

На правах рукописи



**Ненашев Олег Вячеславович**

**РЕИНЖИНИРИНГ ЦИФРОВЫХ УСТРОЙСТВ И ВСТРАИВАНИЕ СРЕДСТВ  
ТЕСТИРОВАНИЯ НА БАЗЕ МНОГОУРОВНЕВЫХ МОДЕЛЕЙ**

Специальность 05.13.05 –  
Элементы и устройства вычислительной техники и систем управления

**АВТОРЕФЕРАТ**

диссертации на соискание учёной степени кандидата технических наук

Работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Санкт-Петербургский политехнический университет Петра Великого».

Научный руководитель           Кандидат технических наук, доцент  
Филиппов Алексей Семёнович

Официальные оппоненты:   Мурсаев Александр Хафизович  
доктор технических наук, профессор, профессор  
кафедры вычислительной техники ФГАОУ ВПО  
«Санкт-Петербургский государственный  
электротехнический университет «ЛЭТИ» им.  
В.И. Ульянова (Ленина)»

Суворова Елена Александровна  
кандидат технических наук, доцент, начальник  
лаборатории систем-на-кристалле ФГАОУ ВПО  
«Санкт-Петербургский государственный  
университет аэрокосмического  
приборостроения»

Ведущая организация:       ОАО «Научно-производственное объединение  
«Импульс» (г. Санкт-Петербург)

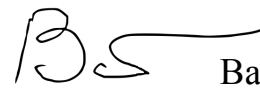
Защита состоится 24 декабря 2015 года в 16:00 в аудитории №325 9-го корпуса ФГАОУ ВО «Санкт-Петербургский политехнический университет Петра Великого» на заседании диссертационного совета Д 212.229.18 при ФГАОУ ВО «Санкт-Петербургский политехнический университет Петра Великого», расположенном по адресу: 195251, г. Санкт-Петербург, ул. Политехническая, д. 29.

С диссертацией можно ознакомиться в библиотеке ФГАОУ ВО «Санкт-Петербургский политехнический университет Петра Великого» и на сайте [www.spbstu.ru](http://www.spbstu.ru).

Автореферат разослан 19 октября 2015 года.

Учёный секретарь  
диссертационного совета

кандидат технических наук, доцент



Васильев Алексей Евгеньевич

## 1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы исследования.** В течение жизненного цикла систем автоматики и вычислительной техники требуется их модификация (реинжиниринг) для обеспечения соответствия меняющимся требованиям по надёжности, точности, тестопригодности и т.д. Задачи реинжиниринга СнК могут решаться многократно, поэтому требуется их алгоритмизация с применением специализированных инструментариев - совокупности моделей, методов и средств реинжиниринга. Актуальны задачи разработки новых моделей устройств, а также методик анализа и преобразования архитектур для широкого класса задач реинжиниринга. Создание научной базы и последующая реализация таких инструментариев могут улучшить технико-экономические и эксплуатационные характеристики СнК за счёт следующих факторов:

- снижения затрат на разработку цифровых систем путём повторного использования существующих наработок и методик реинжиниринга;
- алгоритмизации и формализации ресурсоёмких задач из процесса разработки с целью их последующей автоматизации в системах автоматизации проектирования (САПР);
- унификации процессов разработки устройств за счёт использования единого инструментария для различных задач реинжиниринга устройств;
- повышения качества проектируемых устройств за счёт минимизации влияния человеческого фактора при внесении изменений в устройство.

Одним из классов задач реинжиниринга является встраивание средств внутрисхемного тестирования (ВСВТ) в системы на кристалле (СнК). Имеются примеры решения частных задач при ВСВТ: внедрение тестовых агентов и интерфейсов, генерация подсистем тестирования, применение динамической реконфигурации при тестировании устройств на программируемых логических интегральных схемах (ПЛИС) и пр.

**Степень разработанности темы исследования.** Вопросы построения моделей устройств для задач реинжиниринга и их последующего применения рассматриваются в нескольких ключевых направлениях. Модели на базе низкоуровневых представлений на уровне логических вентилей или примитивов языков описания устройств разрабатывались Вилси, Басаргиным, Менсом, Виллисом. Применение высокоуровневых описаний на базе UML (Unified Modeling Language) рассматривалось ранее в работах Видала, Бурместера, Чена, Рамоса и др. Применение абстрактных синтаксических деревьев исходного описания устройств рассматривалось Хатчингсом, Монсоном и Похлом. Кроме академических исследований, современные САПР (например, SAMATE, DMS Software Reengineering Toolkit) содержат внутренние модели представления устройств, но доступ к информации о них ограничен. В большинстве случаев модели направлены на решение частных задач реинжиниринга в узком классе. Универсальные модели на базе UML обладают высокой сложностью и недостаточной степенью формализации, что затрудняет их применение в задачах реинжиниринга. В области разработки и

реинжиниринга программного обеспечения многоуровневые модели, сочетающие преимущества различных подходов, рассматривались Кларком, Франком, Баторием и другими. При этом практически отсутствуют работы, посвященные созданию многоуровневых моделей цифровых устройств.

Встраивание средств внутрисхемного тестирования является хорошо проработанной областью. Автоматизированный синтез подсистем тестирования для цифровых систем с процессором рассматривается в работах Путри, Катлера, Зайцева и Чена. Встраивание независимых от процессора компонентов тестирования и самодиагностики рассматривается в работах Сетхурама, Чена, Мелехина. Также существует множество примеров ВСВТ для частных задач, возникающих при проектировании цифровых устройств. Эмуляция сбоев памяти рассматривается в работах Сигалла, Делонга, Мадейры и других. Аналогично можно привести примеры для других классов задач. При этом основную проблему представляет универсализация подходов, для чего актуальна задача их адаптации к другим моделям представления устройств, используемых при реинжиниринге систем.

**Целью исследования** является разработка новых моделей и методов анализа, трансформации, а также контроля устройств вычислительной техники и систем управления для последующего использования в маршрутах проектирования с целью снижения затрат на разработку и улучшения технико-экономических и эксплуатационных характеристик.

**Объектом исследования** являются процессы проектирования, анализа, модификации и контроля элементов и устройств автоматики и вычислительной техники. **Предметом исследования** являются расширяемые метамодели устройств, а также методы их обработки и специализации для решения широкого класса прикладных задач реинжиниринга цифровых устройств. Для достижения поставленной цели решены следующие научные задачи:

1. Формирование критериев оценки и анализ существующих подходов к построению моделей устройств для задач реинжиниринга устройств.
2. Разработка новой многоуровневой модели устройств, предназначенной для решения задач реинжиниринга цифровых устройств.
3. Разработка методик построения модели на основе существующих языков описания устройств и её применения в задачах анализа, трансформации и верификации устройств.
4. Специализация модели для решения задач реинжиниринга в области встраивания средств внутрисхемного тестирования СнК.

