

# Обзор программных средств с открытым исходным кодом для исследования современных архитектур ПЛИС Xilinx

А. Строгонов, д. т. н.<sup>1</sup>, М. Кривчун<sup>2</sup>, П. Городков<sup>3</sup>

УДК 004.4 | ВАК 05.13.12

Исследование гипотетических архитектур ПЛИС используя САПР с открытым исходным кодом дает возможность научному сообществу выявлять проблемы существующих ПЛИС, разрабатывать новые концепции и подходы к созданию коммерческих САПР. За последнее время вышли модернизированные инструменты с открытым исходным кодом для реализации проектов на основе новых архитектур коммерческих ПЛИС Intel FPGA (Altera) и Xilinx. В статье представлен обзор открытых программных инструментов, используемых в академической среде для изучения современных архитектур ПЛИС Xilinx, в том числе RapidSmith2 – для реализации алгоритмов трассировки и размещения, VTR – для академических ПЛИС с гипотетической архитектурой, набор библиотек Tincr для Vivado, RapidWright с поддержкой инструментов Vivado, Torc – для открытых реконфигурируемых вычислений, OpenPR – для частичной реконфигурации ПЛИС. Кроме того, рассмотрены особенности иерархического построения архитектуры современных ПЛИС Xilinx.

**И**сследования архитектур зарубежных ПЛИС проводятся, главным образом, в двух направлениях: первое относится к коммерческим ПЛИС и созданию инструментов для САПР, а второе фокусируется на изучении гипотетических архитектур ПЛИС и разработке собственных САПР. Большая часть работ с гипотетической архитектурой и исследованием САПР выполняется с использованием САПР VTR (Verilog-to-Routing).

Эксперименты на коммерческих ПЛИС вносят огромный вклад в развитие САПР с дополнительными возможностями, например такими, как обнаружение

и устранение новых уязвимостей в безопасности ПЛИС. Подобные исследования помогают производителям ПЛИС находить решения проблем, с которыми сталкиваются разработчики проектов на их основе.

## RapidSmith2 – РАЗВИТИЕ ИНСТРУМЕНТОВ RapidSmith

Разработка исходной версии RapidSmith началась в 2010 году. Этот программный продукт, созданный группой ученых из частного американского университета Бригама Янга, обеспечивает простой программный интерфейс приложения (API) для реализации экспериментальных идей и алгоритмов для САПР в современных ПЛИС Xilinx [1–3]. На основе RapidSmith в Xilinx Lab был разработан инструмент RapidWright [4], призванный устранить недостатки первой версии RapidSmith.

В новой версии – RapidSmith2 – реализован ряд существенных нововведений. В частности, обеспечена поддержка проектов из САПР Vivado. Исходный код распространяется по свободной лицензии и может быть доступен из публичного репозитория на веб-сервисе

<sup>1</sup> Воронежский государственный технический университет, профессор кафедры полупроводниковой электроники и нанoeлектроники, тел. +7 4732 43-76-95, andreistrogonov@mail.ru.

<sup>2</sup> Международная аутсорсинговая IT-компания UAB Andersen Soft, разработчик программного обеспечения (Go Developer), hedhyw@yahoo.com.

<sup>3</sup> Воронежский государственный технический университет, аспирант кафедры полупроводниковой электроники и нанoeлектроники, тел. +7 4732 43-76-95, gorodkoff@gmail.com.

GitHub [1]. С помощью RapidSmith2 можно разрабатывать собственные инструменты для САПР, которые позволяют:

- экспортировать проект из Vivado (в формате RSCP);
- анализировать дизайн проекта;
- произвольно изменять дизайн проекта;
- импортировать изменения обратно в Vivado (в формате TSP).

Дизайн проекта можно создать с нуля прямо в RapidSmith2 без необходимости экспорта из Vivado, но последующий импорт в Vivado необходим для генерации битового потока для конфигурирования ПЛИС.

Другое ключевое нововведение в RapidSmith2 – изменение типа дизайна. Вместо XDL-вида с экземплярами (Instance) и секциями (Cell) RapidSmith2 использует представление Vivado с секциями и базовыми элементами (BEL). Это позволяет раскрыть фактический дизайн проекта и обеспечивает новые возможности в исследовании САПР, которые было сложно реализовать в старой версии RapidSmith.

RapidSmith2 нацелен на исследования современных коммерческих ПЛИС Xilinx класса 7К. Пользователи RapidSmith2 должны быть знакомы с архитектурой ПЛИС Xilinx и САПР Vivado, знать язык программирования TCL (Tool Command Language).

