

DOI: 10.15514/ISPRAS-2023-35(5)-10



Применение нейронных сетей для сегментации изображений в задаче быстрой трассировки интегральных схем

Р.А. Соловьев, ORCID: 0000-0003-0312-452X <roman.solovyev.zf@gmail.com>

Т.М. Кадирлиев, ORCID: 0009-0004-7872-5164 <perrysquaredz@gmail.com>

Д.В. Тельпухов, ORCID: 0000-0001-9551-0748 <telpukhov@ippm.ru>

*Институт проблем проектирования в микроэлектронике РАН (ИППМ РАН),
124365 Москва, Зеленоград, ул. Советская, дом 3*

Аннотация. В работе исследуются возможности применения нейросетевых методов для решения задачи глобальной трассировки интегральных схем. Разработан алгоритм генерации обучающей выборки на основе волнового алгоритма Ли, позволяющий синтезировать трехмерные матрицы с препятствиями и точками, которые нужно соединить. Для обучения выбрана полносверточная нейронная сеть U-Net, эффективная для семантической сегментации изображений. Проведена оценка качества результатов на тестовой выборке. Показано значительное сокращение времени трассировки по сравнению с волновым методом, однако доля маршрутов без разрывов составила лишь 37%. Предложены пути улучшения обучающей выборки и адаптации подхода под реальные условия с использованием файлов DEF и GUIDE. В работе продемонстрирован потенциал нейросетевых методов для ускорения задачи трассировки, однако требуется продолжение исследований для повышения качества и надежности результатов. Работа полезна для специалистов в области проектирования интегральных схем и машинного обучения.

Ключевые слова: машинное обучение; нейронные сети; волновой алгоритм; глобальная трассировка;

Для цитирования: Соловьев Р.А., Кадирлиев Т.М., Тельпухов Д.В. Применение нейронных сетей для сегментации изображений в задаче быстрой трассировки интегральных схем. Труды ИСП РАН, том 35, вып. 5, 2023 г., стр. 145–156. DOI: 10.15514/ISPRAS-2023-35(5)-10.

Application of Neural Networks for Image Segmentation in the Problem of Fast Global Routing

R.A. Soloviev, ORCID: 0000-0003-0312-452X <roman.solovyev.zf@gmail.com>

T.M. Kadirliev, ORCID: 0009-0004-7872-5164 <perrysquaredz@gmail.com>

D.V. Telpukhov, ORCID: 0000-0001-9551-0748 <telpukhov@ippm.ru>

*Institute of Design Problems in Microelectronics, RAS (IPPM RAS),
3, Sovetskaya st., Moscow, Zelenograd, 124365, Russia.*

Abstract. The paper explores the possibilities of using neural network methods to solve the problem of global routing for VLSI ASIC design. An algorithm has been developed for generating a training dataset based on the Lee algorithm, which allows one to synthesize three-dimensional matrices with obstacles and points that need to be connected. The U-Net fully convolutional neural network, effective for semantic segmentation of images, was selected for training. The quality of the results was assessed using a validation data. A significant reduction in routing time compared to the Lee algorithm was shown, but the share of unbroken routes was only 37%. Ways to improve the training dataset and adapt the approach to real conditions using DEF and GUIDE files are

proposed. In general, the work demonstrated the potential of neural network methods to speed up the global routing task, but continued research is required to improve the quality and reliability of the results. The work is useful for specialists in the field of integrated circuit design and machine learning.

Keywords: machine learning; neural networks; lee algorithm; global routing.

For citation: Soloviev R.A., Kadriev T.M., Telpukhov D.V. Application of Neural Networks for Image Segmentation in the Problem of Fast Global Routing *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 5, 2023. pp. 145-156 (in Russian). DOI: 10.15514/ISPRAS-2023-35(5)-10.

1. Введение

Возрастающая сложность современных цифровых интегральных схем обуславливает новые задачи и повышающиеся требования в области средств автоматизации проектирования. Одним из способов улучшения качества получаемых результатов является применение новых быстрых алгоритмов и моделей, которые позволяют сократить время проектирования, а значит – открывают возможности для более широкого исследования пространства возможных решений поставленной задачи.

Одним из наиболее трудоемких этапов процесса проектирования интегральных схем является трассировка – этап, следующий за размещением элементов схемы и заключающийся в соединении полученных логических элементов проводниками. Эффективная и быстрая трассировка критически важна для обеспечения корректной работы проектируемого устройства.

