

Анализ работы программных инструментов с открытым исходным кодом OpenLANE для разработки микросхем

А. В. Зыков, Р. Ф. Ильясов

АНО ВО «Университет Иннополис», г. Иннополис

a.zykov@innopolis.ru, r.iliasov@innopolis.ru

Аннотация — Целью исследования является анализ применения имеющихся свободно распространяемых программных инструментов с открытым исходным кодом для полного цикла проектирования цифровых микросхем от HDL описания до геометрии GDSII. Результаты сравниваются с результатами коммерческого программного обеспечения Cadence.

Ключевые слова — микроэлектроника, VLSI, САПР, маршрут проектирования, OpenROAD.

I. ВВЕДЕНИЕ

Современные программные инструменты разработки микросхем достигли высокого уровня развития. В то же время коммерческие инструменты имеют высокую стоимость. Это не представляет проблем для крупных разработчиков. Но становится серьёзной преградой для небольших коллективов желающих спроектировать собственную микросхему для своих целей.

Один пример – крупная компания разрабатывает передовую микросхему, которая будет производиться в больших объёмах по наиболее современной технологии (например, 5 нм – 10 нм). Разработчики вкладывают значительные ресурсы в то, чтобы добиться высоких показателей этой микросхемы (тактовая частота, потребление энергии и прочее). В этой ситуации более чем оправдано использование коммерческих программных инструментов с широкими возможностями.

Другой пример – небольшой коллектив разрабатывает электронную систему, для которой требуется цифровая микросхема. Эта электронная система будет производиться в сравнительно небольших объёмах (например, несколько сотен или несколько тысяч единиц). Есть выбор использовать имеющиеся на рынке микросхемы – микроконтроллеры, ПЛИС. Альтернативный вариант, спроектировать собственную микросхему (ASIC) специально для этой системы. Специализированная микросхема будет иметь гораздо более высокие характеристики по сравнению с микросхемами общего назначения, даже такими как ПЛИС, и даже если технология изготовления будет не самой современной (например, 130 нм или возможно 65 нм, 28 нм). В этой ситуации высокая стоимость коммерческих

программных инструментов становится серьёзной преградой. Ситуацию может исправить свободное программное обеспечение для полного маршрута проектирования микросхем.

Сам по себе маршрут проектирования имеет несколько шагов: синтез, оптимизация, размещение, трассировка, статический анализ временных задержек, формирование геометрии, верификация. Для каждого шага есть один или несколько, обычно независимых, свободных инструментов. Объединение нескольких инструментов в целостный маршрут проектирования может представлять собой нетривиальную задачу. Поэтому представляют особый интерес уже готовые системы инструментов собранные в целостный маршрут проектирования.

С точки зрения импортозамещения и технологического суверенитета важно наличие отечественных инструментов для разработки микросхем. При небольшом объёме отечественных разработок в области микроэлектроники, с экономической точки зрения использование инструментов с открытым исходным кодом представляется более реалистичным, чем построение отечественной коммерческой линейки инструментов для проектирования микросхем. В этом случае можно самостоятельно произвести ответвление от основной ветки разработки. Во-первых, это сэкономит трудозатраты, т. к. разработка будет происходить не с нуля. Во-вторых, это позволит самостоятельно добавлять нужный функционал в инструменты, не обращаясь к зарубежным коммерческим разработчикам.

II. ПО С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ ДЛЯ ПРОЕКТИРОВАНИЯ МИКРОСХЕМ

На текущий момент существует несколько систем инструментов для маршрута проектирования основанных на программных инструментах с открытым кодом. Но не все они полезны для проектирования микросхем (ASIC). Некоторые, такие как VTR (Verilog-to-Routing) и F4PGA (ранее известный как SymbiFlow), ориентированы на синтез образа для программирования ПЛИС. Такие системы невозможно использовать для проектирования микросхем без серьёзной переработки.

