

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра «Системы автоматики, автоматизированное управление
и проектирование»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ С.В. Ченцов
« ____ » _____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА

15.03.04 – Автоматизация технологических процессов и производств

**РЕШЕНИЕ ЗАДАЧИ РАЗМЕЩЕНИЯ ПРИ ЦИФРОВИЗАЦИИ
ПРОЕКТИРОВАНИЯ РАДИОЭЛЕКТРОННЫХ ИЗДЕЛИЙ**

Руководитель	_____	___06.2020 г.	доцент, канд. техн. наук Е.Е. Носкова
Выпускник	_____	___06.2020 г.	И.В. Паршинцев
Нормоконтролер	_____	___06.2020 г.	Т.А. Грудинова

Красноярск 2020

РЕФЕРАТ

Выпускная квалификационная работа по теме «Решение задачи размещения при цифровизации проектирования радиоэлектронных изделий» содержит 67 страниц текстового документа, 61 иллюстрацию, 7 использованных источников.

ПРОГРАММНАЯ ИНТЕГРАЦИЯ, МЕТОД ОБРАТНОГО РАЗМЕЩЕНИЯ, C#.

Объектом разработки является программная интеграция метода обратного размещения с программными продуктами фирмы Cadence.

Цель работы: повышение функциональности систем автоматизированного проектирования печатных плат за счет решения задачи размещения.

Задачи, которые решались в ходе выполнения данной бакалаврской работы:

- 1) Анализ методов и алгоритмов решения задачи размещения при проектировании радиоэлектронных комплексов;
- 2) Анализ функциональных возможностей современных САПР радиоэлектронных устройств при решении задач синтеза конструкций;
- 3) Программная реализация метода размещения при цифровизации проектирования радиоэлектронных комплексов;
- 4) Интеграция метода размещения с продуктами фирмы Cadence.

В результате выполнения выпускной квалификационной работы проведен анализ методов и алгоритмов решения задачи размещения, выполнен обзор программных продуктов фирмы Cadence, получена программная реализация метода обратного размещения с целью расстановки элементов на коммутационном поле.

СОДЕРЖАНИЕ

Введение.....	4
1 Цифровизация проектных работ при решении задач синтеза конструкции	5
1.1 Цифровые технологии проектирования как основа технологий Индустрии 4.0	5
1.1.1 Общее представление технологий индустрии четвертого поколения.....	5
1.1.2 Цифровые технологии проектирования	10
1.2 Задачи синтеза конструкций в рамках этапа конструирования радиоэлектронных изделий.....	12
1.3 Методы и алгоритмы решения задачи размещения при ее цифровизации	16
1.3.1 Критерии размещения	16
1.3.2 Методы размещения	19
2 Разработка программного решения для выполнения задачи размещения	25
2.1 Интеграция программного решения с продуктами линейки Cadence.	25
2.1.1 Разработка принципиальной схемы в САПР OrCAD.....	25
2.1.2 Интеграция с графическим редактором Capture.....	33
2.2 Технологии программной реализации	37
2.2.1 Метод обратного размещения.....	37
2.2.2 Программная реализация задачи размещения	40
3 Применение метода обратного размещения при разработке печатных узлов.....	42
3.1 Разработка печатных узлов в САПР OrCAD.....	42
3.2 Применение метода обратного размещения при разработке печатных узлов	53
Заключение	67
Список использованных источников	68

ВВЕДЕНИЕ

Технологии цифрового проектирования и моделирования радиоэлектронных средств и комплексов на современном этапе входят в состав технологий Индустрии 4.0. Разработка сложных радиоэлектронных изделий на современной элементной базе требует создания программных средств, позволяющих решать проектные задачи в рамках сквозного проектирования на всех уровнях и этапах процесса разработки. При этом цифровые технологии конструирования радиоэлектронных изделий, предлагаемых ведущими мировыми и российскими разработчиками САПР, зачастую не позволяют проектировщику решать задачи конструирования на должном уровне из-за низкого качества получаемых решений. Особенно это касается решения задачи размещения компонентов коммутационной схемы на заданном коммутационном поле при выполнении печатного монтажа.

Современные САПР печатных плат разных производителей не решают автоматически задачу размещения компонентов на заданном коммутационном поле в виде контура печатной платы. Отчасти это связано с тем, что при разработке сложных печатных узлов с большим количеством элементов и связей, что характерно для современной элементной базы, возникает слишком много условий, не поддающихся формализации. Автоматическое размещение не используется в практике проектирования радиоэлектронных изделий на предприятиях региона, занимающихся проектно-производственной деятельностью. Проектировщики занимаются размещением компонентов вручную, несмотря на высокую сложность и трудоемкость этой задачи. Поэтому разработка программных решений, осуществляющих размещение элементов печатного монтажа как в автоматическом, так и в интерактивном режимах, является неотъемлемым шагом в развитии цифровых технологий проектирования радиоэлектронных изделий в настоящее время.