TCL-интерфейс, реализованный в Vivado компанией Xilinx, – мощное дополнение к существующим инструментам. Теоретически он обеспечивает полный функционал для создания собственных алгоритмов САПР, таких как размещение, трассировка и т. д. Однако следует отметить ряд недостатков TCL, которые мотивируют использовать другие инструменты САПР, в частности RapidSmith2:

- TCL – медленный интерпретируемый язык, на нем достаточно долго выполняются такие комплексные алгоритмы трассировки, как PathFinder;
- на TCL неудобно и сложно писать программы, он не является объектно-ориентированным языком, что приводит к трудностям при написании сложных алгоритмов (TCL удобен только для создания скриптов автоматизации);
- TCL-интерфейс в Vivado не предоставляет полное описание ПЛИС, в нем нет доступа к субсекциям трассировочных ресурсов. Это значительно ограничивает возможности инструментов САПР, которые могут быть написаны на TCL.

Таким образом, к полезным свойствам RapidSmith2 относятся способность экспортировать дизайн проекта из Vivado, изменять его на более мощном (по сравнению с TCL) языке Java, а затем импортировать обратно в Vivado. RapidSmith2 (совместно с набором библиотек Tincr) абстрагирует этот процесс до нескольких простых вызовов функций.

## VTR – САПР ДЛЯ АКАДЕМИЧЕСКИХ ПЛИС С ГИПОТЕТИЧЕСКОЙ АРХИТЕКТУРОЙ

САПР с открытым исходным кодом VTR (Verilog-to-Routing) – совместная разработка представителей международного научного сообщества ряда стран и вузов, предназначенная для исследований архитектуры ПЛИС, а также исследований и разработок в области САПР [3, 5, 6]. Более подробно с VTR можно ознакомиться в работе [3].

VTR можно рассматривать как развитие инструмента VPR, которому посвящены многочисленные публикации [3]. САПР VTR8.0 позволяет обрабатывать крупные Verilog-проекты с умножителями и блоками памяти, создавать и исследовать новые архитектуры академических ПЛИС, по техническим характеристикам близкие к промышленным ПЛИС Intel (Altera) серии Stratix IV GX и Xilinx с 6-входовыми LUT с адаптивными логическими блоками.

В качестве входных данных САПР VTR принимает представление цифровой схемы на языке Verilog и описание целевой архитектуры ПЛИС. VTR умеет совершенствовать и синтезировать дизайн с помощью программного инструмента ODIN II с открытым исходным кодом; оптимизировать логику и проводить технологическое отображение в k-LUT с помощью программного инструмента ABC; упаковывать булевы функции в логические блоки (кластеризация), выполнять размещение и трассировку с помощью инструмента VPR (Versatile Place and Route).

Маршрут проектирования в VTR предусматривает два основных этапа. Синтезатор ODIN II преобразует Verilog-проект в сплошной список соединений, состоящий из логических элементов и черных ящиков, которые представляют гетерогенные блоки. Затем программный инструмент ABC выполняет независимую от технологического базиса (заказная БИС или ПЛИС) логическую оптимизацию с последующим отображением в базис ПЛИС – LUT и триггеры логических блоков. Результатом работы ABC является список соединений LUT, триггеров и черных ящиков в формате.blif.

**ООО "Руднев-Шиляев"**

Разработка и производство:

- платы сбора данных
- измерительные приборы
- виброакустические системы
- инструментальные решения задач заказчика

Москва (495) 787-63-67  
(495) 787-63-68

www.rudshel.ru  
adc@rudshel.ru

Далее VPR упаковывает этот список соединений в более крупные логические блоки (кластеры), размещает и выполняет трассировку ПЛИС, генерируя выходные файлы на каждом из этапов. VPR версии 8.0 использует алгоритм упаковки AARack. Упаковщик AARack реализует кластеризацию взаимосвязанных логических элементов в структурно-зависимые блоки ПЛИС, основан на «жадном» эвристическом алгоритме упаковки с использованием затравочного механизма роста кластера. На кристалле ПЛИС кластеры размещаются с помощью алгоритма «имитации отжига», а в процессе трассировки используется модификация алгоритма PathFinder.

VPR может проанализировать Verilog-проект, реализованный в базис ПЛИС с гипотетической архитектурой, и представить различные статистические данные и оценки, такие как минимальное количество межсоединений на трассировочный канал, необходимых для успешной маршрутизации; общая длина межсоединений; производительность; занимаемая площадь и потребляемая мощность.