A. Системы, ориентированные на разработку микросхем

ALLIANCE/CORIORIS — разрабатывается институтом LIP6 (Университет Сорбонны, Франция) [1]. Активно разрабатывался в 1990–2000 годы. Сейчас в состоянии поддержки.

Qflow — разрабатывался Тимати Эдвардсом (R. Timothy Edwards) с 2013 года в группе Open Circuit Design [2].

OpenROAD — амбициозный проект под финансированием DARPA, стартовавший летом 2018, [3]-[5]. Разрабатывается группами в нескольких университетах США под руководством профессора Эндрю Канга, Калифорнийский Университет Сан-Диего (Prof. Andrew Kahng, UC San Diego VLSI CAD Laboratory). Целью проекта является создание единого программного инструмента выполняющего все этапы проектирования RTL-to-GDSII (от HDL описания до геометрии чипа), работающего в автоматическом режиме «без человека внутри цикла». На данный момент проект не завершен. Репозиторий GitHub проекта OpenROAD [6] содержит подраздел OpenROAD-flow-scripts организующий полный маршрут проектирования на основе инструмента OpenROAD и двух других инструментов в тех областях, где инструмент OpenROAD пока не реализован. Для синтеза Verilog HDL в логическую схему используется инструмент Yosys. Для финальной генерации геометрии в формате GDSII используется инструмент KLayout.

OpenLANE — система инструментов в значительной степени основанная на OpenROAD. Разрабатывается компанией Efabless с осени 2019 года, [7]-[9]. Над проектом работает так же Тимати Эдвардс (разработчик системы Qflow), как сотрудник компании Efabless. OpenLANE разрабатывался для инициативы chipIgnite, которой занимается компания Efabless совместно с заводом по производству чипов SkyWater при финансовой поддержке Google. В рамках этого проекта небольшим группам разработчиков (стартапы, университетские группы и прочее) предоставляется возможность спроектировать и изготовить небольшую партию чипов (100 штук в корпусе QFN или 300 штук в корпусе WCSP) на заводе SkyWater по технологии 130 нм (площадь на чипе 10 мм²). Для проектирования предлагается использовать систему инструментов с открытым исходным кодом OpenLANE, но по желанию можно использовать и свою коммерческую систему. Заявленная цель инициативы chipIgnite — создание экосистемы открытого исходного кода в полупроводниковой области. Участники обязаны выложить исходный код своих полупроводниковых проектов в открытый доступ.

В этом исследовании был выбран маршрут проектирования OpenLANE, как наиболее перспективный — не устаревший, реально работающий, проверенный на практике в проекте chipIgnite при проектировании реальных чипов, изготовленных на фабрике SkyWater.

B. Наиболее значимые программные инструменты

Yosys — инструмент для синтеза логических схем из Verilog HDL и их оптимизации. Разрабатывается с 2013 года Клиффордом Вольфом (Вена, Австрия), [10], [11]. Используется в OpenLANE и практически безальтернативно в других системах тоже. Так, например изначально Qflow использовал другой инструмент ODIN II [12], но потом полностью перешел на Yosys.

ABC — библиотека оптимизации логических схем, активно разрабатываемая в университете Беркли (Alan Mishchenko) [13]. Безальтернативно используется во всех других инструментах, таких как Yosys, ODIN II. Так же используется внутри OpenROAD для локального ресинтеза модулем «Restructure» (RMP).

Magic и KLayout — программные инструменты для генерации геометрии в формате GDSII. В OpenLANE используется Magic для генерации и проверки физических ограничений (design rule check, DRC). KLayout используется для сравнения с результатом от Magic для дополнительной верификации.

Netgen — дополнительный инструмент верификации в наборе OpenLANE. Производит сравнение геометрии со схематикой (layout versus schematic, LVS).

CVC — дополнительный инструмент верификации в наборе OpenLANE. Производит оценку верности схемы (circuit validity check).

C. Система инструментов OpenLANE

Система OpenLANE устанавливается из GitHub репозитория в виде набора скриптов и набора прекомпилированных приложений в докер-контейнере. Основные приложения, используемые для синтеза — Yosys, OpenROAD, Magic.