1 Цифровизация проектных работ при решении задач синтеза конструкции

1.1 Цифровые технологии проектирования как основа технологий Индустрии 4.0

1.1.1 Общее представление технологий индустрии четвертого поколения

Четвёртая промышленная революция - переход на автоматизированное цифровое производство, управляемое в режиме реального времени интеллектуальными системами при постоянном взаимодействии с внешней средой, выходящее за границы одного предприятия, с перспективой преобразования в общую промышленную сеть вещей и услуг.

Индустрия четвертого поколения получила свое название от инициативы, возглавляемой бизнесменами, политиками и учеными, которые определили ее в 2011 году как возможность повышения конкурентоспособности германской обрабатывающей промышленности через повышенную интеграцию «киберфизических систем» в заводские процессы. Подобные изменения затрагивают такие грани жизни человека как: рынок труда, жизненная среда, политические системы, технологический уклад, человеческую идентичность.[2]

Четвёртая промышленная революция обычно описывается путем указания ключевых технологий, выход из НИОКР которых планируется в ближайшее время. К данным технологиям причисляют следующие собирательные понятия грядущих технологий:

- Большие данные;
- Интернет вещи;
- 3D-печать;
- Печатная электроника;
- Квантовые вычисления;

- Блокчейн;
- Виртуальная и дополненная реальность.

Большими данными считаются обозначения структурированных и неструктурированных массивов данных огромных объёмов, эффективно обрабатываемых специальными программными инструментами. В широком смысле о «больших данных» говорят, как о этапе, связанном с появлением технологической мощности, способной на анализ данных, сопоставимых с мировым объёмом и вытекающих из этого трансформационных последствий.

Концепция интернет вещей, как вычислительной сети между физическими предметами, оснащёнными технологиями для взаимодействия друг с другом, способная перестроить экономические и общественные процессы, исключая из части действий и операций необходимость участия человека.

Концепция сформулирована еще в 1999 году как осмысление перспектив широкого применения средств радиочастотной идентификации для взаимодействия физических предметов между собой и с внешним окружением. Наполнение концепции многообразным технологическим содержанием и внедрение практических решений для её реализации начиная с 2010-х годов считается устойчивой тенденцией в информационных технологиях, прежде всего, это происходит из-за распространения беспроводных сетей и выполнения облачных вычислений, развитию технологий межмашинного взаимодействия, началу активного перехода на адреса интернет протокола шестого поколения и освоению программно-определяемых сетей.

3D - печатью называется послойное создание детали, использующее цифровую модель прототипа. Изготавливаемым объектом может служить любая вещь, от медицинских протезов и до строительных блоков. В настоящее время распространяется печать несколькими материалами одновременно. Технология имеет большой потенциал, связанный с отсутствием

необходимости организации полноценного технологического процесса изделия. Для создания требуется только цифровая модель изделия.

Область электроники, разрабатывающая реализацию электронных схем с помощью печатного оборудования, называется печатной электроникой. Печатное оборудование наносит токопроводящие и полупроводниковые чернила на поверхность плоской подложки, формирующие в соответствии с электрической схемой активные, пассивные элементы и межэлементные соединения.

Появление данной технологии связано с доступностью получения органических полимеров и наночастиц металлических соединений, которые при определённых условиях способны заменить кремний в электронных и компьютерных технологиях. Их можно добавлять в жидкости, выполняющие функции красок или чернил, которые затем наносятся на подложку и создают устройства, такие как тонкоплёночные транзисторы или резисторы.

Для обычной печати краски наносятся на бумагу, но в случае печатной электроники она является непригодной, связано это с шероховатой поверхностью и быстрым поглощением ею воды. Для печатной электроники принято использовать такие материалы, как пластик, керамика или кремний.

Для самой печати обычно используется типовое печатное оборудование: для трафаретной печати, флексографии, ротогравюрной и офсетной печати, а также струйные принтеры. Специальные чернила наносятся слоями друг на друга, и сильно зависят от согласованного режима работы принтера. Методы печати являются важнейшей задачей данного направления.[6] Выбор требуемого печатного метода определяется свойствами печатных материалов, а также экономическими и техническими соображениями в терминах печатной продукции.

В некоторых областях, таких как производство светодиодов, печатная электроника не влияет на производительность изделий, но большинство устройств, созданных средствами печатной электроники, уступают по характеристикам обычным электронным приборам, однако являются более

экономически выгодными. Данный параметр является наиболее важным преимуществом, когда сравнение происходит для крупномасштабного производства. Прогнозируется, что изделия не требующие высокую производительность электронных схем получат большое распространение из-за очень низкой стоимости. К таким изделиям можно отнести гибкие дисплеи, радиочастотную идентификация, декоративные и анимированные плакаты, активные покрытия.

