DOI: 10.22184/1992-4178.2018.174.3.146.154

САПР с открытым кодом для реализации проектов в академических ПЛИС

А. Строгонов, д. т. н.[1], М. Горлов, д. т. н.[2], П. Городков[3]

УДК 621.3.049 | ВАК 05.27.00

Использование программных инструментов индустриальных САПР для таких задач, как синтез, размещение и трассировка при исследовании академических ПЛИС, связано с трудностями или даже невозможно, поскольку компании – разработчики программных интерфейсов индустриальных САПР не предоставляют достаточной информации о программных интерфейсах ПО, ссылаясь на коммерческую тайну. Базы данных индустриальных ПЛИС также являются собственностью компаний, что усложняет доступ к ним. Поэтому исследования новых архитектур проводятся на гипотетических (академических) ПЛИС с использованием программных инструментов САПР с открытым кодом. Рассмотрим особенности применения наиболее распространенных программных инструментов для исследования академических ПЛИС.

Большая часть работ по академическим архитектурам и исследованиям САПР выполняется с использованием инструмента размещения и трассировки в базис ПЛИС VPR (Versatile Place and Route). Изначально в VPR версии 5.0 не было встроенных средств логического синтеза. Впоследствии VPR вошел в состав САПР VTR 8.0 (Verilog to Routing), который является совместной разработкой трех университетов: Торонто (Канада), Нью-Брансвика (Канада) и Калифорнийского в Беркли (США) [1–6].

В некоторых индустриальных САПР ПЛИС предусмотрена возможность связи с академическими САПР. Например, САПР Altera с помощью программного интерфейса Quartus II University Interface Program (QUIP) производит выгрузку несложных HDL-проектов для некоторых серий ПЛИС в формате blif (Berkeley Logic Interchange Format), который поддерживается многими инструментами с открытым кодом. Это позволяет выполнить оценку HDL-проектов, реализованных с помощью указанных инструментов. Для этого извлекают информацию о задержках в критичных путях и сравнивают с данными, полученными для HDL-проектов, созданных с помощью индустриальных САПР.

В САПР VTR 8.0 предусмотрены различные архитектурные xml-файлы, которые позволяют разрабатывать и исследовать ПЛИС, близкие по техническим характеристикам к ПЛИС компании Altera (Intel FPGA) серии Stratix IV с адаптивными логическими элементами, умножителями и блоками памяти. В документации к VTR 8.0 приводится фрагмент архитектурного файла, описывающий секцию конфигурационных логических блоков (КЛБ) ПЛИС компании Xilinx серии Virtex-6.

В xml-файлах используются различные конструкции языка, позволяющие создавать как логические блоки, так и межсоединения. Логические блоки необходимы для представления основных логических, вычислительных (умножители) и запоминающих устройств (однопортовая и двухпортовая память) в ПЛИС. Межсоединения обеспечивают связность внутри и между логическими блоками.

Сложные блоки можно создавать посредством иерархии тегов <pb_type>. Межсоединения в ПЛИС описываются с помощью тега <interconnect>. А тег <mode> позволяет строить логические блоки с различными режимами работы: например, блоки памяти ПЛИС могут быть сконфигурированы как 512×8 или 1024×4. Кроме того, с помощью этого тега можно задавать режимы конфигурации умножителя, в частности, один умножитель с размерностью операндов 18×18 или два умножителя с размерностью 9×9. Можно построить перестраиваемый логический элемент (в VTR 8.0 логический элемент принято обозначать BLE), таблица перекодировки (LUT) которого функционирует в двух режимах – как одна 6-входовая или как две 5-входовых с тремя общими входами [4, 5].

На рис. 1 показана внутрикластерная (локальная) коммутация в используемом в VTR 8.0 КЛБ академической ПЛИС с применением коммутатора 60×60. КЛБ состоит из 10 ЛЭ (BLE), а ЛЭ имеет 6-входовую перестраиваемую таблицу перекодировок, которая может быть сконфигурирована

[1] Воронежский государственный технический университет, профессор кафедры полупроводниковой электроники и наноэлектроники, тел. +7 4732 43-76-95, andreistrogonov@mail.ru.

[2] Воронежский государственный технический университет, профессор кафедры полупроводниковой электроники и наноэлектроники, тел. +7 4732 43-76-95, m-gorlov@mail.ru.

[3] Воронежский государственный технический университет, аспирант кафедры полупроводниковой электроники и наноэлектроники, тел. +7 4732 43-76-95, gorodkoff@gmail.com.