При работе OpenLANE выполняет следующие основные шаги:

- 1) Yosys — синтез логических схем из HDL
- 2) OpenROAD — статический анализ задержек (static timing analysis, STA)
- 3) OpenROAD — инициализация размещения элементов (floorplan)
- 4) OpenROAD — размещение элементов ввода-вывода
- 5) OpenROAD — размещение элементов tap/decap
- 6) OpenROAD — генерация сети распределения питания (power distribution network, PDN)
- 7) OpenROAD — глобальное размещение элементов
- 8) OpenROAD — оптимизация по изменению размеров
- 9) OpenROAD — детальное размещение элементов
- 10) OpenROAD — генерация дерева тактового сигнала (clock tree synthesis, CTS)

Результаты полного синтеза в OpenLANE после получения геометрии

Модуль	Площадь (мкм ²)	Количество ячеек	Количество флип-флопов	Максимальная частота (МГц)
арбитр 4	10138	36	12	515,5
арбитр 8	8551	74	24	436,7
арбитр 12	8551	115	36	401,6
арбитр 16	8134	153	48	337,8
арбитр 32	7647	308	96	185,2
арбитр 48	11418	469	144	119,5
арбитр 64	15258	619	192	88,9
арбитр 80	19335	782	240	73,7
арбитр 96	23332	943	288	63,4
AES	1519574	18753	2994	76,0
AES-core	837443	18720	2476	91,7

- 11) OpenROAD – изменение размеров для оптимизации задержек
- 12) OpenROAD – детальное размещение элементов
- 13) OpenROAD – глобальная трассировка
- 14) OpenROAD – вставка заполнителей
- 15) OpenROAD – детальная трассировка
- 16) OpenROAD – экстракция паразитных ёмкостей и сопротивлений
- 17) OpenROAD – статический анализ задержек
- 18) Magic – генерация геометрии GDSII
- 19) KLayout – генерация геометрии GDSII
- 20) OpenROAD – сравнение результатов Magic и KLayout
- 21) Magic – экстракция электрических цепей SPICE
- 22) Netgen – верификация геометрии по сравнению со схематикой (LVS)
- 23) Magic – проверка физических ограничений (DRC)
- 24) OpenROAD – проверка наличия дефектов «антенна»
- 25) CVC – валидация электрических цепей

Стоит заметить, что в вычислительном плане самый дорогой шаг – это шаг номер 15 «детальная трассировка» (detailed routing). Например, для тестового модуля AES полное время обработки заняло 3792 секунды. Из них 2359 секунд (62%) ушло на шаг номер 15. Следующий за ним по затратам времени – шаг номер 23 (Magic, DRC), на который ушло 483 секунды (13%). Так же пиковое потребление памяти

было самым высоким на шаге номер 15, который потребовал 6.6 Гб для тестового дизайна AES. После синтеза этот дизайн содержал 20155 библиотечных ячеек.

III. СИСТЕМЫ ДЛЯ ТЕСТИРОВАНИЯ МАРШРУТА ПРОЕКТИРОВАНИЯ

Для анализа возможностей системы OpenLANE были использованы модули арбитров с размером от 4 запросчиков до 96 запросчиков.

Использовался вариант простого приоритетного арбитра. Запросчик с нулевым номером имеет наивысший приоритет. За ним следует номер один. За ним следует номер два и так далее. Простой арбитр в свою очередь может быть использован для построения более сложных арбитров.

Дополнительно были использованы два более крупных модуля входящих в набор тестов OpenLANE – модуль AES и модуль AES-core. Эти два модуля имеют один и тот же HDL код выполняющий функционал AES шифрования, но отличаются в параметрах синтеза.

Основной параметр на этапе RTL-синтеза – это ограничение на максимальное количество входов вентиля, на которые нагружен выход данного логического вентиля (maximum fanout). Для всех модулей арбитров использовался параметр 5. В OpenLANE этот параметр задаётся Tcl-командой:

```
set ::env(SYNTH_MAX_FANOUT) 5
```

Для модулей AES были использованы значения, заданные в самих тестах – значение 8 для AES и значение 6 для AES-core.

