

DOI: 10.15514/ISPRAS-2021-33(6)-8



Сравнение открытых маршрутов проектирования цифровой аппаратуры: qFlow, OpenLANE, Coriolis, SymbiFlow

^{1,2,3,4,5} А.С. Камкин, ORCID: 0000-0001-6374-8575 <kamkin@ispras.ru>

^{1,2} С.А. Смолов, ORCID: 0000-0003-0173-3081 <smolov@ispras.ru>

^{1,2} М.М. Чупилко, ORCID: 0000-0002-8772-5631 <chupilko@ispras.ru>

¹ Институт системного программирования РАН им. В.П. Иванникова,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25

² Российский экономический университет им. Г.В. Плеханова,
117997, Россия, г. Москва, Стремянный пер., д. 36

³ Московский государственный университет имени М.В. Ломоносова,
119991, Россия, г. Москва, Ленинские горы, д. 1

⁴ Московский физико-технический институт,
114701, Россия, Московская область, г. Долгопрудный, Институтский пер., 9

⁵ НИУ Высшая школа экономики,
101000, Россия, г. Москва, ул. Мясницкая, д. 20

Аннотация. В данной работе сделан обзор открытых инструментов логического синтеза, трассировки и размещения элементов моделей цифровой аппаратуры, анализа временных характеристик и синтеза топологических схем. Среди инструментов были выбраны системы проектирования qFlow, OpenLANE, Coriolis и SymbiFlow, поддерживающие полные маршруты: от RTL-модели до двоичных образов для ПЛИС или исходных данных для полупроводниковых фабрик. Для экспериментальной оценки инструментов была взята модель микропроцессора с архитектурой RISC-V под названием PicoRV32. Результаты испытаний показали, что открытые инструменты пригодны для создания топологических схем реалистичных примеров. Однако коммерческие инструменты позволяют создавать более эффективные с точки зрения производительности топологические модели.

Ключевые слова: цифровая аппаратура; микропроцессор; проектирование; открытое программное обеспечение; ПЛИС; СБИС; RISC-V; qFlow; OpenLANE; Coriolis; SymbiFlow

Для цитирования: Камкин А.С., Смолов С.А., Чупилко М.М. Сравнение открытых маршрутов проектирования цифровой аппаратуры: qFlow, OpenLANE, Coriolis, SymbiFlow. Труды ИСП РАН, том 33, вып. 6, 2021 г., стр. 111-130. DOI: 10.15514/ISPRAS-2021-33(6)-8

Comparison of Open Flows for Digital Hardware Development: qFlow, OpenLANE, Coriolis, and SymbiFlow

^{1,2,3,4,5} A.S. Kamkin, ORCID: 0000-0001-6374-8575 <kamkin@ispras.ru>

^{1,2} S.A. Smolov, ORCID: 0000-0003-0173-3081 <smolov@ispras.ru>

^{1,2} M.M. Chupilko, ORCID: 0000-0002-8772-5631 <chupilko@ispras.ru>

¹ *Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*

² *Plekhanov Russian University of Economics,
36, Stremyanny lane, Moscow, 117997, Russia*

³ *Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia*

⁴ *Moscow Institute of Physics and Technology,
9, Institutskiy per., Dolgoprudny, Moscow region, 114701, Russia*

⁵ *National research university Higher school of economics,
20, Myasnitskaya street, Moscow, 101000, Russia*

Abstract. This paper reviews open-source tools for the logical synthesis, place-and-route, static timing analysis and topology generation hardware design stages. The following tools have been described: qFlow, OpenLANE, Coriolis, and SymbiFlow. These tools are aimed to synthesize RTL models into FPGA bitstreams or GDS II physical layouts. A PicoRV32 implementation of RISC-V microprocessor has been used for experimental evaluation of these flows. The results show that open-source flows are capable to produce physical layouts for realistic examples. At the same time, commercial CADs allow generating more effective designs in terms of clock frequency.

Keywords: digital hardware, microprocessor; computer-aided design; open source; FPGA; ASIC; RISC-V; qFlow; OpenLANE; Coriolis; SymbiFlow

For citation: Kamkin A.S., Smolov S.A., Chupilko M.M. Comparison of Open Flows of Digital Hardware Development: qFlow, OpenLANE, Coriolis, and SymbiFlow. *Trudy ISP RAN/Proc. ISP RAS*, vol. 33, issue 6, 2021, pp. 111-130 (in Russian). DOI: 10.15514/ISPRAS-2021-33(6)-8

1. Введение

Процесс проектирования аппаратуры включает в себя следующие этапы: (1) логический синтез списка соединений технологических логических элементов из RTL-модели (logic synthesis), (2) размещение логических элементов на схеме и трассировка соединений между ними (place and route), (3) статический временной анализ (static timing analysis), (4) синтез физического топологического представления (physical layout generation).

Входной информацией для первого этапа является RTL-модель аппаратуры, написанная на языке Verilog или VHDL. Логический синтез начинается с синтаксического анализа RTL-модели и составления внутреннего представления, которое затем оптимизируется и преобразуется с помощью технологической библиотеки в список соединений (netlist) логических элементов (gates) или, иначе, ячеек (cells). На втором этапе элементы (ячейки) списка соединений отображаются на ресурсы ПЛИС или размещаются на будущей СБИС. Одновременно происходит трассировка соединений между элементами: сначала общая (global), затем – детальная (detail). На третьем этапе проводится проверка того, что схема после размещения и трассировки удовлетворяет заданным временным ограничениям. Если проблем с ограничениями не возникло, осуществляется генерация физического топологического представления аппаратуры.

В случае ПЛИС физическое представление выглядит как двоичный образ, описывающий конфигурации логических элементов ПЛИС (таблиц поиска, lookup tables) и соединения

между ними. В случае СБИС топологическое описание, представленное в виде плоских геометрических фигур в формате GDS II [1], должно пройти этап проверки физических производственных ограничений (design rule checking, DRC) [2]. DRC показывает технологическую корректность топологического описания и его пригодность к использованию на фабрике для производства СБИС.

При получении различных представлений аппаратуры возникает проблема проверки их эквивалентности. Для ее решения существует два подхода.

- 1) Логическая проверка эквивалентности (Logic equivalence checking, LEC) – используется для проверки отношения соответствия между двумя логическими моделями (например, RTL-моделью и соответствующим списком соединений) [2].
- 2) Топология-схема (Layout vs schematic, LVS) – используется для проверки того, что физическое топологическое представление (возможно, оптимизированное) эквивалентно исходному списку соединений. Здесь используется обратное построение списка соединений по топологическому представлению [2]. Проверка LVS имеет смысл только при производстве СБИС.

Коммерческие инструменты, с помощью которых в настоящее время доводится до производства основная масса электроники, поддерживают осуществление каждого из перечисленных этапов проектирования аппаратуры. Однако и среди программ с открытым исходным кодом существуют инструменты, решающие как отдельные из рассмотренных задач, так и объединяющие в себе целые маршруты проектирования. Далее в работе такие инструменты будут называться открытыми САПР.

Статья организована следующим образом: во второй главе рассматриваются открытые САПР как для случая СБИС, так и для ПЛИС; в третьей главе описаны открытые инструменты логического синтеза, трассировки и размещения элементов цифровой аппаратуры; в четвертой главе приводятся результаты испытаний САПР на открытой реализации микропроцессора, после чего следуют выводы.

2. Открытые САПР

Основной задачей САПР, рассматриваемых в статье, является преобразование RTL-модели в топологическое представление в формате GDS II или двоичный образ для ПЛИС. В статье рассматриваются следующие системы:

- 1) для проектирования СБИС: qFlow, OpenLANE, Alliance/Coriolis;
- 2) для получения двоичных образов для ПЛИС: SymbiFlow.

Основным языком разработки всех рассматриваемых САПР является C++. Сводная информация о рассматриваемых системах представлена в табл. 1.

Табл. 1. Открытые САПР
Table 1. Open-source CADs

Инструмент	Поддержка	Временные рамки разработки	Текущая версия	Лицензия
qFlow	Р.Т. Эдвардс	2013 – н. вр.	1.4.98	GPL
OpenLANE	Сообщество разработчиков	2020 – н. вр.	02.00.18	Apache 2.0
Alliance/Coriolis	Сообщество разработчиков	1990 – н. вр.	28.10.2019	GPL, LGPL, Apache 2.0
SymbiFlow	Сообщество разработчиков	2018 – н. вр.	–	ISC

2.1. qFlow

qFlow [3] – открытая САПР, предназначенная для проектирования СБИС по RTL-моделям, написанным на языке Verilog. qFlow включает в себя как сторонние открытые инструменты: Magic, graywolf, qrouter, Yosys, так и небольшие вспомогательные программы, которые запускают сторонние инструменты и конвертируют различные представления аппаратуры. Каждый из отдельных используемых инструментов распространяется под собственной лицензией, а для программного кода, соединяющего эти инструменты в единый маршрут, используется только лицензия GPL [4].