**Специальность, которой соответствует диссертация.** Поставленные научные задачи исследования относятся к специальности 05.13.05 «Элементы и устройства вычислительной техники и систем управления», так как реинжиниринг устройств и встраивание средств внутрисхемного тестирования входят в маршруты проектирования современных цифровых устройств и позволяют обеспечить новые свойства и характеристики элементов, схем и устройств вычислительной техники и систем управления.

**Методы исследования.** При проведении исследований и апробации разработанных в диссертационной работе моделей и методик использованы методы системного анализа, теории алгоритмов, теории программирования и теории графов. Также применены методы экспертных оценок для сравнительного анализа существующих моделей представления устройств.

**Личный вклад автора диссертации:**

1. Постановка задач исследования и первичный анализ выполнены автором совместно с к.т.н. А.С. Филипповым и ст. преп. С.Л. Максименко.
2. Разработка многоуровневой (“гибридной”) метамодели устройств и методик её специализации выполнены автором лично.
3. Методики построения гибридной модели разработаны автором лично.
4. Разработка методов реинжиниринга устройств для частных задач повышения отказоустойчивости и ВСВТ выполнена автором лично.

**Личный вклад автора диссертации при прототипировании и внедрении:**

1. Прототипирование модели, методик и средств выполнено автором лично.
2. Разработка методик внесения неисправностей в память устройств выполнена совместно с О.В. Мамутовой (ст. преп. каф. КСПТ).
3. Прототипирование методов анализа структуры нетлистов и поиска элементов выполнена совместно с И.В. Егоровым (аспирант каф. КСПТ)
4. Внедрение САПР на базе прототипа инструментария для задач ВСВТ в устройства проводилось совместно с инженерами ООО “ЭсДиСи”.
5. Иные задачи по прототипированию и внедрению решены лично автором.

**Основные положения, выносимые на защиту:**

1. Разработана новая гибридная метамодель цифровых устройств, входящая в класс многоуровневых моделей, отличающаяся от других совмещением двух уровней представления и применимая в широком классе задач анализа, модификации и верификации архитектур цифровых устройств.
2. Предложены и обоснованы функционально-полный набор базовых операций над моделью, а также методики построения модели из исходных описаний и специализации модели, позволяющие эффективно описывать алгоритмы реинжиниринга устройства.
3. Разработаны методы встраивания средств внутрисхемного тестирования и самодиагностики, которые снижают затраты на тестирование и риск возникновения ошибок, а также могут быть интегрированы в типовые маршруты проектирования однокристалльных цифровых устройств.
4. Разработана методика совместной верификации модели устройств в системе моделирования и аппаратных прототипов на базе единого набора тестов, в отличие от других не требующая процессорных ядер в системе.

**Теоретическая значимость** исследования заключается в разработке новых моделей представления устройств и методик их использования, позволяющих решать задачи реинжиниринга для различных уровней и языков описания устройств. **Новизна** определяется отсутствием аналогичных моделей и методик, учитывающих специфику проектирования и реинжиниринга СнК в условиях ограничений по системным ресурсам и временным характеристикам.

**Практическая значимость** работы заключается в возможности расширить область применения реинжиниринга за счёт применения разработанных моделей, методов и инструментальных средств. Это позволяет:

- улучшить технико-экономические характеристики цифровых СнК за счёт повторного использования выполненных ранее разработок, что позволяет снизить затраты на проектирование и контроль их качества;
- упростить задачу прототипирования новых архитектур и инженерных решений за счёт алгоритмизации изменений в СнК;
- создавать инструментарии, решающие частные задачи реинжиниринга и встраиваемые в существующие САПР и маршруты проектирования.

Практическая значимость результатов исследования подтверждена внедрением в проектах по разработке сложных СнК в компаниях ООО «Синописис СПб» (Synopsys Inc.) и ООО «ЭсДиСи». Результаты работы также внедрены в учебный процесс на кафедре компьютерных систем и программных технологий и использованы в проекте №2.1.2/12647 «Исследование фундаментальных свойств асинхронных многопроцессорных вычислительных структур в базисе перепрограммируемых логических кластеров». Имеются соответствующие акты о внедрении. При прототипировании моделей и методик получен результаты, имеющие самостоятельную практическую ценность в области проектирования цифровых устройств. Подана заявка на регистрацию разработанного прототипа САПР как программы для ЭВМ.

**Апробация работы.** Результаты работы докладывались и обсуждались на 12 научных конференциях и семинарах, в том числе: IX Европейской конференции про разработке программного обеспечения (ESEC, СПб., 2013), I Всероссийском конгрессе молодых учёных (СПб., 2013), симпозиуме по автоматизированной верификации встраиваемых систем (VES2013, СПб., 2013), II Технической Конференции по производству, разработке и испытанию изделий (СПб., 2013), симпозиуме по автоматизации проектирования СнК (Хайдарабад, 2014), VI Всероссийской научно-технической конференции "Проблемы разработки перспективных микро- и наноэлектронных систем" (MES, Москва 2014), Международной конференции по электрическим цепям, системам и цифровой обработке сигналов (ICCSSP, СПб., 2014) и других научных мероприятиях. Результаты внедрения прототипа PHRT также были представлены на практической конференции, посвящённой системе непрерывной интеграции Jenkins CI (JUC, Лондон, 2015).

**По теме диссертации опубликовано** 12 работ. 3 публикации - в изданиях, рекомендованных ВАК Минобрнауки РФ. 5 работ опубликованы на английском языке, две из которых входят в индекс Scopus.

**Структура и объём диссертации.** Диссертация состоит из введения, 5 глав, заключения, списка использованных сокращений и определений, списка использованных источников и 7 приложений. Объём диссертации составляет 195 страниц, в том числе 133 страницы основной части. Работа содержит 42 рисунка, 13 таблиц и список использованных источников из 90 наименований.

## 2. ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении показана актуальность выбранной темы и степень её разработанности, сформулированы цели и задачи исследования. Указаны методология и методы исследования. Изложены основные результаты работы, показаны их новизна, степень достоверности, научная и практическая значимость. Сформулированы положения, выносимые на защиту.

В первой главе проведён анализ проблематики построения моделей устройств в задачах встраивания средств тестирования (СТ) и реинжиниринга в целом. Рассмотрены подходы к построению моделей устройств, с помощью экспертных методов сформированы критерии их оценки, произведён сравнительный анализ и выбор подхода для реализации.

В работе использована классическая трёхстадийная схема реинжиниринга систем (рисунок 1). Алгоритмы анализа и трансформации архитектур устройств описываются на базе некоторой модели представления устройств, к которой предъявляются следующие требования:

1. Возможность создания алгоритмов реинжиниринга на базе модели.
2. Возможность специализации для частных классов задач реинжиниринга.
3. Независимость от исходного языка описания устройства.