Учитывая, что на трассировку приходится значительная доля временных затрат всего процесса физического синтеза, актуальной задачей представляется разработка подходов, позволяющих ускорить данный этап. Одним из перспективных направлений является применение методов машинного обучения, базирующихся на выявлении зависимостей между входными и выходными данными.

В данной статье предлагается подход к созданию и оценке качества работы нейронной сети, способной имитировать работу волнового алгоритма трассировки. Полученные результаты позволяют оценить потенциал использования предложенного подхода для ускорения процесса трассировки интегральных схем в реальных маршрутах проектирования.

2. Реализация волнового алгоритма

В качестве базового алгоритма трассировки в данном исследовании был выбран волновой алгоритм ввиду двух его ключевых преимуществ: во-первых, данный алгоритм гарантирует нахождение пути между двумя заданными точками (при условии его существования); во-вторых, найденный им путь является кратчайшим. Указанные свойства являются критически важными для формирования корректного набора данных, необходимого для обучения нейронной сети, несмотря на то, что волновой алгоритм не относится к числу наиболее высокоскоростных [1].

На начальном этапе алгоритма формируется трехмерное пространство поиска, в котором генерируются препятствия и случайным образом выбираются стартовая и финишная ячейки. Далее от стартовой ячейки осуществляется переход в соседние ячейки в рамках трёхмерной окрестности фон Неймана (без учёта диагональных ячеек). При этом проверяется, являются ли данные ячейки свободным пространством и не были ли они помечены как посещённые на предыдущих итерациях алгоритма.

На этапе распространения "волны" в успешно прошедшие проверку ячейки заносится значение, равное номеру текущей итерации (количеству "шагов" от стартовой ячейки). Данные ячейки становятся новыми источниками, относительно которых выполняются последующие итерации описанной проверки. Процесс повторяется до тех пор, пока

очередная распространяющаяся "волна" не достигнет финишной ячейки либо не будет обнаружено свободного пространства для дальнейшего распространения.

Восстановление кратчайшего пути выполняется в обратном направлении: на каждой итерации выбирается соседняя ячейка с маркировкой на единицу меньше текущей, что позволяет последовательно приближаться к стартовой ячейке. Алгоритм завершает работу при достижении стартовой ячейки [1].

Результатом выполнения данного подхода будет являться матрица, содержащая ячейки, обозначающие путь между стартом и финишем. Для простоты визуализации, на рис. 1 представлен пример работы данного подхода в двумерном пространстве.

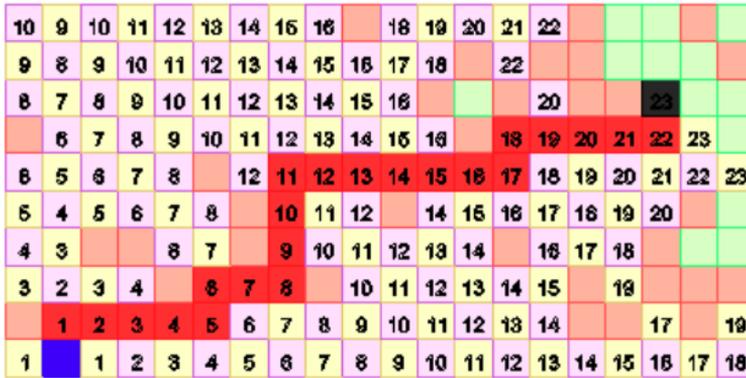


Рис.1. Все шаги волнового алгоритма в двумерном пространстве с выделенным восстановленным путем.

Fig.1. All steps of Lee's algorithm in a two-dimensional space with a highlighted reconstructed path.

3. Создание тренировочного набора данных

3.1 Входные данные

В качестве информации, которая подается на вход нейронной сети, используется специально созданный синтетический датасет, который является набором данных для обучения нейронной сети. Датасет при данном подходе состоит из пар задача-решение, которые используются для построения зависимостей, на основе которых нейронной сетью производятся предсказания [2]. Входом для задачи является трехмерная матрица с размерами 64x64x3 (3 – количество слоев металлизации) на которой размещаются препятствия, а также стартовая и финишная ячейки, которые используются волновым алгоритмом для построения пути.

Алгоритм генерации датасета можно условно разбить на 3 части:

- Создание препятствий;
- Размещение стартовой и финишной ячейки;
- Построение пути.

Генерация препятствий основывается на использовании дождевых матриц (рис. 2). В качестве первого этапа формируются двумерные матрицы заданного размера, в которых в случайно выбранных столбцах генерируются линии различной длины. Для каждого из трех слоев металлизации создаются две такие матрицы: одна с вертикальными линиями и вторая, полученная транспонированием первой и содержащая горизонтальные линии. Затем указанные пары матриц объединяются в единую матрицу, включающую препятствия обоих направлений для данного слоя металлизации.

