

Научная статья

Статья в открытом доступе

УДК 621.3.049.771

doi: 10.30987/2658-6436-2026-2-10-20

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНЫХ РЕШЕНИЙ ДЛЯ АВТОМАТИЗАЦИИ РАСПОЗНАВАНИЯ ТОПОЛОГИИ ИНТЕГРАЛЬНЫХ СХЕМ

Алексей Алексеевич Малаханов¹, Кристина Максимовна Азарова^{2✉},
Кирилл Андреевич Моисеев³

^{1, 2, 3} Брянский государственный технический университет, г. Брянск, Россия

¹ malakhan@yandex.ru, <https://orcid.org/0000-0003-2512-8206>

² kristina.azarova04@mail.ru, <https://orcid.org/0009-0003-3057-9156>

³ yiffik@yandex.ru, <https://orcid.org/0009-0005-6941-0440>

Аннотация. Целью исследования является сравнительный анализ коммерческого и авторского программных средств для автоматизированного распознавания элементов топологии интегральных микросхем в задачах обратного проектирования. Задачей является определение специфических профилей ошибок каждого из рассматриваемых инструментов (коммерческого программного комплекса Pix2Net и авторского кода на языке Python/OpenCV) и формулирование рекомендаций по их улучшению. Методы исследования: экспериментальное распознавание элементов микросхем, изготовленных по БиКДМОП техпроцессу; анализ ошибок по методу Парето с классификацией на три категории (пропуск элемента, ложное срабатывание, геометрическая ошибка). Новизна работы состоит в сопоставлении нейросетевого и шаблонного подходов к распознаванию элементов топологии интегральных схем с выявлением компромисса между полнотой обнаружения и точностью детекции. Результаты исследования: Pix2Net обеспечивает точность 98 %, но порождает 78,9 % ложных срабатываний; авторский подход при точности 72 % демонстрирует доминирование пропусков (60 %) при низком уровне ложных детекций. Выводы: каждый подход обладает специфическим профилем ошибок; анализ Парето позволяет целенаправленно определить приоритетные направления улучшения, при этом авторский инструмент является перспективным для учебных и исследовательских целей благодаря отсутствию зависимости от GPU и коммерческих лицензий.

Ключевые слова: реверс-инжиниринг, интегральные схемы, распознавание образов, Pix2Net, OpenCV, СЭМ-изображения, netlist, GDSII, анализ Парето, БиКДМОП

Для цитирования: Малаханов А.А., Азарова К.М., Моисеев К.А. Сравнительный анализ программных решений для автоматизации распознавания топологии интегральных схем // Автоматизация и моделирование в проектировании и управлении. 2026. №2 (32). С. 10-20. doi: 10.30987/2658-6436-2026-2-10-20.

Original article

Open Access Article

COMPARATIVE ANALYSIS OF SOFTWARE SOLUTIONS FOR AUTOMATED RECOGNITION OF INTEGRATED CIRCUIT TOPOLOGY

Aleksey A. Malakhanov¹, Kristina M. Azarova^{2✉}, Kirill A. Moiseev³

^{1, 2, 3} Bryansk State Technical University, Bryansk, Russia

¹ malakhan@yandex.ru, <https://orcid.org/0000-0003-2512-8206>

² kristina.azarova04@mail.ru, <https://orcid.org/0009-0003-3057-9156>

³ yiffik@yandex.ru, <https://orcid.org/0009-0005-6941-0440>

Abstract. The aim of this study is a comparative analysis of commercial and proprietary software tools designed for the automated recognition of integrated circuit (IC) topology elements in reverse engineering tasks. The objective is to determine the specific error profiles of each tool under consideration, namely the commercial Pix2Net software package and an author-developed Python/OpenCV code and to formulate recommendations for their improvement. The research methods include experimental recognition of IC elements fabricated using the BiCDMOS process; error analysis based on the Pareto principle with classification into three categories, namely element omission, false positive detection, and geometric error. The novelty of the work lies in comparing neural network-based and template-matching approaches

to recognizing IC topology elements, revealing a trade-off between detection completeness and accuracy. The research results show that Pix2Net achieves an accuracy of 98% but generates 78.9% false positives, whereas the proprietary approach demonstrates an accuracy of 72% with omissions dominating at 60% while maintaining a low level of false detections. The conclusions indicate that each approach possesses a specific error profile. Pareto analysis allows for targeted identification of priority areas for improvement. Furthermore, the proprietary tool proves promising for educational and research purposes due to its independence from GPU hardware and commercial licenses.

Keywords: reverse engineering, integrated circuits, pattern recognition, Pix2Net, OpenCV, SEM images, netlist, GDSII, Pareto analysis, BiCDMOS

For citation: Malakhanov A.A. Azarova K.M., Moiseev K.A. Comparative Analysis of Software Solutions for Automated Recognition of Integrated Circuit Topology. Automation and modeling in design and management, 2026, no. 2 (32). pp. 10-20. doi: 10.30987/2658-6436-2026-2-10-20.

Введение

Реверс-инжиниринг интегральных схем (ИС) или обратное проектирование ИС, представляет собой комплекс методов, направленных на восстановление структуры и функциональности микроэлектронных устройств по их физическим образцам. Актуальность методов обусловлена возрастающей потребностью в обеспечении аппаратной безопасности, независимой верификации проектных решений, а также в восстановлении документации на устаревшие компоненты, снятые с производства [1]. Современные ИС могут содержать миллиарды транзисторов, поэтому их «ручной» анализ становится крайне трудоёмким. В связи с этим особую значимость приобретают автоматизированные методы анализа топологии и восстановления схмотехнической структуры ИС [2].

Для обоснования значимости автоматизации анализа топологии целесообразно обратиться к уже существующим подходам к поиску скрытых радиоканалов на кристалле ИС. В работе [3] показано, что задача идентификации блока радиоканала может решаться в рамках трёх основных подходов: сопоставления с эталонной базой топологических образов, классификации по конструкционным, схмотехническим и топологическим признакам, а также обратного проектирования с восстановлением принципиальной схемы. Рассмотрение этих подходов позволяет задать методическую основу, в рамках которой анализируемая система автоматизированного распознавания элементов топологии может быть интерпретирована как частный случай одного из направлений реверс-инжиниринга и как возможный компонент более общей процедуры восстановления структуры ИС.

