняется расчет количества стадий конвейера для последовательной и максимально-параллельной схемы. Изменение степени параллелизма предполагает изменение количества стадий конвейера и связанное с этим изменение ресурса.

Оценка крайних вариантов производится по двум метрикам: длине конвейера L_k (задержке схемы) и максимальному количеству параллельных операций в одной стадии конвейера P_k.

$$P_k = \max_{i=\{1..N\}}\{|V_i|\},$$

где V_i – подмножество вершин i-го яруса ЯПФ формы графа.

Между этими двумя вариантами возможно преобразование параллелизма и получение промежуточных вариантов по задержке схемы и параллелизму (занимаемым ресурсам).

Для оценки задержки выполнения схемы разработан следующий **алгоритм** поиска критического пути в управляющем графе:

1. Сгенерировать УГ для последовательного и максимально-параллельного варианта управления вычислениями;
2. Провести оптимизирующие преобразования полученных графов;
3. Вычислить критический путь графа.

Управляющий граф в данном алгоритме синтезируются по HDL-графу, так как HDL-граф является оптимизированным и содержит типы данных. Учет типов данных позволяет точно оценить количество параллельных операций в параллельных участках схемы - их размерность известна.

Для учета задержки выполнения при синтезе УГ каждой вершине присваивается метрика, равная задержке выполнения данной вершины. Список предложенных метрик приведен в таблице 2.