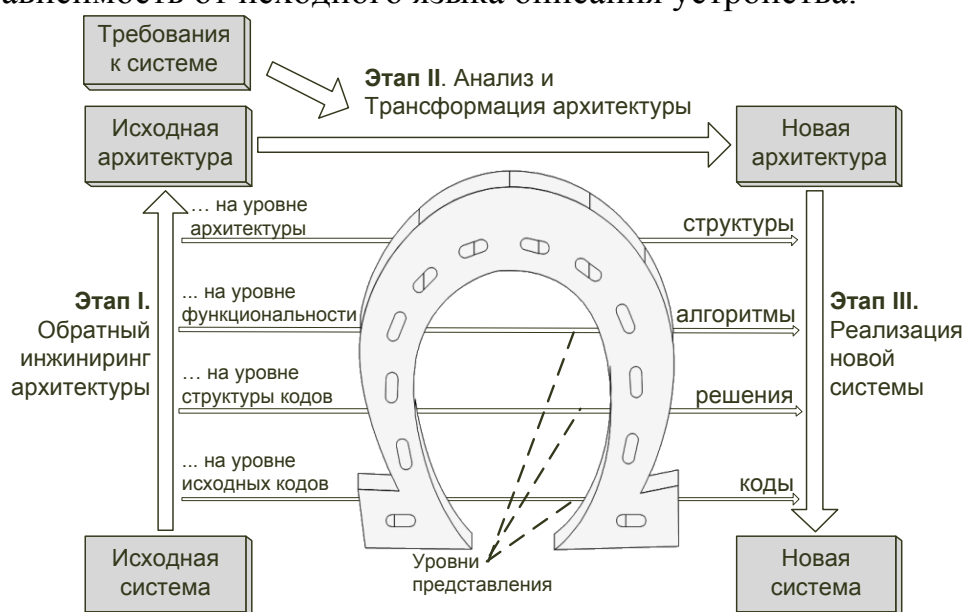


Рисунок 1 - Процесс реинжиниринга. Модель “подковы” (Кацман, 1998)

Для проведения реинжиниринга требуются модели устройств на различных уровнях представления. Путём экспертного опроса инженеров и исследователей сформированы требования к модели представления устройств и критерии оценки применимости моделей в общих задачах реинжиниринга. Сравнительный анализ существующих подходов показал, что они не позволяют построить модель для широкого класса задач. Предложен новый подход – использование многоуровневой модели как универсального описания, на базе которого строятся все алгоритмы работы с ней. Поскольку универсальные модели не учитывают специфику частных задач реинжиниринга, предлагается использовать многоуровневую метамодель, которая может быть специализирована для частных задач. К ней предъявлены следующие требования:

- хранение структурного описания устройства в модели;
- использование архитектурного графа для представления устройства;
- ограниченный набор типов базовых элементов модели;
- возможность параметризации элементов модели;
- наследование свойств элементов через расширение свойств;
- поддержка одновременной работы с несколькими устройствами;
- близость к одному из языков описания устройств (HDL).

Одной из областей применения реинжиниринга при проектировании СнК является встраивание СТ с целью контроля качества устройств и их диагностики при последующей эксплуатации. Задача актуальна даже при проектировании устройств с учётом всех требований к тестопригодности (подход DFT – design for test), так как могут меняться требования к системе и ставиться задачи с большими аппаратными затратами. Применение гибридных моделей позволяет эффективно описывать тестовую инфраструктуру внутри СнК и учитывать требования по системным ресурсам при реинжиниринге.

При верификации сложных СнК актуальна задача повторного использования тестов с целью снижения затрат. В этом случае целесообразно применять инструментарии реинжиниринга для генерации и встраивания тестовых средств по описаниям тестов. В отличие от подходов с запуском программных тестов для процессорных СнК, реинжиниринг позволяет реализовать тесты на любой платформе и при этом добиваться лучшего покрытия и скорости выполнения тестов. На основании анализа в первой главе сформированы задачи исследования, перечисленные выше.

**Во второй главе** описана новая «гибридная» метамодель устройств, которая относится к классу многоуровневых моделей и включает низкоуровневый структурный (нетлисты) и высокоуровневый архитектурный уровни описания. Метамодель используется как основа для построения специализированных моделей устройств для частных задач реинжиниринга. В ряде случаев метамодель может быть использована без специализации. Метамодель строится на базе формальных средств и неформальных предложений по итогам экспертного опроса, введены следующие ограничения:

1. Метамодель предназначена для структурных преобразований в архитектурах устройств с использованием дополнительной информации для анализа и принятия решений.
2. Метамодель рассчитана на реинжиниринг однокристалльных устройств, специфика распределённых архитектур в работе не рассматривается.
3. Метамодель не включает полную информацию об устройствах для частных задач. Данная задача решается при специализации модели.
4. Устройства имеют статическую структуру с постоянными связями между элементами (не рассматривается динамическая реконфигурация ПЛИС).
5. Значимые для реинжиниринга поведенческие и иные высокоуровневые описания выражаемы в структурной форме путем частичного синтеза.

На рисунке 2 приведена разработанная структура типов элементов гибридной метамодели, удовлетворяющая поставленным требованиям. Серым



цветом отмечено структурное подмножество гибридной метамодели, белым – высокоуровневые элементы. Связи между элементами реализуются через механизмы групп, ссылок и наследования для упрощения анализа модели.

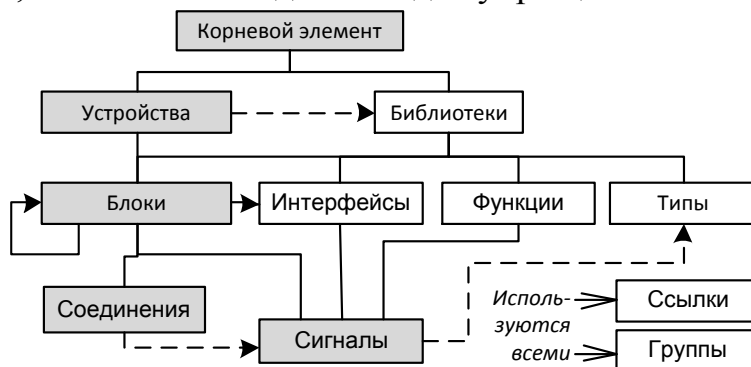


Рисунок 2 - Типы элементов гибридной метамодели и производных моделей

Модель представляется как множество элементов с корневым узлом (1.а), каждый элемент представляет собой сложный объект, предназначенный для программной обработки. Возможна реализация новых типов элементов через наследование свойств одного из базовых типов. Любой тип элементов может быть описан следующим множеством (1.б):

$$a) G = (V, V_{root}) \quad б) V = (A, I, P, L, C, M), \quad (1)$$

где:

- $A$  – внутренние свойства (атрибуты), отличные для каждого типа;
- $I$  – ссылка на базовый объект, от которого наследуются свойства;
- $P$  – ссылка на родительский элемент;
- $L$  – список внешних ссылок на элемент;
- $C$  – массив ссылок на дочерние элементы;
- $M$  – множество метаданных, передающих дополнительную информацию об элементе между шагами алгоритма реинжиниринга.