Типовой рабочий процесс реверс-инжиниринга включает послойное препарирование кристалла, съёмку каждого топологического слоя при помощи сканирующего электронного микроскопа (СЭМ), получение оптического образа топологического слоя или всей топологии целиком, сшивку фрагментов в мозаику, сегментацию и экстракцию полигональных контуров, идентификацию стандартных ячеек и генерацию «электрического списка цепей» (*netlist*) [4]. На каждом этапе возникают специфические трудности: артефакты съёмки, смещения между слоями, сложность выделения границ элементов на фоне шумов.

Среди существующих средств автоматизированного распознавания элементов топологии ИС можно выделить как коммерческие решения, в частности *Pix2Net* (MITRE) [5], *Degate* [6] и инструментарий *TechInsights* [7], так и академические разработки, основанные на методах глубокого обучения [8]. Коммерческие программные продукты, как правило, характеризуются закрытостью исходного кода и высокой стоимостью, тогда как академические подходы нередко требуют значительных объёмов размеченных данных для обучения. В связи с этим целью настоящей статьи является сравнительный анализ двух подходов к распознаванию элементов топологии ИС – коммерческого комплекса *Pix2Net* и разработанного авторского инструмента на базе *OpenCV*, который может быть развёрнут на стандартном вычислительном оборудовании без применения дорогостоящих графических ускорителей.

Материалы, модели, эксперименты и методы

Теоретические основы. Основным объектом исследования и последующей обработки являются высокодетализированные изображения послойных срезов кристалла и топологии технологических слоёв интегральных микросхем, полученные методом сканирующей электронной микроскопии (СЭМ). Сканирующая электронная микроскопия обеспечивает увеличение до 40 000 крат при площадном сканировании и до 1 000 000 крат при локальной съёмке, визуализируя транзисторы вплоть до норм 7 нм [9]. Для преобразования растровых СЭМ-изображений в векторную форму применяются методы бинарной сегментации – пороговая обработка (метод Оцу), адаптивная бинаризация и морфологические операции открытия и закрытия [10]. Полученные бинарные маски служат основой для экстракции полигонов, описывающих геометрию элементов и структур интегральных микросхем.

Одним из перспективных направлений в задачах идентификации элементов топологии является использование нейросетевых классификаторов, адаптированных к специфике полутоновых изображений ИС. Например, в [11] предложена модификация неоконгитрона – нечёткая нейронная сеть, способная идентифицировать объекты топологии на изображениях с искажениями яркости и смещения. Экспериментальное тестирование на реальных топологиях микросхем показало среднюю точность идентификации 98,3 %, что на 16,4 % выше по сравнению с базовым неоконгитроном [11].

Свёрточные нейросети, в частности архитектура *U-Net*, показали конкурентные результаты при сегментации СЭМ-изображений полупроводниковых структур, достигая *IoU* свыше 99 % [12]. Развиваются и методы неконтролируемой сегментации, не требующие ручной разметки [13]. Для идентификации повторяющихся элементов используется шаблонное сопоставление методом нормализованной взаимной корреляции (*NCC*), обеспечивающее инвариантность к изменениям общей яркости и контраста [14]. Устойчивость к повороту достигается перебором дискретных углов или тензорными методами [15].

Результатом распознавания является *netlist* – формальное описание электрической связности схемы. Для передачи геометрических данных на этапах проектирования и производства ИС используется формат *GDSII* – отраслевой стандарт представления планарной геометрии ИС [16]. Формирование корректного *netlist* из полигонов представляет задачу графового сопоставления, для решения которой разработаны алгоритмы на основе кодов связности транзисторов.

Программный комплекс Pix2Net. *Pix2Net v2.0 (MicroNet Solutions / MITRE)* представляет собой коммерческий интегрированный программный комплекс, предназначенный для реверс-инжиниринга интегральных схем на основе СЭМ- и *GDS*-изображений [17]. Его функциональные возможности включают сшивку СЭМ-фрагментов, выравнивание слоёв, экстракцию полигонов с использованием как классических фильтров, так и нейросетевых методов, идентификацию стандартных ячеек по встроенной библиотеке, генерацию *netlist* и экспорт результатов в формат *GDSII* [18]. Как правило, данный комплекс разворачивается на производительных рабочих станциях; например, в качестве рабочей конфигурации может использоваться система на базе *Intel Core i5-9300HF*, 16 ГБ оперативной памяти и видеокарты *NVIDIA GeForce GTX 1650*. При этом существенным требованием для работы нейросетевого модуля является поддержка *CUDA 8.0* [19].

Обработка изображений в *Pix2Net* включает три основных этапа, каждый из которых иллюстрируется отдельными рисунками.

Первый этап заключается в импорте послойных оптических изображений ИС, формировании мозаики и геометрическом выравнивании по системе якорных точек. На рис. 1 показано оптическое изображение части кристалла ИС до сшивки для двух слоёв: точечные кружки соответствуют контактам, вытянутые тёмные дорожки – шинам металлизации. Две фотографии, полученные после послойного удаления слоёв, изначально смещены и по-разному растянуты, поэтому простого сдвига одного изображения относительно другого недостаточно для их точного совмещения.