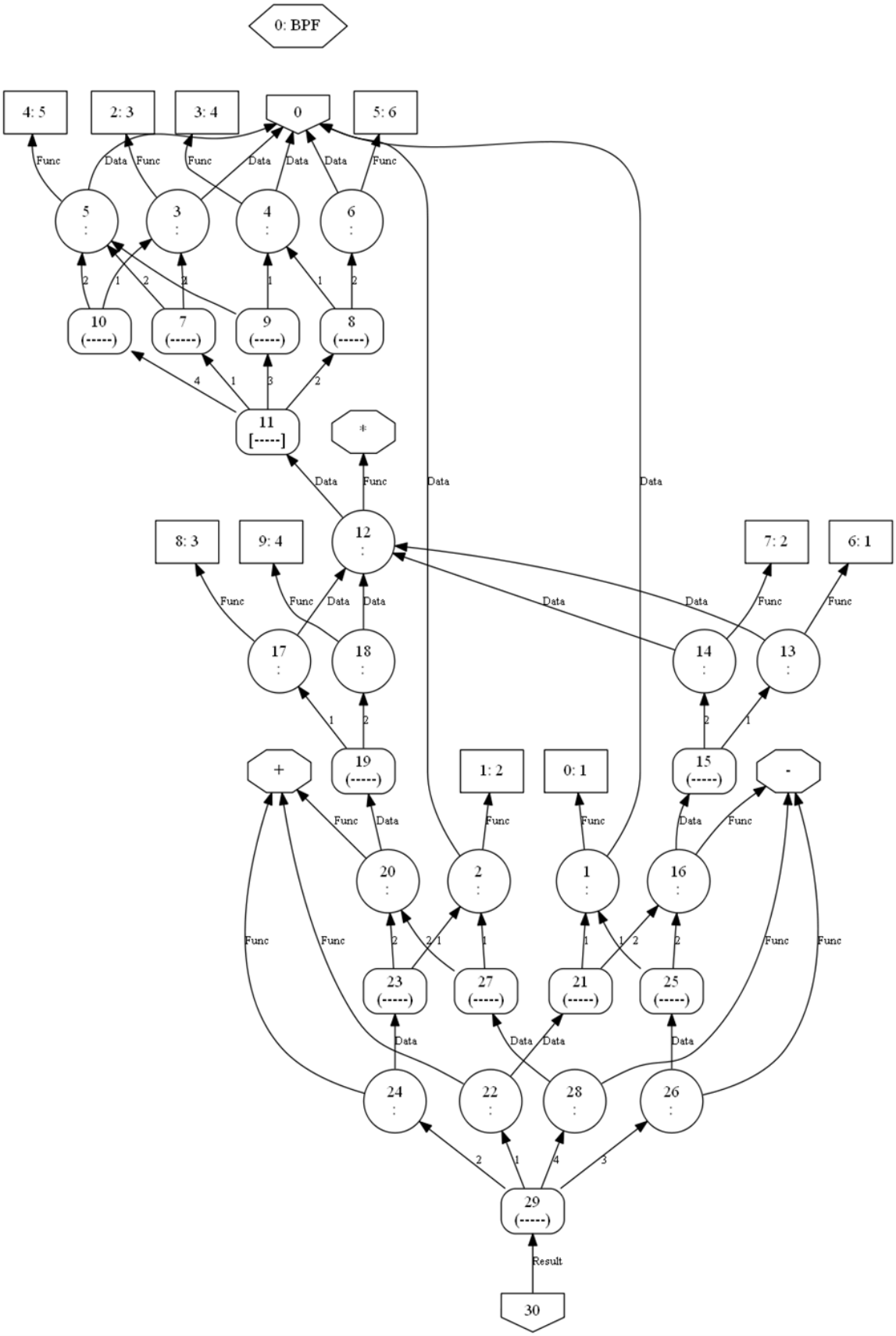

Рисунок 4 – Информационный граф БПФ

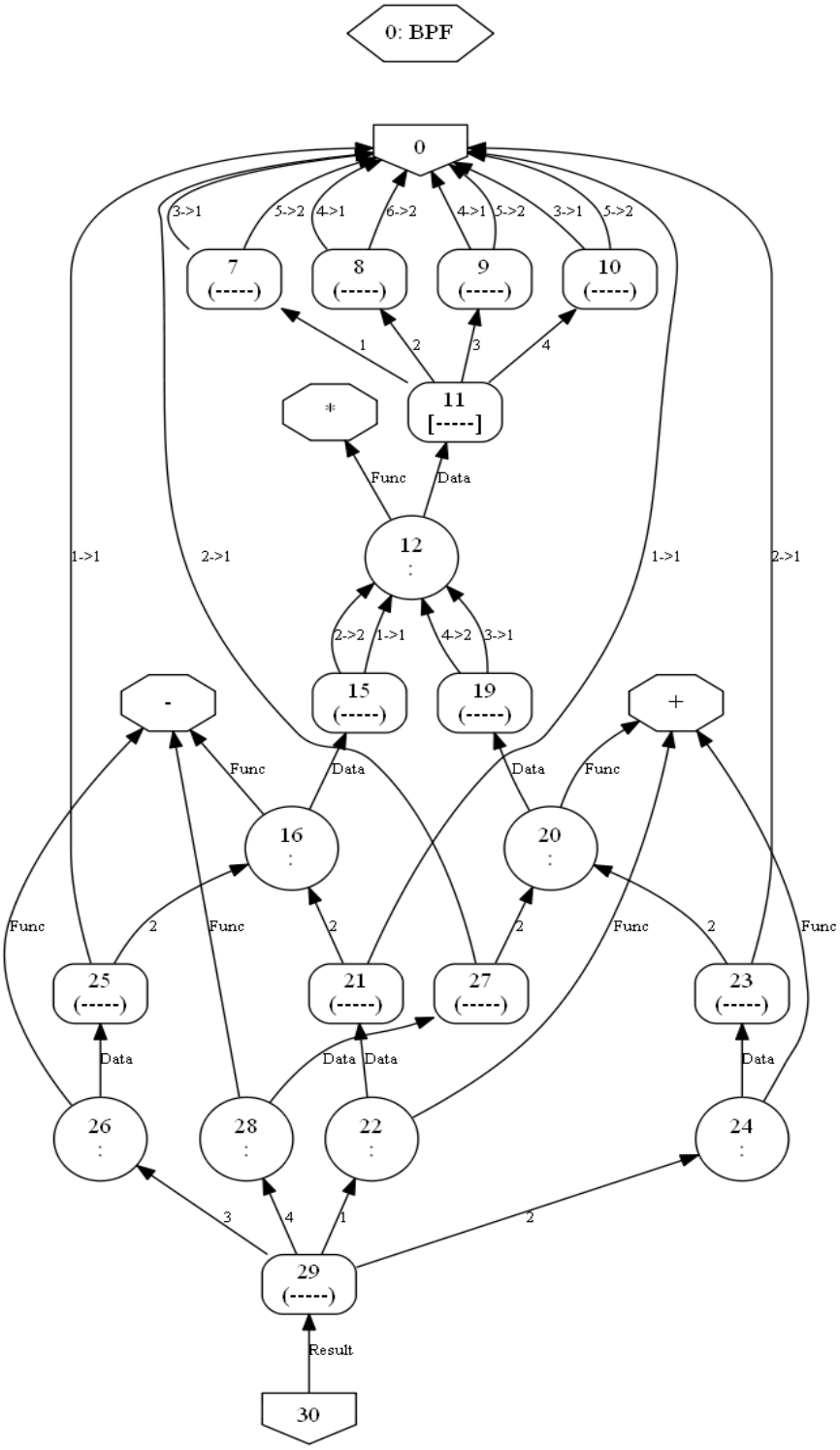

Рисунок 5 – HDL граф БПФ

Таблица 2. Значения метрик для определения границ изменения параллелизма

Тип вершины графа	**Последовательный вариант**	**Максимально-параллельный вариант**
Аргумент	1	1
Результат	1	1
Список данных	1	1
Интерпретация функции	Задержка функции	Задержка функции
Интерпретация параллельного списка	Длина списка	1

Для формирования последовательного варианта при генерации УГ выполняется обход узлов информационного графа в последовательности, не нарушающей корректность вычислений. Максимально-параллельный вариант формируется при генерации УГ для варианта по готовности данных (вариант, генерируемый по умолчанию), при условии одновременной готовности всех данных на входе функции. Длина конвейера в максимально-параллельной форме Lk_min и в последовательной форме Lk_max.