Основное различие между VPR и RapidSmith/RapidSmith2 заключается в том, что инструменты RapidSmith нацелены на коммерческие ПЛИС Xilinx и предоставляют возможность взаимодействия со стандартным маршрутом проектирования Xilinx на любом его этапе. Все функции, доступные из TCL-интерфейса, доступны также в RapidSmith и RapidSmith2. VPR же ограничен функциями в архитектурном описании объектов VPR и плохо приспособлен для исследования архитектур коммерческих ПЛИС, хотя существуют примеры такого применения, например для ПЛИС Xilinx серии Virtex-6.

## TINCR – НАБОР БИБЛИОТЕК ДЛЯ VIVADO

Tincr представляет собой набор TCL-библиотек, предназначенных для САПР Vivado от Xilinx. Они дают возможность разработчикам создавать собственные инструменты САПР на основе Vivado. Процесс упрощается благодаря двум методологиям, реализованным в отдельных TCL-библиотеках – TincrCAD и TincrIO [7, 8].

TincrCAD – это программный интерфейс приложения (API) TCL, надстроенный поверх собственных команд TCL в Vivado. Он содержит набор команд, которые часто используются при разработке собственных инструментов САПР, и предоставляет разработчикам более высокие уровни абстракций, повышая производительность и детализацию информации.

TincrCAD был разработан, чтобы максимально нарастить производительность. При разработке инструментов САПР, предназначенных для ПЛИС, сокращение на миллисекунды времени выполнения часто встречающейся команды на критическом пути внутреннего

цикла может сэкономить часы выполнения алгоритма на реальном проекте. С этой целью TincrCAD использует несколько механизмов кэширования для быстрого получения важных данных.

Вторая особенность TincrCAD – простота программирования. Vivado предоставляет множество команд для реализации инструментов САПР, но они сложные и многословные. Программисту может понадобиться час на написание одной команды. TincrCAD решает эту проблему, преобразуя команды низкого уровня в наборы функций высокого уровня. Эти функции организуются в «ансамбли» (ensembles), которые предоставляют программисту возможность инкапсуляции.

TincrIO предлагает набор команд для выгрузки данных дизайна проекта в Vivado и ПЛИС в открытый формат с возможностью последующего анализа. С помощью TincrIO пользователи могут генерировать описание проекта в ПЛИС в формате XDLRC и экспортировать проекты из Vivado в Tincr-снимок (Tincr checkpoint) вместе с EDIF, данными размещения, маршрутизации и т. д.

TincrIO работает только с двумя типами файлов: Tincr-снимком и файлами отчетов XDLRC. Снимок Tincr – архив файлов, которые хранят список соединений проекта и его отображение на физические компоненты в ПЛИС. TincrIO предоставляет функции для импорта и экспорта контрольных точек в Vivado и из него. В настоящее время формат снимков Tincr является экспериментальным, поэтому при импорте в Vivado возможны ошибки.

Файлы XDLRC (или файлы отчетов XDL) – наследники предыдущих инструментов проектирования Xilinx. Файл XDLRC содержит полное описание устройства и его ресурсов вплоть до уровня базового элемента (BEL). TincrIO может генерировать полный файл XDLRC самостоятельно. Несмотря на невозможность получения списка соединений в пределах секции, TincrIO может быть объединен с инструментом RapidSmith, чтобы просто и быстро сгенерировать эту информацию вручную.

## RapidWright – ПЛАТФОРМА С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ ОТ XILINX RESEARCH LABS

Инструменты RapidWright напрямую поддерживаются САПР Vivado [4]. На рис. 1 показана взаимосвязь компонентов RapidWright и Vivado. RapidWright позволяет настраивать реализации проектов исходя из пользовательских задач и предоставляет методологию проектирования с применением предварительно созданных модулей.

RapidWright был переработан из исходного кода RapidSmith. Модель устройства, реализованного в ПЛИС, стала детальнее, производительность и эффективность



# ЭЛЕКТРО

МИНПРОМТОРГ  
РОССИИ



29-я международная выставка  
«Электрооборудование. Светотехника.  
Автоматизация зданий и сооружений»

# 8-11.06.2020

Россия,  
Москва

ЦВК «ЭКСПОЦЕНТР»,  
Краснопресненская наб., 14

Павильон №2  
(залы 1, 2)

[www.elektro-expo.ru](http://www.elektro-expo.ru)