Привлекательность печатных электроники связана с возможностью реализации новых или более совершенных функциональных технологий: создание многослойных микроструктурированных заготовок более простым и экономически эффективным способом, печать на гибких подложках для создания изогнутого изделия.

За счет явления квантовой механики суперпозиции в квантовых вычислениях используются особые единицы информации – кубиты, имеющие сразу два состояния 0 и 1, в отличие от байта. Благодаря этому является возможным производить обработку всех состояний сразу, получая большое преимущество над привычным компьютером. В настоящее время существует несколько прототипов, исполняющие фиксированные алгоритмы небольшой сложности.

Блокчейном называется непрерывная последовательность блоков информации, выстроенная по определенным правилам. Связь между блоками обеспечивается хеш-суммами, каждый блок содержит информацию о своей хеш-сумме и сумме предыдущего блока. Чтобы изменить информацию, хранящуюся таким образом, необходимо редактировать и все последующие блоки. Цепочки данных в виде блоков обычно хранятся на множестве независимых друг от друга компьютеров. Из-за этого изменение информации становится сильно затруднительным процессом.

Данный термин появился как название полностью реплицированной распределённой базы данных, реализованной в системе «Биткойн», из-за чего блокчейн часто относят к транзакциям в различных криптовалютах, однако

технология цепочек блоков может быть распространена на любые взаимосвязанные информационные блоки.[5] Биткойн стал первым применением технологии блокчейн в октябре 2008 года.

Сейчас же блокчейн находит применение в таких областях, как финансовые операции, идентификация пользователей или создание технологий кибербезопасности. Блокчейн-технологиями интересуются банковские учреждения и государственные организации.

Виртуальная и дополненная реальность – технологии, позволяющие симулировать определенную среду или дополнять естественную через специальные цифровые средства.[4] Самый востребованный на сегодня вариант применения VR (*virtual reality*) и AR (*augmented reality*) технологий в промышленности – обучение персонала и демонстрация информационных моделей. Для компаний со множеством офисов и площадок по всему миру это еще и вопрос унификации производственных операций. Массовое внедрение в производство технологий виртуальной и дополненной реальности в настоящее время могут себе позволить лишь большие компании. Связанно это с потребностью в серьезных инвестициях.

Профильных приложений для промышленности на рынке фактически нет. Чаще всего компаниям приходится заниматься разработкой данных приложений самостоятельно и реализовывать его только внутри себя, не продвигая их на разных цифровых площадках. Так происходит из-за того, что настроенное технологическое обеспечение под определенные задачи сложно изменять, интегрировать и использовать для другого сценария.

При согласовании и демонстрации информационных моделей изделий производственным компаниям становится удобно пользоваться VR и AR технологиями, нежели привычным способом. Связанно это с тем, что разработка и выпуск нового оборудования обращается огромное количество чертежей в виде 3D контента, которые необходимо утверждать. Гораздо проще ссылаться на 3D модель, для которой делаются все чертежи и расчеты. Если она верна, то и все документы по ней верны. Но демонстрация на

обычных мониторах недостаточно информативна. Вдобавок виртуальная реальность позволяет подгрузить в модель несколько участников.

Подобная технология работает безупречно. Недостатки ее в том, что нет инструментов для ее быстрой реализации, современные САПР не производят высокоточные модели, которые позволяли бы достаточно наглядно ее оценить. Из-за этого демонстрационные модели приходится проектировать вручную.

Потенциал применения технологий виртуальной и дополненной реальности в промышленном производстве огромен, однако текущее состояние технологических решений VR и AR, удовлетворяющих в основном потребности индустрии развлечений, накладывает на внедрение их в производство ряд ограничений.

Имеющиеся на рынке устройства пока обладают низким разрешением и низкой автономностью в работе. Отсутствуют механизмы интеграции с инфраструктурой компаний. Нет отработанных процессов конвертации контента из инженерного формата в VR и AR.

Все представленные технологии в скором будущем станут частью жизни человека, как в производстве, так и при повседневных делах, поэтому необходимо подготовить соответствующий конструктив для успешного интегрирования с ними.

1.1.2 Цифровые технологии проектирования

Проектирование технического объекта – представление образа объекта, который еще не существует, в требуемой форме. Инженер формирует данный образ или он создается с помощью некоторых формализованных алгоритмов. В случае выраженной потребности общества в определенных технических объектах проектирование будет являться необходимым и включает в себя разработку технического задания, отражающего эти потребности, и реализацию их в виде проектной документации.[1]

Наиболее частым случаем является предоставление технического

задания в виде определенной группы документов и называется первичным описанием объекта. Результатом проектирования является полный комплект документации, содержащий информацию для создания объекта в заданных условиях. Такая документация называется окончательным описанием объекта.

Во время проектирования появляется ряд описаний, которые подводят итоги решения промежуточных задач и принятия определенных решений. Такие описания называют проектными решениями.