При определении максимального количества параллельных операций Pk управляющий граф для максимально-параллельного варианта приводится к ярусно-параллельной форме (ЯПФ) с высотой, равной Lk_min, при этом определяется максимальная ширина ЯПФ. В результате на выходе алгоритма получаются оценки максимальной (Lk_max) и минимальной (Lk_min) задержки схемы, которым соответствует максимальный (Pk_max) и минимальный (Pk_min) занимаемый на системе ресурс.

В третей главе приведены результаты разработки инструментальных средств архитектурно-независимой поддержки проектирования и проектирования СБИС.

Предложен маршрут проектирования, состоящий из следующих основных этапов:

1. Разработка алгоритмов функционирования СБИС на языке ФПП программирования. На данном этапе осуществляется высокоуровневое архитектурно-независимое представление исходных алгоритмов с максимальной степенью параллелизма.

2. Тестирование программного кода, отладка алгоритмов функционирования СБИС без привязки к целевой платформе.

3. Синтез промежуточного представления СБИС в виде информационного и управляющего графов.

4. Типизация данных.

5. Оптимизация промежуточных представлений.

6. Синтез управляющего и информационного HDL-графа. На этом этапе выполняется промежуточный синтез графа данных и управления из информационного и управляющего графа.

7. Решение задачи планирования и распределения ресурсов с изменением степени параллелизма реализации алгоритма, выбор схемы управления вычислениями.

8. Синтез схемы обработки данных и сигналов управления.

9. Синтез описания на HDL языке.

На рисунке 6 изображен состав и взаимосвязь программных средств.

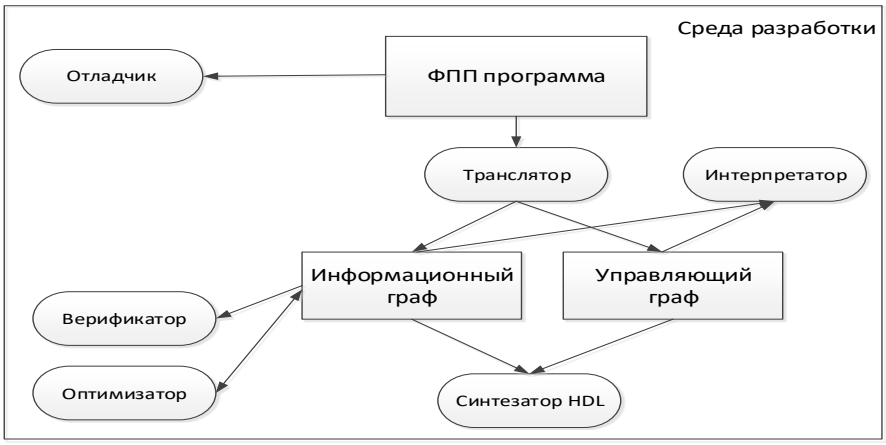

Рисунок 6 - Структура инструментальных средств поддержки проектирования

При синтезе схемы обработки данных по HDL графу соблюдаются следующие правила:

– Одна вершина интерпретации – один такт;
– Одна функция – один модуль;
– Каждый вызов одной и той же функции – экземпляр модуля;
– Вершины аргумента и результата – начальная и конечная стадия конвейера функции.

Алгоритм синтеза состоит из двух основных стадий: синтеза схемы обработки данных и синтеза схемы управления.

Алгоритм синтеза схемы обработки данных состоит из следующих шагов:

- преобразование HDL графа в ярусно-параллельную форму;
- обход графа от вершины аргумента к вершине результата;
- преобразование каждой вершины в соответствующую схему обработки данных СБИС.

Список операторов языка и соответствующие им схемы на СБИС для второго этапа преобразования приведены в таблице 3.

Таблица 3. Схемы синтеза основных операторов языка

Оператор	**Схема**
Арифметические операции	Соответствующий блок арифметической операции (кроме операции деления)
Операции сравнения	Логическая функция с однобитовым выходом
Оператор ?	N параллельных сравнений (компараторов)
Оператор dup	Набор регистров и синхронное присвоение каждому регистру первого элемента (A) аргумента
Оператор #	Мультиплексирование
Целое/булевское значение в качестве функции	Мультиплексор, количество входов данных определяется как $\log_2 N$, где N – разрядность значения, используемого в качестве функции
Оператор ()	Асинхронные присвоения
Внешняя функция	Экземпляр модуля

Алгоритм синтеза схемы управления заключается в синтезе автоматов для каждой вершины управляющего графа и состоит из следующих шагов:

- преобразование управляющего графа к ЯПФ форме;
- синтез схемы управляющего автомата для каждой вершины (таблица 4);
- соединение выходных сигналов управляющих автоматов с соответствующими вершинами HDL графа.

Количество автоматов обработки сигналов управления соответствует количеству типов вершин информационного графа. Список автоматов и схемы соответствующих автоматов приведены в таблице 4.