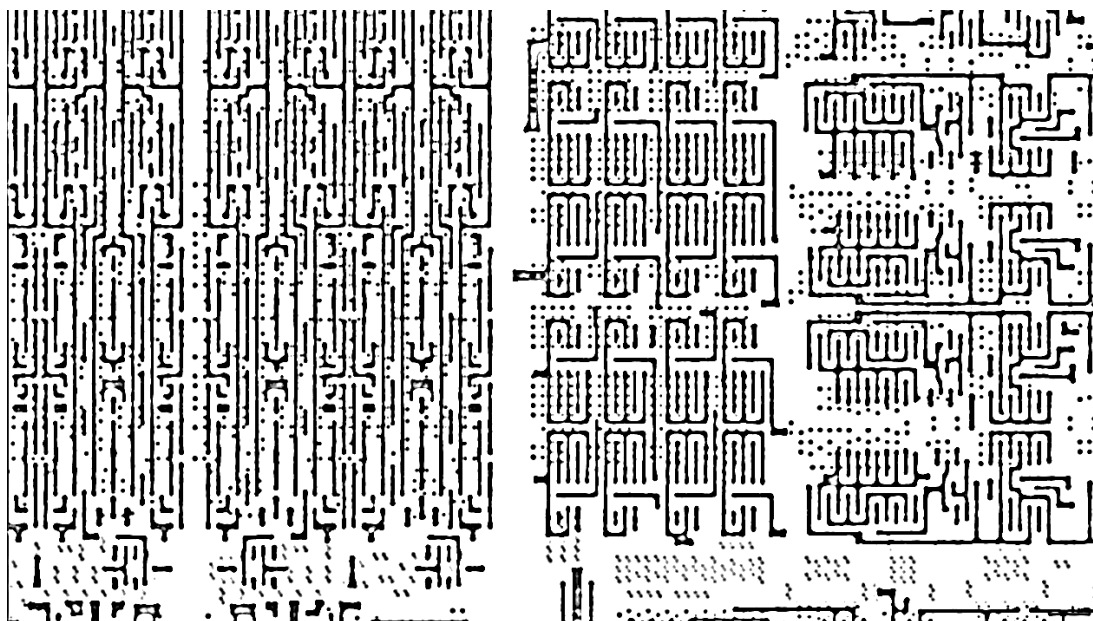


Рис. 1. Оптическое изображение кристалла до сшивки для двух слоёв:
точечные кружки – контакты, вытянутые черные дорожки – шины металлизации

*Fig. 1. Optical image of the crystal before crosslinking for two layers:
dotted circles – contacts, elongated black tracks – metallization tires*

Для коррекции этих искажений используются якорные точки: оператор выбирает на обоих слоях одни и те же опорные фрагменты, и программа деформирует «движущийся» слой (не только сдвигая, но и локально растягивая или сжимая его) так, чтобы соответствующие области совпали. На рис. 2 показан результат такого выравнивания: контактные кружки верхнего слоя совмещены с шинами металлизации нижнего слоя. Качество выравнивания оценивается по величине «процента изгиба» между парами якорей, т.е. по относительному отклонению линии, соединяющей соответствующие точки, от идеальной прямой; этот показатель характеризует остаточные геометрические искажения.

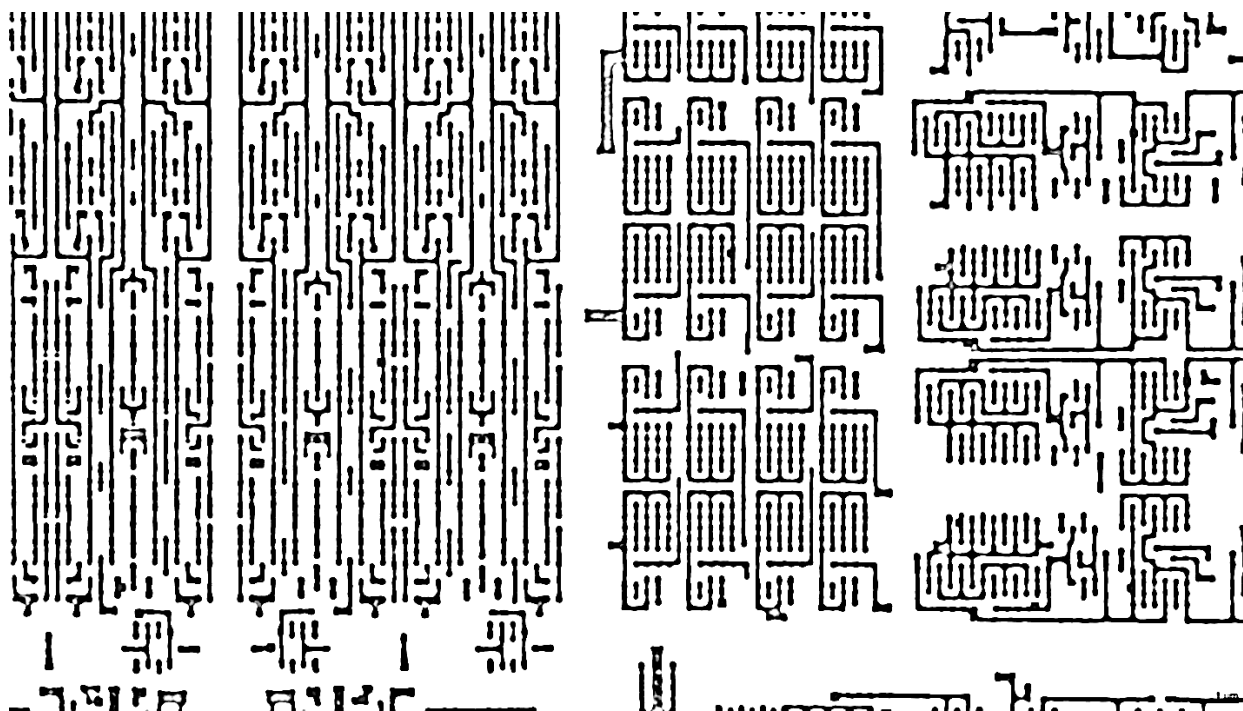


Рис. 2. Выравнивание мозаики по якорным точкам на примере двух слоёв: контактные кружки совмещены с шинами металлизации

Fig. 2. Alignment of the mosaic by anchor points using the example of two layers: the contact circles are aligned with the metallization tires

Второй этап – экстракция полигонов с использованием четырёх категорий фильтров: пиксельных, пороговых, морфологических и нейросетевых. На рис. 3 показан типичный результат обработки: *а* – исходное изображение слоя, *б* – результат применения фильтра, *в* – выделенные полигоны на отфильтрованном изображении, *г* – финальный извлечённый полигон. На рис. 4 приведён пример ручной корректировки: если нейросетевой модуль некорректно определил границы областей, пользователь может удалить лишние фрагменты или дорисовать недостающие участки. Для обучения нейросетевого модуля на практике достаточно разметить 3...4 опорных снимка; за счёт параметров аугментации (до 50 % вариации масштаба, шаг скользящего окна 500 пикселей) обеспечивается устойчивость работы на остальном наборе изображений.