Рассмотрим инструменты, используемые qFlow, упорядочив их по этапам проектирования аппаратуры (иллюстрация схемы работы qFlow версии 1.1, за исключением этапа анализа временных характеристик, находится на рис. 1).

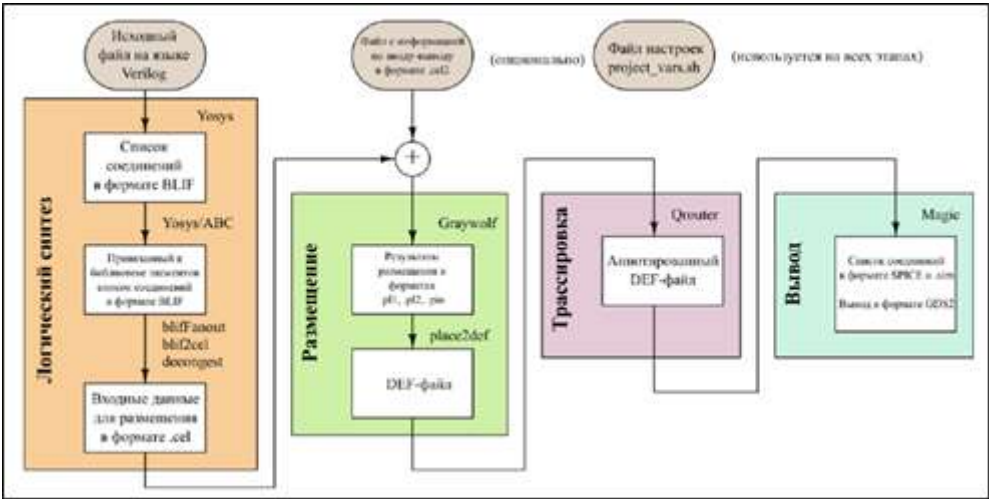


Рис. 1. Схема работы qFlow версии 1.1

Fig. 1. The qFlow 1.1 structure

- Логический синтез: Yosys [5] или Odin II [6] (оба используют инструмент оптимизации логических схем ABC [7]). Результатом работы является список соединений логических элементов, взятых из технологической библиотеки, в формате BLIF (формат записи логических схем, предложенный Калифорнийским университетом в Беркли (University of California, Berkeley) [8]). С помощью вспомогательного инструмента blif2cel список соединений транслируется в формат описания логических ячеек cel (входной формат для инструмента graywolf).
- Размещение логических элементов и трассировка соединений: graywolf [9] для размещения и общей трассировки, qrouter [10] для детальной трассировки. Для связи инструментов между собой результат работы graywolf транслируется вспомогательным инструментом place2def в формат DEF (Design Exchange Format) [11] – открытый формат представления модели аппаратуры на топологическом уровне, предложенный компанией Cadence. Инструмент qrouter является простым детальным трассировщиком соединений с ограниченными возможностями, не масштабируемым на сложные промышленные проекты. Результат его работы представляется также в формате DEF.
- Статический временной анализ: OpenSTA [12] или vesta [13]. Эти инструменты более подробно рассматриваются в разделе 3.4.
- Физический топологический синтез и DRC: Magic [14], использующий среду моделирования электротехнических характеристик SPICE [15]. Инструмент Magic более

подробно рассматривается в разделе 3.3.

- е. LEC/LVS: Yosys для LEC и Netgen [16] для LVS. Эти инструменты более подробно рассматриваются в разделах 3.1 и 3.3 соответственно.

Итак, qFlow поддерживает полный маршрут проектирования от Verilog до GDS II. САПР распространяется с библиотеками логических элементов ИТ/OSU 2.7, поддерживающими технологические нормы 180-500 нм, а также с библиотекой FreePDK45 с нормой 45 нм [17], не предназначенной для синтеза GDS II-файлов. qFlow предоставляет пользователю простой графический интерфейс, позволяющий настраивать около десяти параметров синтеза. Отметим, что архитектура qFlow является расширяемой и адаптируемой к новым инструментам и библиотекам элементов. В подтверждение возможностей данной САПР ее автор опубликовал результат синтеза RTL-модели микропроцессора Raven RISC-V [18].

2.2. OpenLANE

OpenLANE [19] – САПР, предназначенная для получения физического топологического представления моделей аппаратуры на основе RTL-моделей, написанных на языке Verilog. OpenLANE использует технологическую библиотеку SkyWater PDK [20], поддерживающую технологическую норму 130 нм. Архитектура OpenLANE близка к qFlow, а одним из основных разработчиков САПР является Р. Т. Эдвардс – создатель qFlow. Как и qFlow, OpenLANE использует различные сторонние инструменты, которые могут распространяться под своими собственными лицензиями.

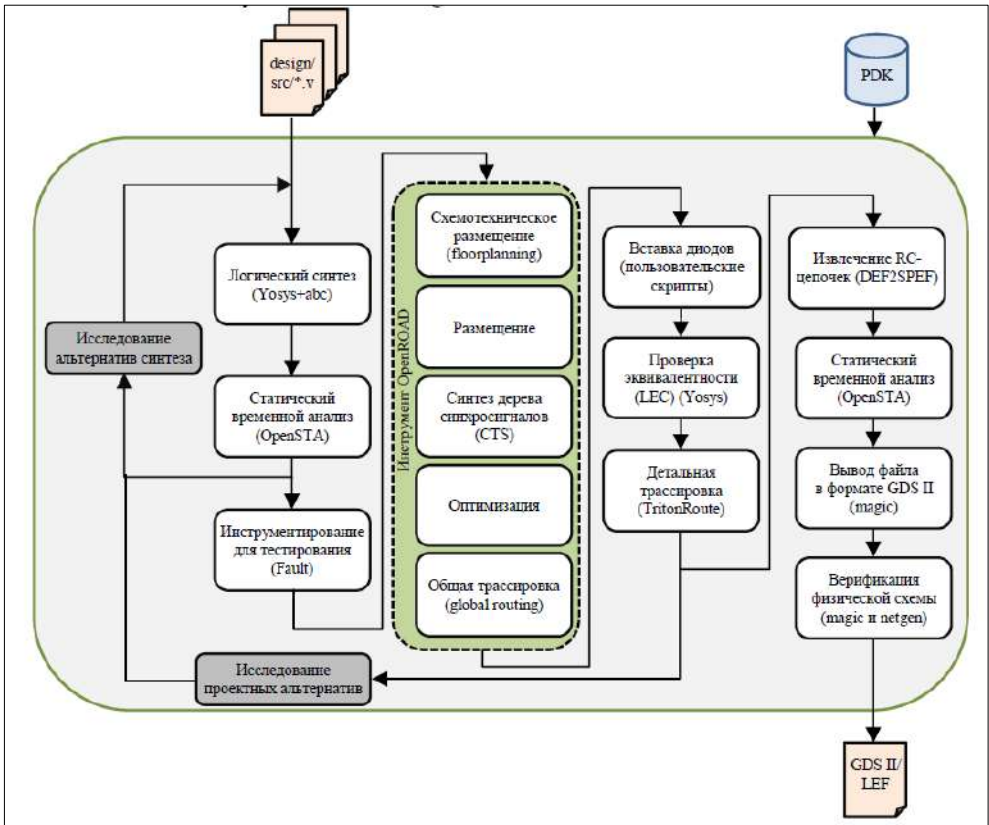


Рис. 2. Схема работы САПР OpenLANE

Fig. 2. The OpenLANE workflow

Рассмотрим основные инструменты, входящие в состав САПР и расположенные по этапам проектирования аппаратуры (см. рис. 2):

- a. логический синтез: Yosys и ABC.
- b. размещение логических элементов и трассировка соединений: OpenROAD [21], состоящий из нескольких инструментов, выполняющих задачи размещения и общей (глобальной) трассировки, и TritonRoute [22], осуществляющий детальную трассировку.
- c. статический временной анализ: OpenSTA.
- d. физический топологический синтез и DRC: Magic или KLayout [23];
- e. LEC/LVS: Yosys для LEC и Netgen для LVS.

OpenLANE и qFlow используют похожие наборы внешних инструментов. Основными отличиями являются использование в OpenLANE инструментов OpenROAD вместо graywolf и TritonRoute вместо qroute, а также использование библиотеки элементов SkyWater PDK вместо ИТ/OSU 2.7 и FreePDK45. Необходимо отметить, что OpenLANE использовалась для производства 40 систем-на-кристалле (СнК) в рамках проекта Open MPW Shuttle Program [24] производства на фабрике SkyWater любительских моделей аппаратуры, спонсируемого компанией Google. Среди выпущенных СнК присутствуют, например, Caravel_RISC-V_OSU и Softshell, реализующие системы-на-кристалле с ядрами архитектуры RISC-V.

В целом, САПР OpenLANE выглядит более предпочтительной для промышленного применения, чем qFlow, за счет относительно современных технологических норм библиотеки элементов (130 нм) и более современных специализированных инструментов для этапов размещения и трассировки. Обе указанных САПР используют инструменты Yosys и ABC на этапе логического синтеза.