Так как задачей в данном проекте является матрица $64 \times 64 \times 3$, то аналогичный процесс повторяется для каждого из трех слоев металлизации.

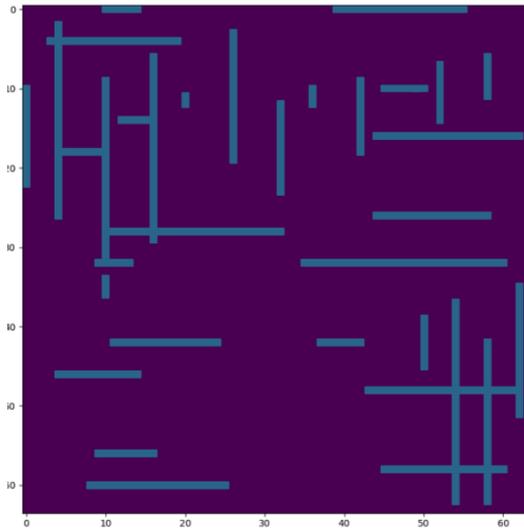


Рис.2. Пример матрицы препятствий для одного слоя металлизации.
Fig.2. Example of a matrix with obstacles in one metallization layer.

Сгенерированные с помощью дождевых матриц препятствия обозначаются значением «-1», чтобы их наличие не нарушало работы волнового алгоритма, который работает только с положительными значениями. По той же причине стартовая и финишная ячейки обозначаются как «-2» и «-3», а ячейки пустого пространства заданы нулями.

Размещение стартовой и финишной ячеек осуществляется путём генерации случайным образом трёх координат (X, Y и Z) в диапазоне размерностей матрицы для каждой ячейки. Далее для выбранных ячеек дополнительно проверяется отсутствие наложения на имеющиеся препятствия, с повторной генерацией координат при необходимости.

Построение пути производится описанным ранее волновым алгоритмом, на вход которого поступает трехмерная матрица с размещенными препятствиями и ячейками старта и финиша. В качестве результата работы алгоритма формируется матрица такой же размерности, заполненная нулевыми значениями. На позиции ячеек, соответствующих координатам пути, найденного с помощью волнового алгоритма, присваивается маркерное значение (в данном случае – "1"). Пример исходной и результирующей матриц показан на рис. 3.

4. Описание архитектуры нейронной сети

4.1 Общие сведения о подходе обучения нейронной сети

В работе использовался подход "обучение с учителем" (supervised learning), заключающийся в построении нейронной сетью функциональной зависимости между входными данными (представляющими задачу) и соответствующими им выходными данными (решениями) [3]. Цель обучения в рамках данного подхода состоит в минимизации среднеквадратичного отклонения между выходами нейронной сети и эталонными решениями по всей совокупности объектов тренировочной выборки. Иными словами, в процессе обучения происходит подбор оптимальных параметров нейронной сети, позволяющих максимально точно аппроксимировать зависимость между входными и выходными данными.

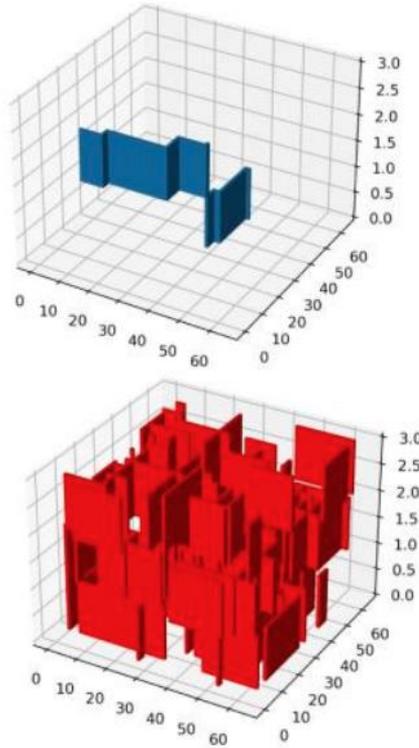


Рис.3. Матрица с маршрутом, построенным волновым алгоритмом (сверху) и исходная матрица с препятствиями (снизу), красным цветом изображены препятствия.

Fig.3. Matrix with the path created by Lee algorithm (top) and initial matrix with the obstacles (bottom), obstacles are marked with red color.