Проектирование, в котором проектные решения являются результатом действий проектировщика на основе формальных и формализованных алгоритмов называют автоматизированным проектированием. Системы, которые обладают интерфейсами для автоматизированной работы проектировщика называют САПР (Системы автоматизированного проектирования).

Цифровое проектирование – одно из актуальнейших направлений современной промышленности, позволяющее осуществлять разработку в минимальные сроки и создавать продукцию нового поколения. Связанно это с возможностью интеграции одной цифровой модели на всех этапах проектирования. Весь разрабатывающий продукт персонал имеет к нему доступ. Основой цифрового проектирования является использование сложных математических моделей, обладающих достаточным уровнем адекватности реальным материалам для того, чтобы производить разработку цифровых двойников продукции и производственных процессов.

Цифровая модель является совокупностью примитивов, каждый из которых обладает определенными уникальными свойствами (плотность, проводимость). Такие модели позволяют производить оценку проектируемой системы без реальных опытов, заменяя их на вычисления определенными математическими методами, так же уменьшая стоимость продукта, и зависящие только от вычислительных ресурсов, которые используются для выполнения данного расчета.

1.2 Задачи синтеза конструкций в рамках этапа конструирования радиоэлектронных изделий

Конструированием электронных устройств называется раздел проектирования, занимающийся разработкой конструкторской документации для создания электронных изделий на производстве. Конструирование начинается по завершению формирования принципиальных схем, полученных на этапе функционального проектирования.

Конструкторское проектирование определяется группой задач, связанных с преобразованием функциональных или принципиальных схем в совокупность конструктивных узлов, между которыми устанавливаются необходимые пространственные и электрические связи. Использование функционально-узлового метода, предполагающего выделение конструктивных узлов различной степени сложности, находящихся в отношении соподчиненности, является основой конструирования.

В результате, конструкция электронного устройства представляет собой разноуровневую структуру, в которой узлы низших уровней объединяются в узлы высшего уровня. Благодаря развитию методов микроминиатюризации электронных компонентов и использованию интегральных микросхем модульный принцип конструирования получил широкое распространение.

При конструировании электронных устройств используются следующие названия конструктивов электронных устройств.

Элемент – конструктив, представляющий из себя отдельное изделие, такое как: транзистор, диод, резистор, интегральная микросхема, гибридная интегральная схема.

Ячейка или типовой элемент замены – группировка элементов в едином исполнении на основе печатной платы, в виде микроэлектронного узла. В большинстве случаев ячейка реализует сложную функционально законченную схему и является отдельным физическим модулем. Элементы в ячейке соединены между собой печатными соединениями.

Следующие иерархические уровни занимают конструктивы,

называемые устройствами (панелями, субблоками), блоками (стойками). Каждый из конструктивов последующего уровня состоит из нескольких конструктивов предыдущего уровня, размещаемых в некоторой несущей конструкции (каркасе) и соединяемых между собой проводным монтажом, гибкими шлейфами или многослойными печатными объединительными платами.

В конструкторском проектировании выделяют три группы задач:

- 1) синтез конструкций;
- 2) контроль полученных решений;
- 3) оформление документации конструкторской и технологической, включая выпуск носителей информации.

Основными задачами синтеза конструкций ЭУ являются следующие коммутационно-монтажные задачи:

1) компоновка конструктивов i -го уровня в конструктив $(i - 1)$ -го уровня (компоновка простейших функциональных элементов логических схем в корпуса микросхем, компоновка интегральных микросхем (ИМС) и других электрорадиоэлементов в ячейки, ячеек в блоки и устройства);

2) размещение конструктивов i -го в конструктивах $(i - 1)$ -го уровня (размещение цифровых и аналоговых бескорпусных ИМС по подложке гибридной интегральной схемы (ГИС), размещение электрорадиоэлементов на печатной плате, размещение ячеек в блоке и т. д.);

3) трассировка монтажных соединений между конструктивами на всех уровнях (соединения между кристаллами ИМС, печатный монтаж, межплатные соединения).

Решение задач синтеза конструкций: компоновки, размещения и трассировки осуществляется на основе алгоритмов конструирования в составе САПР. Среди алгоритмов конструкторского проектирования выделяют два основных класса: конструктивные и итерационные.

Конструктивные алгоритмы формируют проектное решение за ряд последовательных шагов:

выбирается один элемент схемы рассматриваемого уровня;
к выбранному элементу по определенным правилам присоединяется
второй;

к полученному комплексу элементов добавляется третий и т.д.

Алгоритмы, использующие подобную методологию, называются последовательными. Алгоритмы, в которых формируются несколько групп элементов в пределах одного шага, называются параллельными.

Итерационные алгоритмы требуют задания начального приближения решения задачи конструкторского проектирования, которое затем улучшается. Начальное решение задается инженером-проектировщиком (пользователем САПР) или является результатом работы конструктивного алгоритма.