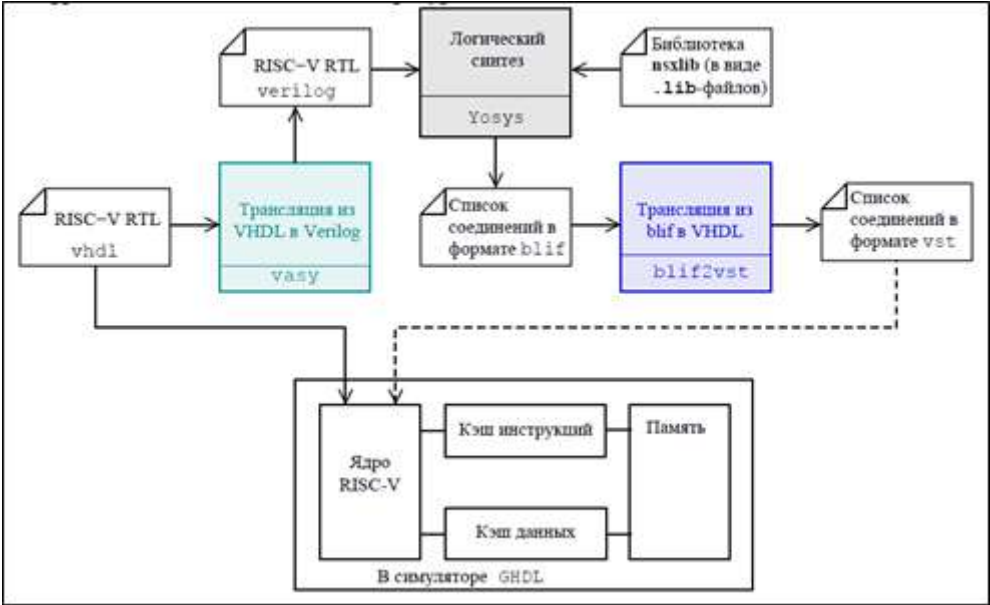


Рис. 3. Логический синтез в САПР Alliance/Coriolis
Fig. 3. Logic Synthesis in Alliance/Coriolis

2.3. Alliance/Coriolis

Alliance/Coriolis [25] – открытая САПР разработки СБИС. Поддерживает полный маршрут получения GDS II-файлов из RTL-моделей, написанных на языке VHDL. Со сменой

собственного логического анализатора Alliance на Yosys САПР стала поддерживать маршрут получения GDS II-файлов из RTL-моделей, написанных на языке Verilog. Отличительной особенностью данной САПР является поддержка разработки как цифровой, так и аналого-цифровой и чисто аналоговой аппаратуры.

Необходимо также заметить, что Alliance/Coriolis поддерживает концепцию OpenAccess [37], предложенную Cadence, а именно позволяет импортировать информацию для размещения и трассировки из LEF- (Library Exchange Format, формат библиотечного обмена) и DEF-файлов (Design Exchange Format, формат обмена моделями) [11].

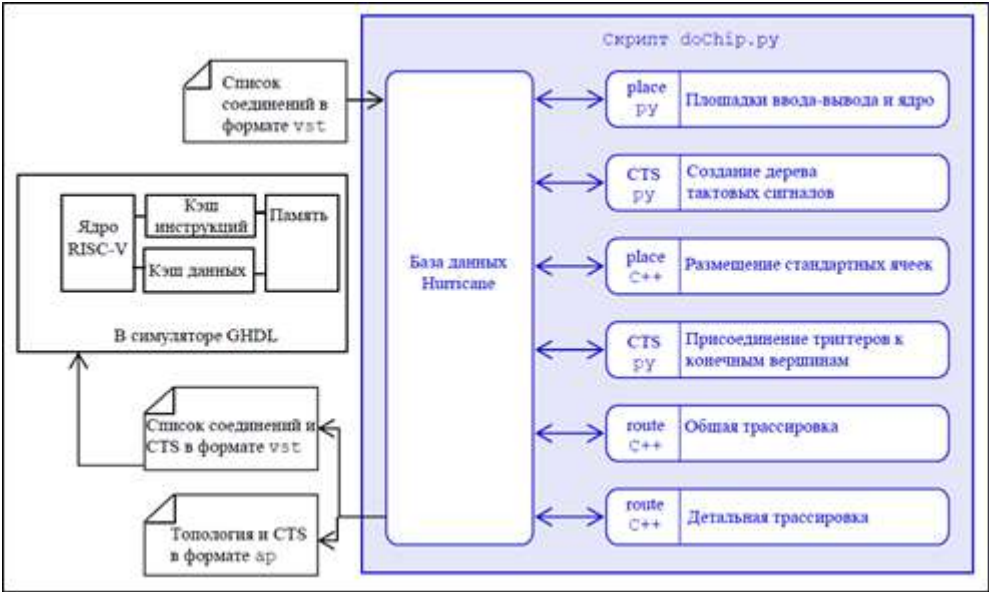


Рис. 4. Размещение и трассировка в САПР Alliance/Coriolis
Fig. 4. Place and route in Alliance/Coriolis



Рис. 5. Этап валидации и получения GDS II-файла в САПР Alliance/Coriolis
Fig. 5. Stages of validation and GDS II generation in Alliance/Coriolis

САПР Alliance/Coriolis применялась в процессе производства ряда сложных моделей аппаратуры, например, для исследовательского суперскалярного микропроцессора StaCS,

включавшего 875 000 транзисторов, и 1-Гигабитного маршрутизатора IEEE Gigabit HSL Router, включавшего 400 000 транзисторов. Однако разработчики САПР отмечают, что Alliance/Coriolis не пригодна для техпроцесса с нормой менее 130-180 нм.

Подводя итог рассмотрения САПР Alliance/Coriolis, подчеркнем, что на протяжении долгого времени (до появления qFlow в начале 2010-х) она являлась единственной открытой системой, реализующей весь маршрут проектирования цифровой аппаратуры. Также отметим, что инструменты (за исключением Yosys), входящие в состав Alliance/Coriolis, являются специфичными для данной среды.

2.4. VTR и SymbiFlow

VTR (Verilog-to-Routing) 8.0.0 [38,39] является открытой САПР, распространяемой по лицензии MIT [40] и предназначенной для подготовки RTL-моделей аппаратуры к размещению на ПЛИС. Ядро VTR – инструмент VPR (Versatile Place and Route) [41] – создается с середины 1990-х годов, при этом сам VTR – с 2010-х годов; последняя версия была выпущена 24.03.2020. Для представления результата работы VTR использует формат FASM (FPGA Assembly) [42], который пригоден для синтеза двоичных образов, загружаемых на ПЛИС, с помощью сторонних инструментов. В качестве такого инструмента создателями VTR предлагается использовать открытую САПР SymbiFlow (см. ниже).

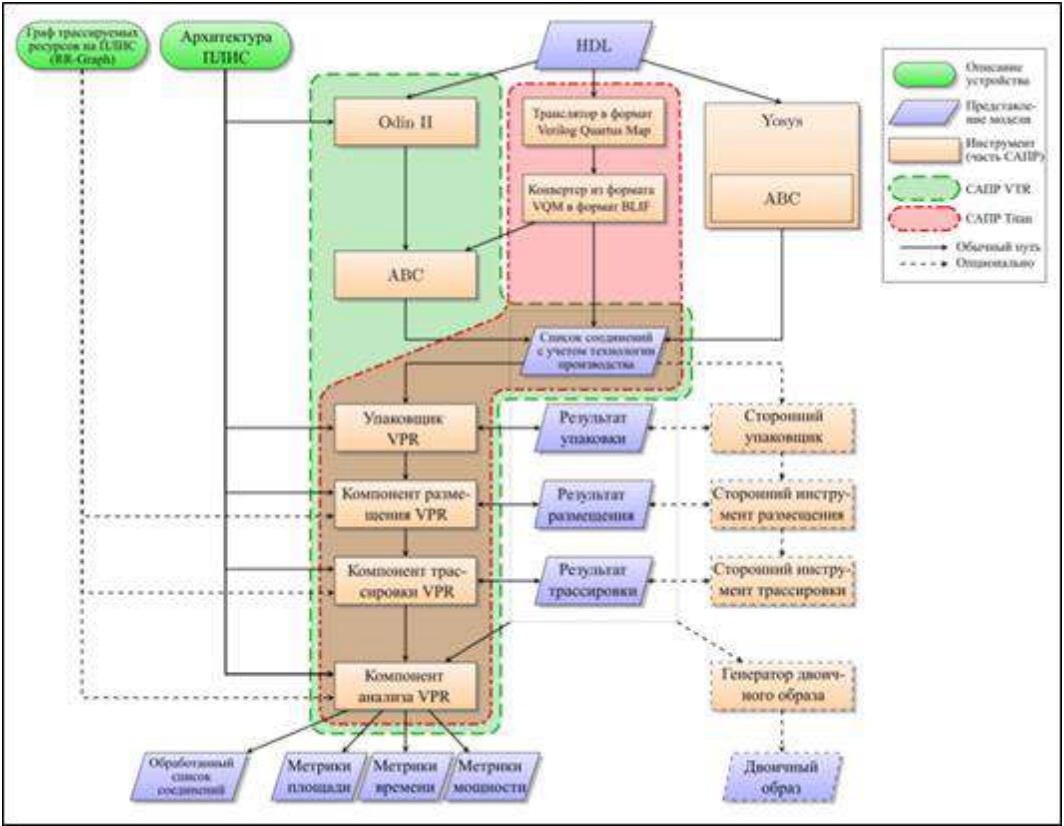


Рис. 6. Схема работы САПР VTR
Fig.6. The VTR workflow

Рассмотрим этапы генерации двоичного образа для ПЛИС с помощью VTR (см. рис. 6, выделение зеленым цветом).