Два других важных параметра относятся к этапу размещения элементов (floorplan, placement). Они задают целевой процент использования площади ядра под сами элементы в виде Tcl-команд:

```
set ::env(FP_CORE_UTIL) 45
set ::env(PL_TARGET_DENSITY) [ expr
($::env(FP_CORE_UTIL)+5) / 100.0 ]
```

Параметр FP_CORE_UTIL задаёт целевой процент использования площади чипа на этапе планирования размещения (floorplan). Параметр PL_TARGET_DENSITY задаёт целевую плотность размещения на этапе размещения элементов (placement).

Для арбитров размером 32 и выше использовался параметр 45. Но для арбитров размером от 4 до 16 потребовалось уменьшать этот параметр из-за ошибок на этапе генерации сети распределения питания. Видимому, код отвечающий за генерацию сети питания не рассчитан на такие маленькие модули. Поэтому площадь занимаемая этими маленьким арбитрами получилась выше, чем площадь арбитра размером 32.

Для модулей AES и AES-core использовались значения, заданные в тестах – значение 15 для AES и значение 25 для AES-core.

IV. АНАЛИЗ РЕЗУЛЬТАТОВ ПРОЕКТИРОВАНИЯ

В таблице 1 представлены результаты полного синтеза (после размещения элементов, трассировки и формирования геометрии) нескольких модулей – простых арбитров с разным количеством запросчиков и два варианта модуля AES.

Для каждого модуля представлены площадь на чипе (в квадратных микрометрах), количество функциональных ячеек (включая буферные ячейки), количество флип-флопов и максимальная тактовая частота (в МГц).

Арбитры размером меньше 32 занимают высокую площадь из-за маленького значения параметра FP_CORE_UTIL. По этой же причине модуль AES занимает площадь больше, чем модуль AES-core, т. к. для него этот параметр имеет значение 15 в отличие от 25 для второго.

Указанное количество ячеек включает логические вентили и буферы (в том числе буферы дерева распределения тактового сигнала), и не включает флип-флопы и технические ячейки, как например tap cell, decap.

На рисунке 1 приведен пример изображения размещения элементов на кристалле для модуля AES после формирования геометрии GDSII в инструменте Magic.

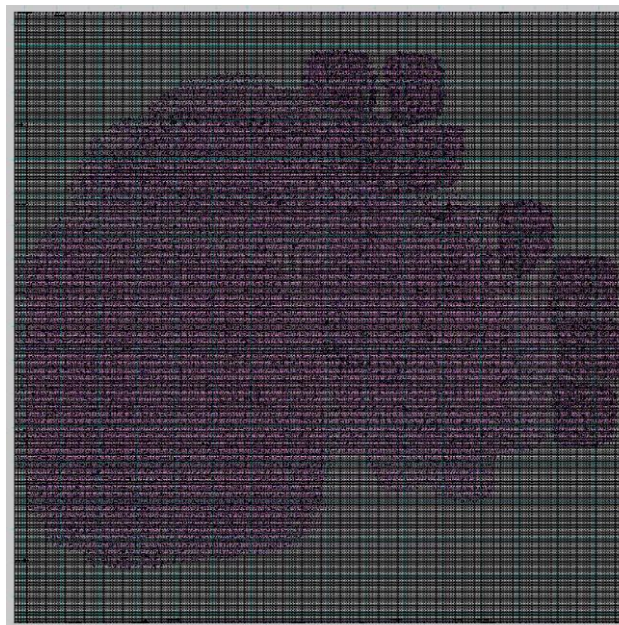


Рис. 1. Изображение GDSII результата в Magic

А. Сравнение с Cadence GENUS

В таблице 2 представлены результаты сравнения работы этапа RTL-синтеза в OpenLANE (выполняется инструментом Yosys) и коммерческого инструмента Cadence GENUS. В отличие от таблицы 1 эти результаты получены до размещения элементов и трассировки. Соответственно, площадь и задержки рассчитываются по упрощённым моделям без учёта соединений.