12+



Реклама

**6G** ЭКСПОЦЕНТР



**ЭЛЕКТРО  
МАРКЕТ**

ВАЖНЫЕ СВЯЗИ  
ДЛЯ ВАЖНЫХ ДЕЛ



**ЭЛЕКТРО  
TALK**

РАЗГОВОРЫ  
С ТОЛКОМ



**ЭЛЕКТРО  
SKILLS**

ПРОКАЧАЙ НАВЫКИ  
И КОМПЕТЕНЦИИ

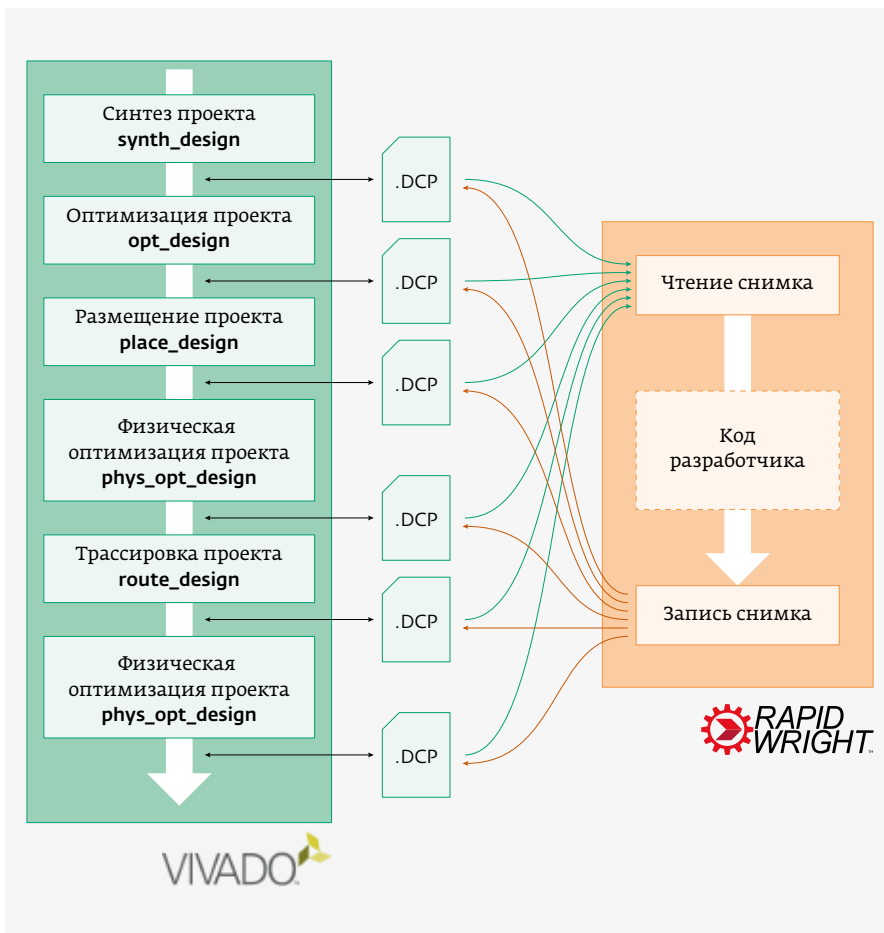


Рис. 1. Интеграция инструментов САПР Vivado и RapidWright

повысились, требования к памяти снизились, добавилось несколько функций и возможностей, основанных на концепции проекта Vivado. Отличительная функция RapidWright – способность читать и записывать незашифрованные файлы снимка дизайна Vivado (Vivado Design Checkpoint). RapidWright поддерживает полное представление как о логическом, так и физическом списке соединений проектов ПЛИС.

## ТОРС – НАБОР ИНСТРУМЕНТОВ ДЛЯ ОТКРЫТЫХ РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛЕНИЙ

Набор инструментов для реконфигурируемых вычислений с открытым исходным кодом Torcs был выпущен под лицензией GNU GPL 3.0 [9]. Он подходит для разработки собственных инструментов САПР и исследования архитектуры ПЛИС.

Инфраструктура набора инструментов Torcs (рис. 2) может считывать, записывать и манипулировать списками соединений EDIF и XDL, а также пакетами битового потока Xilinx (но не внутренними объектами конфигурации).

Torcs предоставляет исчерпывающую информацию, полученную из непатентованных источников, о внутренних межсоединениях и внутренней логике основных семейств ПЛИС Xilinx. В настоящее время поддерживаются все ПЛИС семейств Virtex, VirtexE, Virtex2, Virtex2 Pro, Virtex4, Virtex5, Virtex6, Virtex6L, Spartan3E, Spartan6 и Spartan6L.