Таблица 4. Управляющие автоматы

Тип вершины	**Схема**
Аргумент	Логическое “И” размерности количества аргументов
Результат	-//-
Обработка данных (интерпретация)	Регистры хранения сигналов готовности аргумента и функции и логическое “И” формирования сигнала управления
Список данных	Счетчик до длины списка данных, сигнал формируется при переполнении счетчика
Параллельный список	Задержка для каждого элемента

Разработаны алгоритмы синтеза HDL описания СБИС из промежуточного представления ФПП программы и алгоритмы синтеза конструкций ФПП языка в блоки СБИС. На основе алгоритмов синтеза разработан синтезатор с ФПП языка в HDL описание схемы СБИС. Синтезатор позволяет преобразовывать неизменное промежуточное представление алгоритма на ФПП языке в различные варианты схем СБИС с использованием различных размерностей и типов данных и различным уровнем параллелизма схемы.

В четвертой главе рассмотрены результаты практической реализации СБИС, описана разработанная методика сравнительного анализа полученных решений и набор тестовых задач для сравниваемых методов. Изложены результаты оценки проектов.

Для сравнения методов синтеза выделены следующие классы исследуемых параметров, применяемых при идентичных условиях тестирования:

- Метрики получаемого результата разработки (быстродействие, объем, энергопотребление и т.д.);
- Косвенные метрики процесса разработки (трудоемкость разработки/тестирования, время/затраты на перенос проекта с платформы на платформу).

Быстродействие схемы определяет количественную меру результата, выдаваемого СБИС в единицу времени. Введены следующие метрики быстродействия: максимальная тактовая частота схемы и задержка схемы в тактах до выдачи результата (латентность). Сравнение этих метрик отдельно у разных схем некорректно, так как при различных алгоритмах синтеза у конвейерных схем для одного и того же случая могут быть получены результаты с более длинным конвейером (большей латентностью в тактах) и большей тактовой частотой и наоборот. Поэтому для корректного

сравнения предложено ввести композитную временную метрику – время от появления данных на входе в схему до выдачи результата, рассчитываемую по формуле:

$$T_w = C_l/F_{max},$$

где F_{max} – максимальная тактовая частота схемы, а C_l – задержка схемы в тактах (длина конвейера).

При ранжировании метрик наибольшее значение имеет метрика производительности (T_w) и метрики занимаемого ресурса по различным классам ресурсов. Улучшение данных метрик приводит к увеличению трудоемкости и времени разработки. В эталонном с точки зрения производительности варианте описания схем для каждой платформы на языках HDL время разработки увеличивается кратно количеству платформ, для которых разрабатывается проект. В разработанном в рамках работы методе (ФПП) и методах, выбранных для сравнения (MATLAB, HLS), увеличение времени при разработке для нескольких платформ практически не происходит (+/-10%). При этом для существующих методов высокоуровневой разработки HLS, MATLAB существенно (в несколько раз) увеличивается ресурс либо падает в 2-3 раза производительность схемы. Исходя из вышеизложенного критерием эффективности метода является сравнимая с ручной разработкой на HDL производительность (разница ~20%) и существенное уменьшение количества требуемого ресурса СБИС по сравнению с существующими методами (HLS, MATLAB).

Оценка метрики занимаемого ресурса производилась в базисе ПЛИС. Использовались следующие основные группы ресурсов: логические ячейки, триггеры, встроенные специализированные блоки (блочная память, специализированные арифметические блоки – умножители, DSP и блоки ввода/вывода). Для сравнения выбраны наиболее распространённые методы синтеза: традиционный маршрут проектирования с описанием проекта на языке HDL (на диаграммах рисунки 7-10 обозначен как xHDL), высокоуровневый синтез на C-подобном языке (HLS) и высокоуровневый синтез в среде MATLAB с последующим автоматическим синтезом проекта в HDL описание.

Для тестовых задач выделены два класса алгоритмов: алгоритмы обработки данных и управления. Типы задач взяты из распространённого пакета CHStonex. При тестировании использованы следующие классы задач: операции с комплексными числами, операции с матрицами и алгоритмы цифровой обработки сигналов.

Для сравнения результаты были приведены к относительному масштабу, где для производительности за 100% принимался самый быстрый вариант, а для сравнения ресурсов за 100% принимался результат с наименьшим количеством занимаемого ресурса по категориям логических ячеек, триггеров и умножителей (рисунки 7-10).

В результате анализа по производительности на всём наборе тестовых задач методы распределились в следующем порядке (в порядке убывания): HDL, ФПП, MATLAB, HLS.

Из диаграмм видно, что метод HLS занимает самый большой ресурс, многократно превышающий остальные (в 3 задачах по ресурсу логических ячеек, в 4 задачах по ресурсу триггеров и 1 задаче по умножителям). Это связано с неэффективностью распараллеливания изначально последовательного описания алгоритма.

Реализации тестовых задач в среде MATLAB по занимаемому ресурсу логических ячеек и триггеров в целом отличается от HDL реализации в 2-4 раза.

Неоптимальное использование ресурса в последовательных вариантах показывается неудовлетворительную эффективность преобразования параллелизма в

MATLAB. Предложенный ФПП-метод показывает сравнимое использование ресурса, при сравнимой с HDL реализациями производительностью.