4.2 Архитектура Unet

Сеть Unet (см. рис. 4), является полностью сверточной нейронной сетью, разработанной для биомедицинской сегментации изображений. Ее архитектура основана на парадигме кодировщик-декодировщик с пропускающими соединениями (skip connections), которые позволяют точно локализовать объекты [4].

Архитектура сети Unet состоит из двух основных частей: энкодера и декодера. Энкодер следует типичной структуре сверточной нейронной сети и состоит из повторяющихся сверточных и подвыборочных слоев [4]. Этот путь направлен на захват контекста и извлечение высокоуровневых признаков из входного изображения. С другой стороны, декодер повышает размеры карт признаков для восстановления сегментированного изображения.

Уникальным аспектом сети Unet являются пропускающие соединения, которые объединяют соответствующие слои между сжимающим и расширяющим путями. Эти соединения позволяют сети объединять низкоуровневые и высокоуровневые признаки, обеспечивая точную локализацию и детальную сегментацию, что является важным аспектом для поставленной нами задачи, так как в ее основе лежит взаимодействие с объектами малых размеров.

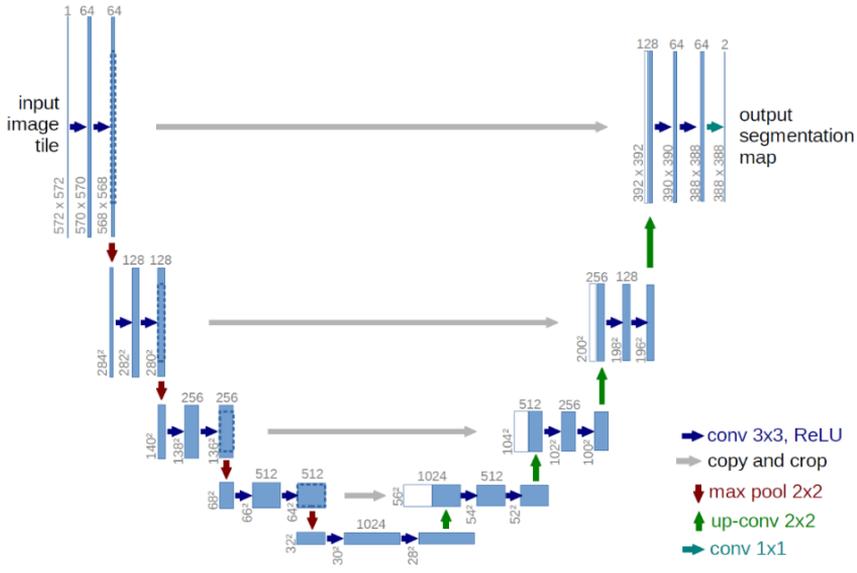


Рис.4. Архитектура Unet.
Fig.4. Unet architecture.

4.3 Поддача данных в Unet

Используя тот факт, что обычные изображения рассматриваются как двумерные матрицы, у которых каждый цвет подается в отдельный входной канал, можно применить схожий подход для датасета описанного в разделе 3.1, но вместо цвета каждый канал Unet будет получать один слой металлизации, как для задачи, так и для решения.

Следовательно, результатом работы нейронной сети будет являться матрица с размерами идентичными исходным и содержащая только предсказание пути, полученное с помощью нейронной сети Unet.

5. Оценка работы нейронной сети

В качестве метрики функции потерь в предложенном методе используется коэффициент игральной кости Dice [5] – одна из наиболее часто применимых метрик оценки результатов сегментации свёрточных нейронных сетей.

$$D = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

В формуле коэффициента значения p_i и g_i являются парами соответствующих значений для предсказанных (predicted) и реальных значений (ground truth). В случае, когда нужно обнаружить путь, данные значения будут либо «0», либо «1» (если пиксель является путем, то значение будет равно 1). Соответственно знаменатель – это сумма всех пикселей пути, как предсказанного, так и решения, соответственно числитель – это сумма корректно предсказанных пикселей, поэтому сумма увеличивается только если оба значения p_i и g_i совпадают (оба равны 1). Следовательно, данный коэффициент лежит в диапазоне от 0 до 1, где 1 – абсолютное совпадение множеств, 0 – отсутствие каких-либо пересечений между множествами. Так как функция потерь должна стремиться к нулю, то вид самой функции потерь, используемой нейронной сетью во время работы будет $1-D$. Соответственно функция валидации будет являться просто самим коэффициентом D , так как она должна показывать процент совпадений между предсказаниями и предоставленными решениями.

6. Тестирование

На этапе тренировки была использована часть датасета размерностью 90 тысяч элементов и функция потерь 1-Dice, а на этапе валидации часть датасета размерностью 10 тысяч элементов и коэффициент Dice.