1. Логический синтез: Odin II или Yosys (оба используют ABC). Результат этого этапа представлен в формате BLIF.
2. Размещение и трассировка: VPR. Результат представлен в формате FASM и включает информацию об использовании ресурсов целевой ПЛИС, энергопотреблении, а также временные характеристики.
3. LEC: Yosys.

Некоторые компоненты САПР VTR находят применение и в коммерческих САПР, примером которых может служить Titan [43] (см. рис. 6, выделение красным цветом).

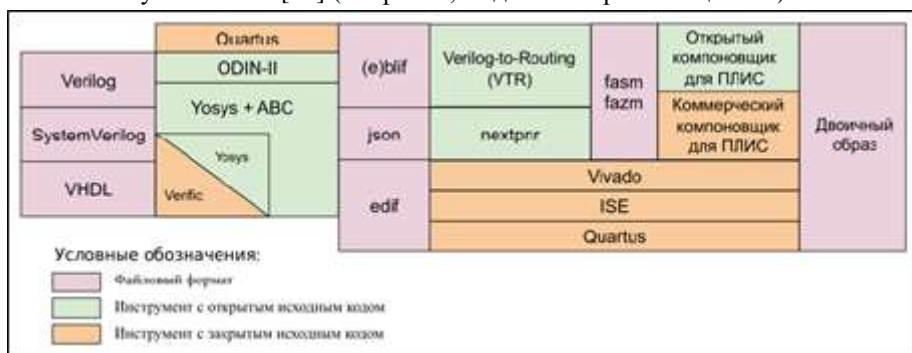


Рис. 7. Схема работы САПР SymbiFlow

Fig. 7. The SymbiFlow workflow

САПР SymbiFlow [44] предназначена для получения двоичных образов, пригодных для загрузки на ПЛИС. Рассмотрим основные этапы работы САПР SymbiFlow для получения двоичных образов (см. рис. 7).

- 1) Логический синтез: Odin II или Yosys для моделей на языке Verilog; коммерческий инструмент Verific [45] для моделей на языке VHDL.
- 2) Размещение и трассировка: VTR или nextpnr [46] (оба инструмента создают представление аппаратуры в формате FASM, которое трансформируется в двоичный образ открытым инструментом OpenFPGA [47] или коммерческими инструментами, такими как Vivado [48] и ISE [49] от компании Xilinx, Quartus [50] от компании Intel).

Помимо открытого инструмента OpenFPGA сообщество разработчиков SymbiFlow поддерживает ряд распространяемых по лицензии ISC генераторов двоичных образов с открытым исходным кодом для отдельных серий ПЛИС: Project IceStorm [51] (для ПЛИС Lattice iCE40), Project Trellis [52] (для ПЛИС Lattice ECP5), Project Oxide [53] (для ПЛИС Lattice Nexus), Project Apicula [54] (для ПЛИС Gowin LittleBee), а также Project X-Ray [55] (для ПЛИС Xilinx 7-й серии).

Резюмируя, отметим, что, хотя использование VTR и SymbiFlow позволяет получить двоичный образ, который можно загрузить в некоторые современные ПЛИС, эти САПР едва ли применимы для сложных проектов. Так, в недавней работе [56], посвященной сравнению работы SymbiFlow и коммерческих САПР, был сделан вывод о необходимости доработки алгоритмов размещения и трассировки, используемых в SymbiFlow. В той статье на примере ПЛИС Stratix IV от компании Intel (500 000 LUT-таблиц) было показано, что результаты работы SymbiFlow и САПР Quartus от Intel в целом сравнимы. Однако для ПЛИС Artix-7 от компании Xilinx (50 000 LUT-таблиц), было показано двукратное повышение использование ресурсов ПЛИС по сравнению с САПР Vivado от Xilinx.

3. Инструменты проектирования аппаратуры

Данный раздел включает в себя обзор отдельных инструментов, которые могут быть использованы в открытых маршрутах проектирования. Инструменты сгруппированы по следующим задачам: логический синтез, размещение и трассировка, синтез физического топологического представления и его проверка, статический временной анализ. Отдельно рассматриваются существующие открытые библиотеки элементов.

3.1. Инструменты логического синтеза

В данном подразделе рассматриваются инструменты Yosys, Odin II и GHDL (см. табл. 2). Yosys 0.10 (Yosys Open SYntethis Suite) от 27.09.2021 – открытый инструмент логического синтеза RTL-моделей, написанных на языке Verilog (поддерживается синтезируемое подмножество языка, описанного в стандарте IEEE 1364-2005), в модели в формате BLIF. Для проведения оптимизаций используется инструмент ABC. Инструмент поддержкой большого количества инструкций Verilog, изначально создан К. Вульфom (C. Wolf). Odin II (выпущен как часть VTR 8.0.0 от 24.03.2020) – открытый инструмент логического синтеза RTL-моделей, написанных на языке Verilog. Был создан в Университете Нью-Брансуика (University of New Brunswick). Как и в Yosys, в Odin II используется инструмент ABC. Odin II не ориентирован на поддержку RTL-моделей большого размера, поддерживает не все синтезируемые конструкции стандарта IEEE 1364-2005 (например, repeat, deassign, edge, forever, disable), почти не поддерживает конструкции моделирования Verilog на уровне вентилей (за исключением buf) [57]. GHDL 1.0.0 от 03.02.2021 [58] – открытый инструмент логического анализа и моделирования RTL-моделей, написанных на языке VHDL (полностью поддерживается стандарт IEEE 1076-2002, а IEEE 1076-2008 – только частично). Поддержка инструмента осуществляется сообществом разработчиков. Проект был создан Т. Гингольдом (T. Gingold) в середине 2000-х годов. Реализация логического синтеза в данном инструменте находится в экспериментальной стадии и для использования в открытых маршрутах требует доработки. Заметим, что ни один из открытых инструментов логического синтеза не поддерживает язык SystemVerilog, несмотря на активное его использование в промышленной разработке в целях проектирования и верификации.

Табл. 2. Инструменты логического синтеза
Table 2. Tools for logic synthesis

Инструмент	Входная нотация	Язык разработки	Поддержка	Временные рамки разработки	Лицензия
Yosys	Verilog	C++	Сообщество разработчиков	Начало 2010-х – н. вр.	ISC
Odin II	Verilog	C++	Университет Нью-Брансуика	Середина 2000-х – н. вр.	MIT
GHDL	VHDL	Ada	Сообщество разработчиков	Середина 2000-х – н. вр.	GPL

3.2 Инструменты размещения и трассировки

В данном подразделе рассматриваются следующие инструменты (см. табл. 3): graywolf, OpenROAD, FastRoute, BoxRouter, qrouter, TritonRoute и VPR. graywolf 0.1.6 от 12.08.2018 – открытый инструмент размещения и общей (глобальной) трассировки, предназначенный для получения логической схемы в формате списка

соединений на основе модели в формате BLIF. Данный инструмент является ответвлением проекта TimberWolf 6.3.5 [59], разработанного в Йельском университете (Yale University). OpenROAD 2.0 от 17.07.2021 – зонтичный проект для инструментов размещения логических элементов, общей трассировки, создания дерева тактовых сигналов, статического временного анализа и других задач. Поддерживаются форматы LEF, DEF, Verilog. Если считать TritonRoute, описанный ниже, частью данного проекта, то в таком случае OpenROAD поддерживает все преобразования от размещения элементов до детальной трассировки.

Табл. 3. Инструменты размещения и трассировки

Table 3. Tools for place-and-route

Инструмент	Входная нотация	Язык разработки	Поддержка	Временные рамки разработки	Лицензия
graywolf	BLIF	Си	Сообщество разработчиков	1980-е гг. – н. вр.	GPL
OpenROAD	LEF/DEF/Verilog	C++	Сообщество разработчиков	2018 г. – н. вр.	BSD
FastRoute	Текстовый файл	Си	–	2006 г. – 2012 г.	BSD
BoxRouter	Текстовый файл	C++	–	2006 г. – 2007 г.	BSD
qrouter	LEF/DEF	C++	Р.Т. Эдвардс	2011 г. – н. вр.	GPL
TritonRoute	LEF/DEF	C++	Сообщество разработчиков	2018 г. – н. вр.	BSD
VPR	BLIF	C++	Сообщество разработчиков	1997 г. – н. вр.	MIT

FastRoute 4.1 – инструмент общей трассировки с открытой лицензией [60,61]. Использует несколько алгоритмов трассировки, основным из которых является алгоритм волновой трассировки Ли [64]. FastRoute изначально создавался для соревнований между инструментами трассировки, поэтому входные и выходные форматы не являются стандартными. Исходный код инструмента выдается по запросу.