Исключение составили задачи обработки матриц параллельно по ресурсу триггеров. Анализ полученных вариантов показал, что избыточное использование ресурса триггеров возникает из-за большего количества стадий конвейера в этих задачах.

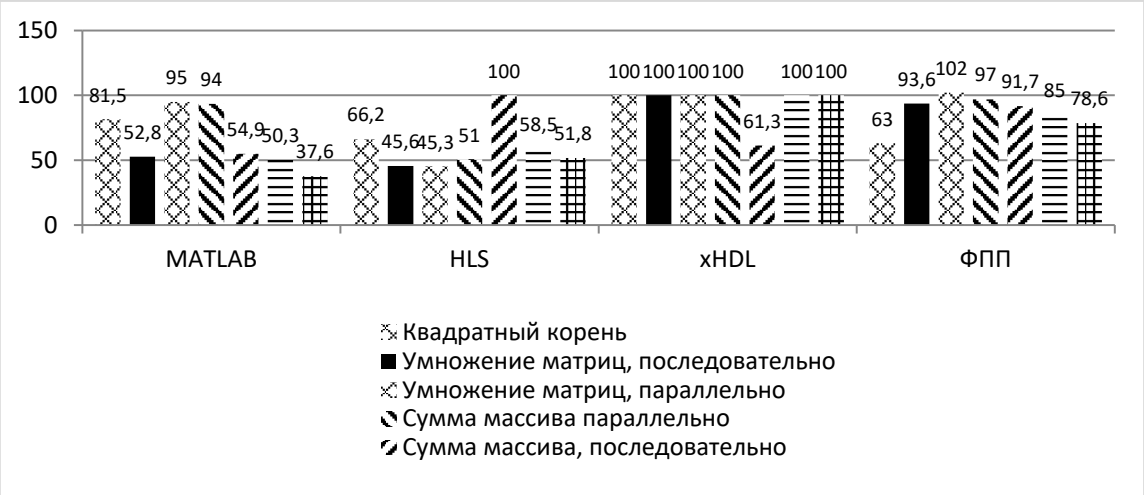

Рисунок 7 – Производительность методов на тестовых задачах

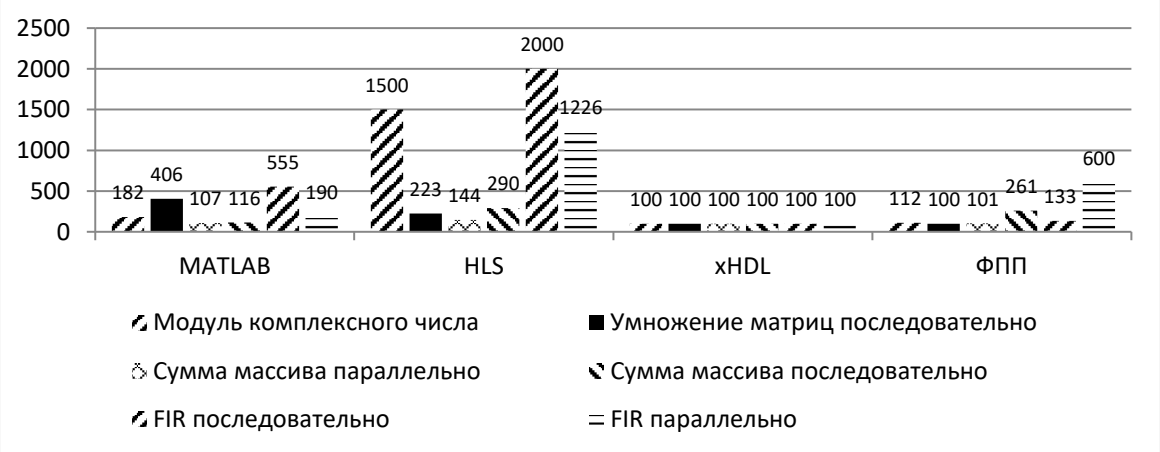

Рисунок 8 – Ресурс логических ячеек

Большее количество стадий конвейера возникает из-за изначального описания с максимальным параллелизмом на уровне операций. Эти примеры показывают, что в данном случае эффективнее объединить стадии конвейера без потери производительности.