Torcs состоит из четырех основных API-интерфейсов (на рис. 2 показаны на синем фоне) и инструментов САПР (показаны на сером фоне). Модули, отмеченные красной точкой, находятся в разработке. Блоки, выделенные серым цветом, в настоящее время не планируются либо являются внешними инструментами, с которыми Torcs мог бы взаимодействовать.

## OpenPR – НАБОР ИНСТРУМЕНТОВ ДЛЯ ЧАСТИЧНОЙ РЕКОНФИГУРАЦИИ ПЛИС XILINX

OpenPR представляет собой набор инструментов с открытым исходным кодом для частичной реконфигурации ПЛИС Xilinx [10]. OpenPR функционально практически идентичен другому набору инструментов для исследования ПЛИС – Xilinx Partial Reconfiguration Early Access для ISE9.2i (Xilinx PR Toolkit), который не входил в официальный выпуск ISE, но была возможность внедрения модуля в поток разработки. Xilinx PR Toolkit позволяет частично реконфигурировать ПЛИС на основе слотов – фиксированных (по размеру и по положению) геометрических областей для выбранной ПЛИС.

ООО  
СМП

ИНТЕРНЕТ-МАГАЗИН  
www.SMD.ru

электронные  
для поверхностного  
монтажа

КОМПОНЕНТЫ

НОВОЕ В ПРОГРАММЕ ПОСТАВОК

- Низкоомные резисторы типоразмеров 0402 и 0603
- Керамические конденсаторы большой емкости

Москва, Ленинградский пр., 80 к. 32, e-mail: sale@smd.ru  
Тел.: +4991 158-7396, +4991 940-6244, +4991 943-8782