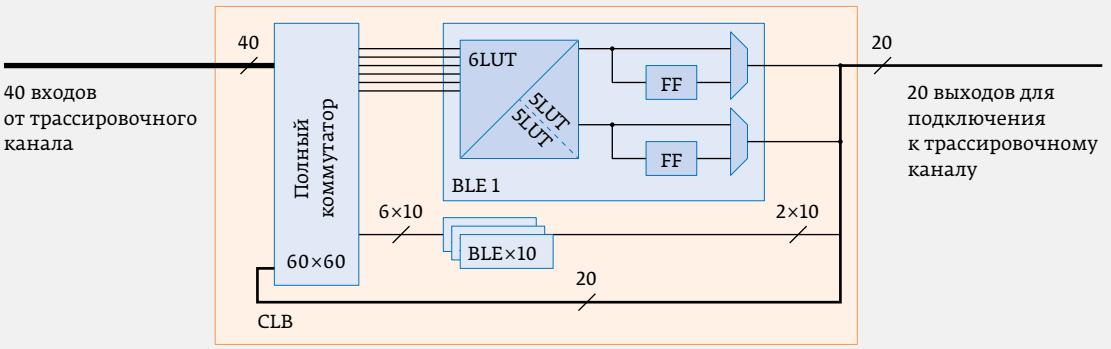

Рис. 1. Конфигурационный логический блок в САПР VTR 8.0 (внутрикластерная коммутация реализуется с использованием полных коммутаторов)

как 6-входовая LUT или как две 5-входовых LUT с пятью общими входам.

Используемый в VTR синтаксический анализатор ODIN-II на основе описания архитектуры в XML-файле определяет гетерогенные блоки (умножители и блоки памяти) с точки зрения входов, выходов и возможной конфигурации, а также создает файл в blif-формате, где описаны логика устройства и черные ящики, которые определяют соединения с остальной частью системы.

С помощью программного инструмента ABC проводятся аппаратно-независимая логическая оптимизация схемы и технологическое отображение (размещение в логические боки) в базис LUT ПЛИС.

Выделяют следующие этапы обработки Verilog-кода синтезатором ODIN-II: построение абстрактного синтаксического дерева (AST), препроцессинг, оптимизация AST, конвертация AST в плоский нетлист и его дальнейшая оптимизация, частичное отображение в базис ПЛИС (рис. 2).

Затем логическая сеть отображается на целевой размер LUT ПЛИС. ABC не использует информацию о целевой архитектуре ПЛИС, кроме размера LUT. Результирующая логическая сеть отображается в технологический базис ПЛИС. ABC выводит оптимизированный и обработанный blif-файл.

VPR, который является ядром VTR, обеспечивает размещение кластеров и гетерогенных блоков на кристалле, оптимально организует глобальные и локальные трассировочные ресурсы для меж- и внутрикластерной связи логических блоков. КЛБ размещаются по алгоритму «имитация отжига», который может быть реализован на основе четырех ключевых компонентов: представления состояния текущего решения, набора перемещений из одного состояния в другое, целевой функции стоимости для оценки каждого состояния и схемы охлаждения, определяющей переход от начального поиска к локальной оптимизации.

Для разводки электрических связей в VPR используется модифицированный алгоритм PathFinder для выполнения полного либо управляемого процесса трассировки. VPR может выполнять трассировку Verilog-проекта с заданной шириной трассировочного канала или определить необходимую для этого минимальную ширину.

В [7] VTR доработан до возможности генерации синтезируемой RTL-модели архитектуры академической ПЛИС и битового потока для этой архитектуры. Текущие ограничения учитывают только то, что архитектурный файл позволяет создать гомогенную ПЛИС с системой коммутации, как у ПЛИС серии Stratix III, без использования блоков памяти и умножителей. Но это многообещающий шаг к более легкой разработке пользовательских архитектур. Предполагается, что RTL-модель подойдет для проектирования ПЛИС по 65-нм технологическому КМОП-процессу кремниевой фабрики TSMC с использованием метода стандартных ячеек.

Для исследования ПЛИС применяется еще один инструмент с открытым кодом – RapidSmith, который обеспечивает возможность подключения VPR к коммерческим ПЛИС Xilinx. В настоящее время доступна версия RapidSmith 2 с новым упаковочным инструментом RSVPack, позволяющим принимать информацию со стадии синтеза в САПР Xilinx с поддержкой Vivado для новых серий ПЛИС [7, 8].

ПО RapidSmith содержит новый набор инструментов и API для исследования, создания и тестирования новых алгоритмов размещения и трассировки на индустриальных ПЛИС Xilinx. RapidSmith, написанный на Java, основан на языке XDL. Интерфейс QUIP САПР Quartus II напоминает XDL, однако QUIP, в отличие от инструмента Xilinx xdl, не предоставляет сведения, необходимые для построения детальной трассировки.

К сожалению, XDL не совсем понятен многим пользователям и его потенциал часто не реализуется. RapidSmith способен преодолеть некоторые проблемы использования языка XDL, которые затрудняли его применение в прошлом.