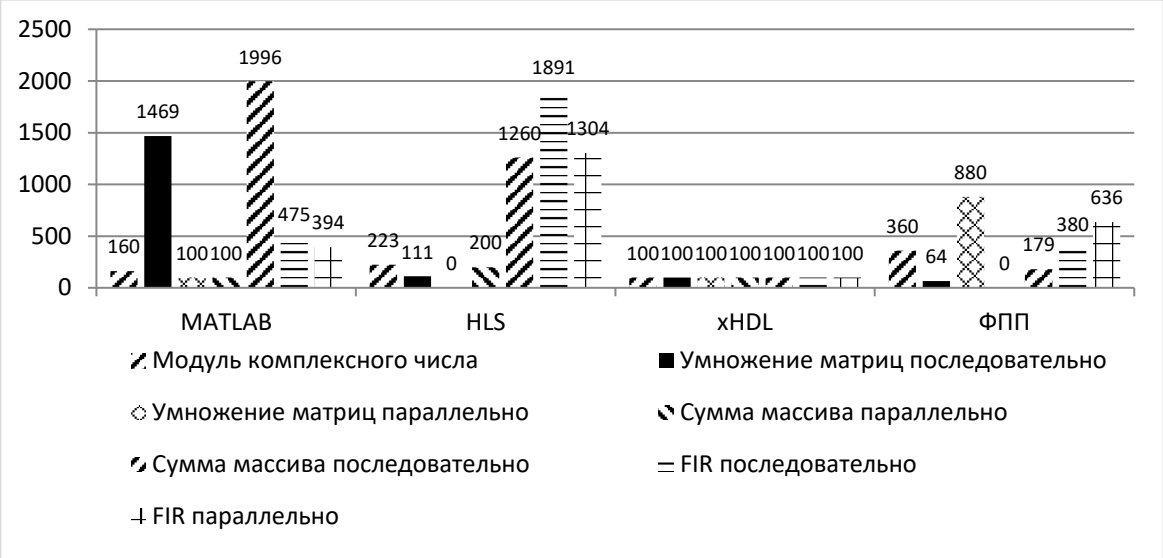

Рисунок 9 – Ресурс триггеров

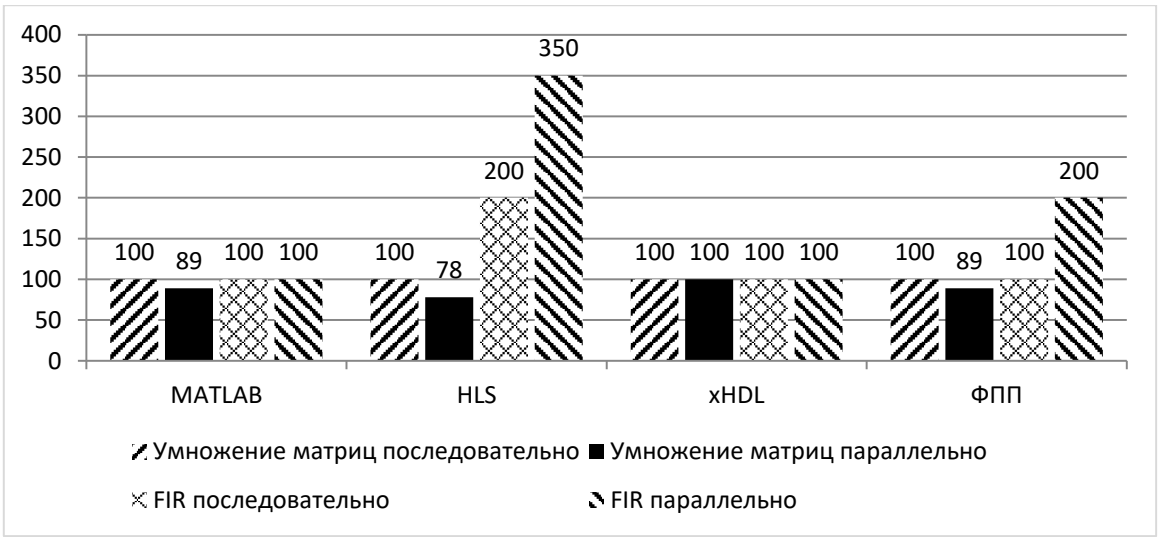

Рисунок 10 – Ресурс DSP

С использование разработанного комплекта ПО производилась разработка 3 комплектов СБИС в рамках контрактов 14.578.21.0116, 02.G25.31.0202 СБИС угломерного навигационного приемника и 14.578.21.0021 СБИС бортового комплекса управления для малых космических аппаратов.

В рамках первых двух контрактов разрабатывалась СБИС цифровой обработки сигналов для первичной обработки сигналов угломерного навигационного приемника. СБИС первичной обработки сигналов угломерного навигационного приемника реализовывалась для 2 платформ: ПЛИС и БМК (базовый матричный кристалл) Алмаз-13. Исходное описание алгоритма было выполнено на ФПП языке с использованием предложенного высокоуровневого метода синтеза. С помощью разработанного ПО из одного исходного описания был произведен синтез 2 вариантов схемы под разные платформы: ПЛИС и БМК. Для платформы ПЛИС существовал используемый для отладки вариант описания СБИС приемника, выполненный ручным описанием на языке HDL, с которым производилось сравнение эффективности разработки с использованием разработанного метода. Результаты показали сравнимую производитель-

ность и ресурс СБИС при двукратной экономии времени за счет использования одного описания для обоих платформ ПЛИС и БМК.

В рамках проекта СБИС бортового комплекса управления для малых космических аппаратов разрабатывалось описание СБИС для 2 платформ: отладочной на базе ПЛИС и радиационно-стойкой СБИС. В результате оба варианта СБИС были реализованы из одного высокоуровневого описания с уменьшением времени разработки в 2 раза.