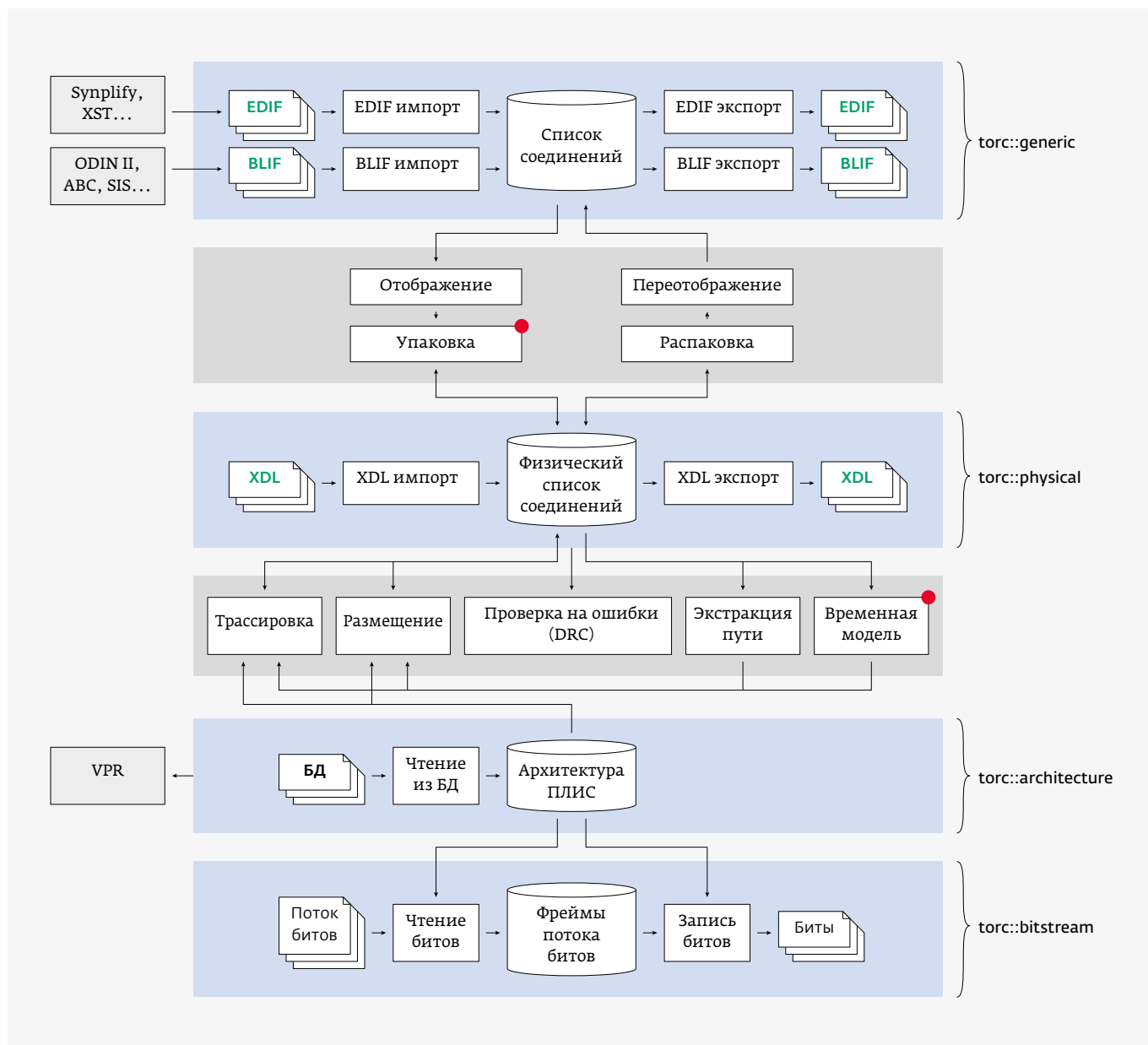


Рис. 2. Инфраструктура Torc

OpenPR позволяет создавать базовый статический дизайн с одним или несколькими частично реконфигурируемыми слотами. Он выпускается по открытой лицензии, поэтому другие исследователи могут расширять проект, чтобы преодолеть ограничения ISE. OpenPR отличается от других инструментов частичной реконфигурации, таких как JBits, тем, что не зависит от информации, являющейся собственностью Xilinx.

### ИЕРАРХИЯ В АРХИТЕКТУРЕ ПЛИС XILINX КЛАССА SпK

Все современные серии ПЛИС Xilinx имеют одинаковую архитектуру, состоящую из шести основных уровней

Санкт-Петербург, Россия  
ул. Матроса Железняка,  
д. 57, лит. А, пом. 126-Н  
Телефон: 7-812-3259792

Москва, Россия  
Лужницкая набережная, 2/4,  
строение 19, офис 119  
Телефон: 7-095-1477590

**VITAL-IC**

Поставки электронных компонентов  
широкой номенклатуры  
Системы RFID: поставка и консультации

**XILINX** **Mini-Circuits**  
**ALTERA**

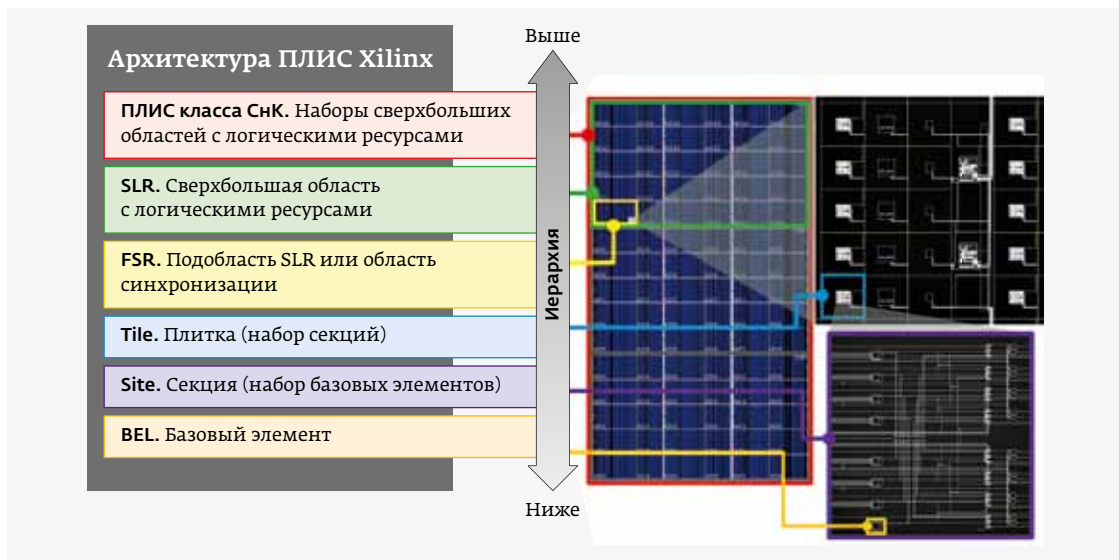


Рис. 3. Уровни иерархии в ПЛИС Xilinx

иерархии (рис. 3). На данный момент САПР Vivado поддерживает три серии ПЛИС Xilinx: Series 7, UltraScale и UltraScale+. Каждая серия может содержать несколько семейств, оптимизированных по стоимости, мощности, производительности, размеру и другим параметрам [8].