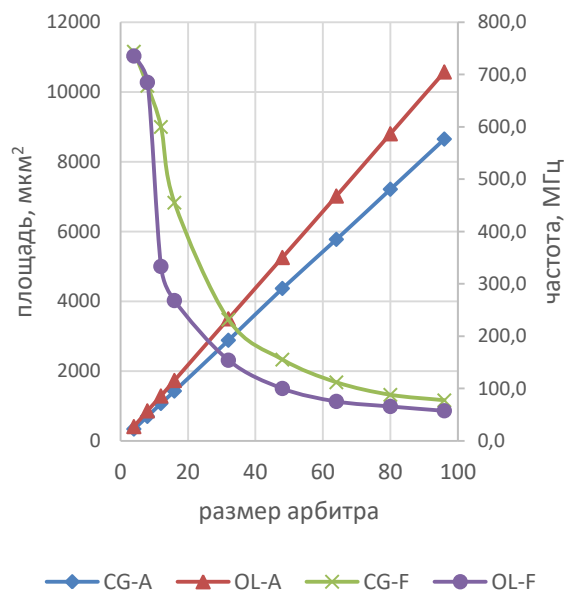


Рис. 2. Синтез в Cadence GENUS и OpenLANE

На рисунке 2 представлены графики площади и частоты для арбитров разного размера при синтезе в Cadence GENUS (CG-A – площадь, CG-F - частота) и в OpenLANE (OL-A – площадь, OL-F - частота).

Сравнение результатов RTL-синтеза в OpenLANE и Cadence GENUS (OpenLANE / GENUS)

Модуль	Площадь (мкм ²)	Количество ячеек	Количество флип-флопов	Максимальная частота (МГц)
арбитр 4	409 / 339	21 / 13	12 / 12	735,3 / 743,5
арбитр 8	860 / 702	50 / 29	24 / 24	684,9 / 678,2
арбитр 12	1286 / 1072	79 / 47	36 / 36	333,3 / 599,7
арбитр 16	1730 / 1428	107 / 61	48 / 48	268,1 / 455,0
арбитр 32	3497 / 2883	222 / 126	96 / 96	154,1 / 231,5
арбитр 48	5254 / 4368	337 / 199	144 / 144	100,1 / 155,2
арбитр 64	7015 / 5774	451 / 275	192 / 192	75,4 / 111,8
арбитр 80	8800 / 7211	566 / 355	240 / 240	65,8 / 87,9
арбитр 96	10571 / 8648	681 / 434	288 / 288	57,5 / 77,4
AES	228303 / 162288	17161 / 11993	2994 / 2987	* / 144,5
AES-core	209968 / 169987	16881 / 14192	2476 / 2987	* / 157,7

Для сравнимости результатов к Cadence GENUS было подключено то же описание технологии (PDK), что входит в OpenLANE по умолчанию – «sky130A». Это технология 130 нм фабрики SkyWater. Как для OpenLANE, так и для Cadence GENUS представленные результаты были получены при использовании библиотеки «sky130_fd_sc_hd» (здесь «hd» означает «high density» - высокая плотность). Коммерческий инструмент производит результат с более высокой степенью оптимизации, чем свободно распространяемый инструмент, что ожидаемо. Для больших модулей AES и AES-core расчёт временных задержек после этапа RTL-синтеза в OpenLANE происходит некорректно (в таблице вместо этих частот «звёздочки»). Впоследствии, после размещения элементов, эта ситуация нормализуется. Для небольших модулей арбитров такой проблемы не возникает.

V. ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен маршрут проектирования микросхем OpenLANE основанный на инструментах с открытым исходным кодом. Этот маршрут проектирования является на данный момент наиболее перспективом, так как он активно разрабатывается и уже протестирован при производстве реальных микрочипов (инициатива chipIgnite).