ЗАКЛЮЧЕНИЕ

Работа посвящена повышению надежности и скорости разработки цифровых СБИС посредством разработки метода и инструментальных средств для высокоуровневого синтеза цифровых однокристальных систем. Полученные результаты имеют значение для развития вычислительной техники, включающих языки программирования, технологии программирования, системное и прикладное программное обеспечение для проектирования вычислительной техники.

1. Основным научным результатом диссертационных исследований является метод архитектурно-независимого высокоуровневого синтеза цифровых однокристальных систем, базирующийся на функционально-потоковой парадигме параллельных вычислений, позволяющий обеспечить архитектурную независимость исходного описания СБИС.

2. Предложен метод синтеза описания СБИС из функционально-потокового описания исходного алгоритма с массовым параллелизмом. ФПП-модель изменена с целью её адаптации для разработки СБИС.

3. Разработаны алгоритмы оценки границ изменения параллелизма исходного ФПП описания при переходе к различным вариантам реализации схемы на СБИС. Предложенные алгоритмы позволяют реализовывать схемы от полностью последовательного варианта до максимально-параллельного.

4. Разработаны алгоритмы преобразования (синтеза) языковых конструкций и промежуточного представления ФПП языка в описание схемы СБИС на языках описания аппаратуры.

5. На основе разработанных алгоритмов синтеза создано ПО синтезатора с ФПП языка на языки HDL.

6. Предложена методика сравнительного анализа методов синтеза, определены метрики тестирования и получены результаты решения практических задач. Результаты сравнительного анализа показали, что предлагаемый метод синтеза обеспечивает архитектурную независимость и переносимость и при этом показывает сравнимую с реализацией на HDL языках производительность. При этом получение различных вариантов по степени параллелизма из одного описания алгоритма на ФПП языке не сказывается на эффективности реализации.

7. Получены практические результаты реализации сложно-функциональных блоков и кристаллов СБИС на основе разработанного метода, показана эффективность разработанного метода.

Развитие исследований по данному направлению возможно в сторону разработки алгоритмов преобразования параллелизма и изменений стратегий управления вычислениями при переходе к архитектурным вариантам. В разрабатываемых алгоритмах необходимо реализовать учет метрик производительности (обмен метриками между ФПП и HDL этапами синтеза), получаемых после синтеза архитектурного решения из HDL описания. Это позволит учесть метрики производительности схемы на

этапе преобразования параллелизма, что дополнительно увеличит эффективность алгоритма синтеза.

СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

В изданиях, рекомендованных ВАК:

1. Рыженко И.Н. Технология архитектурно-независимого, высокоуровневого синтеза сверхбольших интегральных схем / Рыженко И.Н., Непомнящий О.В., Легалов А.И. // Доклады Академии наук высшей школы Российской Федерации. – 2014. - № 1(22). - С. 93–103.
2. Рыженко И.Н. Оптимизация параллельных списков функционально-потокового языка программирования ПИФАГОР / Васильев В.С., Матковский И.В., Рыженко И.Н. // Системы. Методы. Технологии. – 2014. - № 3. - С. 102–107.
3. Рыженко И.Н. Метод архитектурно-независимого высокоуровневого синтеза / Рыженко И.Н., Непомнящий О.В., Легалов А.И. // Известия ЮФУ. Технические науки – 2018 – 8 – С. 36 – 47.
4. Рыженко И.Н. Метод высокоуровневого синтеза и программный инструментарий для описания алгоритмов функционирования СБИС. / Рыженко И.Н., Непомнящий О.В. // Программная инженерия. – 2020 – т.11 №1. – С.35-39.
5. Рыженко И. Н. Методы преобразования параллелизма в процессе высокоуровневого синтеза СБИС / А. И. Легалов, Непомнящий О. В., Рыженко И. Н., Шайдуров В. В. // Моделирование и анализ информационных систем. - 2022. - Т. 29. - № 1. - С. 60-72

В других изданиях и материалах конференций:

6. Рыженко И.Н. Проблемы и решения для проектирования специализированных бортовых вычислительных систем / Рыженко И.Н., Непомнящий О.В., Андреев А.С., Комаров А.А. // Интеллект и наука: труды XIII Междунар. науч. – практ. конф. – Железногорск – 2013 - С. 36–37.
7. Рыженко И.Н. Увеличение максимальной корректируемой ошибки при реализации алгоритма Фитца / Рыженко И.Н., Андреев А.С., Комаров А.А., Леонова А.В. // Решетневские чтения - Красноярск. – 2015 - Т. 1. № 19. - С. 234-236.
8. Рыженко И.Н. Реализация принципа повторного использования частотного ресурса на базе универсальной платформы спутникового модема / Рыженко И.Н., Андреев А.С., Комаров А.А., Леонова А.В. // II Всероссийская научно-техническая конференция «Системы связи и радионавигации», г. Красноярск. - 2015 г. - С. 40-44.
9. Рыженко И.Н. Частотная синхронизация в режиме ACM в сетях на базе протокола DVB-S2/S2X / Рыженко И.Н., Комаров А.А., Конюшко С.В. // III Всероссийская научно-техническая конференция «Системы связи и радионавигации», г. Красноярск - 2016 г. - С. 43-47.
10. Ryzhenko I.N**.** Methods and algorithms for a high-level synthesis of the very-large-scale integration / Nepomnyashchy O.V., Legalov A.I., Tyapkin V. Ryzhenko I.N**.,** Shaydurov V.V. // WSEAS Transactions on Computers ISSN / E-ISSN: 1109-2750 / 2224-2872 – 2016 - Volume 15, Art. #22 - pp. 239-247 **(индексируется в Scopus).**
11. Рыженко И.Н. Методы и средства определения частотной ошибки сигнала спутниковой связи в режиме реального времени / Непомнящий О.В., Хабаров В.А., Рыженко И.Н., Комаров А.А. // Известия вузов. Приборостроение. - С. Петербург. Спб ИТМО – 2014 - Т. 57. № 3. – С.35-39. (ВАК).
12. Рыженко И.Н. Многомерное пространственное кодирование видеоданных в каналах спутниковой связи для малых космических аппаратов / Непомнящий О.В., Митюков В.А., Рыженко И.Н., Хабаров В.А. // Наукоемкие технологии - 2015. - Т.16. №