На высшем уровне иерархии архитектуры Xilinx (см. рис. 3) – **интегральная схема (ПЛИС)**, которая представляет собой двумерный массив, состоящий из **подобласти с логическими ресурсами FSR (Fabric Sub Region) или области синхронизации** для одной **сверхбольшой области с логическими ресурсами SLR (Super Logic Regions)** или двух либо более SLR в случае ПЛИС класса СнК. В свою очередь, FSR содержит двумерный массив **функциональных блоков (плиток, или Tile)**. Например, в ПЛИС UltraScale все области синхронизации имеют высоту 60 конфигурируемых логических блоков (КЛБ), но их ширина варьируется в зависимости от сочетания используемых типов плиток.

Уровень иерархии SLR есть только в ПЛИС Xilinx, которые используют технологию многоуровневого соединения кремниевых пластин (SSIT), представляющих собой 3D-сборку с использованием коммутационной платы (интерпозер) и TSV-технологии.

Атомарная единица ПЛИС Xilinx – **базовый элемент (BEL)**. Выделяют два типа BEL: базовые элементы логики (Basic Elements of Logic) и трассировочные базовые элементы (Routing BELs). BEL первого типа объединяются в секции, которые поддерживают реализацию ячейки (cell) дизайна проекта. Каждый базовый элемент поддерживает один или более типов UNISIM-ячеек, которые описаны в руководстве к библиотекам для ПЛИС серии Series 7 и ПЛИС серии UltraScale. Сопоставление между конечной ячейкой в списке соединений и базовым элементом секции называют «размещением» ячейки. Таким

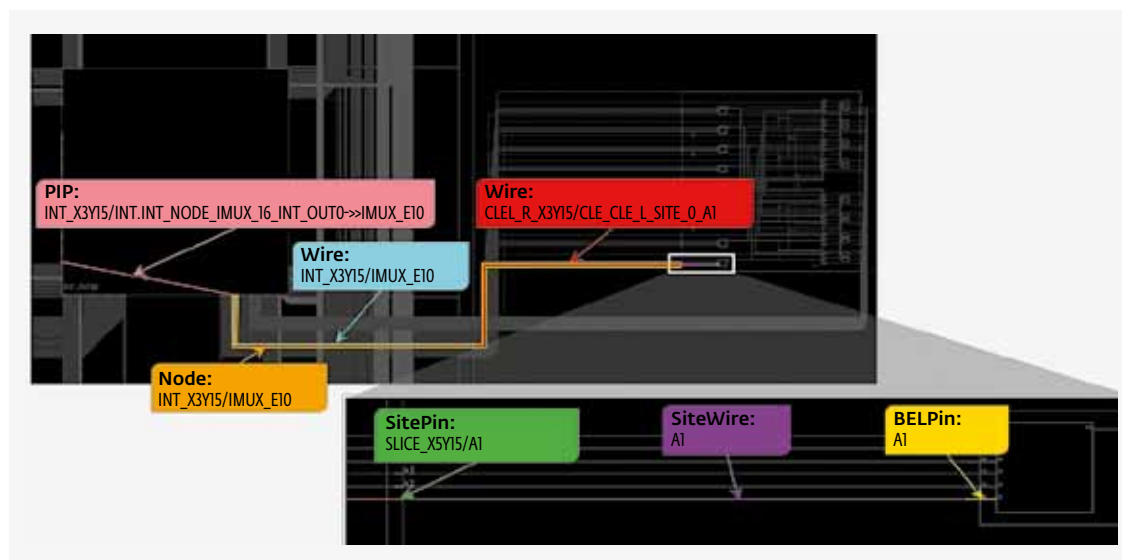
образом, запуск TCL-команды Vivado place\_design, по сути, сопоставляет все конечные ячейки в списке соединений на совместимые и допустимые секции базовых элементов.

Трассировочные базовые элементы – это программируемые коммутаторы на мультиплексорных структурах, предназначенные для маршрутизации сигналов между базовыми элементами. Они не поддерживают элементы дизайна проекта, а используются для трассировки сигналов в ПЛИС. Однако некоторые трассировочные базовые элементы имеют опциональную инверсию.