В отличие от моделей Вилси и Бенца, гибридная метамодель не содержит поведенческих описаний в качестве отдельных элементов. Эта информация хранится в функциях и метаданных элементов, низкоуровневая структура которых формируется за счёт частичного синтеза. Гибридную метамодель можно свести к древовидному графу в структурном представлении или к направленному ациклическому графу (орграфу) - в семантическом (рисунок 3). Это позволяет использовать аппарат теории графов при анализе и трансформации модели. На схеме а) показан пример орграфа устройства  $D$  со ссылкой на блок  $B_L$  в библиотеке. На схеме б) данная ссылка разворачивается копированием элементов библиотеки в иерархию. При этом модель сводится к древовидному графу, что позволяет использовать иной математический аппарат (например, для навигации по модели и поиска элементов).

$$a) G = (V, E) \quad б) G = (V, A), \quad (2)$$

где:

- $V$  – непустое множество элементов;
- $E$  – множество рёбер (ссылок);
- $A$  – ориентированное множество вершин (дуг).

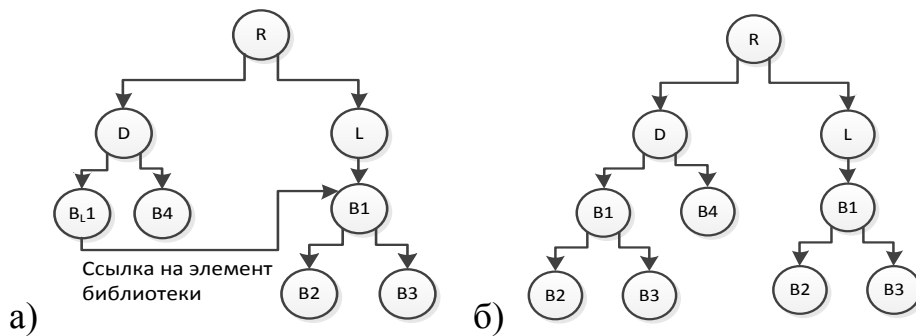


Рисунок 3 - Примеры графов для простого устройства  
(а – DAG; б – развёрнутый древовидный граф)

Элементы метамодели описывают базовую структуру устройства, которая может быть дополнена информацией при помощи рассмотренных в третьей главе механизмов специализации. На верхнем уровне находятся устройства и библиотеки, использование элементов которых возможно через ссылки или наследование. Блоки реализуют уровни в иерархии устройства и интерфейсы, описывающие порты и статические параметры. Блоки содержат сигналы, подключаемые к другим сигналами, портам или функциям посредством одно- или двунаправленных соединений. Элементы объединяются в группы (например, шины и агрегации сигналов). Функции описывают элементы, которые не могут быть выражены в структурной форме: поведенческие описания в VHDL, макросы в SystemC и т.д. Возможно задание отношения наследования между элементами, при котором один элемент неявно включает свойства и структуру другого. Оливьером доказано, что подобный подход позволяет поддерживать иерархическую семантику высокоуровневых языков (SystemC и др.). Примером наследования могут быть блоки, включающие диагностические выводы без изменения функциональных исходного блока.

В работе доказана совместимость модели с произвольными исходными описаниями посредством рассмотрения вырожденного случая. Предложен общий подход к импорту моделей из HDL. Рассмотрены частные случаи для языков VHDL, Verilog, SystemC и System Verilog, а также формата нетлистов EDIF. Доказательство является конструктивным и выполнено путём анализа элементов языка и выявления эквивалентов в гибридной модели.

**В третьей главе** описываются методики работы с предложенной моделью представления устройств при решении типовых задач реинжиниринга цифровых систем. Рассмотрены задачи специализации модели для частных задач реинжиниринга, ввод и вывод многоуровневых описаний, набор операций над моделью, а также задачи верификации и валидации модели.

При импорте описаний заключается в построении гибридной модели по исходным описаниям и компиляции описаний в синтезируемые представления. При использовании одного исходного описания на HDL модель строится по рекомендациям из второй главы. В случае нескольких исходных описаний требуется их объединение и последующая оптимизация. На рисунке 4 приведены маршруты импорта и экспорта модели для обоих случаев. В

качестве промежуточного представления используются абстрактные синтаксические деревья (АСД). Предложены пути решения следующих задач:

- восстановление иерархической структуры устройств из нетлистов;
- выявление однотипных блоков в нетлистах;
- выявление оптимизаций, сделанных сторонними САПР при синтезе.

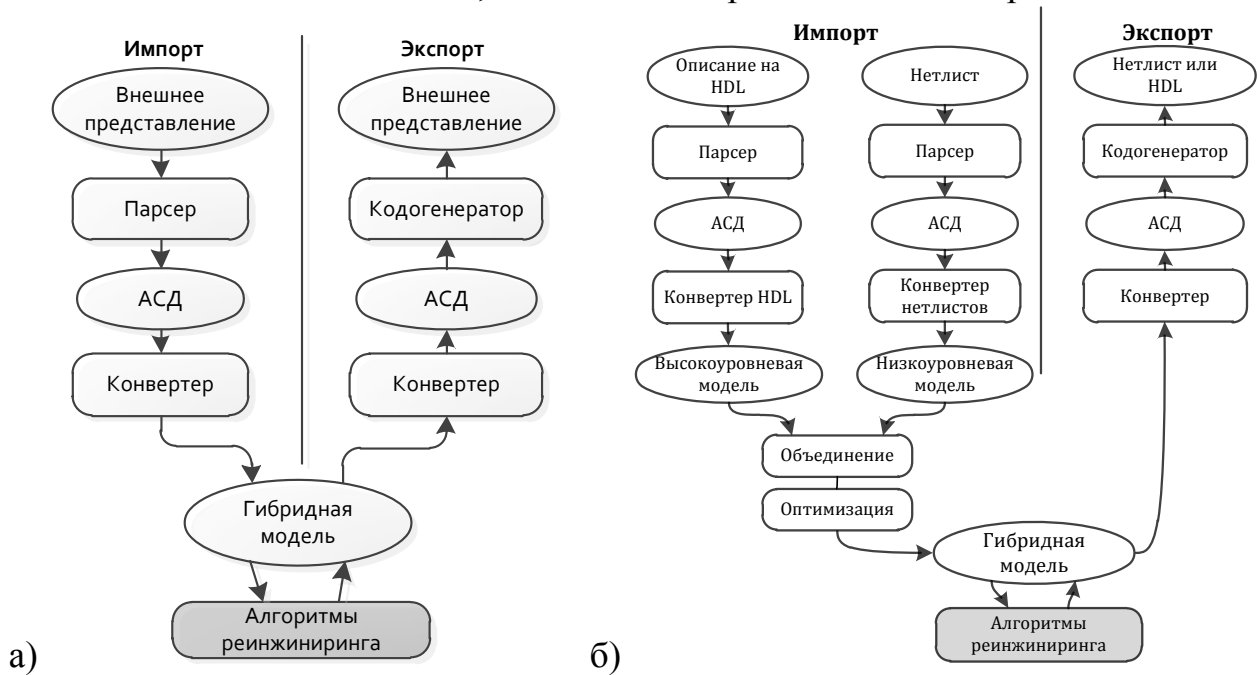


Рисунок 4 - Методика ввода-вывода описаний

(а – одно описание; б – гибридная модель на базе нескольких описаний)

Согласно работам Чакраборти, Экмана и др. нетлисты IP-ядер уязвимы к реверс-инжинирингу, и актуален вопрос защиты нетлистов для продуктовых компонентов устройств (IP-ядер). Предложенная гибридная модель и методики применимы для извлечения информации и внесения “закладок”, поэтому в работе рассмотрен вопрос защиты от восстановления архитектуры устройства.