Также для полученных предсказаний было необходимо провести постобработку, заключающуюся в переопределении значений матриц предсказаний. Необходимо было выбрать порог, превышение которого свидетельствовало бы о наличии пути, а недостижение порога – его отсутствию.

Для тестирования разработанной модели на новом наборе данных помимо проверки на схожесть между предсказанной матрицей и матрицей пути, было необходимо проверить предсказанный путь на разрывы. Это необходимо так как даже при большой корреляции с волновым алгоритмом, отсутствие одной ячейки может означать отсутствие пути между стартовой и финишной ячейкой. Для оценки матриц на разрывы был использован рекурсивный алгоритм:

- 1) Алгоритм инициализируется в стартовой ячейке.
- 2) По аналогии с волновым алгоритмом проверяются 6 соседних ячеек, но не на наличие свободного пространства, а на наличие пути, то есть «1».
- 3) Если такие ячейки найдены, в ячейку, вокруг которой происходил поиск путей, записывается «0» для предотвращения циклического бесконечного вызова функции самой себя.
- 4) В обнаруженных в пункте 2 ячейках, для каждой (если их несколько), вызываются функции проверки 6 соседних ячеек.
- 5) Пункты 2,3,4 повторяются до тех пор, пока не будет достигнута финишная ячейка, что даст на выходе значение «1», либо пока на очередном вызове пункта 2 не будет обнаружено ни одной ячейки – что будет обозначать разрыв пути.

Помимо проверки на разрывы было проведено сравнение длин путей, полученных в результате работы, как волнового алгоритма, так и предсказаний нейронной сети. Для каждого подхода средняя длина была найдена сложением всех единиц во всех матрицах и делением полученного значения на размер датасета.

7. Анализ полученных результатов

Размер датасета, созданного для тренировки нейронной сети равен 100 тысячам элементов с разбиением 90% на тренировочную часть и 10% на валидационную часть. Процесс тренировки нейронной сети длился 40 эпох.

График коэффициента Dice в режиме валидации, то есть средний процент пересечения между предсказанными матрицами и матрицами решения (рис. 5).

Рассмотрим визуализацию успешного предсказания пути с помощью нейронной сети (см. рис. 6).

Также рассмотрим и неудачное предсказание, полученное нейронной сетью (см. рис. 7). На нем можно заметить, как при наличии более комплексных препятствий начинают возникать трудности с построением пути при необходимости прохода через несколько слоев металлизации.

Для оценки работы нейронной сети на предсказание путей без разрывов было создано 3 датасета различной размерности, содержащих 1000, 5000 и 10000 матриц, по ним был вычислен средний процент успешно проложенных путей. Для каждого из них также была проведена оценка на время работы по сравнению с волновым алгоритмом и оценка средней длины путей для всех 16000 матриц, см. табл. 1.

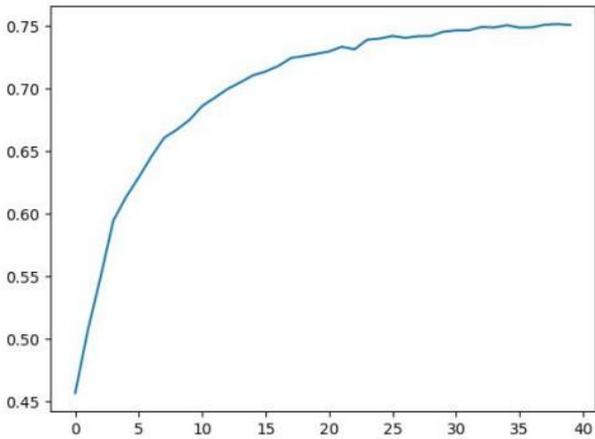


Рис.5. График коэффициента Dice на этапе валидации.
Fig.5. Graph of dice coefficient for validation stage.

По результатам, представленным в таблице, можно сделать вывод, что в среднем скорость работы нейронной сети выше скорости работы волнового алгоритма в 57 раз.

Также можно отметить, что вычислительная сложность волнового алгоритма $O(N^2)$, то есть, чем больше схема, тем в среднем длиннее будут пути и выполнение алгоритма будет занимать гораздо больше времени.

После применения рекурсивного алгоритма на 16 тысяч матриц количество путей без разрывов, то есть корректно построенных путей, было равно 37%.

Средняя длина пути для волнового алгоритма: 45,23.

Средняя длина пути для предсказания: 43,45.