Базовые элементы содержат входные и выходные выводы, снабжены конфигурируемыми соединениями, которые могут соединять входные выводы с выходными. Такие соединения называют программируемыми точками коммутации (PIP). Как логические базовые элементы, так и трассировочные базовые элементы имеют секционные программируемые межсоединения. Однако в случае логических элементов секция должна быть незанятой ячейкой, чтобы можно было использовать маршрутизацию. Часто эти секционные программируемые межсоединения при реализации в базовых элементах логики называют маршрутными мостами (route through). При трассировке проекта иногда необходимо провести соединения через неиспользованную LUT или другой базовый элемент логики с помощью программируемого межсоединения секции.

Группу связанных элементов и их соединения называют **секцией (Site)**. Секция может содержать три основные категории объектов: базовый элемент (логический или трассировочный); выводы секции (Site Pin) – внешний вход и выход секции; межсоединения секции (Site Wire) – соединяют элементы между собой и выводы секции.

Секция, которая является экземпляром типа, присваивается уникальное имя с суффиксом \_X#Y#,



**Рис. 4.** Различные трассировочные ресурсы в ПЛИС Xilinx, доступные в RapidWright

определяющим ее местоположение на сетке типов секций. Каждый тип секции имеет свою собственную, независимую от других, координату в XY-системе. Исключения составляют секции типов SLICEL и SLICEM, которые делят одно сеточное пространство. Секции SLICEL и SLICEM – основные программируемые логические строительные блоки, содержат LUT и триггеры, формируют скелет структуры ПЛИС.

Секции располагаются по всей площади кристалла, каждый экземпляр секции соответствует какому-либо типу секции из архитектуры семейства ПЛИС. Кроме того, секции ПЛИС иногда способны принимать разные типы, однако, если функциональный блок обозначен, назначается первичный тип секции.

На абстрактном уровне ПЛИС Xilinx созданы с помощью сборки сетки из функциональных блоков (плиток, или Tile). Аналогично секции каждый функциональный блок является экземпляром какого-либо типа и каждый функциональный блок имеет уникальное имя с суффиксом \_X#Y#. Функциональные блоки используются при реализации проекта на ПЛИС. Они спроектированы так, чтобы примыкать друг к другу. Поскольку у функциональных блоков нет портов ввода или вывода, существуют узлы (nodes), которые описывают соединения между функциональными блоками. Узел является набором электрически соединенных проводов, которые перекрывают один или более функциональных блоков. На рис. 4 показаны различные трассировочные ресурсы в ПЛИС Xilinx, доступные в RapidWright: PIP, Node, Wire, SitePin, SiteWire, BELPin.

\*\*\*

Создание собственных программных инструментов на основе зарубежных САПР для коммерческих ПЛИС является сложной, практически не выполнимой задачей.

Закрытые базы данных зарубежных ПЛИС и отсутствие поддержки программных интерфейсов коммерческих САПР тормозят исследования новых архитектур ПЛИС СНГ. Рассмотренные в статье программные инструменты с открытым кодом, широко распространенные в научной среде, помогают исследовать перспективные архитектуры ПЛИС.

## ЛИТЕРАТУРА

1. Github repository of RapidSmith2. – <https://github.com/byuccl/RapidSmith2/blob/master/docs/VSRTUserGuide.pdf>
2. **Строгонов А., Кривчун М.** Исследование трассировочной способности архитектур промышленных ПЛИС Xilinx с помощью программного инструмента RapidSmith2 // Компоненты и технологии. 2019. № 5. С. 90–94.
3. **Строгонов А., Горлов М., Городков П.** САПР с открытым кодом для реализации проектов в ПЛИС // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. 2018. № 3. С. 1–8.
4. RapidWright.Documentation. – <https://www.rapidwright.io/docs/Introduction.html#what-is-rapidwright>
5. **Строгонов А. В.** Реализация цифровых устройств в базе программируемых логических интегральных схем. Учебное пособие. – Саратов: Ай Пи Эр Медиа, 2019. 151 с. <http://www.iprbookshop.ru/83658.html>
6. VTR Github Repository. – <https://github.com/verilog-to-routing/vtr-verilog-to-routing>
7. Tincr github repository. – <https://github.com/byuccl/tincr>
8. **Townsend T. J.** Vivado Design Interface: Enabling CAD-Tool Design for Next Generation Xilinx FPGA Devices. All Theses and Dissertations, 2017. – <https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=7492&context=etd>
9. Torc. Tools for open reconfigurable computing. – <http://torc-isi.sourceforge.net>
10. **Sohanghpurwala A. A.** OpenPR: An Open-Source Partial Reconfiguration Tool-Kit for Blacksburg, Virginia, 2010.