Анализ процесса конструирования электронных устройств (ЭУ) на основе существующих методов можно проводить, используя разные точки зрения. На входе процесса проектирования ЭУ имеется функциональная схема, которая содержит информацию о базовых элементах (в зависимости от уровня проектирования), о связях между элементами и внешних связях проектируемого ЭУ. Кроме того, задаются технологические параметры, например, при конструировании печатного узла – размер печатной платы, разрешенные для применения в данной разработке серии микросхем, шаг сетки трассировки и т.п. В результате проектирования должен получиться рисунок трассировки.

Общей целевой функцией φ всего процесса проектирования печатной платы (ПП) следует считать число проведенных связей. Целевая функция φ зависит от случайных входных параметров ζ (например, числа базовых элементов, числа задействованных выводов элементов схем и т.д.), так и от переменных, статистически устойчивых для класса разработок β (размеры ПП, серии микросхем, разрешенных для применения и т.п.). Хотя оптимальный размер ПП может быть определен с достаточной точностью как функция входных параметров, этот вопрос практического применения не нашел, так как

изготовление и применения плат различных размеров в одной разработке связано с изменением технологической оснастки производства.

Действия над входными величинами (ζ, β) в процессе проектирования можно представить следующим образом. Обозначив совокупность используемых в САПР алгоритмов компоновки K , совокупность алгоритмов размещения P и совокупность алгоритмов трассировки T можно представить схему последовательной реализации основных алгоритмов (рисунок 1).

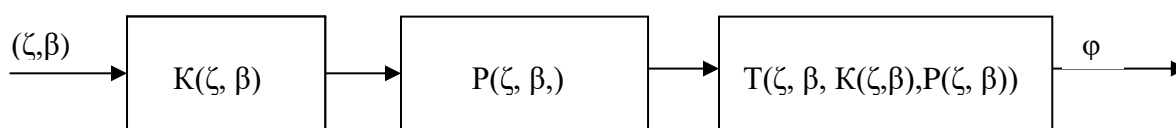


Рисунок 1 – Схема последовательной реализации основных алгоритмов

Возможно использование алгоритмов совместного решения задач компоновки и размещения, размещения и трассировки, компоновки и трассировки (при этом трассировка выполняется на условном коммутационном поле).

Общая задача проектирования формулируется следующим образом: имея заданные условия β и случайные воздействия ζ , найти такую совокупность алгоритмов и критериев $\{K, P, T\}$ которая обеспечивала бы получение максимального конечного значения целевой функции Φ . Таким образом, имеется задача о выборе и принятии решения в условиях неопределенности. Оптимизация решения на каждом шаге отдельно не всегда дает в сумме оптимальное решение, особенно, если на промежуточных этапах слабо учитывается конечный критерий. Указанный недостаток является причиной поиска связей между основными этапами конструкторского проектирования.

Реализованные алгоритмы K, P, T в составе САПР должны симитировать действия конструктора, который решает одновременно все три задачи с оптимизацией. Это требует создания обширного математического обеспечения САПР, позволяющего варьировать различными алгоритмами.

Полная алгоритмизация действий конструктора затруднительна, поэтому задача решается поэтапно, через выстраивание преемственных критериев всех этапов, поэтому при разработке САПР ставятся упрощенные задачи, т.е. на каждом шаге принятия решений приближаться к оптимальной траектории Ф.

Очевидно, что само выделение трех этапов конструкторского проектирования – компоновки, размещения и трассировки, направлено на снижение размерности общей задачи. Декомпозиция проводится следующим образом, что сначала осуществляется компоновка путем оптимальной группировки функциональных узлов, затем размещение компонентов выделенных узлов в их монтажном пространстве с учетом критериев, отражающих оптимальность последующего этапа – трассировки соединений.

1.3 Методы и алгоритмы решения задачи размещения при ее цифровизации

1.3.1 Критерии размещения

Размещение – проектная процедура определения расположения множества элементов на коммутационном пространстве.[3] Проектирование печатного узла связано с размещением не только электронных компонентов в коммутационном поле, но и распределением электрических связей по слоям платы, чтобы будущие печатные проводники были наиболее короткими и количество переходных отверстий было минимальным. Для этого компоненты с общими связями располагают рядом с учетом количества общих связей. В первую очередь происходит размещение крупных элементов, таких как: микросхемы, разъемы. После этого размещаются мелкие элементы: конденсаторы, резисторы. Рекомендуется тщательно изучать связи и в соответствии с этим делать оптимальное размещение, так как от размещения компонентов зависят результаты трассировки.

При решении задачи размещения элементов ЭУ находят применение следующие критерии размещения:

- суммарная длина соединений;

- расстояние между элементами, соединенными наибольшим числом связей;
- число пересечений проводников на КП;
- длина наиболее длинных связей;
- число цепей с возможно более простой конфигурацией;
- число перегибов проводников;
- число межслойных переходов;
- параметры паразитных связей между элементами и проводниками;
- равномерность температуры по поверхности КП.