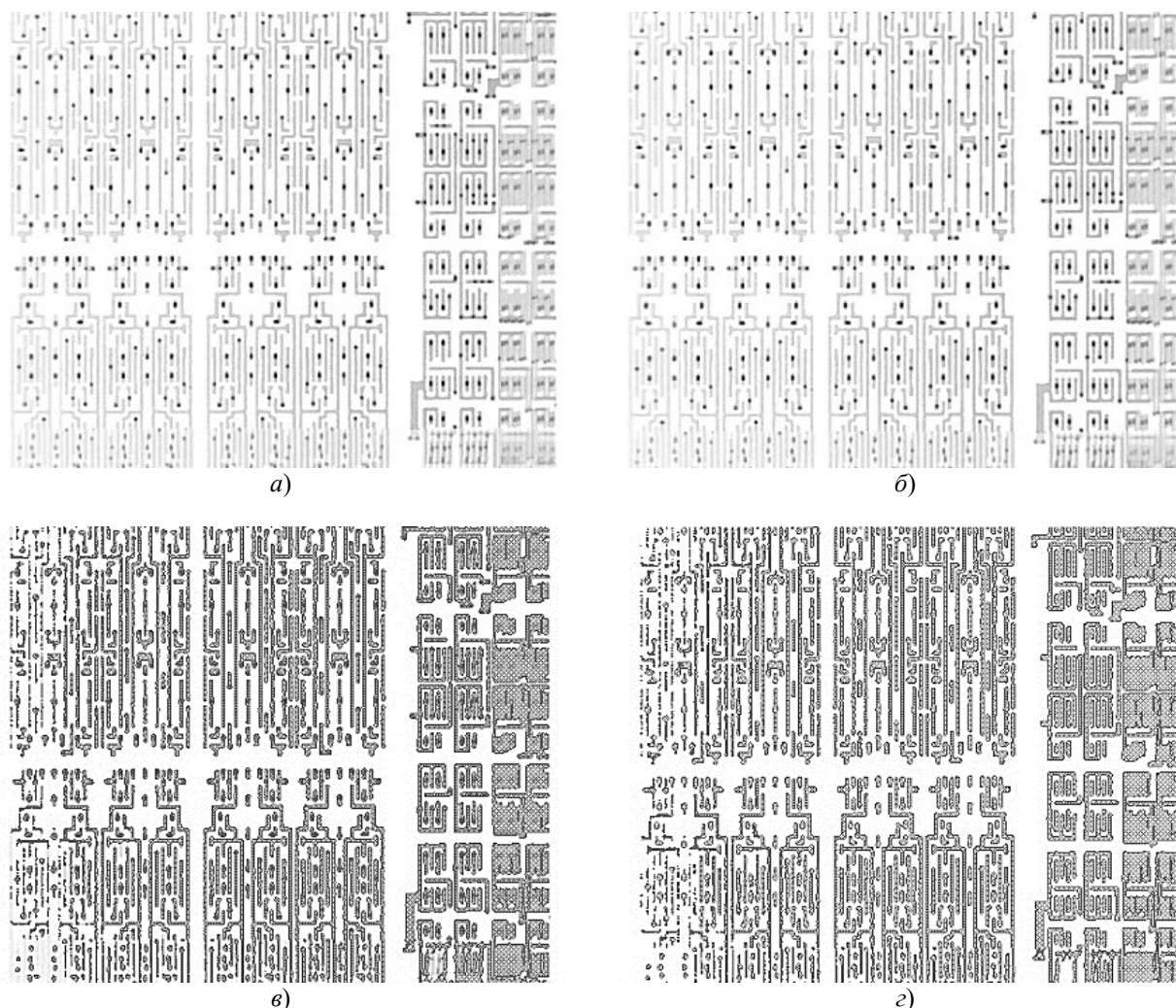


Рис. 3. Результат фильтрации и выделения полигональных контуров топологии:

а – исходное изображение слоя, *б* – результат применения фильтра, *в* – выделенные полигоны на отфильтрованном изображении, *г* – финальный извлечённый полигон

Fig. 3. The result of filtering and highlighting polygonal contours of the topology:

a – the original image of the layer, *b* – the result of applying the filter, *c* – selected polygons in the filtered image, *d* – the final extracted polygon

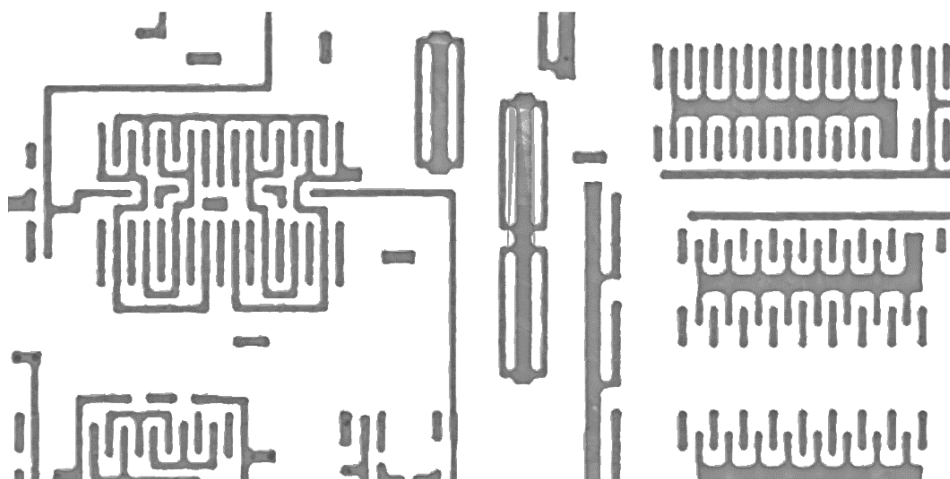


Рис. 4. Ручная корректировка и доработка экстрагированных полигонов
Fig. 4. Manual correction and refinement of extracted polygons

Третий этап предполагает идентификацию стандартных ячеек по библиотеке (инверторная ячейка, логический блок, буфер и др.) с последующей привязкой портов – сопоставлением геометрических контактов на топологии их логическим входам и выходам – и генерацией электрической схемы и *netlist* (рис. 5 – 6).