BoxRouter 2.0 – инструмент общей трассировки с открытой лицензией [62,63]. BoxRouter решает задачу трассировки с помощью целочисленного линейного программирования. Инструмент, как и FastRoute, создавался для соревнований, поэтому его входные и выходные форматы не стандартизованы. Исходный код инструмента выдается по запросу.

qrouter 1.4.85 от 22.10.2021 – вспомогательный инструмент для детальной трассировки, разработанный Р. Т. Эдвардсом. Основан на алгоритме волновой трассировки Ли. Работает с файлами в формате LEF и DEF. Результат сохраняется в формате DEF.

TritonRoute 0.0.6.0 от 16.07.2019 – открытый инструмент детальной трассировки, разработанный выпускниками Калифорнийского университета в Сан-Диего (University of California, San Diego). Работает с файлами в формате LEF и DEF. Данный инструмент является основным для детальной трассировки в OpenROAD и, соответственно, в OpenLANE. Основан на представлении задачи трассировки в виде графов, на которых осуществляется минимизация функции стоимости размещения линий [65].

VPR – открытый инструмент размещения и трассировки моделей аппаратуры, разработанный в 1997 г. в Университете Торонто (University of Toronto). Инструмент ориентирован на ПЛИС. Последняя версия была выпущена в рамках САПР VTR 24.03.2020.

3.3. Инструменты синтеза GDS II-файлов и их проверки

В данном подразделе рассматриваются инструменты Magic, KLayout и Netgen (см. табл. 4).

Magic 8.3.220 от 12.10.2021 – открытый инструмент синтеза многослойных топологических схем в форматах GDS II или CIF на основе логических схем, представленных в форматах LEF/DEF. Изначально Magic разрабатывался в Калифорнийском университете в Беркли, а основным текущим разработчиком является Р.Т. Эдвардс. Ограничением инструмента является поддержка только таких проектов, которые содержат горизонтальные и вертикальные линии (так называемый Манхэттенский стиль), однако в некоторых версиях инструмента добавлялась ограниченная поддержка границ другой формы. Инструмент позволяет проводить проверку DRC.

KLayout 0.27.4 от 26.09.2021 – открытый инструмент анализа топологических схем в форматах GDS II или OASIS [66] с возможностью синтеза файлов в формате GDS II. Поддерживает импорт из файлов в формате LEF. Инструмент поддерживает проведение проверок DRC и LVS. Поддерживается только Манхэттенский стиль соединительных линий.

Netgen 1.5.206 от 24.10.2021 – открытый инструмент проведения LVS, в настоящее время используется только для проверки GDS II-файлов, синтезируемых Magic. Изначально разрабатывался М. Сивилотти (М. Sivilotti), текущим разработчиком является Р. Т. Эдвардс.

Табл. 4. Инструменты синтеза GDS II-файлов
Table 4. Tools for GDS II synthesis

Инструмент	Входная нотация	Язык разработки	Поддержка	Временные рамки разработки	Лицензия
Magic	LEF/DEF	Си	Р. Т. Эдвардс	1980-е гг. – н. вр.	Свободная
KLayout	GDS II/LEF	C++	Сообщество разработчиков	2006 г. – н. вр.	GPL
Netgen	GDS II	Си	Р. Т. Эдвардс	2002 г. – н. вр.	GPL

3.4. Инструменты анализа временных характеристик

В данном подразделе рассматриваются инструменты OpenSTA и vesta (см. табл. 5).

OpenSTA 2.2.0 от 14.09.2020 – открытый инструмент, предназначенный для статической проверки временных характеристик на уровне логических вентилях. Основной разработчик – Дж. Черри (J. Cherry). Помимо открытой лицензии может распространяться по коммерческой лицензии от имени компании Parallax Software, Inc., если использующий OpenSTA проект не является проектом с открытым исходным кодом. Входными данными для инструмента являются список соединений, написанный на языке Verilog, временные ограничения в формате SDC (Synopsys Design Constraints) [67], технологическая библиотека в формате Liberty [68]. OpenSTA анализирует задержки различных путей исполнения, при этом находя недостижимые пути.

vesta (qFlow 1.4.98 от 24.09.2021) – открытый инструмент статической проверки временных характеристик, разработанный Р. Т. Эдвардсом как вспомогательный для проекта qFlow. На вход vesta принимает список соединений, написанный на Verilog, и технологическую библиотеку в формате Liberty, а результатом работы является оценка максимальной задержки сигналов в анализируемой схеме и максимальной возможной тактовой частоты.

Табл. 5. Инструменты анализа временных характеристик
Table 5. Tools for static timing analysis

Инструмент	Входная нотация	Язык разработки	Поддержка	Временные рамки разработки	Лицензия
OpenSTA	Verilog	C++	Parallax Software Inc.	2019 г. – н. вр.	GPL

vesta	Verilog	Си	Р. Т. Эдвардс	2013 г. – н. вр.	GPL
-------	---------	----	---------------	------------------	-----

3.5. Библиотека элементов

Библиотеки элементов (Process Design Kit, PDK) включают различную информацию, необходимую для производства СБИС на контрактных фабриках: примитивы логических элементов (символьное представление, параметры), данные для проведения различных проверок (DRC и LVS), технологические данные (допустимые слои, правила их изображения и другое), модели для проведения физического моделирования (SPICE), файлы с дополнительными ограничениями (например, в формате LEF). Полностью укомплектованная PDK должна включать в себя также и GDS II-представления логических элементов.

В данном подразделе рассматриваются следующие библиотеки (см. табл. 6) для технологических норм от 7 до 500 нм: семейство sxlib, ИТ/OSU, FreePDK45 (и ее расширение 45nm FreePDK), FreePDK15 (и ее расширение Open-Cell 15nm), SkyWater PDK, ASAP PDK.

Семейство sxlib/ssxlib/vxlib/vsclib/vgalib/rgalib/nsxlib [69] – открытые библиотеки элементов, предназначенные для использования в САПР Alliance/Coriolis и созданные в инструменте Graal в рамках работ над указанной САПР. Библиотеки описывают топологию в символическом виде, однако утверждается, что они могут быть отображены на реальный технологический процесс 130 нм, хотя изначально предназначались для 350 нм. Наиболее современной библиотекой, имеющей отношение к данному семейству, является библиотека nsxlib [28], созданная на основе sxlib, проверенная ее создателем на соответствие DRC для FreePDK 45nm, то есть технологическим нормам производства 45 нм.

ИТ/OSU 2.7 от 25.06.2007 – набор библиотек, в состав которого входят библиотеки TSMC 180 нм и 250 нм (теоретически подходят для фабрики TSMC, Тайвань), AMI 350 нм и 500 нм. Разрабатывался под руководством проф. Дж. Стин (J. E. Stine, Jr.) сначала в Технологическом институте Иллинойса (Illinois Institute of Technology), а затем в университете штата Оклахома (Oklahoma State University). Библиотеки распространяются в проприетарном формате компании Cadence, однако также включают GDS II-представления логических элементов, а также Verilog- и VHDL-описания элементов для целей симуляции, и используются в qFlow.

FreePDK45 1.4 от 07.04.2011 – библиотека элементов с технологической нормой 45 нм, созданная университетом штата Оклахома и университетом штата Северная Каролина (North Carolina State University, NCSU). Отметим, что FreePDK45, разработанная OSU в 2007-2008 гг., распространялась без лицензии и встроена в САПР qFlow 1.3. Та же самая FreePDK45 параллельно разрабатывалась в NCSU с 2007 г. [70], ее версии 1.0 и 1.1 распространялись по лицензии GPL. FreePDK45 (NCSU) версии 1.2, выпущенная 10.03.2018, уже включала в себя FreePDK45 от OSU. Соответственно, все последующие версии библиотеки выпускались только NCSU и являются частично коммерческими (лицензия Mentor SVRF на использование технологии верификации от компании Mentor), хотя основная часть библиотеки распространяется по лицензии Apache 2.0 (но только в исследовательских целях). Начиная с версии 1.2, появилась расширенная версия библиотеки, называемая 45nm FreePDK [71] и подходящая для организации производства: в ее создании приняли участие коммерческие компании Nangate (в 2019 г. купленная компанией Silvaco) и Semiconductor Research Corporation, некоммерческая организация (альянс) Silicon Integration Initiative (Si2) [72] и фонд поддержки исследований National Science Foundation (NSF) [73]. Библиотека выдается по запросу. Отметим, что в работе [74] говорится о непригодности библиотеки для производства из-за несогласованности значений в библиотечных Liberty и SPICE-файлах.

FreePDK15 1.2 от 09.06.2017 – открытая библиотека элементов, разработанная в NCSU [75]. Является продолжением работ над FreePDK45. Технологический процесс –15 нм.