В своем текущем состоянии OpenLANE в значительной своей части, отвечающей за физический синтез, полагается на инструмент OpenROAD. В свою очередь OpenROAD, хотя и находится в стадии активной разработки, но пока крайне далёк от завершенности. В перспективе можно ожидать, что OpenROAD достигнет состояния, когда сможет

заменить собой полностью OpenLANE. Так же эта незавершенность сказывается и на работе OpenLANE в текущем виде. Например, этап детальной трассировки, который выполняет OpenROAD в OpenLANE, даже для сравнительно небольших модулей (десятки тысяч логических вентилях) требует значительный вычислительных затрат как по времени вычислений, так и по требуемой памяти. Этап детальной трассировки в OpenROAD выполняет TritonRoute [14], [15].

Сравнение результатов работы свободно распространяемых инструментов OpenLANE и коммерческого инструмента Cadence GENUS показало более высокую степень оптимальности для коммерческого инструмента, при соизмеримых в целом результатах, полученных в этих программных инструментах.

ЛИТЕРАТУРА

- [1] URL: <http://coriolis.lip6.fr> (дата обращения: 15.07.2022)
- [2] URL: <http://opencircuitdesign.com/qflow> (дата обращения: 15.07.2022)
- [3] Samuel K. M., DARPA Plans a Major Remake of U.S. Electronics, IEEE Spectrum, July 2018.
- [4] Ajayi T., Blaauw D., Chan T.-B., Cheng C.-K., Chhabria V. A., Choo D. K., Coltella M., Dobre S., Dreslinski R., Fogaça M, Hashemi S., Hosny A., Kahng A. B., Kim M., Li J, Liang Z., Mallappa U., Penzes P., Pradipta G., Reda S., Rovinski A., Samadi K., Sapatnekar S. S., Saul L., Sechen C., Srinivas V., Swartz W., Sylvester D., Urquhart D., Wang L., Woo M., Xu B., OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain, Proc. Government Microcircuit Applications and Critical Technology Conference, 2019, P. 1105-1110.
- [5] URL: <https://theopenroadproject.org> (дата обращения: 15.07.2022)

- [6] URL: <https://github.com/The-OpenROAD-Project> (дата обращения: 15.07.2022)
- [7] Ghazy A., Shalan M., OpenLANE: The Open-Source Digital ASIC Implementation Flow, Workshop on Open-Source EDA Technology (WOSET), 2020, Article № 21.
- [8] Shalan M., Edwards T., Building OpenLANE: A 130nm OpenROAD-based Tapeout-Proven Flow : Invited Paper, 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), San Diego, CA, USA, 2020, P. 1-6.
- [9] Edwards T., Shalan M., Kassem M., Real Silicon using Open Source EDA, in IEEE Design & Test, 2021, doi: 10.1109/MDAT.2021.3050000.
- [10] Wolf C., Glaser J., Yosys — A Free Verilog Synthesis Suite, In Proceedings of Austrochip 2013.
- [11] Glaser J. and Wolf C., Methodology and Example-Driven Interconnect Synthesis for Designing Heterogeneous Coarse-Grain Reconfigurable Architectures, In Jan Haase, editor, Models, Methods, and Tools for Complex Chip Design. Lecture Notes in Electrical Engineering. Springer, 2013, V. 265, 2014, P. 201-221.
- [12] Jamieson P., Kent K. B., Gharibian F., Shannon L., Odin II - An Open-source Verilog HDL Synthesis tool for CAD Research, Proc.of the IEEE Symposium on Field-Programmable Custom Computing Machines, 2010, P. 149-156
- [13] URL: <http://people.eecs.berkeley.edu/~alanmi/abc/> (дата обращения: 15.07.2022)
- [14] Kahng A. B., Wang L., Xu B., TritonRoute: The Open Source Detailed Router, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, doi:10.1109/TCAD.2020.3003234.
- [15] Kahng A. B., Wang L., Xu B., The Tao of PAO: Anatomy of a Pin Access Oracle for Detailed Routing, Proc. ACM/IEEE Design Automation Conf., 2020, P. 1-6.