В большинстве случаев основным рассматриваемым критерием оптимальности при решении задачи размещения является минимизация суммарной длины соединений (СДС), которая косвенно учитывает другие из перечисленных критериев. Это обуславливается рядом факторов:

- уменьшение длин соединений улучшает электрические параметры схемы;
- как показывает практика, чем меньше суммарная длина соединений, тем в среднем проще их реализация при трассировке;
- уменьшение суммарной длины соединений снижает трудоёмкость изготовления монтажных схем, особенно проводного монтажа.

Основным критерием качества размещения является минимальная суммарная длина соединений.

Пусть даны элементы коммутационной схемы e_1, e_2, \dots, e_n и для каждой пары элементов заданы весовые коэффициенты r_{ij} ($i, j=1, 2, \dots, n$), определяющие «степень связи» элементов друг с другом. Таким образом, схема задана матрицей соединений $R = \|r_{ij}\|_{n \times n}$.

Пусть также имеется некоторый фиксированный набор позиций для размещения элементов l_1, l_2, \dots, l_m ($m \geq n$). Будем полагать, что $m=n$. Если $m > n$, то можно ввести $m-n$ фиктивных элементов, не связанных с остальными. Определим расстояние d_{ij} между парами позиций. В любом случае, если на

коммутационном поле фиксированы позиции для размещения элементов, то можно задать матрицу соединений $D = \|d_{ij}\|_{n \times n}$, в которой элемент d_{ij} равен расстоянию между центрами позиций l_i и l_j . Очевидно, что матрица D – симметрическая с нулевой диагональю ($d_{ii} = 0, i = 1, 2, \dots, n$).

Произвольное размещение элементов в позициях представляет собой некоторую перестановку $p = p(1), \dots, p(i), \dots, p(n)$, где $p(i)$ задает номер позиции, присвоенный i -му элементу. Таким образом, всего имеется $n!$ различных вариантов размещения элементов.

Рассмотрим задачу минимизации суммарной длины (МСВД) соединений при следующих предположениях. Соединения будем считать условно исходящими из геометрических центров элементов. Кроме того, предполагаем совпадение центров элементов и позиций. Как правило, при решении задачи размещения необходимо учитывать предварительное закрепление некоторых элементов в позициях и соединения элементов с внешними выводами. Сопоставляя внешним выводам элемент e_0 и фиксируя расположение элементов, получим упрощенное представление коммутационного поля. Очевидно, что длина соединений между элементами e_i и e_j оценивается величиной $r_{ij}d_{p(i)p(j)}$. Обозначим через L_s множество всех фиксированных элементов, включая элемент e_0 ; тогда суммарная взвешенная длина соединений элемента e_i с элементами из L_s оценивается по формуле

$$a_{ip(i)} = \sum_{s \in L} r_{ij} d_{p(i)p(s)}, \quad (1.1)$$

где $d_{p(i)p(s)}$ – расстояние между элементом e_i , находящимся в позиции $p(i)$, и элементом e_s .

Учитывая вышесказанное, а также симметричность матриц R и D , запишем выражение для суммарной взвешенной длины соединений при произвольном размещении:

$$F(p) = \sum_{i,j=0}^n r_{ij} d_{p(i)p(j)} + a_{ip(i)}, \quad (1.2)$$

таким образом, задача размещения по критерию МСВД соединений состоит в минимизации функционала (1.2) на множестве перестановок p . Данная задача является вариантом общей математической модели, впервые рассмотренной Купмансом и Бекманом и получивший название задачи квадратичного назначения.

1.3.2 Методы размещения

Решение задачи размещения элементов с учетом конструктивно-технологических ограничений должно сформировать такое расположение элементов на коммутационном поле, при котором формируются лучшие условия для решения задачи трассировки их соединений.

Все известные в настоящее время алгоритмы размещения являются приближенными. Связано это с тем, что окончательная длина соединений может быть определена только при решении дальнейшей задачи трассировки. Расчет ведется через расстояние между позициями установки элементов. Каждый элемент представляется в виде точки, являющейся геометрическим центром корпуса элемента. Благодаря такому подходу значительно упрощается решение задачи размещения. При этом есть возможность, при необходимости осуществить переход от точек к элементам – принципиальных алгоритмических трудностей нет.

Выделяют основные группы алгоритмов размещения

- 1) Алгоритмы решения математических задач, являющиеся моделями задачи размещения;
- 2) Конструктивные алгоритмы начального размещения;
- 3) Итерационные алгоритмы улучшения начального варианта размещения;
- 4) Непрерывно-дискретные методы размещения.

К первой группе относятся, прежде всего, метод ветвей и границ для задачи квадратичного назначения, к которой при определённых упрощениях сводится задача размещения: набор позиций считается фиксированным, элементы рассматриваются как геометрические точки, схема соединений представляется взвешенным графом. Другой класс моделей связан с оптимизацией размещения, когда набор позиций для установки заранее не фиксирован.