Для описания алгоритмов анализа и преобразования устройства на базе гибридной модели необходим функционально полный набор операций. Сформированы следующие условия:

1. Возможно преобразование произвольной исходной целостной модели в произвольную новую модель за конечное число операций.
2. Возможен доступ к любому элементу за конечное число операций.
3. Для любого элемента доступны его родительский и дочерние элементы.
4. Для любого элемента могут быть изменены ссылки и свойства.
5. Функциональная полнота алгоритмической части языка.

Ниже приведена формальная запись первых 4-х критериев в модели:

$$\left\{ \begin{array}{l} \forall V \in G(V_{1..m}, V_{root}) \exists f_{1..N}, n \in \mathbb{N} : V = f_n(f_{n-1}(\dots f_1(V_{root}))) \\ \forall V \in G(V_{1..m}, V_{root}), \exists f, n \in \mathbb{N} : (A, I, P, L, C, M) = f(V) \\ \forall T \in Types \exists f_{create\_T} : V_{new} = f_{create\_T}(A, I, P, L, C, M) \\ \forall V_{device} \exists f_{add} : V_{device} \cap f_{add}(V_{root}, V_{device}) \neq \emptyset \\ \forall V_{device} \exists f_{delete} : V_{device} \cap f_{delete}(V_{root}, V_{device}) = \emptyset \end{array} \right. \quad (4)$$

В работе предложен удовлетворяющий требованиям набор операций различных категорий, приведённых в таблице 1. Предлагается использовать универсальный и функционально полный язык программирования (ЯП), широко используемый в современных САПР (например, TCL).

Таблица 1 - Категории операций в предлагаемом языке работы с моделью

Категория	Примеры операций	Комментарии
Операции алгоритма	Ветвления, циклы, вызов процедур, ...	Обеспечены универсальным ЯП и не взаимодействуют с моделью
Навигация по модели	Поиск элементов в иерархии, обход элементов, получение списков ссылок	Операции не вносят изменений в модель устройства. $E = f(G, A)$ ; $E = f(G, P)$ ; $E = f(G, L, C)$
Изменение структуры модели	Добавление, перенос и удаление элементов	Меняется структура графа модели, но объекты в узлах остаются неизменными. $G = f_m(G, V)$
Изменение свойств элементов	Изменение атрибутов, переименование элементов модели	Меняется отдельный элемент модели, структура не меняется. $V = f_m(V, A)$ ; $V = f_m(V, P, L, C)$
Работа с метаданными	Модификация и считывание значений метаданных и атрибутов элементов	Используются для расширения модели нестандартными данными. $M = f(V, M)$ ; $V = f_m(V, M)$
Обработчики событий	Условная генерация элементов, валидация вносимых изменений	Специальные команды обратного вызова при операциях над моделью. $G = f_x(G, E)$

Процесс встраивания СТ может включать множество преобразований модели, поэтому актуальна задача контроля модели и описываемого ею устройства. Сформированы следующие критерии верифицируемости модели:

- модель сводится к целостному древовидному графу, возможен его обход;
  - модель содержит конечное число элементов;
  - структурные блоки имеют строго древовидную иерархию;
  - на любой некорневой элемент существуют внешние ссылки;
- все элементы модели имеют полное внутреннее описание;
- возможен вызов алгоритмов контроля при выполнении операций.

Достаточность условий доказана в диссертации. Необходимость условий обусловлена требованиями к модели, рассмотренными в первой главе.

Гибридная метамодель является основой для построения производных моделей, которые могут быть эффективно использованы в алгоритмах реинжиниринга. Предусмотрены механизмы специализации на трех уровнях: модели, методов и реализации в инструментальных средствах (таблица 2). В таблице 3 приведены примеры специализации модели для задач ВСВТ. Подход позволяет использовать ориентированные на исходную метамодель алгоритмы реинжиниринга в производных моделях, что расширяет область их применения. Для специализации гибридной метамодели для класса частных задач реинжиниринга предлагается следующий метод:

1. Определение области применения разрабатываемой модели: класса задач, системных ограничений, методик описания и маршрута проектирования.
2. Определение характеристик устройства, которые требуется отразить в модели для выполнения анализа и принятия решений о реинжиниринге.
3. Дополнение структурного представления частной модели: введение специальных типов, специализированных блоков (например, ТА).
4. Введение и спецификация метаданных для хранения дополнительной информации о характеристиках элементов модели.
5. Анализ решаемых задач и формирование дополнительного набора операций для инкапсуляции стандартных операций в алгоритмах.
6. Дополнение методик импорта и экспорта модели для использования специализированной модели устройства в маршруте проектирования.
7. Разработка специализированных методов валидации и контроля корректности модели, интеграция методов в набор операций.
8. Разработка специализированных алгоритмов и инструментальных средств для решения выбранного класса задач реинжиниринга

Таблица 2 - Примеры точек расширения для различных уровней специализации

Уровень специализации	Примеры точек расширения модели
Модель устройства	– Наследование элементов в специализированных типах; – Добавление метаданных в элементы; – Задание зависимостей через ссылки.
Методы работы с моделью	– Создание специализированных наборов операций; – Задание операций обратного вызова при изменениях; – Условная генерация компонентов.
Инструментальные средства	– Применение средств объектно-ориентированных языков; – Интеграция со сторонними САПР.

Таблица 3 - Примеры специализации модели для задач ВСВТ

Уровень специализации	Задача ВСВТ для...	
	средств самодиагностики	тестовых агентов (ТА)
Модель устройства	– Метаданные о характеристиках элементов; – Хранение поведенческих описаний и тестовых векторов в элементах.	– Блоки и интерфейсы для тестовых агентов; – Библиотеки типовых параметризуемых элементов.
Методы работы с моделью	– Расширение команд верификации модели при преобразовании.	– Встраивание тестовых точек, интерфейсов и соединений; – Генерация ТА для заданной контрольной точки.
Инструментальные средства	– Частичный синтез устройств.	– Синтез описаний для задач описания тестов.