Распространяется по лицензии BSD. FreePDK15 не включает всю необходимую информацию, необходимую для производственных целей, но существует ее версия, дополненная компанией Silvaco и называемая Open-Cell 15nm [72,76], которая вполне подходит для применения в производстве. Open-Cell лицензируется по Apache 2.0 только для академических целей. Доступ к FreePDK15 и Open-Cell 15nm осуществляется по запросу.

SKY130 PDK 0.0.0 от 08.05.2020 – открытая библиотека элементов, разработанная компанией SkyWater Technology для производства на собственной фабрике при спонсорстве компании Google. Технологический процесс –130 нм. Библиотека поддерживает несколько стратегий размещения элементов на кристалле в зависимости от плотности упаковки, рабочего напряжения и других параметров. Распространяется по лицензии Apache 2.0. Библиотека используется в САПР OpenLANE и проекте MPW производства СБИС, спонсируемом компанией Google.

ASAP 7nm Predictive PDK 1.7 от 02.2021 [77] – открытая библиотека элементов, разработанная в университете штата Аризона (Arizona State University) при участии компании ARM. Технологический процесс – 7 нм, однако библиотека предназначена только для исследовательских целей, но не для производства. Распространяется по некоммерческой лицензии, однако доступна в инструменте OpenROAD (и, соответственно, в САПР OpenLANE по лицензии BSD-3).

Табл. 6. Открытые библиотеки элементов
Table 6. Open technology libraries

Название библиотеки и норма технологического процесса, нм	Формат описания ячеек (без GDS библиотека не предназначена для производства)	Поддержка	Временные рамки разработки	Лицензия
Семейство sxlib; 130	Внутренний формат Alliance; формат Synopsys, Verilog	Университет Пьера и Марии Кюри	2002 г. – н. вр.	GPL
ИТ/OSU; 180, 250, 350, 500	Форматы Magic и Cadence, GDS, Verilog, VHDL, LEF, Liberty, SPICE	Университет штата Оклахома	1999 г. – 2007 г.	GPL для 1.0, далее – некоммерческая
FreePDK45 / 45nm FreePDK; 45	Форматы Cadence и Synopsys, GDS, Verilog, LEF, Liberty, SPICE	Университет штата Северная Каролина / Silvaco	2007 г. – 2011 г.	GPL до 1.2, далее – Apache 2.0 и SVRF (некоммерческая)
FreePDK15 / Open-Cell 15nm; 15	Форматы Cadence и Synopsys, GDS, Verilog, LEF, Liberty, SPICE	Университет штата Северная Каролина / Silvaco	2014 г. – н. вр.	BSD / Apache 2.0 (некоммерческая)
SkyWater PDK 130nm; 130	Форматы Magic, GDS, Verilog, LEF, Liberty, SPICE	SkyWater Technology	2020 г. – н. вр.	Apache 2.0
ASAP 7nm; 7	Форматы Cadence, Liberty, SPICE	Университет штата Аризона	2016 г. – н. вр.	Некоммерческая (BSD для OpenLANE)

4. Испытания маршрутов проектирования

Для проведения испытаний открытых маршрутов проектирования в рамках данной работы был взят небольшой микропроцессор с открытой архитектурой RISC-V [78] PicoRV32 [79], с объемом кода около 5400 строк, реализующий подмножество инструкций RV32IMC.

В рамках эксперимента мы применили qFlow, OpenLANE, Coriolis для получения топологии в формате GDS II, а также SymbiFlow и коммерческий инструмент Vivado для получения двоичных образов для ПЛИС. Отметим, что САПР VTR не позволила получить FASM-файл для Verilog-модели PicoRV32 из-за ограничений используемого им логического анализатора Odin II. Эксперименты со всеми САПР, кроме Coriolis, проводились на компьютере с центральным процессором Intel Core i7-8700 3.2 ГГц, 32 Гб оперативной памяти и ОС Ubuntu 20.04. Для Coriolis использовалась виртуальная машина с ОС Ubuntu 18.04 и 8 Гб памяти, так как данная САПР не поддерживает работу с ОС Ubuntu 20.04.

Для проведения экспериментов для случая СБИС с получением GDS II-файлов были выбраны следующие библиотеки: OSU350 (350 нм) в qFlow, SKY130 (версия fd_sc_hd, 130 нм) в OpenLANE, символическая smos45 и реальная nsxlib (130 нм) в Coriolis. Во всех трех САПР использовался Yosys 0.9 для логического синтеза, который, в свою очередь, использует инструмент ABC для привязки к технологическому базису. В табл. 7 приведено сравнение данных для трех открытых САПР и коммерческой САПР [80], построенной на основе среды VSDFLOW [81]. Сравнение осуществлялось по количеству логических элементов схемы, полученному как до привязки к логическим элементам, находившимся в соответствующей библиотеке, так и после нее, по площади на кристалле, по оценке максимальной тактовой частоты. При этом OpenLANE по умолчанию применял опцию flatter (подстановка объектов иерархической структуры для получения «плоской» модели) в Yosys для упрощения RTL-модели до логического синтеза, а qFlow и Coriolis – после синтеза. Для возможности сравнения результатов опции в qFlow и Coriolis были изменены аналогично OpenLANE.

Табл. 7. Результаты экспериментов с PicoRV32 для СБИС
Table 7. Results of Experiments with PicoRV32 targeted at ASICs

Параметры	САПР			
	qFlow	OpenLANE	Coriolis	VSDFLOW
Технологическая норма, нм	350	130	130	180
Количество логических элементов до работы ABC	17 040	16 841	16 740	н/д
Количество логических элементов (после отображения на базис с помощью ABC)	25 184	14 840	13 839	13 826
Площадь, мкм ²	12 725 184	442 309	1 803 400	н/д
Оценка максимальной тактовой частоты, МГц	87	24	19	378

Данные, представленные в табл. 7, показывают, что инструменты OpenLANE, Coriolis и VSDFLOW синтезировали схемы с примерно одинаковым количеством логических элементов (после отображения на логический базис с помощью ABC значения для двух последних инструментов отличаются менее чем на 1%, OpenLANE потребовалось на 7% больше элементов, чем Coriolis). Схема, синтезированная инструментом qFlow, включает существенно больше элементов (на 70%), чем в случае OpenLANE. Возможно, это связано с тем, что OpenLANE использует при синтезе большее количество типов логических ячеек, чем qFlow, и, следовательно, потенциально способна проводить более эффективные

оптимизации. Подтверждением этой гипотезы могут служить данные, представленные в табл. 8-9. Так, на примере логической операции AND можно заметить, что инструмент qFlow для ее реализации использует один тип ячеек (AND2X2), а OpenLANE – 5 (and2_2, and3_2, and3b_2, and4_2, and4b_2).

Одновременно, оценочные значения тактовой частоты для OpenLANE и Coriolis существенно ниже, чем для qFlow, что может быть связано с излишним пессимизмом инструментов оценки в первом случае и излишним оптимизмом во втором случае. Оценка частоты проводилась на основе самого длительного времени прохождения данных в модели, подсчитанного инструментами vesta (в qFlow), OpenSTA (в OpenLANE), hitas (в Coriolis). Заметим, что оценочное значение тактовой частоты в коммерческом САПР на порядок выше, чем в открытых инструментах.

Табл. 8. Количество логических ячеек при работе qFlow по типам
Table 8. The number of logical cells in qFlow's result, by types

Тип ячеек				Количество ячеек			
AND2X2	808	DFFPOSX1	1 613	NAND3X1	3 009	OAI22X1	263
AOI21X1	2 703	INVX1	2 724	NOR2X1	2 183	OR2X2	330
AOI22X1	444	MUX2X1	303	NOR3X1	281	XNOR2X1	335
BUFX2	339	NAND2X1	3 106	OAI21X1	6 589	XOR2X1	154

Табл. 9. Количество логических ячеек при работе OpenLANE по типам
Table 9. The number of logical cells in OpenLANE's result, by types

Тип ячеек						Количество ячеек					
a211o_2	34	a311o_2	2	buf_1	1 653	nor2b_2	1	o221ai_2	11	or2_2	1 091
a211oi_2	59	a31o_2	39	buf_2	8	nor3_2	41	o22a_2	1 314	or2b_2	24
a21bo_2	150	a31oi_2	4	conb_1	42	o2111a_2	1	o22ai_2	52	or3_2	64
a21boi_2	10	a32o_2	39	dfxtp_2	1 613	o211a_2	71	o2bb2a_2	84	or3b_2	5
a21o_2	60	a41o_2	1	inv_2	1 577	o211ai_2	6	o2bb2ai_2	6	or4_2	98
a21oi_2	251	and2_2	158	mux2_1	1 222	o21a_2	53	o311a_2	6	or4b_2	6
a221o_2	87	and3_2	57	mux2_2	2	o21ai_2	139	o31a_2	18	or4bb_2	2
a22o_2	990	and3b_2	1	mux4_1	221	o21ba_2	208	o31ai_2	1		
a2bb2o_2	1 895	and4_2	346	nand2_2	77	o21bai_2	2	o32a_2	117		
a2bb2oi_2	80	and4b_2	2	nor2_2	532	o221a_2	203	o41a_2	4		

Для проведения экспериментов для случая ПЛИС с получением двоичных образов использовались открытая САПР SymbiFlow и коммерческая САПР Vivado 2020.2 от компании Xilinx. В качестве целевой ПЛИС была выбрана модель Xilinx Arty A7-100T.