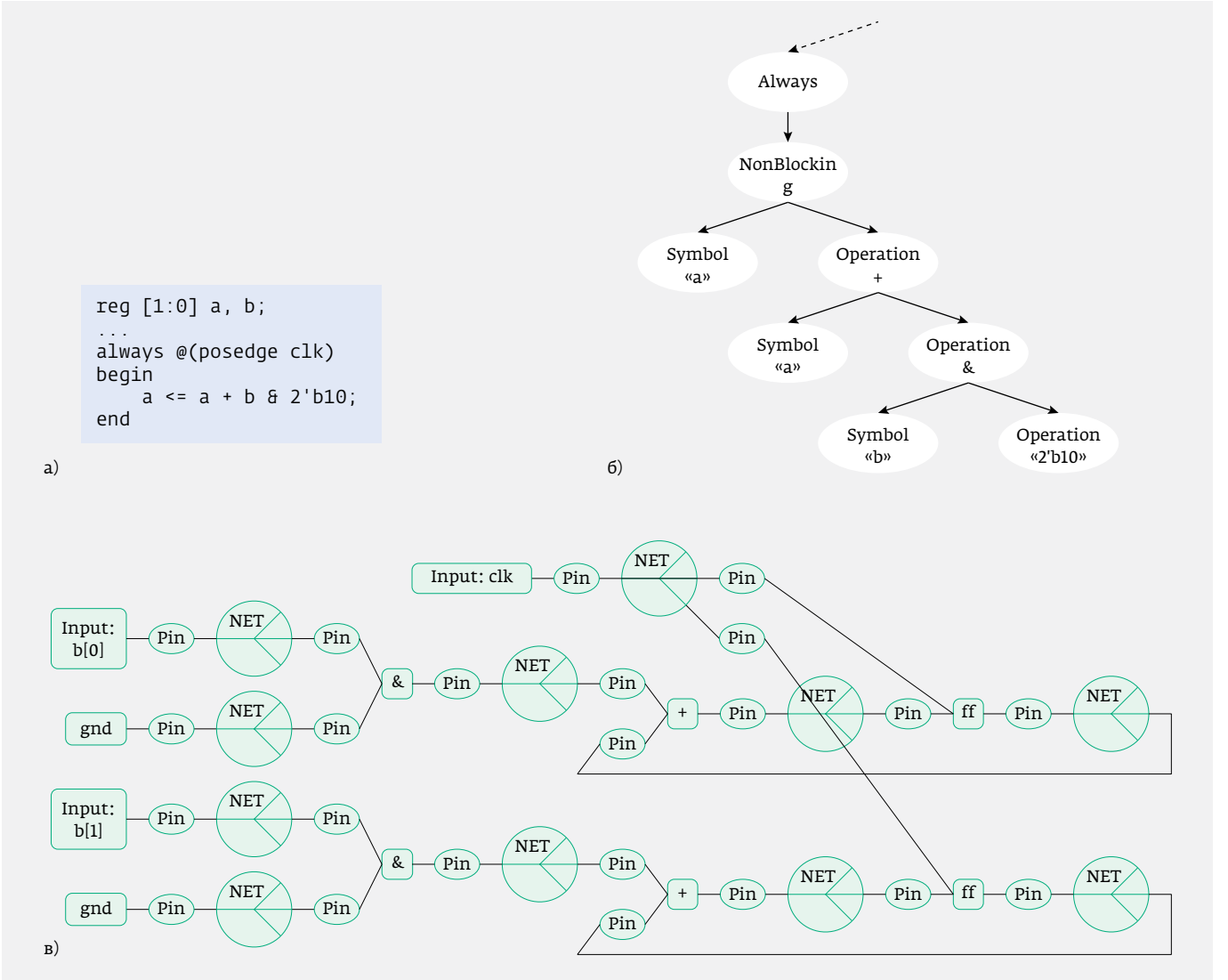

Рис. 2. Представление структурных данных в ODIN II: а) фрагмент Verilog-кода; б) абстрактное синтаксическое дерево (AST); в) плоский нетлист

XDL представляет собой читаемый формат ASCII, совместимый с более широко распространенным форматом NCD (Native Circuit Description) – двоичным форматом Xilinx. XDL обладает большей частью возможностей NCD, и Xilinx предоставляет исполняемый файл xdl, который может преобразовывать проекты NCD в XDL и обратно.

XDL и NCD являются встроенными форматами Netlist Xilinx для описания и представления проектов ПЛИС Xilinx. XDL – это интерфейс, используемый RapidSmith для размещения и извлечения информации о проекте в разных точках маршрута проектирования ПЛИС Xilinx (рис. 3).

RapidSmith преобразует «нечитаемый двоичный формат» NCD-файла, применяемого на различных стадиях проектирования ПЛИС (размещение, трассировка) в удобочитаемый эквивалент XDL-формата (текстовый XDLRC-файл) для последующего синтаксического анализа, манипуляций и экспорта (см. рис. 3). Однако XDL-формат Xilinx не поддерживается внешними средствами. XDLRC-файл содержит описание ресурсов ПЛИС в терминах архитектур Xilinx (КЛБ, блоки ввода-вывода, программируемые межсоединения, блочная память, ЦОС-блоки, микропроцессорные ядра, например PowerPC) и их межсоединений. На рис. 4 показан фрагмент XDLRC-файла, включающий такие конструкции, как логические плитки (tiles, это могут быть как КЛБ, так и умножители, блоки памяти), узлы (primitive site, например секции в КЛБ SLICEL или SLICEM), межсоединения (wire), программируемые точки межсоединений (PIP) внутри маршрутизаторов на мультиплексорных структурах. Логическая плитка состоит из одного или нескольких узлов, а узлы, в свою очередь, – из нескольких базовых логических элементов (BLE), связанных мультиплексорами. Таким образом, в ПЛИС Xilinx используется