Вторая и третья группы содержат в себе приближённые алгоритмы, использующиеся для оптимизации размещения элементов при фиксированном наборе позиций.

Характерной чертой конструктивных алгоритмов является то, что они создают размещение. Тогда как итерационные алгоритмы предполагают задание начального размещения. Конструктивные алгоритмы используют последовательный или параллельно-последовательный процесс установки элементов в позиции при локальной оптимизации функции-критерия размещения.

В итерационных алгоритмах производится перерасположение элементов или их групп с целью минимизации выбранного критерия. Эти алгоритмы требуют существенных затрат машинного времени и используются для получения конечного размещения.

Основной областью применения непрерывно-дискретных методов размещения являются конструкции, в которых позиции для установки элементов заранее не фиксированы. Исходной базой для построения этих алгоритмов являются непрерывные модели.

Наибольший интерес при формировании маршрута проектирования в рамках цифрового проектирования представляют конструктивные алгоритмы поскольку они имитируют действия инженера проектировщика при решении задачи размещения.

Исходной информацией для размещения является схема соединений, параметры конструкции элементов и коммутационного поля. Пусть $E = \{e_1, e_2,$

..., e_n } – множество элементов, подлежащих размещению, а $L = \{l_1, l_2, \dots, l_n\}$ – множество позиций для их установки. Вводится n -шаговый процесс принятия решений, на каждом шаге которого выбирается один из неразмещенных элементов и помещается в одну из незанятых позиций. Структура любого последовательного алгоритма размещения определяется правилами выбора очередного элемента и позиции для его установки.

Пусть E_k – элементы, размещенные до k -го шага, а L_k – позиции, занятые этими элементами; E'_k и L'_k – соответственно неразмещенные элементы и незанятые позиции. Отметим, что перед началом размещения могут быть две ситуации: нет размещенных ранее элементов, внешние выводы узла (контакты, разъемы и т. п.) не закреплены (в этом случае в алгоритме должен быть особо определен способ установки первого элемента); имеется группа заранее размещенных элементов или закрепленных внешних выводов.

В основу большинства последовательных алгоритмов размещения положен эвристический принцип оптимизации целевой функции, сводящийся к выбору на данном шаге локально оптимальной позиции для одного из элементов при неизменности положения ранее размещенных элементов. Поскольку критерий минимума суммарной длины соединений в силу отмеченных ранее причин наиболее распространен, он и будет рассмотрен при описании алгоритмов данной группы. Вместе с тем, используемые в этих алгоритмах тактики размещения могут быть с некоторыми модификациями применены и для других критериев. В алгоритмах размещения по связности элемент и позиция выбираются независимо.

Выбор элемента. Любое правило выбора элемента для размещения основано на вычислении «меры связности» еще неразмещенных элементов с уже размещенными. Естественной мерой связности двух элементов e_i и e_j является количество соединений между ними, заданное в матрице соединений $R = \|r_{ij}\|_{n \times n}$.

Так, в алгоритме «попарных связей» Куртзберга для каждого неразмещенного элемента e_i подсчитывается характеристика

$$c_i = \max_{e_j \in E_k} r_{ij}, \quad (1.3)$$

где r_{ij} рассчитывается по формуле

$$r_{ij} = \sum_{s \in J_{ij}} \frac{p_s + \lambda}{p_s} w_s, \quad (1.4)$$

где J_{ij} – множество цепей, связывающих элементы;

e_i и e_j p_s – размер цепи;

w_s – весовой коэффициент;

λ – целочисленный параметр.

Параметр λ позволяет дифференцировать вклад цепей различного размера. Чем больше значение λ , тем больше влияние цепей с малым значением p_s . Очевидно, что при $\lambda=0$ значение r_{ij} (1.4) равно суммарному весу цепей между элементами e_i и e_j . Поскольку всегда $P_2 \geq 2$, то $\lambda \geq -1$.

Очередным размещаемым элементом является элемент, имеющий максимальную характеристику (1.3), т. е. выбор элемента осуществляется по наибольшему числу связей с уже размещенными элементами.

Другое правило выбора основано на расчете для каждого неразмещенного элемента $e_i \in E'_k$ суммарной связности с уже размещенными элементами. Если задаётся матрица соединений R , то соответствующая характеристика

$$c_i = \sum_{e_j \in E_k} r_{ij}, \quad (1.5)$$

если для схемы задан список цепей (комплексов), то аналогичная характеристика имеет вид

$$c_i = J_i \left(\bigcup_{e_j \in E_k} J_j \right), \quad (1.6)$$

где J_i – множество цепей, связанных с элементом e_i ;

c_i – конъюнкция множества цепей элемента e_i и множества цепей E_k .

Выбор в данном случае осуществляется на основании максимального значения характеристики (1.5) и (1.6).