Analysis of Open-source EDA Tools OpenLANE for ASIC Design

A.V. Zykov, R.F. Iliasov

"University Innopolis", Innopolis

a.zykov@innopolis.ru, r.iliasov@innopolis.ru

Abstract — The objective of this investigation is to explore the application of current open-source EDA RTL-to-GDSII ASIC design flows. The OpenLANE flow was chosen for the analysis as the currently most prominent example. This flow is actively developed, and it was used to produce real microchips through chipIgnite project. The comparison between open-source OpenLANE tools and commercial tool Cadence GENUS demonstrated higher optimization level in commercial tool. Though the result of both tools seems to be comparable enough to prove practical usefulness of the open-source OpenLANE tool chain.

Keywords — VLSI, OpenROAD.

REFERENCES

- [1] URL: <http://coriolis.lip6.fr> (access date: 15.07.2022)
- [2] URL: <http://opencircuitdesign.com/qflow> (access date: 15.07.2022)
- [3] Samuel K. M., DARPA Plans a Major Remake of U.S. Electronics, IEEE Spectrum, July 2018.
- [4] Ajayi T., Blaauw D., Chan T.-B., Cheng C.-K., Chhabria V. A., Choo D. K., Coltella M., Dobre S., Dreslinski R., Fogaça M, Hashemi S., Hosny A., Kahng A. B., Kim M., Li J, Liang Z., Mallappa U., Penzes P., Pradipta G., Reda S., Rovinski A., Samadi K., Sapatnekar S. S., Saul L., Sechen C., Srinivas V., Swartz W., Sylvester D., Urquhart D., Wang L., Woo M., Xu B., OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain, Proc. Government Microcircuit Applications and Critical Technology Conference, 2019, P. 1105-1110.
- [5] URL: <https://theopenroadproject.org> (access date: 15.07.2022)
- [6] URL: <https://github.com/The-OpenROAD-Project> (access date: 15.07.2022)
- [7] Ghazy A., Shalan M., OpenLANE: The Open-Source Digital ASIC Implementation Flow, Workshop on Open-Source EDA Technology (WOSET), 2020, Article № 21.
- [8] Shalan M., Edwards T., Building OpenLANE: A 130nm OpenROAD-based Tapeout-Proven Flow : Invited Paper, 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), San Diego, CA, USA, 2020, P. 1-6.
- [9] Edwards T., Shalan M., Kassem M., Real Silicon using Open Source EDA, in IEEE Design & Test, 2021, doi: 10.1109/MDAT.2021.3050000.
- [10] Wolf C., Glaser J., Yosys — A Free Verilog Synthesis Suite, In Proceedings of Austrochip 2013.
- [11] Glaser J. and Wolf C., Methodology and Example-Driven Interconnect Synthesis for Designing Heterogeneous Coarse-Grain Reconfigurable Architectures, In Jan Haase, editor, Models, Methods, and Tools for Complex Chip Design. Lecture Notes in Electrical Engineering. Springer, 2013, V. 265, 2014, P. 201-221.
- [12] Jamieson P., Kent K. B., Gharibian F., Shannon L., Odin II - An Open-source Verilog HDL Synthesis tool for CAD Research, Proc.of the IEEE Symposium on Field-Programmable Custom Computing Machines, 2010, P. 149-156
- [13] URL: <http://people.eecs.berkeley.edu/~alanmi/abc/> (access date: 15.07.2022)
- [14] Kahng A. B., Wang L., Xu B., TritonRoute: The Open Source Detailed Router, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, doi:10.1109/TCAD.2020.3003234.
- [15] Kahng A. B., Wang L., Xu B., The Tao of PAO: Anatomy of a Pin Access Oracle for Detailed Routing, Proc. ACM/IEEE Design Automation Conf., 2020, P. 1-6.