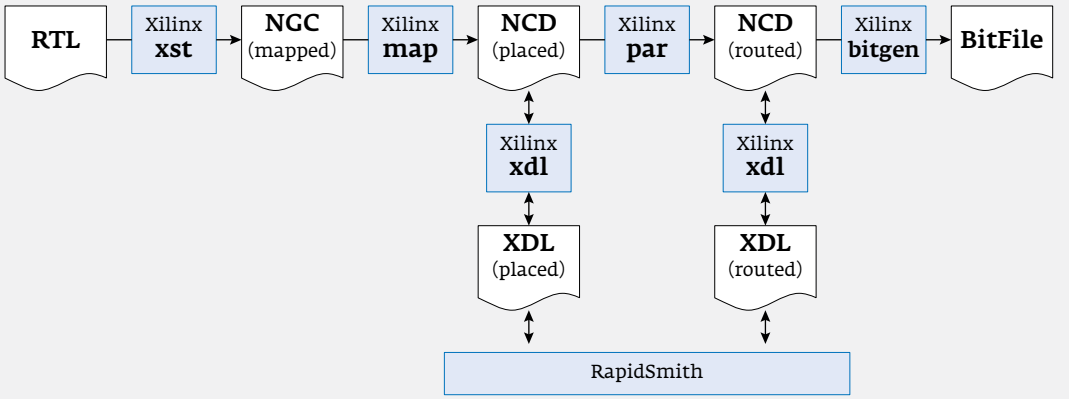

Рис. 3. Цепочка программных модулей САПР Xilinx ISE и точки, в которых NCD-файлы могут быть преобразованы в XDL-файлы и впоследствии использованы программными инструментами САПР RapidSmith

трехуровневая иерархия: плитка, узел и базовый логический элемент.

Программный модуль XST САПР Xilinx ISE объединяет в себе процесс синтеза и технологического отображения, модуль map – упаковку ЛЭ в секции КЛБ и размещение (секции, сформированные на этапе map, получают свои места в матрице КЛБ), модуль par – трассировку между КЛБ и другими блоками. На рис. 3 показаны точки в цепочке программных модулей ISE, в которых NCD-файлы преобразуются в XDL-файлы. Модуль map читает файл NGC после синтеза, технологического отображения и записывает результат в файл NCD (placed). Трассировщик par читает файл NCD (placed) на стадии размещения и записывает результат своей работы в файл NCD (routed) [8].

Подобно предусмотренному в VTR упаковщику AAPack, упаковщик RSVPack в ПО RapidSmith 2 использует «жадный» эвристический алгоритм упаковки, что позволяет применять эти инструменты в составе САПР для разработки архитектур ПЛИС, схожих с индустриальными ПЛИС, такими как Virtex 6. RSVPack поддерживает упаковку таких блоков, как мультиплексоры F7/F8, LUTRAM (LUT в режиме ОЗУ) и SRL (LUT как сдвиговый регистр) в секции SLICEM.

VTR не читает синтезированный список соединений от синтезатора XST САПР Xilinx ISE, так как использует