Разница в 4% между длинами пути объясняется наличием разрывов в полученных предсказаниях.

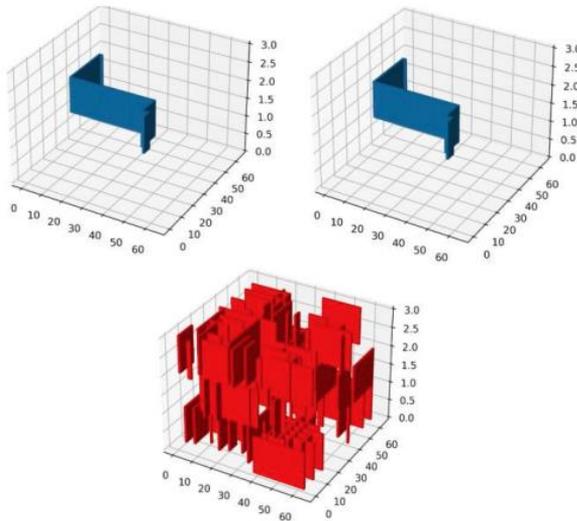


Рис.6. Путь, построенный волновым алгоритмом (слева сверху), пример удачного предсказания (справа сверху) и исходная матрица с препятствиями (снизу), красным цветом изображены препятствия.

Fig.6. Path created by Lee algorithm (top left), example of a successful prediction (top right) and source matrix with obstacles (bottom), obstacles are marked with red color.

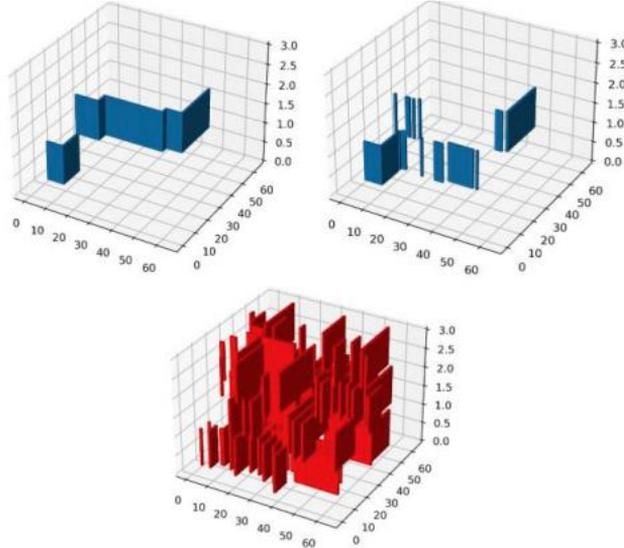


Рис. 7. Путь, построенный волновым алгоритмом (слева сверху), пример неудачного предсказания (справа сверху) и исходная матрица с препятствиями (снизу), красным цветом изображены препятствия.

Fig.7. Path created by Lee algorithm (top left), example of an unsuccessful prediction (top right) and source matrix with obstacles (bottom), obstacles are marked with red color.

Табл. 1. Оценка скорости работы двух методов решения задачи трассировки

Table 1. Evaluation of the completion time for two methods of solving the global routing task

Количество матриц	1000	5000	10000
Время работы волнового алгоритма	140 секунд	728 секунд	1477 секунд
Время работы нейронной сети	3 секунды	12 секунд	23 секунды

8. Возможности практического применения

8.1 Общие сведения об архитектуре OpenLane

Для полноценного применения на практике данную нейронную сеть будет необходимо внедрить в один из существующих открытых маршрутов проектирования интегральных схем. Для этого было выбрано открытое программное обеспечение OpenLane [6].

OpenLane включает в себя множество этапов (см. рис. 8), которые позволяют получить из исходного RTL описания и библиотеки стандартных ячеек, набор данных необходимых для производства интегральных схем (GDSII, LEF и т.п.).

Официальная документация делит работу программы на 7 основных этапов: Synthesis, Floorplaning, Placement, CTS, Routing, Tapeout, Signoff.

Рассмотрим в подробностях три интересующих нас момента:

CTS (Clock Tree Synthesis) – данный этап необходим для равномерного распределения синхросигнала между всеми частями дизайна, что позволяет минимизировать смещение и задержки. Важной для нас деталью здесь является то, что в результате выполнения CTS мы будем иметь состояние интегральной схемы до этапа трассировки.

Routing – этап, включающий в себя глобальную и детальную трассировку, определение паразитных сопротивлений и емкостей, а также оценку задержек и целостности сигналов. Из

данного этапа мы будем извлекать информацию о расположении проводников, полученных в результате работы FastRoute – алгоритма для глобальной трассировки.