В четвёртой главе рассмотрено применение разработанных моделей и методик для решения задач встраивания средств внутрисхемного тестирования и самодиагностики устройств, в том числе с учётом системных ограничений. Предложен подход к совместному тестированию устройства и его программной модели в средствах моделирования. Для класса задач ВСВТ произведена специализация модели по предложенному выше методу, краткая информация об изменениях на различных уровнях специализации приведена в таблице 3.

При ВСВТ необходимо обеспечить полную управляемость и частичную наблюдаемость тестируемого компонента системы, что может требовать встраивания новых функциональных блоков и интерфейсов. Эта операция относится к реинжинирингу (рисунок 5), при ней решаются следующие задачи:

1. Изоляция тестируемого модуля от влияния остальной части системы.
2. Изоляция СнК от влияния тестируемого модуля (или останова системы).
3. Встраивание средств генерации тестовых векторов.
4. Встраивание средств для сброса и установки начального состояния.
5. Встраивание диагностических интерфейсов для контроля состояния СнК.

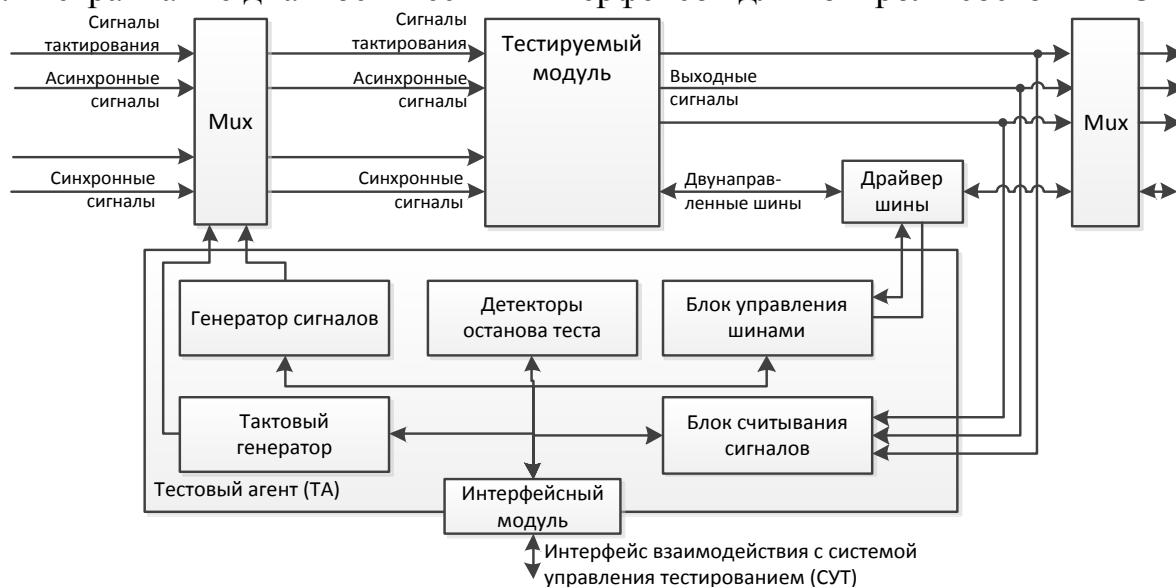


Рисунок 5 - Пример встраивания тестового интерфейса в устройство

В работе приведён алгоритм решения перечисленных задач с использованием гибридных моделей. Показана возможность оптимизации затрат при тестировании сложных устройств и использования инструментариев реинжиниринга в качестве средства поддержки принятия решений при встраивании тестовых интерфейсов.

Для снижения затрат на верификацию устройств актуальна задача повторного использования тестовых алгоритмов из систем моделирования устройств (СМУ). На базе методики ВСВТ предложен метод проведения совместного тестирования с использованием СМУ и прототипа на ПЛИС на базе единого набора тестов. На рисунке 6 приведена схема подключения тестовых агентов к СМУ, реализация интерфейса прикладного программирования (ИПП) и средств сопряжения не рассматривается. Ниже приведены этапы проведения тестирования в предлагаемом алгоритме:

1. Выполнение тестов в системе моделирования.
2. Извлечение списка сигналов, которые изменяются или считываются при выполнении тестов в системе моделирования.
3. Формирование списка блоков  $M$ , в которые требуется встроить ТА.
4. Встраивание СТ посредством реинжиниринга устройства:
  - a. Добавление тестового интерфейса  $T_I$ ;
  - b. Встраивание ТА для каждого проверяемого модуля ( $T_{A1..N}$ );
  - c. Подключение тестовых агентов к  $T_I$ ;
  - d. Формирование выходного описания, описывающего адресацию к встроенным ТА через тестовый интерфейс  $T_I$ .

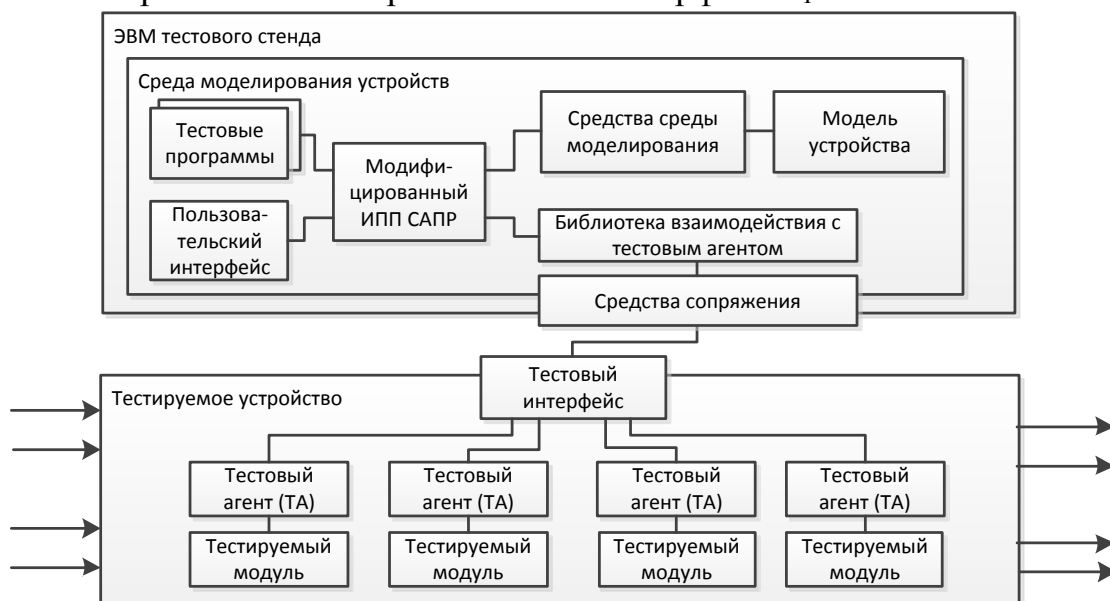


Рисунок 6 - Совместное тестирование в СМУ и прототипа устройства

В работе предложен подход к встраиванию СТ в устройства при наличии ограничений по ресурсам СнК (площадь кристалла в СБИС, число базовых элементов в ПЛИС и БМК). Выделены следующие шаги алгоритма:

1. Поиск минимального множества ТА, обеспечивающих необходимый уровень наблюдаемости и управляемости.
2. Независимый синтез устройства, СУТ и ТА для получения оценки по используемым ресурсам. При нарушении требований изменяются целевая платформа или набор тестов.
3. Выбор способа подключения ТА для обеспечения требуемой скорости проведения тестирования: независимое подключение, использование системных шин или сетей на кристалле.
4. Оптимизация сети подключения тестовых агентов.
5. Итерационная оптимизация архитектуры устройства посредством синтеза в САПР и принятия решений о новом цикле реинжиниринга.