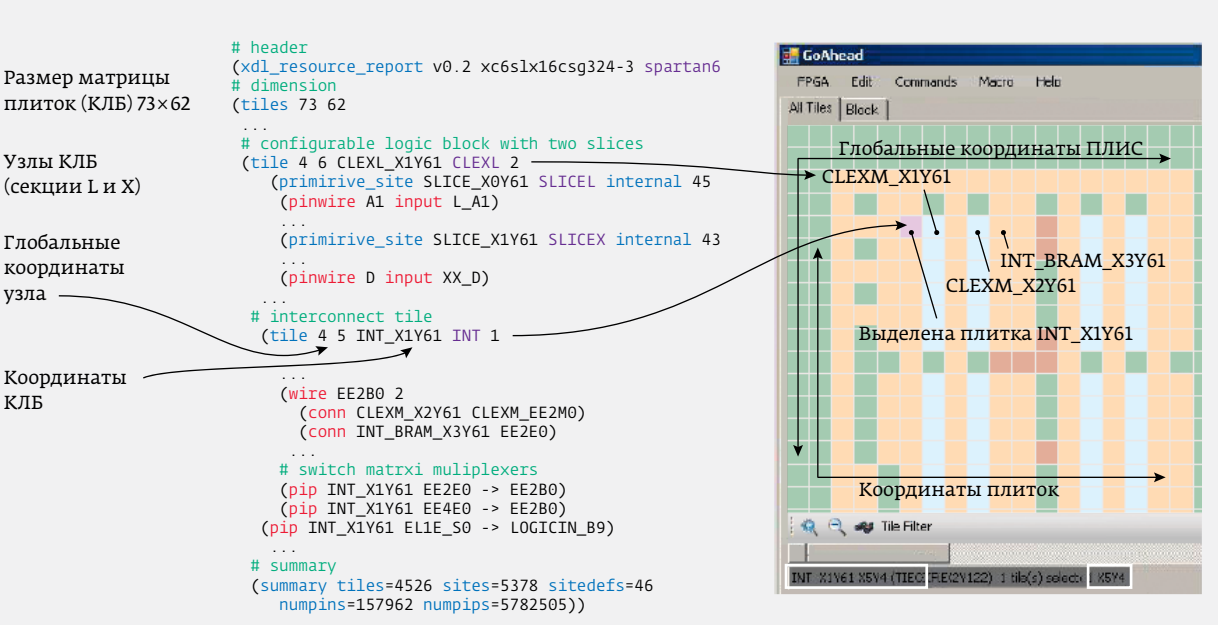

Рис. 4. Фрагмент XDLRC-файла с отображением ресурсов на кристалле ПЛИС серии Spartan-6

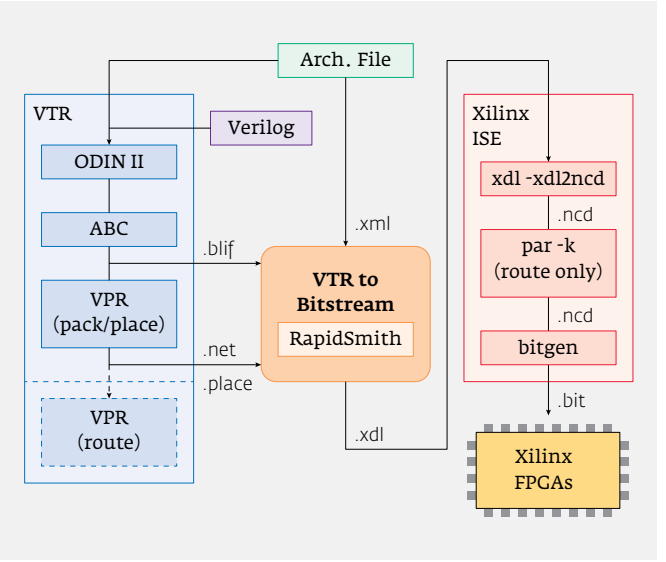

Рис. 5. Реализация академических ПЛИС, разработанных в VTR, в базисе индустриальных ПЛИС Xilinx

синтезатор ODIN II. Поэтому на основе ПО RapidSmith был разработан программный инструмент VTR to Bitstream (VTB), позволяющий реализовывать созданные в VTR академические ПЛИС в базисе индустриальных ПЛИС Xilinx [8].

Комбинации файлов упаковочного списка соединений (.net), размещения (.place) и трассировки может быть достаточно для того, чтобы VPR генерировал битовый поток, но в архитектурном XML-файле ПЛИС отсутствует определение того, как будет реализована конфигурация каждой логической функции, трассировки и гетерогенных блоков в целевой ПЛИС. ПО RapidSmith читает эти файлы и сопоставляет их с популярными сериями ПЛИС Xilinx для генерации битового потока с применением инструментов манипулирования битовым потоком САПР Xilinx.

Программный инструмент VTR to Bitstream использует тот же архитектурный файл – blif-файл (LUT плюс триггеры), созданный с помощью ABC, что и VTR после аппаратно-независимой логической оптимизации и технологического отображения в базис LUT, net-файл (упаковка LUT и триггеров в КЛБ и их соединения), файл размещения в формате.place, созданный VPR, и выдает файл в формате .xdl. С помощью исполняемого файла xdl формат .xdl преобразуется в двоичный формат.ncd, который используется в САПР Xilinx ISE. Затем САПР Xilinx ISE выполняет трассировку и генерацию битового потока (файл .bit) для программирования ПЛИС (рис. 5).

Для реализации в базисе индустриальных ПЛИС Xilinx серии Virtex-6 необходимо изменить базовый КЛБ, используемый в САПР VTR 8.0. В [9] предлагается модель КЛБ с разряженным коммутатором (рис. 6). В более поздних экспериментах с архитектурами [10] были введены аппаратные сумматоры на выходах LUT (XADDER) для организации вертикальных цепей переноса для каскадирования сумматоров, а также доработана модель трассировочных ресурсов с использованием однонаправленных и двунаправленных межсоединений с различной длиной сегментации, проходящих непрерывно через L=1, 2, 4 и 16 кластеров в горизонтальном, вертикальном, диагональном и изогнутых направлениях (рис. 7). Для более точного соответствия индустриальным ПЛИС серии Virtex-6 локальные