Табл. 10. Результаты экспериментов с PicoRV32 для ПЛИС
Table 10. Results of Experiments with PicoRV32 targeted at FPGAs

Параметры	САПР	
	SymbiFlow	Vivado
Использование логических ячеек	4 192 из 63 400 (7%)	2 626 из 63 400 (4%)
Использование блочной памяти, число битов	73 728 из 4 239 360 (2%)	36 864 из 4 239 360 (<1%)
Оценка максимальной тактовой частоты, МГц	62.5	83.3

Результаты экспериментов представлены в табл. 10.

Двоичные образы, синтезированные обеими САПР, используют сопоставимые количества логических ячеек ПЛИС. Максимальная тактовая частота схемы, синтезированной открытой САПР, на 25% ниже, чем у схемы, синтезированной с помощью коммерческой САПР. Что касается использования памяти, то отметим, что на данной ПЛИС она выделяется блоками по 36 Кбит. Это означает, что схема, синтезированная для примера PicoRV32, использовала один (Vivado) или два (SymbiFlow) блока памяти. В целом можно говорить о несколько более высокой эффективности алгоритмов синтеза, реализованных в коммерческой САПР.

5. Выводы

В настоящее время тренд на развитие и использование открытого ПО коснулся также и области проектирования цифровой аппаратуры. В данной статье рассмотрены открытые САПР для проектирования цифровой аппаратуры (qFlow, OpenLANE, Coriolis, VTR, SymbiFlow), а также ряд отдельных инструментов и библиотек, используемых в данных маршрутах. Можно сказать, что открытые САПР способны покрыть весь маршрут проектирования аппаратуры для ПЛИС и его значительную часть – для СБИС. К недостаткам открытых инструментов можно отнести устаревшие технологические нормы используемых библиотек логических элементов, пригодных для создания GDS II-файлов. Также в инструментах отсутствует полноценная поддержка современного языка проектирования и верификации моделей аппаратуры SystemVerilog.

Было проведено экспериментальное сравнение результатов работы данных САПР с коммерческими аналогами на примере небольшого RISC-V ядра PicoRV32. Результаты экспериментов показывают, что среди открытых САПР для СБИС OpenLANE создает топологические представления с наименьшей площадью, а qFlow – с наибольшей максимальной тактовой частотой. По использованию логических элементов (в случае проектирования СБИС) инструменты OpenLANE и Coriolis продемонстрировали результаты, сопоставимые с коммерческой САПР VSDFLOW, а по использованию логических ячеек (в случае проектирования для ПЛИС) открытая САПР SymbiFlow незначительно уступила коммерческому инструменту Vivado. Слабым местом открытых САПР является быстроедействие синтезируемых схем – при использовании коммерческих инструментов максимальная тактовая частота, как правило, выше (в 1.3-20 раз).

Список литературы / References

- [1]. Graphic Design System User's Operating Manual First Edition. Available at: http://www.bitsavers.org/pdf/calma/GDS_II_Users_Operating_Manual_Nov78.pdf, accessed: 25.10.2021.
- [2]. L. Lavagno, G. Martin, L. Scheffer. Electronic Design Automation for Integrated Circuits Handbook – 2 Volume Set. CRC Press, 2006, 1152 p.
- [3]. qFlow. Available at: <http://opencircuitdesign.com/qflow>, accessed: 25.10.2021.
- [4]. GNU General Public License. Available at: <https://www.gnu.org/licenses/gpl-3.0.ru.html>, accessed: 25.10.2021.
- [5]. Yosys Open SYnthesis Suite. Available at: <http://www.clifford.at/yosys>, accessed: 25.10.2021.
- [6]. Odin II. Available at: <https://docs.verilogtorouting.org/en/latest/odin>, accessed: 25.10.2021.
- [7]. ABC: A System for Sequential Synthesis and Verification, Available at: <http://www.eecs.berkeley.edu/~alanmi/abc>, accessed: 25.10.2021.
- [8]. Berkeley Logic Interchange Format (BLIF), Available at: <http://www.cs.columbia.edu/~cs6861/sis/blif/index.html>, accessed: 25.10.2021.
- [9]. Graywolf. Available at: <https://github.com/rubund/graywolf>, accessed: 25.10.2021.
- [10]. Qrouter. Available at: <http://opencircuitdesign.com/qrouter>, accessed: 25.10.2021.
- [11]. LEF/DEF Version 5.7. Available at: <http://www.ispd.cc/contests/18/lefdefref.pdf>, accessed: 25.10.2021.

- [12]. OpenSTA. Available at: <https://github.com/The-OpenROAD-Project/OpenSTA>, accessed: 25.10.2021.
- [13]. Vesta. Available at: <https://github.com/RTimothyEdwards/qflow/blob/master/src/vesta.c>, accessed: 25.10.2021.
- [14]. Magic VLSI Layout Tool. Available at: <http://opencircuitdesign.com/magic>, accessed: 25.10.2021.
- [15]. SPICE. Available at: <http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE>, accessed: 25.10.2021.
- [16]. Netgen. Available at: <http://opencircuitdesign.com/netgen>, accessed: 25.10.2021.
- [17]. Oklahoma State University System on Chip (SoC) Design Flows, Available at: <https://vlsiarch.ecen.okstate.edu/flow>, accessed: 25.10.2021.
- [18]. T. Edwards, M. Kassem, C. Wolf. PicoSoC: How we created a RISC-V based ASIC processor using a full open-source foundry targeted RTL-to-GDS flow, and how you can, too! 7th RISC-V Workshop, 2017. Available at: <https://riscv.org/wp-content/uploads/2017/12/Wed-1142-RISCV-Tim-Edwards.pdf>, accessed: 25.10.2021.
- [19]. OpenLANE. Available at: <https://openlane.readthedocs.io>, accessed: 25.10.2021.
- [20]. Open source process design kit for usage with SkyWater Technology Foundry's 130nm node. Available at: <https://github.com/google/skywater-pdk>, дата обращения 25.10.2021.
- [21]. OpenROAD, Available at: <https://github.com/The-OpenROAD-Project/OpenROAD>, accessed: 25.10.2021.
- [22]. UCSD Detailed Router, Available at: <https://github.com/The-OpenROAD-Project/TritonRoute>, accessed: 25.10.2021.
- [23]. KLayout Layout Viewer and Editor. Available at: <https://www.klayout.de>, accessed: 25.10.2021.
- [24]. MPW-ONE Projects. Available at: https://efabless.com/projects/shuttle_name/MPW-ONE, accessed: 25.10.2021.
- [25]. Coriolis VLSI CAD Tools. Available at: <http://coriolis.lip6.fr>, accessed: 25.10.2021.
- [26]. M.-M. Louerat, R. Chotin et al. RISC-V design using FOSS. Available at: <https://open-src-soc.org/2019-10/media/slides/2nd-RISC-V-Meeting-2019-10-01-10h00-Jean-Paul-Chaput.pdf>, accessed: 25.10.2021.
- [27]. GNU Lesser General Public License. Available at: <https://www.gnu.org/licenses/lgpl-3.0.ru.html>, accessed: 25.10.2021.
- [28]. N. Shimizu. FOSS for Free HW with Japanese technology process, Available at: https://www.lip6.fr/public/2018-03-09_Shimizu.pdf, accessed: 25.10.2021.
- [29]. Hurricane. Available at: <https://gitlab.lip6.fr/vlsi-eda/coriolis/-/tree/devel/hurricane>, accessed: 25.10.2021.
- [30]. Coriolis Etesian, Available at: <https://github.com/Coloquinte/Coloquinte>, accessed: 25.10.2021.
- [31]. Katana. Available at: <https://gitlab.lip6.fr/vlsi-eda/coriolis/-/tree/devel/katana>, accessed: 25.10.2021.
- [32]. FLUTE. Available at: <http://home.eng.iastate.edu/~cnchu/flute.html>, accessed: 25.10.2021.
- [33]. Symbolic Layout. Available at: <http://coriolis.lip6.fr/pages/symbolic-layout.html>, accessed: 25.10.2021.
- [34]. HiTas. Available at: <https://www-soc.lip6.fr/en/team-cian/software/tasyagle>, accessed: 25.10.2021.
- [35]. s2r. Available at: <https://gitlab.lip6.fr/vlsi-eda/coriolis/-/blob/devel/cumulus/src/plugins/s2r.py>, accessed: 25.10.2021.
- [36]. druc. Available at: <https://gitlab.lip6.fr/vlsi-eda/alliance/-/tree/master/alliance/src/druc>, accessed: 25.10.2021.
- [37]. OpenAccess. Available at: <https://si2.org/oa-tools-utils-libs>, accessed: 25.10.2021.
- [38]. Verilog-to-Routing, Available at: <https://docs.verilogtorouting.org/en/latest>, accessed: 25.10.2021.
- [39]. K.E. Murray, O. Petelin et al. VTR 8: High Performance CAD and Customizable FPGA Architecture Modelling. *ACM Transaction on Reconfigurable Technology and Systems*, vol. 13, issue 2, 2020, Article 9, 55 p.
- [40]. The MIT License. Available at: <https://opensource.org/licenses/MIT>, accessed: 25.10.2021.
- [41]. VPR. Available at: <https://docs.verilogtorouting.org/en/latest/vpr>, accessed: 25.10.2021.
- [42]. FPGA Assembly (FASM) Output. Available at: <https://docs.verilogtorouting.org/en/latest/utils/fasm>, accessed: 25.10.2021.
- [43]. K.E. Murray, S. Whitty, S. Liu, J. Luu, V. Betz. Timing-Driven Titan: Enabling Large Benchmarks and Exploring the Gap between Academic and Commercial CAD. *ACM Transaction on Reconfigurable Technology and Systems*, vol. 8, issue 2, 2015, Article 10, 18 p.
- [44]. SymbiFlow. Available at: <https://symbiflow.readthedocs.io/en/latest/toolchain-desc/design-flow.html>, accessed: 25.10.2021.
- [45]. Verific Design Automation. Available at: <https://www.verific.com>, accessed: 25.10.2021.