На базе предложенных моделей и методов разработан прототип инструментального средства PHRT (Programmable Hardware Reengineering Toolkit – программируемый инструментарий реинжиниринга устройств). Архитектура прототипа и методики его применения имеют самостоятельную практическую ценность, но не выносятся на защиту в виде отдельных

положений. В пятой главе приведены примеры решения задач реинжиниринга на базе предложенных модели и методик:

- встраивание тестовых агентов и интерфейсов в устройства;
- встраивание средств внесения однократных сбоя в блочную память;
- внесение структурной избыточности и блоков самодиагностики;
- замена блочной и регистровой памяти на отказоустойчивые аналоги;
- анализ путей распространения сигналов и деревьев синхронизации;
- построение интегрированного инструментария отладки устройства на ПЛИС для встраивания СТ и самодиагностики (рисунок 8).

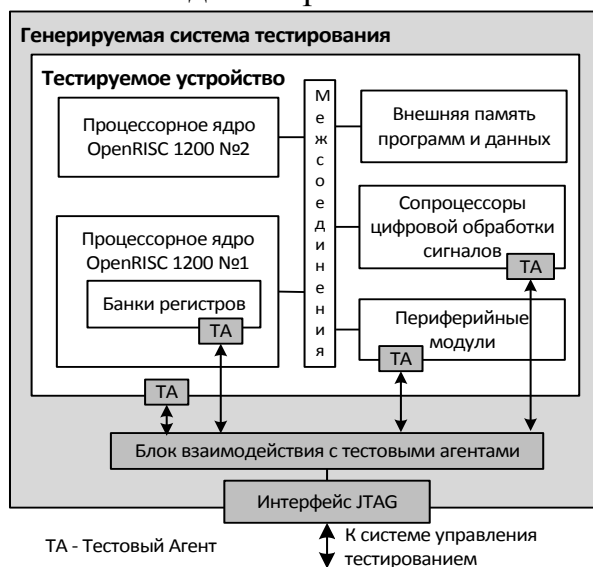


Рисунок 7 - Пример ВСВТ в СнК

Таблица 4 – Использование ресурсов\*

Эксперимент	F <sub>max</sub> , МГц	Логические ячейки	Регистры	Блочная память, бит
Исходное устройство	39,31	8055	3105	53888
Тестовые интерфейсы	56,14	516	270	0
Ввод-вывод в САР без изменений	37,18	8247	3105	53888
Ручное изменение	39,15	8752	3417	53888
Автоматическое изменение в САР	37,02	8916	3417	53888

\*результаты для JTAG и 10 ТА, встраиваемых в банки регистров OpenRISC на ПЛИС Altera Cyclone IV

На рисунке 7 и в таблице 4 приведён пример встраивания СТ для одной из рассмотренных СнК. Результаты апробации подтвердили применимость предложенных методик для реальных устройств без значительного изменения характеристик системы по сравнению с иными рассмотренными подходами. Для задач встраивания средств самодиагностики возможна доработка прототипа с целью уменьшения влияния СТ на характеристики устройства.

На рисунке 8 приведён пример реализации интегрированной системы разработки для задач внутрисхемного тестирования устройств. Разработчик задаёт описания на HDL и тестовые векторы в редакторах, после чего запускает их через модуль PHRT. Среда производит реинжиниринг и программирование ПЛИС, затем выполняет тесты и выводит результаты. Полностью решается задача подготовки и проведения внутрисхемного тестирования. Внедрение прототипа в проектах ООО «ЭсДиСи» позволило вдвое снизить общие затраты на проведение тестирования и снизить время обнаружения ошибок за счёт отказа от прогона длительных тестов в системах моделирования. На примере системы непрерывной интеграции Jenkins предложен подход к встраиванию предложенного решения в маршруты проектирования устройств.

На базе алгоритма ВСВТ совместно с О.В. Макутовой предложен подход к оценке и контролю устойчивости СнК к однократным сбоям памяти, основанный на предложенной методике встраивания тестовых агентов.



Особенностью подхода является применение встроенного в СнК процессора в качестве СУТ, что позволяет снизить аппаратные затраты на ВСВТ. На рисунке 9 приведён фрагмент программы встраивания средств тестирования блочной памяти устройства, использующей разработанные расширения PHRT, и решающий задачу в соответствии с разработанными алгоритмами.