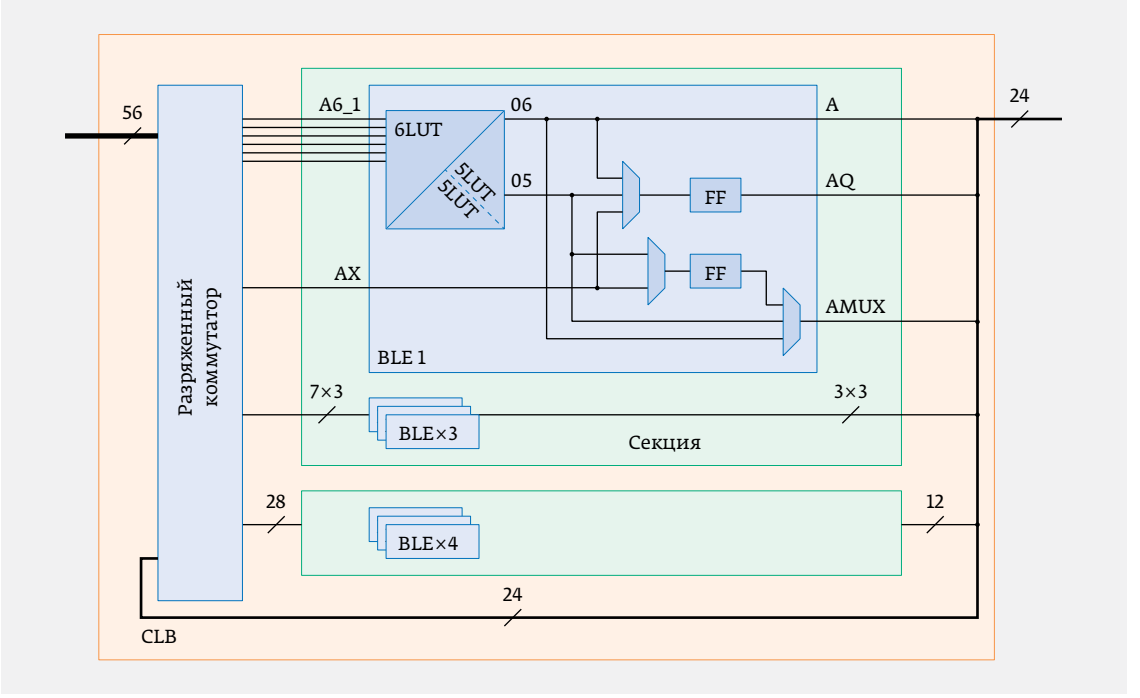

Рис. 6. Предлагаемая модель КЛБ ПЛИС серии Virtex-6 с разряженным коммутатором для реализации в САПР VTR

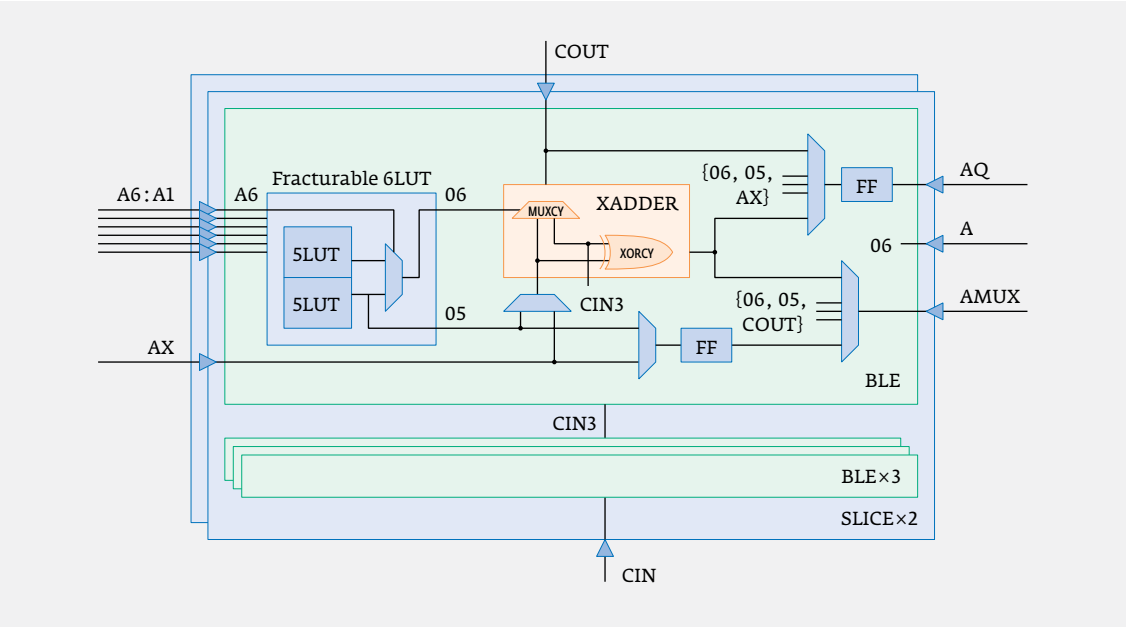

Рис. 7. Доработанный КЛБ, используемый в архитектурном файле VTR для соответствия ПЛИС серии Virtex-6

коммутаторы в КЛБ вынесены в маршрутизаторы трассировочных каналов.