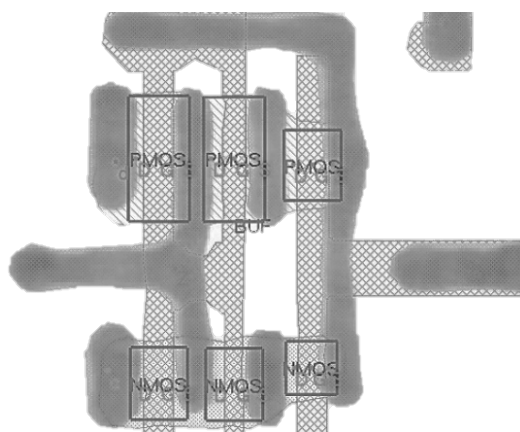


Рис. 5. Пример идентифицированной стандартной ячейки в области топологии
Fig. 5. An example of an identified standard cell in the field of topology

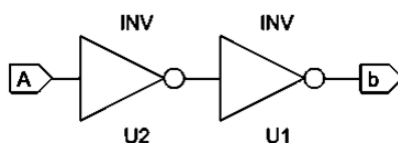


Рис. 6. Фрагмент схемы цифрового блока, автоматически восстановленной из *netlist*
Fig. 6. A fragment of the digital block diagram, automatically restored from the netlist

На рис. 5 показан пример идентифицированной стандартной ячейки в области топологии – инвертирующего буфера. На рис. 6 приведён фрагмент схемы цифрового блока, автоматически сформированный на основе *netlist*, полученного по результатам идентифицированных ячеек. На рис. 7 показан результат работы функции автоматического поиска: по одному выбранному логическому блоку найдены и выделены все аналогичные элементы на кристалле, включая логические элементы «ИЛИ», «ИЛИ-НЕ», «И», «И-НЕ» и буферы.

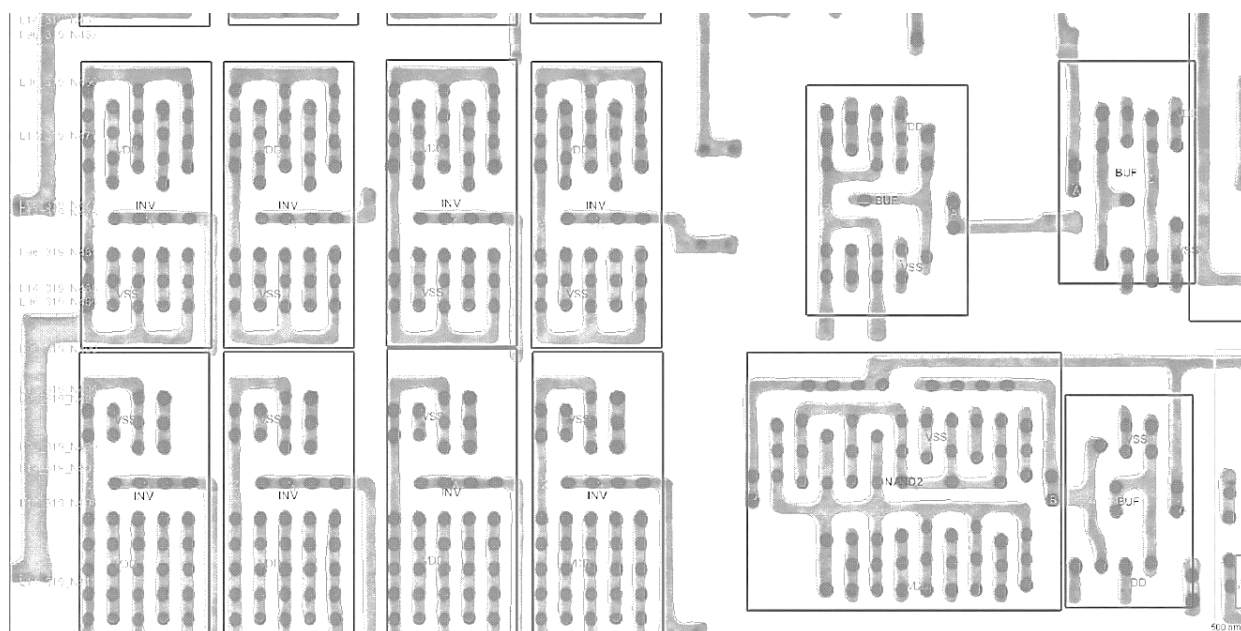


Рис. 7. Автоматически обнаруженные элементы на кристалле
 Fig. 7. Automatically detected elements on a chip

Авторский инструмент распознавания элементов топологии ИС на базе *OpenCV*. Авторами разработан алгоритм распознавания элементов топологии ИС и создана программа с графическим интерфейсом, реализованная на языке *Python 3.11* в среде *Visual Studio Code* [20]. Для функционирования программы используются стандартные библиотеки *Python*: *OpenCV*; *NumPy*; *tkinter*.

OpenCV отвечает за всё, что связано с изображением: открытие файлов, фильтрация, поиск нужных фрагментов. *NumPy* используется для быстрых вычислений с «таблицами» чисел, в которые преобразуется изображение. Библиотека *tkinter* нужна для создания окон программы, кнопок и полей ввода, т.е. для простого графического интерфейса.

Принципиальное отличие разработанной программы от коммерческого комплекса *Pix2Net* заключается в отсутствии необходимости использования графического ускорителя: все вычислительные операции и обработка изображений выполняются средствами центрального процессора. После загрузки изображения слоя ИС программа последовательно выполняет его предварительную обработку, включающую сглаживание для снижения уровня шума, повышение резкости и преобразование в градации серого, что упрощает последующий анализ.