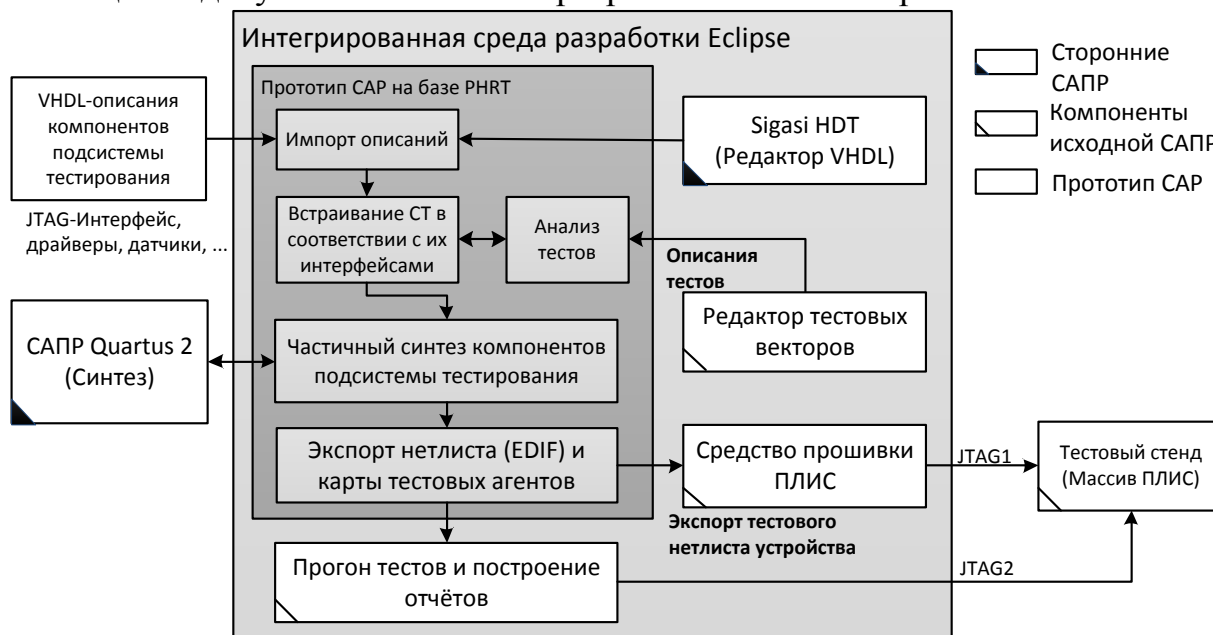


Рисунок 8 - Прототип САПР для тестирования устройств на базе PHRT

```
# Загрузка расширений и импорт устройства
phrt::system::load_extensions {quartus vhdl_netlist memfault_inject }
phrt::vhdl_netlist::import openRISC1200.vhd /test/
# Поиск блочной памяти в устройстве и инструментирование
set mem_list [phrt:find_refs /test/or1200_top -R ALTSYNCRAM]
phrt::memfault_inject::instrument $mem_list
# Экспорт модифицированного нетлиста
phrt::vhdl_netlist::export /test out.vhd
```

Рисунок 9 - Фрагмент программы встраивания СТ в устройство в PHRT

Результаты прототипирования и решения задач реинжиниринга устройств подтверждают применимость разработанных моделей и методик, в диссертации описаны разработанные для частных задач алгоритмы и инструментальные средства. В предлагаемых моделях и методиках выявлен ряд недостатков и ограничений, предложены пути их устранения путём доработки модели и создания новых методик её применения.

**В заключении** представлены полученные в работе научные и практические результаты, показаны их научная новизна, достоверность, теоретическая и практическая значимость, рассмотрены пути дальнейшего развития исследования.

**В приложениях** представлены используемые в работе термины и определения, материалы по применению методик в частных задачах реинжиниринга, дополнительная информация о САПР и результатах его апробации. В таблице 5 приведён перечень приложений к диссертации.

Таблица 5 - Список приложений к диссертации

№	Число страниц	Название / Содержание
A	8	Рассмотрены классификации средств реинжиниринга цифровых устройств инструментариев по различным признакам.
B	6	Обзор средств инструментариев реинжиниринга устройства. Проведён анализ используемых ими методик и моделей на основании открытой информации.
C	14	Прототип инструментария PHRT. Описан прототип PHRT и его расширения, разработанные при апробации предложенных моделей и методик на примерах частных задач реинжиниринга.
D	8	Набор операций над гибридной моделью в прототипе САПР. Приведено краткое описание набора базовых операций над предлагаемой моделью, которые предложены для САПР
E	7	Особенности использования VHDL-нетлистов в среде Quartus II и прототипе, которые выбраны в прототипе PHRT в качестве входного формата описания устройств.
F	8	Пример введения структурной избыточности в устройство. Детализация описанных в пятой главе примеров использования PHRT при выполнении частных задач реинжиниринга
G	3	Акты о внедрении полученных результатов в ООО «Синописис СПб», ООО «ЭсДиСи» и ФГАОУ ВО «СПбПУ»

При прототипировании САПР и методик реинжиниринга цифровых устройств на базе гибридных моделей разработан ряд расширений для средства PHRT, которые решают задачи реинжиниринга и ВСВТ с использованием предложенных в четвертой и пятой главах диссертации подходов. Краткие описания расширений приведены в таблице 6.

Таблица 6 - Основные расширения PHRT, созданные при прототипировании

Название	Назначение / Описание
vhdl_basic	Базовая поддержка VHDL – импорт и экспорт описаний.
vhdl_netlist	Ввод-вывод VHDL-нетлистов, генерируемых САПР Quartus II.
edif	Ввода-вывод нетлистов в форматах EDIF 2 0 0 и EDIF 3 0 0.
quartus	Частичный синтез компонентов модели в среде Quartus II.
signal_analysis	Решает задачи анализа структуры нетлистов, выявления деревьев и цепей тактирования, зависимостей между сигналами.
reliability	Решает задачу повышения отказоустойчивости через внесение структурной избыточности и замену блоков памяти аналогами.
test_insertion	ВСВТ и самодиагностики в соответствии с методами, описанными в четвёртой главе.
memfault_inject	Библиотека для встраивания средств внесения сбоя в память устройство на основании алгоритмов, разработанных совместно с Мамутовой О.В. и описанных в пятой главе.

### 3. ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Ненашев О.В. Расширяемый инструментарий для автоматизации реинжиниринга цифровых систем на кристалле. // Университетский Научный Журнал.- СПб.: Санкт-Петербургский университетский консорциум, 2014.- №5.- С. 194-203
2. Ненашев О.В. Методы встраивания средств тестирования в устройства с использованием средств автоматизации реинжиниринга. // Проблемы разработки перспективных микро- и наноэлектронных систем.- М.: ИППМ РАН, 2014.- № 2.- С. 101–106.
3. Мамутова О.В., Ненашев О.В., Филиппов А.С. Автоматизация низкоуровневого оснащения СнК средствами эмуляции внесения сбоя в память. // Известия вузов. Электроника.- М.:МИЭТ, 2015.- № 2.- С. 50-57
4. Ненашев О.В. Программное средство для анализа и трансформации архитектуры устройств, описанных на HDL // I Всероссийский конгресс молодых учёных.- СПб.: Издательство НИУ ИТМО, 2012.- С. 42-48
5. Ненашев О.В. Автоматическое добавление средств тестирования и самодиагностики устройства с использованием средств автоматизации реинжиниринга // Сборник статей Тринадцатой международной научно-практической конференции “Фундаментальные и прикладные исследования, разработка и применение высоких технологий в промышленности и экономике”.- СПб.: Изд-во СПбГПУ, 2012.- С. 46–49.
6. Ненашев О.В. Проблемы разработки программного инструментария для средства реинжиниринга устройства // Лучшие доклады XXXIX Недели Науки.- СПб.: Издательство СПбГПУ, 2011.- С. 72–75.
7. Nenashev O. Automated synthesis of in-circuit testing components with use of programmable hardware reengineering tools // Proceedings of the Youth Science Symposium.- СПб.: Изд-во СПбГПУ, 2012.- С. 37–42.
8. Nenashev O. PHRT: A programmable tool for automated hardware reengineering and verification // Proceedings of the VES2013 workshop, Saint Petersburg.- 2013.- С. 5-13.
9. Nenashev O. PHRT: A Model and Programmable approach to Hardware Reengineering Automation // Proceeding of 9th European Software Engineering Conference. 2013. С.- 719-722.
10. Mamoutova O.V., Nenashev O.V., Filippov A.S. In-circuit Emulation of Memory Fault Injection // Recent Advances in Electrical Engineering and Computer Science.- 2014.- С. 105–107.