В [10] исследовалось использование программных инструментов VTR, Yosys и VTR to Bitstream (рис. 8) в виде маршрутов проектирования (маршруты 2, 3, 4, 5) и программных модулей индустриального САПР Xilinx ISE (маршрут 1) при реализации Verilog-проектов в базис индустриальных ПЛИС Xilinx серии Virtex-6.

В маршрутах 3, 4 и 5 используется новая версия программного инструмента VTR to Bitstream V2 с RapidSmith. Маршрут 2 предполагает совместное использование Yosys / ABC с последующим применением утилит map и par и трассировкой связей между сигналами секций САПР Xilinx ISE. Входным файлом утилиты ngdbuild является edif-файл, формируемый Yosys. Выходной файл утилиты par – .ncd, который преобразуется утилитой bitgen в битовый поток .bit. Маршрут 3 использует связку Yosys + VPR + par, маршрут 4 – Yosys + VPR, маршрут 5 – VTR, маршрут 1 – утилиты ISE. Проводились тесты производительности (Verilog-проекты) bgm, stereo2, mcml, des50, LU32PE и AESx3.

В качестве оценки выбиралось среднее геометрическое значение числа задействованных секций КЛБ ПЛИС Viertex-6 XC6VLX240t тестов производительности для разных маршрутов (метрика Geomean). Максимальный разрыв метрики для маршрутов 2, 3 и 4 по отношению к метрике маршрута 1 с использованием утилит ISE составляет 82%.

Как показал статистический анализ временных задержек с помощью утилиты Xilinx TRCE для маршрутов 1, 2, 3, 4, на этапе синтеза задержки при реализации Verilog-проектов в базис ПЛИС Virtex-6 XC6VLX240t для маршрута 2 больше на 31% относительно маршрута 1, на этапе кластеризации и размещения – на 10% и на этапе трассировки – на 15%. Следовательно, VTR производит синтез более низкого качества, чем САПР Xilinx ISE. В ходе тестирования выяснилось, что ODIN II не может синтезировать проекты des50 и AESx3, а Yosys синтезирует все шесть проектов. Кроме того, в процессе синтеза HDL-кода может образовываться избыточная (неиспользуемая) логика и наблюдаться рост числа используемых LUT и триггеров в проекте, в результате снижается рабочая частота проекта. Например, в документации на САПР Quartus II рекомендуется в некоторых случаях повторный синтез проекта, то есть перевод синтезированного проекта обратно на вентильный уровень и последующий синтез с помощью другого индустриального синтезатора, например Synplify Pro от Synopsys. Считается, что повторный синтез позволяет сократить число используемых логических ресурсов ПЛИС на 4–5%. Для оставшихся четырех проектов VTR показал повышенное число секций КЛБ при реализации в базисе ПЛИС Virtex-6 XC6VLX240t.

* * *

В заключение отметим, что САПР VTR 8.0 позволяет задавать и исследовать сегментацию межсоединений и топологию маршрутизаторов в трассировочных каналах, внутрикластерную коммутацию, структуру логических элементов, коммутацию внутри логического элемента, размеры LUT, различные связи, например между блоками памяти и умножителями в ПЛИС, строить планировку кристалла и многое другое.

Однако VTR не может генерировать битовый поток и не предоставляет средств для создания RTL-моделей архитектур академических ПЛИС, что не позволяет по структурному Verilog-коду синтезировать физическую топологию кристалла для последующего его изготовления.