В основе работы авторской программы лежит метод шаблонного поиска с учётом поворота. Пользователь выделяет на изображении фрагмент, который принимается в качестве эталона искомого элемента, например, резистора, транзистора, конденсатора или другого элемента. Далее алгоритм последовательно поворачивает выбранный шаблон на углы от 0° до 330° с шагом 30° и на каждом этапе выполняет поиск сходных фрагментов на исходном изображении. На рис. 8 в качестве примеров представлены результаты распознавания резисторов (рис. 8, а) и транзисторов (рис. 8, б). Для исключения многократного учёта одного и того же объекта близко расположенные совпадения объединяются, после чего каждому подтверждённому элементу автоматически присваивается обозначение, заданное пользователем.

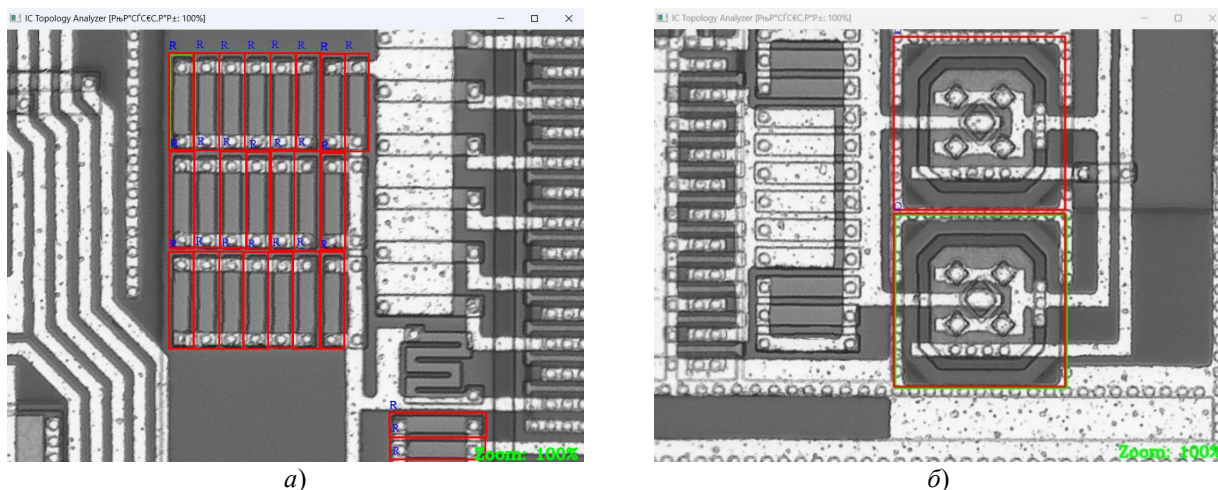


Рис. 8. Результаты поиска элементов на топологии ИС:
a – резисторы, *б* – транзисторы
 Fig. 8. Search results for elements on the IC topology:
a – resistors, *b* – transistors

Дополнительно реализована функция расчёта площадей элементов и оценки сопротивления резисторов по результатам распознавания. Для выделенных резистивных участков программа измеряет их длину и ширину, вычисляет их отношение и умножает на известное поверхностное сопротивление слоя $R_{\square}$, после чего пользователю выводятся рассчитанные значения сопротивления для всех отмеченных резисторов (рис. 9 – 10).

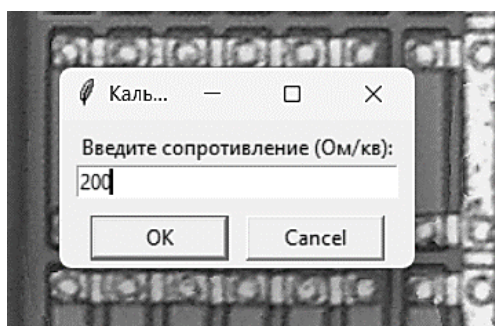


Рис. 9. Окно задания поверхностного сопротивления резистивного слоя $R_{\square}$
 Fig. 9. The window for setting the surface resistance of the resistive layer $R_{\square}$

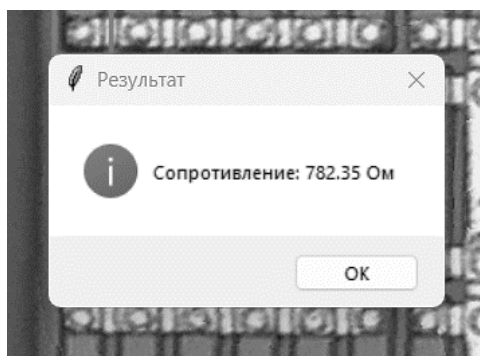


Рис. 10. Окно вывода рассчитанных значений сопротивления выделенных резисторов
 Fig. 10 The output window for the calculated resistance values of the selected resistors

Результаты

Оба инструмента были использованы на одном и том же наборе изображений микросхем, изготовленных по БиКДМОП [21] техпроцессу, для автоматизированного выделения и идентификации характерных элементов топологии. В рамках эксперимента с авторским программным обеспечением была выполнена детальная обработка области с 25-ю

резистивными структурами методом шаблонного сопоставления с учётом поворота; алгоритм уверенно находит элементы в различных ориентациях, однако менее чувствителен к слабоконтрастным объектам.

Для количественной оценки качества распознавания применён анализ Парето – инструмент управления качеством на основе принципа «80/20» [22]. Ошибки классифицированы на три категории: пропуск элемента, ложное срабатывание и геометрическая ошибка.

В авторской программе было зафиксировано 10 ошибок: 6 пропусков (60 %), 2 ложных срабатывания (20 %) и 2 геометрические ошибки (20 %); совокупный вклад пропусков и ложных срабатываний достигает 80 % всех отклонений, т.е. именно эти типы ошибок определяют основное направление доработки. Для *Pix2Net* зарегистрировано 19 ошибок: 15 ложных срабатываний (78,9 %), 3 геометрические ошибки (15,8 %) и 1 пропуск (5,3 %), причём доминирующим фактором становятся избыточные детекции.