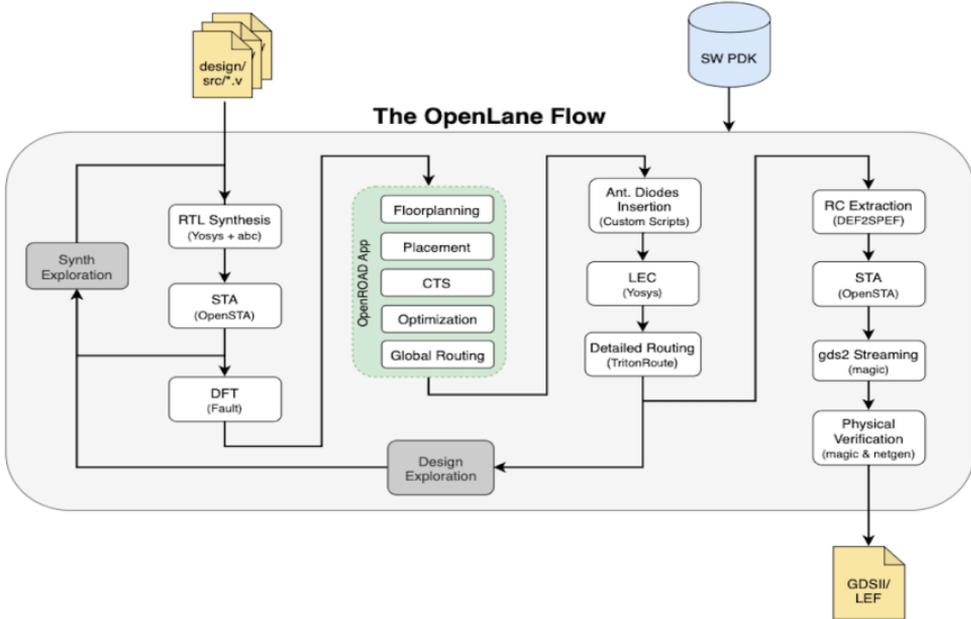


Рис.8. Маршрут синтеза от RTL до GDS – OpenLane.
Fig.8. Synthesis flow from RTL to GDS – OpenLane.

8.2 Выборка файлов

Рассмотрим сами файлы, выбранные нами ранее из различных этапов работы OpenLane. Из CTS будет использоваться DEF (Design Exchange Format) файл – он хранит в себе логическую и физическую информацию о дизайне [7] – то есть внутренние соединения, информацию о группах и контактах, физические ограничения (локация, ориентация и геометрия компонентов). Если посмотреть на то, как выглядят препятствия, сгенерированные дождевыми матрицами (рис. 9), то можно заметить сходства между изображениями.

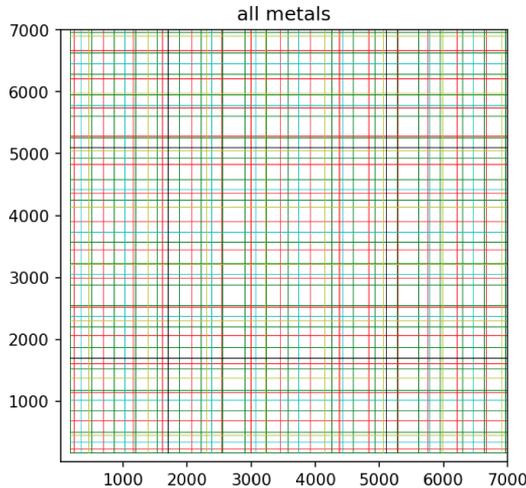


Рис.9. Треки извлеченные из DEF файла для всех слоев металлизации.
Fig.9. Tracks extracted from the DEF file for all metallization layers.

Из Routing будет взят GUIDE файл – в нем содержится информация о путях, проложенных в результате глобальной трассировки. Они разбиты на множество прямоугольных фигур на различных слоях металлизации, и хранятся в виде координат для двух противоположащих углов прямоугольных фигур и слоя металлизации для данной части пути. Данные, полученные из этого файла, будут являться результатом работы программы FastRoute – алгоритма глобальной трассировки, в отличие от рассматриваемого в данной статье волнового алгоритма.