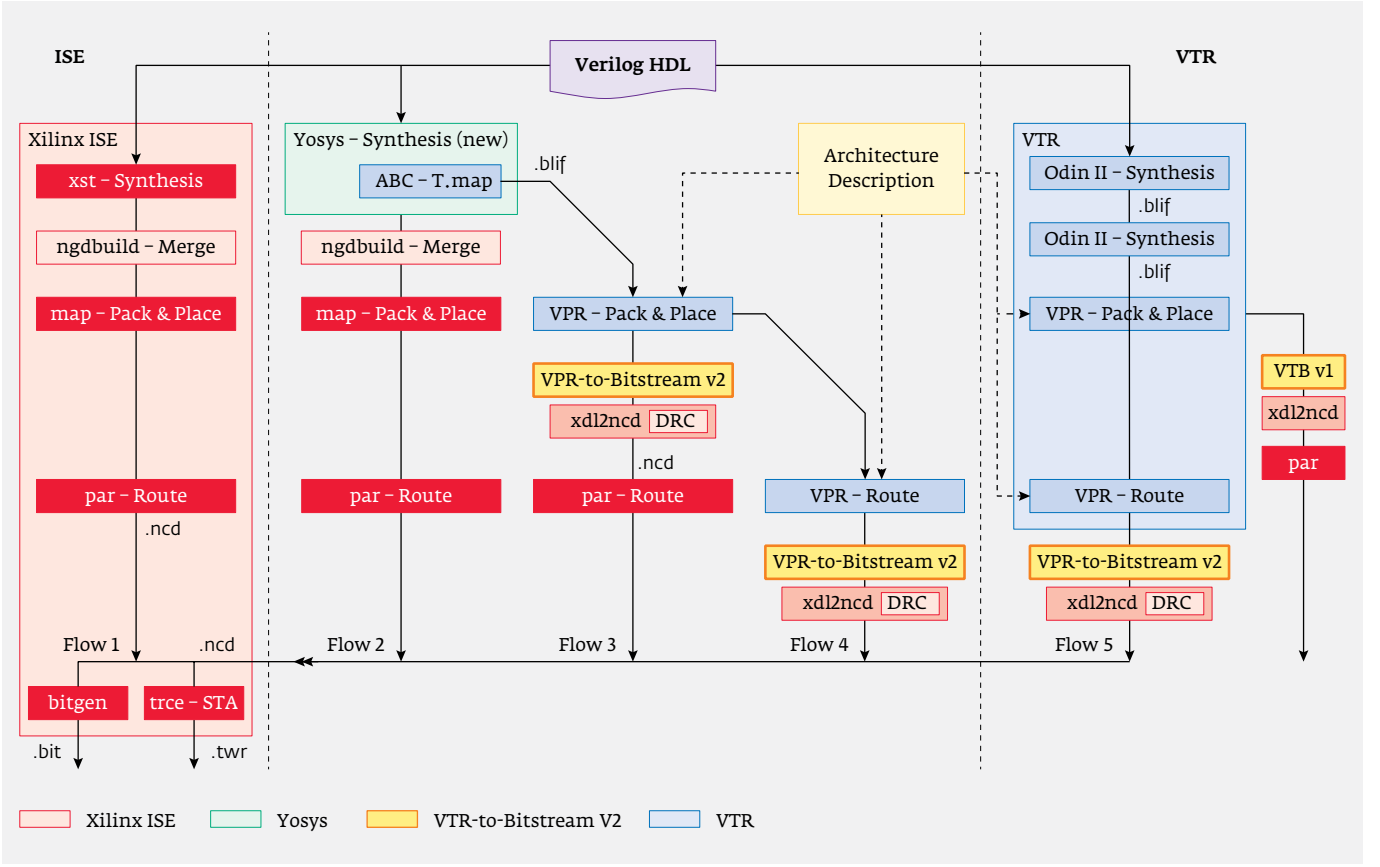

Рис. 8. Маршруты проектирования с использованием индустриального САПР Xilinx ISE и программных инструментов с открытым кодом

В то же время предприняты успешные попытки доработки VTR для генерации RTL-моделей в базисе заказных БИС. Использование ПО RapidSmith позволяет создавать различные инструменты для исследования академических ПЛИС в базисе индустриальных ПЛИС Xilinx. Тем не менее, как отмечает ряд исследователей, реализация ПЛИС серии Virtex-6 в VTR сопряжена с большими трудностями, так как некоторые особенности ЛЭ Virtex-6 не поддерживаются синтезом VTR, например мультиплексоры F7/F8, LUTRAM (LUT в режиме ОЗУ) и SRL (LUT как сдвиговый регистр) в секции SLICEM.

ЛИТЕРАТУРА

1. **Luu J., Anderson J., Rose J.** Architecture Description and Packing for Logic Blocks with Hierarchy, Modes and Complex Interconnect. – FPGA'11, February 27 – March 1, 2011, Monterey, California, USA.
2. **Luu J., Goeders J., Wainberg M. and others.** VTR7.0: Next Generation Architecture and CAD System for FPGAs. – ACM Transactions on Reconfigurable Technology and Systems, Vol. 7, No. 2, Article 6, Publication date: June 2014.
3. Verilog-to-Routing Documentation. Release 7.0.7. Aug 23, 2017.
4. **Строгонов А., Цыбин С., Городков П.** САПР VTR7 для проектирования академических ПЛИС // Компоненты и технологии. 2016. № 3.
5. **Строгонов А., Городков П.** Реализация Verilog-проектов в базисе академических ПЛИС с применением САПР VTR7.0 // Компоненты и технологии. 2017. № 5.
6. **Строгонов А., Городков П.** САПР VTR8 как инструмент исследования новых архитектур ПЛИС // Компоненты и технологии. 2017. № 10.
7. **Kim J.H., Anderson J. H.** Synthesizable Standard Cell FPGA Fabrics Targetableby the Verilog-to-Routing CAD Flow. ACM Trans. Reconfigurable Technol. Syst. 10, 2, Article 11 (April 2017), 23 pages. DOI: http://dx.doi.org/10.1145/3024063
8. **Haroldsen T. D.** Academic Packing for Commercial FPGA Architectures (2017). All Theses and Dissertations. 6526. https://scholarsarchive.byu.edu/etd/6526
9. **Hung E., Eslami F., Wilton S. J. E.** Escaping the Academic Sandbox: Realizing VPR Circuits on Xilinx Devices. Field-Programmable Custom Computing Machines (FCCM), 2013.
10. **Hung E.** Examining where academic FPGA tools lag behind industry. Field Programmable Logic and Applications (FPL). 2015.