Сопоставление профилей ошибок показывает, что два подхода по-разному реагируют на сложные участки изображения: авторский алгоритм на основе порогового шаблонного сопоставления склонен «терять» элементы с пониженным контрастом, тогда как *Pix2Net* с высокочувствительной нейросетевой сегментацией чаще включает в результат шумовые и паразитные структуры. Такая картина хорошо иллюстрирует типичный сдвиг баланса между полнотой обнаружения и избирательностью распознавания в системах компьютерного зрения.

Таблица 1
Table 1

Сравнительная характеристика авторского кода и *Pix2Net*
Comparison of the copyright code and Pix2Net

Критерий	Авторский код	<i>Pix2Net</i>
Точность распознавания	72 %	98 %
Общее число ошибок (25 рез.)	10	19
Доминирующий тип ошибки	Пропуск (60 %)	Ложное срабат. (78,9 %)
Скорость обработки	~30 с	~60 с
Требования к <i>GPU</i>	Нет	<i>CUDA</i> 8.0
Стоимость	Затраты разработчика	Коммерч. лицензия

Заключение

Для авторской программы приоритетом является снижение числа пропусков путём предварительного шумоподавления, понижения порога корреляции и введения многомасштабного поиска (*image pyramid*). Чтобы уменьшить число ложных срабатываний, к обработке изображений имеет смысл добавить ещё один шаг – проверку формы найденных объектов. Для этого для каждого обнаруженного элемента и для эталона можно вычислять так называемые моменты X_u – набор чисел, которые не зависят от положения, поворота и масштаба фигуры. Если значения этих чисел у найденного объекта и у эталонного элемента сильно отличаются, такой объект можно считать ошибочным срабатыванием и исключить из результата. Для *Pix2Net* необходимо ужесточить пороги бинаризации, добавить фильтрацию по площади и соотношению сторон, а также выполнить дообучение нейросети на конкретном технологическом узле.

Прямое сравнение по одной метрике точности не является исчерпывающим. *Pix2Net* требует значительной постобработки для устранения ложных срабатываний, а лицензия и зависимость от устаревшей *CUDA* создают барьеры внедрения. Авторский код легко модифицируется и развёртывается на любом современном ПК, что делает его перспективным для учебных и исследовательских целей.

Проведённое исследование продемонстрировало, что задача распознавания элементов топологии ИС может решаться как средствами коммерческого ПО с нейросетевым ядром, так и при помощи алгоритмов шаблонного сопоставления. Каждый подход обладает специфическим профилем ошибок: *Pix2Net* характеризуется высокой полнотой, но генерирует 78,9 % ложных срабатываний; авторский код склонен к пропуску слабоконтрастных элементов

(60 % ошибок), но демонстрирует низкий уровень ложных детекций.

Проведённый анализ показал, что авторский программный инструмент способен поддерживать базовые операции по распознаванию элементов топологии, но нуждается в дальнейшей доработке. Перспективными направлениями развития являются уточнение настроек порогов и фильтров, расширение набора реализованных функций расчёта и внедрение более продвинутых методов анализа формы и контуров для повышения надёжности распознавания. Дальнейшее совершенствование таких механизмов должно способствовать повышению точности и устойчивости работы программы без увеличения требований к вычислительным ресурсам.

Список источников:

1. Kimura A.G., Elliott A.S., Perkins D.A. Recovery of a hierarchical functional representation of an integrated circuit: pat. US11651126B2 USA. 2023.
2. Dizon-Paradis O., Wilson R., Koblah D. et al. Hands-On Introduction to AI in Hardware Security: IC Reverse Engineering using Image Processing, Computer Vision, and Machine Learning. Florida Institute for National Security, 2025.
3. Лучинин В.В., Гасников А.О. Анализ возможных методов поиска и идентификации блока радиоканала на кристалле интегральной микросхемы // Известия СПбГЭТУ «ЛЭТИ». – 2012. – № 8. – С. 14-19.
4. Lippmann B., Junghanns A. System and method for integrated circuit planar netlist interpretation: pat. US7937678B2 USA. 2011.
5. MITRE. Pix2Net™. URL: <https://www.mitre.org/our-impact/intellectual-property/mitre-pix2net>
6. Bachelot D. Degate: multi-platform software for semi-automatic VLSI reverse engineering of digital logic in chips. URL: <https://www.degate.org>
7. TechInsights. Reverse Engineering Solutions. URL: <https://www.techinsights.com/capabilities/reverse-engineering>
8. Rothaug L., Klix J. et al. Towards Unsupervised SEM Image Segmentation for IC Layout Extraction. Proc. Conf. on Hardware Security. 2023.
9. iST. How to use SEM full-vision imaging technology to reverse-perceive a nano-scale IC circuit. 2018. URL: https://www.istgroup.com/en/tech_20181120/
10. Nahum T. et al. Automatic Defect Segmentation by Unsupervised Anomaly Learning. Proc. 29th IEEE ICIP. 2022. P. 306-310.
11. Дудкин А.А. Нечёткая нейронная сеть для идентификации объектов на изображениях топологических слоёв интегральных микросхем // Системный анализ и прикладная информатика. – 2015. – № 4. – С. 11-15.
12. Azimi R., Kong Y. et al. SEMU-Net: A Segmentation-Based Corrector for Fabrication Process Variations of Nanophotonics with Microscopic Images. Proc. IEEE/CVF WACV. 2025.
13. Almira J.M., Phelippeau H., Martinez-Sanchez A. Fast Normalized Cross-Correlation for Template Matching with Rotations. J. Applied Math. and Computing. 2024. DOI: 10.1007/s12190-024-02157-6.