В итоге, рассмотрев информацию, полученную из DEF и GUIDE файлов, можно сделать вывод о том, что подход, который использовался в данной работе, может быть применен и для информации, которую можно получить из реального маршрута проектирования. Для создания датасета с реальными данными будет необходимо провести следующие шаги:

- 1) Выбрать случайную группу контактов, относящихся к одному пути.
- 2) Удалить из матрицы полученной из GUIDE файла путь, относящийся к данным контактам.
- 3) Скомбинировать матрицу с выбранной группой контактов, матрицу с удаленным путем и матрицу с состоянием интегральной схемы до трассировки.
- 4) В результате выполнения этих действий для различных схем, будет сформирован датасет, на основе которого будет обучена модель сегментации на основе UNET.

9. Выводы

Предложенный подход обозначил достоинства и недостатки применения нейронных сетей для задачи глобальной трассировки интегральных схем, в целом показав свою применимость и перспективность. К преимуществам предложенного подхода можно отнести высокое быстродействие уже обученной модели: в 57 раз быстрее по сравнению с методом волнового алгоритма. При этом количество корректно построенных путей оказалось невысоким: всего 37%. Однако решение данной проблемы возможно следующими способами:

- 1) улучшение датасета за счёт расширения методики генерации исходных матриц;
- 2) включение признака разводимости в функцию потерь при обучении модели;
- 3) выбор более эффективной архитектуры нейронной сети.

Анализ современных маршрутов проектирования свидетельствует о возможности адаптации предложенного подхода к реальным открытым маршрутам проектирования, что вкпе с обозначенными возможностями для усовершенствования обуславливает широкие перспективы для использования нейронных сетей в задаче глобальной трассировки интегральных схем.

Список литературы / References

- [1]. F. Rubin, "The Lee Path Connection Algorithm," in IEEE Transactions on Computers, vol. C-23, no. 9, pp. 907-914, Sept. 1974, doi: 10.1109/T-C.1974.224054.
- [2]. James, Gareth & Witten, Daniela & Hastie, Trevor & Tibshirani, Robert. (2013). An Introduction to Statistical Learning: With Applications in R., Springer, New York, NY <https://doi.org/10.1007/978-1-4614-7138-7>.
- [3]. O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint [arXiv:1511.08458](https://arxiv.org/abs/1511.08458).
- [4]. Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28.
- [5]. Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M. (2017). Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In: Cardoso, M., et al. Deep

Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA ML-CDS 2017 2017. Lecture Notes in Computer Science, vol 10553. Springer, Cham. https://doi.org/10.1007/978-3-319-67558-9_28

- [6]. Ghazy, A.A., & Shalan, M. (2020). OpenLANE: The Open-Source Digital ASIC Implementation Flow. in Proc. Workshop on Open-Source EDA Technol. (WOSET), 2020, Art. no. 21.
- [7]. LEF/DEF 5.8 Language Reference. Доступно по ссылке: <https://coriolis.lip6.fr/doc/lefdef/lefdefref/DEFSyntax.html>, дата обращения — апрель 2023.

Информация об авторах / Information about authors

Тимур Маратович Кадирлиев – бакалавр, инженер-исследователь в Институте Проблем Проектирования в Микроэлектронике РАН. Сфера научных интересов: машинное обучение и нейронные сети, работа с большими данными, веб разработка, автоматизация проектирования цифровых СБИС.

Timur Maratovich Kadirliiev – university graduate, engineer-researcher of the Institute of Design Problems in Microelectronics RAS. Research interests: machine learning and neural networks, data science, web development, automation of digital VLSI design.

Дмитрий Владимирович ТЕЛЬПУХОВ – доктор технических наук, заместитель директора по научной работе Института Проблем Проектирования в Микроэлектронике РАН. Сфера научных интересов: автоматизация проектирования цифровых СБИС, логический синтез, радиационно-стойкое проектирования, машинное обучение и нейронные сети, система остаточных классов.

Dmitry Vladimirovich TELPUKHOV – Dr. Sci. (Tech.), Deputy Director for Scientific Work of the Institute of Design Problems in Microelectronics RAS. Research interests: automation of digital VLSI design, logic synthesis, radiation-resistant design, machine learning and neural networks, residual class system. Роман Александрович СОЛОВЬЁВ – чл.-кор. РАН, доктор технических наук, главный научный сотрудник Института Проблем Проектирования в Микроэлектронике РАН. Сфера научных интересов: автоматизация проектирования цифровых СБИС, нейронные сети и машинное обучение, система остаточных классов, физический синтез интегральных схем.

Roman Aleksandrovich SOLOVYEV – Dr. Sci. (Tech.), member-corr. RAS, chief researcher at the Institute of Design Problems in Microelectronics RAS. Area of scientific interests: automation of digital VLSI design, neural networks and machine learning, residue number system, physical synthesis of integrated circuits.