3 - С. 66-70. (ВАК).

13. Ryzhenko I.N. The VLSI High-Level Synthesis for Building Onboard Spacecraft Control Systems / Nepomnyashchiy O.V., Ryzhenko I.N., Shaydurov V.V., N.Y. Sirotinina N.Y., Postnikov A.I. //Proceedings of the Scientific-Practical Conference "Research and Development - 2016" - Springer, Cham – 2016. (**индексируется в Scopus).**

14. Ryzhenko I.N. Microprogram control of the switching voltage regulator with the minimum duration of transient phenomena / Yablonskiy A.P., Latyshev R.A., Komarov A.A., Ryzhenko I.N. // EUROPEAN RESEARCH, сборник статей победителей VIII международной научно-практической конференции. 2017. С. 73-76.

15. Ryzhenko I.N. Carrier compensation mode implementation in satellite communication channels / Ryzhenko I.N., Lutsenko A.E., Varygin O.G., Nepomnyashchiy O.V. // IEEE: 2019 International Siberian Conference on Control and Communications (SIBCON). Tomsk - 2019 (**индексируется в Scopus и Web of Science**).

16. Рыженко И.Н. Вопросы разработки доверенного когерентного оптического транспондера для передачи сигналов 100 Гбит/с. / Волков С.А., Овсянкин С.В., Рыженко И.Н. // Москва. Наноиндустрия – 2020 – т.13, №99 – С. 324-325.

17. Рыженко И.Н. Результаты сравнения методов высокоуровневого синтеза СБИС / Непомнящий О.В., Рыженко И.Н. // Фундаментальные основы инновационного развития науки и образования. Сборник статей IX Международной научно-практической конференции, Пенза, 2020г., с. 46-51.

Свидетельства о государственной регистрации программ для ЭВМ

1. Рыженко И.Н., Комаров А.А., Непомнящий О.В. Программа синтеза описания схем на языках описания аппаратуры HDL с языка функционально-параллельного программирования «Пифагор» // Свидетельство о государственной регистрации ПО для ЭВМ № 2015619175, 26.08.2015.

2. Непомнящий О.В., Комаров А.А., Рыженко И.Н**.** Программа драйвера бортовой сети космического аппарата // Свидетельство о государственной регистрации ПО для ЭВМ №2015616896, 25.06.2015.

3. Непомнящий О.В., Рыженко И.Н**.** и др**.** Сложно-функциональный блок генераторов псевдослучайной последовательности //Свидетельство о государственной регистрации ПО для ЭВМ № 2016662259, 3.11.2016.

4. Непомнящий О.В., Комаров А.А., Рыженко И.Н**.** Свидетельство о государственной регистрации ПО для ЭВМ № 2016619836 Сложно-функциональный блок интерфейсов вычислительного узла // Свидетельство о государственной регистрации ПО для ЭВМ № 2016619836, 31.08.2016.

5. Рыженко И.Н., Комаров А.А., Андреев А. С. Программное обеспечение спутникового модема «ЯР-1040» // Свидетельство о государственной регистрации ПО для ЭВМ № 2015614726, 27.04.2015.

6. Непомнящий О.В., Рыженко И.Н., Романова Д.С., Легалов А.И. Транслятор архитектурно-независимого описания автоматных и комбинационных схем. //Свидетельство о государственной регистрации ПО для ЭВМ № 2021610682, 01.02.2021.

7. Свидетельство о государственной регистрации ПО для ЭВМ № 2014660589 “Приёмник спутникового сигнала стандарта DVB-S/S2”, Комаров А.А., Рыженко И.Н., Андреев А. С. и др., 10.10.2014.

8. Свидетельство о государственной регистрации ПО для ЭВМ № 2016619714 “Сложно-функциональный блок понижающего сумматора-ограничителя”, Непомнящий О.В., Комаров А.А., Рыженко И.Н. и др., 26.08.2016.