- [46]. nextpnr. Available at: <https://github.com/YosysHQ/nextpnr>, accessed: 25.10.2021.
- [47]. OpenFPGA. Available at: <https://github.com/lnis-uofu/OpenFPGA>, accessed: 25.10.2021.
- [48]. Vivado ML. Available at: <https://www.xilinx.com/products/design-tools/vivado.html>, accessed: 25.10.2021.
- [49]. ISE Design Suite. Available at: <https://www.xilinx.com/products/design-tools/ise-design-suite.html>, accessed: 25.10.2021.
- [50]. Intel Quartus Prime. Available at: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/download.html>, accessed: 25.10.2021.
- [51]. Project IceStorm. Available at: <http://www.clifford.at/icestorm>, accessed: 25.10.2021.
- [52]. Project Trellis. Available at: <https://github.com/YosysHQ/prjtrellis>, accessed: 25.10.2021.
- [53]. Project Oxideio. Available at: <https://github.com/gatecat/prjoxide>, accessed: 25.10.2021.
- [54]. Project Apicula. Available at: <https://github.com/YosysHQ/apicula>, accessed: 25.10.2021.
- [55]. Project X-Ray. Available at: <https://github.com/SymbiFlow/prjxray>, accessed: 25.10.2021.
- [56]. K.E. Murray et al. SymbiFlow and VPR: An Open-Source Design Flow for Commercial and Novel FPGAs. *IEEE Micro*, vol. 40, issue 4, 2020, pp. 49-57.
- [57]. Odin II Verilog Support. Available at: https://docs.verilogtorouting.org/en/latest/odin/verilog_support, accessed: 25.10.2021.
- [58]. GHDL. Available at: <https://github.com/ghdl/ghdl>, accessed: 25.10.2021.
- [59]. C. Sechen, A. Vincentelli. The Timber Wolf Placement and Routing Package. *IEEE Journal of Solid-State Circuits*, vol. 20, issue 2, 1985, pp. 510-522.
- [60]. FastRoute. Available at: <http://home.eng.iastate.edu/~cnchu/FastRoute.html>, accessed: 25.10.2021.
- [61]. M. Pan, Y. Xu et al. FastRoute: an efficient and high-quality global router. *VLSI Design*, 2012, Article 14, 14 p.
- [62]. BoxRouter. Available at: <https://www.cerc.utexas.edu/utda/download/BoxRouter.htm>, accessed: 25.10.2021.
- [63]. M. Cho, K. Lu et al. BoxRouter 2.0: Architecture and Implementation of a Hybrid and Robust Global Router. In *Proc. of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2007, pp. 503-508.
- [64]. C.Y. Lee. An Algorithm for Path Connections and Its Applications. *IRE Transactions on Electronic Computers*, vol. EC-10, issue 3, 1961, pp. 346-365.
- [65]. A.B. Kahng, L. Wang, B. Xu. TritonRoute: The Open-Source Detailed Router. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, issue 3, 2021, pp. 547-559.
- [66]. SEMI P44 – Specification for OASIS. Available at: <https://store-us.semi.org/products/p04400-semi-p44-specification-for-open-artwork-system-interchange-standard-oasis-%C2%AE-specific-to-mask-tools>, accessed: 25.10.2021.
- [67]. Synopsys Design Format. Available at: https://www.intel.com/content/www/us/en/programmable/quartushelp/13.0/mergedProjects/reference/glossary/def_sdc.htm, accessed: 25.10.2021.
- [68]. Liberty. Available at: https://people.eecs.berkeley.edu/~alanmi/publications/other/liberty07_03.pdf, accessed: 25.10.2021.
- [69]. Alliance standard cell libraries. Available at: http://www.vlsitechnology.org/html/sx_description.html, accessed: 10.11.2021.
- [70]. FreePDK45. Available at: <https://research.ece.ncsu.edu/eda/freepdk/freepdk45>, accessed: 25.10.2021.
- [71]. 45nm FreePDK. Open-Cell 15nm, Available at: <https://si2.org/open-cell-library>, accessed: 25.10.2021.
- [72]. Silicon Integration Initiative. Available at: <https://si2.org>, accessed: 25.10.2021.
- [73]. Nangate and Si2 Release Unprecedented Free 45nm Open Source Digital Cell Library. Available at: <https://www.src.org/newsroom/press-release/2008/46>, accessed: 25.10.2021.
- [74]. L.T. Clark, V. Vashishtha et al. Design flows and collateral for the ASAP7 7nm FinFET predictive process design kit. In *Proc. of the 2017 IEEE International Conference on Microelectronic Systems Education (MSE)*, 2017, pp. 1-4.
- [75]. FreePDK15. Available at: <https://research.ece.ncsu.edu/eda/freepdk/freepdk15>, accessed: 25.10.2021.
- [76]. M. Martins, J. M. Matos et al. Open Cell Library in 15nm FreePDK Technology. In *Proc. of the International Symposium on Physical Design*, 2015, pp. 171-178.
- [77]. SAP7 PDK. Available at: <https://github.com/The-OpenROAD-Project/asap7>, accessed: 25.10.2021.
- [78]. RISC-V Architecture, Available at: <https://riscv.org>, accessed: 25.10.2021.

- [79]. PicoRV32, Available at: <https://github.com/cliffordwolf/picorv32>, accessed: 25.10.2021.
- [80]. K.P. Ghosh, A.K. Ghosh. Technology mediated tutorial on RISC-V CPU core implementation and sign-off using revolutionary EDA management system (EMS) – VSDFLOW. In *Proc. of the China Semiconductor Technology International Conference (CSTIC)*, 2018, pp. 1-3.
- [81]. VSDFLOW. Available at: <https://github.com/kunalg123/vsdfLOW>, accessed: 25.10.2021.

Информация об авторах / Information about authors

Александр Сергеевич КАМКИН – кандидат физико-математических наук, ведущий научный сотрудник отдела технологий программирования ИСП РАН, ведущий научный сотрудник научной лаборатории «Гетерогенные компьютерные системы» РЭУ им. Г.В. Плеханова; преподает в МГУ, МФТИ и НИУ ВШЭ. Область научных интересов: автоматизация проектирования цифровой аппаратуры, верификация и тестирование.

Alexander Sergeevich KAMKIN is a leading researcher at the Software Engineering Department ISP RAS, leading researcher at the Plekhanov RUE. He is also a lecturer at MSU, MIPT, and HSE. His research interests include digital hardware design automation, verification and testing. Alexander has a PhD degree in Physics and Mathematics.

Сергей Александрович СМОЛОВ – научный сотрудник отдела технологий программирования ИСП РАН, старший научный сотрудник научной лаборатории «Гетерогенные компьютерные системы» РЭУ им. Г.В. Плеханова. Область научных интересов: автоматизация проектирования цифровой аппаратуры, верификация и тестирование.

Sergey Aleksandrovich SMOLOV is a researcher at the Software Engineering Department of ISP RAS, senior researcher at the Heterogeneous Computing Systems research lab of Plekhanov RUE. His research interests include digital hardware design automation, verification and testing.

Михаил Михайлович ЧУПИЛКО – кандидат физико-математических наук, старший научный сотрудник отдела технологий программирования ИСП РАН, старший научный сотрудник научной лаборатории «Гетерогенные компьютерные системы» РЭУ им. Г.В. Плеханова. Область научных интересов: автоматизация проектирования цифровой аппаратуры, верификация и тестирование.

Mikhail Mikhaylovich CHUPILKO is a senior researcher at the Software Engineering Department of Ivannikov Institute for System Programming of ISP RAS, senior researcher at the Heterogeneous Computing Systems research lab of Plekhanov RUE. His research interests include digital hardware design automation, verification and testing. Mikhail has a PhD degree in Physics and Mathematics.