References:

1. Kimura A.G., Elliott A.S., Perkins D.A. Recovery of a Hierarchical Functional Representation of an Integrated Circuit. Patent US No. 11651126B2; 2023.
2. Dizon-Paradis O., Wilson R., Koblah D. et al. Hands-On Introduction to AI in Hardware Security: IC Reverse Engineering Using Image Processing, Computer Vision, and Machine Learning. Florida Institute for National Security; 2025.
3. Luchinin V.V., Gasnikov A.O. Analysis of the Possible Methods for Search and Identification of the Unit Crystal Radio Channel on the Integrated Circuits. Proceedings of Saint Petersburg Electrotechnical University. 2012;8:14-19.
4. Lippmann B., Junghanns A. System and Method for Integrated Circuit Planar Netlist Interpretation. Patent US No. 7937678B2; 2011.
5. MITRE. Pix2Net™ [Internet]. Available from: <https://www.mitre.org/our-impact/intellectual-property/mitre-pix2net>.
6. Bachelot D. Degate: Multi-Platform Software for Semi-Automatic VLSI Reverse Engineering of Digital Logic in Chips [Internet]. Available from: <https://www.degate.org>
7. TechInsights. Reverse Engineering Solutions [Internet]. Available from: <https://www.techinsights.com/capabilities/reverse-engineering>
8. Rothaug L., Klix J. et al. Towards Unsupervised SEM Image Segmentation for IC Layout Extraction. Proc. Conf. on Hardware Security; 2023.
9. iST. How to Use SEM Full-Vision Imaging Technology to Reverse-Perceive a Nano-Scale IC Circuit [Internet]. 2018. Available from: https://www.istgroup.com/en/tech_20181120/
10. Nahum T. et al. Automatic Defect Segmentation by Unsupervised Anomaly Learning. Proc. of 29th IEEE ICIP. 2022. p. 306-310.
11. Doudkin A.A. Fuzzy Neural Network for Object Identification in Integrated Circuit Layouts. Systems Analysis and Applied Informatics Science. 2015;4:11-15.
12. Azimi R., Kong Y. et al. SEMU-Net: A Segmentation-Based Corrector for Fabrication Process Variations of Nanophotonics with Microscopic Images. Proc. IEEE/CVF WACV; 2025.
13. Almira J.M., Phelippeau H., Martinez-Sanchez A. Fast Normalized Cross-Correlation for Template Matching with Rotations. J. Applied Math. and Computing. 2024. doi: 10.1007/s12190-024-02157-6.

14. OpenCV Documentation. Template Matching. URL: https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html

15. Rajarathnam R.S., Lin Y., Jin Y., Pan D.Z. ReGDS: A Reverse Engineering Framework from GDSII to Gate-level Netlist. Proc. IEEE HOST. 2020.

16. LayoutEditor Documentation. GDSII: The Standard for IC Layout Data. URL: <https://layouteditor.org/layout/file-formats/gdsii>

17. MicroNet Solutions. Pix2Net Suite: Reverse Engineering Software. URL: <http://micronetsol.net>

18. Zhang Y. et al. A rotation invariant template matching algorithm based on Sub-NCC. Math. Biosci. Eng. – 2022. – Vol. 19. – No. 9. – P. 9505-9519.

19. NVIDIA. CUDA Toolkit 8.0 Release Notes. 2016. URL: <https://developer.nvidia.com/cuda-toolkit-archive>

20. Visual Studio Code: редактор исходного кода [компьютерная программа] / Microsoft Corporation. – Режим доступа: <https://code.visualstudio.com>

21. Белоус А.И., Ефименко С.А., Турцевич А.С. Полупроводниковая силовая электроника. – М.: Техносфера, 2013. – 216 с. – ISBN 978-5-94836-367-7. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/73530> (дата обращения: 13.03.2026).

22. Pareto Charts in Quality Control: A Snapshot. Quality Magazine. 2023.

14. OpenCV Documentation. Template Matching [Internet]. Available from: https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html

15. Rajarathnam RS, Lin Y, Jin Y, et al. ReGDS: A Reverse Engineering Framework from GDSII to Gate-Level Netlist. Proc. IEEE HOST; 2020.

16. LayoutEditor Documentation. GDSII: The Standard for IC Layout Data [Internet]. Available from: <https://layouteditor.org/layout/file-formats/gdsii>

17. MicroNet Solutions. Pix2Net Suite: Reverse Engineering Software [Internet]. Available from: <http://micronetsol.net>

18. Zhang Y, et al. A Rotation Invariant Template Matching Algorithm Based on Sub-NCC. Math. Biosci. Eng. 2022;19(9):9505-9519.

19. NVIDIA. CUDA Toolkit 8.0 Release Notes [Internet]. 2016. Available from: <https://developer.nvidia.com/cuda-toolkit-archive>

20. Visual Studio Code: The Open Source AI Code Editor. Computer Program [Internet]. Microsoft Corporation. Available from: <https://code.visualstudio.com>

21. Belous A.I., Efimenko S.A., Turtsevich A.S. Semiconductor Power Electronics. Moscow: Tekhnosfera [Internet]. 2013 [cited 2026 Mar 13]. Available from: <https://e.lanbook.com/book/73530>

22. Pareto Charts in Quality Control: A Snapshot. Quality Magazine; 2023.

Информация об авторах:

Малаханов Алексей Алексеевич

кандидат технических наук, доцент, руководитель учебного дизайн-центра (УДЦ) «РИИРИЭКБ», заведующий кафедрой «Электронные, радиоэлектронные и электротехнические системы» Брянского государственного технического университета.

Азарова Кристина Максимовна

инженер УДЦ «РИИРИЭКБ».

Моисеев Кирилл Андреевич

инженер УДЦ «РИИРИЭКБ».

Information about the authors:

Malakhanov Aleksey Alekseevich

Candidate of Technical Sciences, Associate Professor, Head of the Educational Design Centre (EDC) Development, Testing and Reverse Engineering of Electronic Component Base, Head of the Department of Electronic, Radio-Electronic, and Electrical Systems of Bryansk State Technical University.

Azarova Kristina Maksimovna

Engineer of EDC Development, Testing and Reverse Engineering of Electronic Component Base.

Moiseyev Kirill Andreevich

Engineer of EDC Development, Testing and Reverse Engineering of Electronic Component Base.

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.

Contribution of the authors: the authors contributed equally to this article.

Авторы заявляют об отсутствии конфликта интересов.

The authors declare no conflicts of interests.

Статья поступила в редакцию 18.03.2026; одобрена после рецензирования 14.04.2026; принята к публикации 20.04.2026.

The article was submitted 18.03.2026; approved after reviewing 14.04.2026; accepted for publication 20.04.2026.