Можно использовать другое правило выбора элемента, основанное на оценке числа связей не размещённого элемента $e_i \in E'_k$ как с размещёнными, так и с неразмещёнными элементами:

$$c_i = \sum_{e_j \in E_k} r_{ij} - \sum_{e_j \in E'_k} r_{ij}, \quad (1.7)$$

в этом случае выбирается элемент с максимальным значением c_i (1.7). Заметим, что данный выбор аналогичен принципу максимальной конъюнкции - минимальной дизъюнкции, применяемому в алгоритмах компоновки.

К этому же типу относится характеристика относительной связности:

$$c_i = \sum_{e_j \in E_k} r_{ij} / \sum_{e_j \in E_k} r_{ij}, \quad (1.8)$$

на очередном шаге алгоритма размещается элемент, имеющий максимальный коэффициент относительной связности.

В связи с тем, что при размещении элементов число соединений между отдельными группами элементов не может быть строго определено (оно зависит от способа последующей трассировки соединений), в некоторых алгоритмах используются вероятностные оценки для подсчета меры связности.

Выбранный для размещения элемент e_{i0} должен быть установлен в одну из незанятых позиций из множества L'_k – Эта позиция выбирается с учетом минимизации критерия размещения, в частности, МСД соединений. Естественно, что при последовательном процессе размещения может быть оценена лишь суммарная длина частичных монтажных соединений данного элемента e_{i0} с уже размещёнными элементами E_k .

В принципе возможен подход, когда при установке элемента в позицию рассчитываются трассы соответствующих соединений. Длина этих соединений является критерием для выбора позиции. Однако большие затраты машинного времени делают этот подход нереальным, по крайней мере, при

конструировании узлов с печатными соединениями, и ограниченно применимым при конструировании монтажных схем проводных соединений.

Учитывая вышесказанное, используют приближенные оценки длин монтажных соединений. Наиболее простая из них и наименее точная сводится к расчету расстояний между позициями.

2 Разработка программного решения для выполнения задачи размещения

2.1 Интеграция программного решения с продуктами линейки Cadence

2.1.1 Разработка принципиальной схемы в САПР OrCAD

Cadence Design Systems – компания, являющаяся крупнейшим поставщиком технологий для проектирования электронных устройств, основанная в 1988 году, занимающаяся разработкой программного обеспечения для автоматизации проектирования электронных устройств и предоставлением инженерных услуг. Программное обеспечение для проектирования печатных плат и микросхем является основным продуктом компании.

Продукты Cadence типы проектирования и верификации:

— Virtuoso Platform – инструменты для проектирования интегральных схем; в него входит структурное представление, поведенческое моделирование, симуляция схемы, полный макет, физическая верификация. Используется, в основном, для проектирования аналоговых схем;

— Genus Platform – инструменты для проектирования цифровых интегральных схем. Сюда входит проектирование, тестирование и синтез;

— Xcelium Platform – инструменты для симуляции. Сюда входит формальная верификация, проверка формальной эквиваленции и эмуляция;

— Allegro Platform – инструменты для совместного проектирования интегральных схем и печатных плат;

— OrCAD/PSpice – Инструменты для небольших компаний проектирования и индивидуальных разработчиков.

OrCAD – пакет компьютерных программ, использующийся для создания электронных версий печатных плат, производства электронных схем и их моделирования, предназначен для автоматизации проектирования

электроники.

В составе пакета следующие модули:

- Capture – редактор принципиальных схем,
- Capture CIS Option – менеджер библиотек Active Parts,
- PSpice Analog Digital – пакет аналого-цифрового моделирования,
- PSpice Advanced Analysis – пакет параметрической оптимизации,
- PSpice SLPS option – интерфейс связи с пакетом Matlab,
- PCB Designer – редактор топологий печатных плат,
- SPECCTRA for OrCAD – программа автоматической и интерактивной трассировки,
- Signal Explorer – модуль анализа целостности сигналов и перекрестных искажений.

PSpice является программой симуляции аналоговой и цифровой логики, описанной на языке SPICE, которая предназначена для персональных компьютеров. Разработана компанией MicroSim и используется в автоматизации проектирования электронных приборов. Компания-разработчик была приобретена фирмой OrCAD, а затем Cadence Design Systems. В настоящее время программа также может симулировать и смешанные аналого-цифровые схемы.

В настоящее время PSpice удовлетворяет множеству требований индустрии и интегрирован в маршруты проектирования от OrCAD и Cadence Allegro.

Рассмотрим этапы формирования коммутационной схемы на примере заданной электронной схемы в САПР OrCAD.

Проектирование начинается с создания нового проекта (рисунок 2). Создаётся новый проект с помощью команды *File/New/Project*, после чего в открывшемся диалоговом окне в строке Name указывается имя проекта (без использования символов кириллицы и пробелов), а в строке Location – имя

подкаталога расположения проекта.

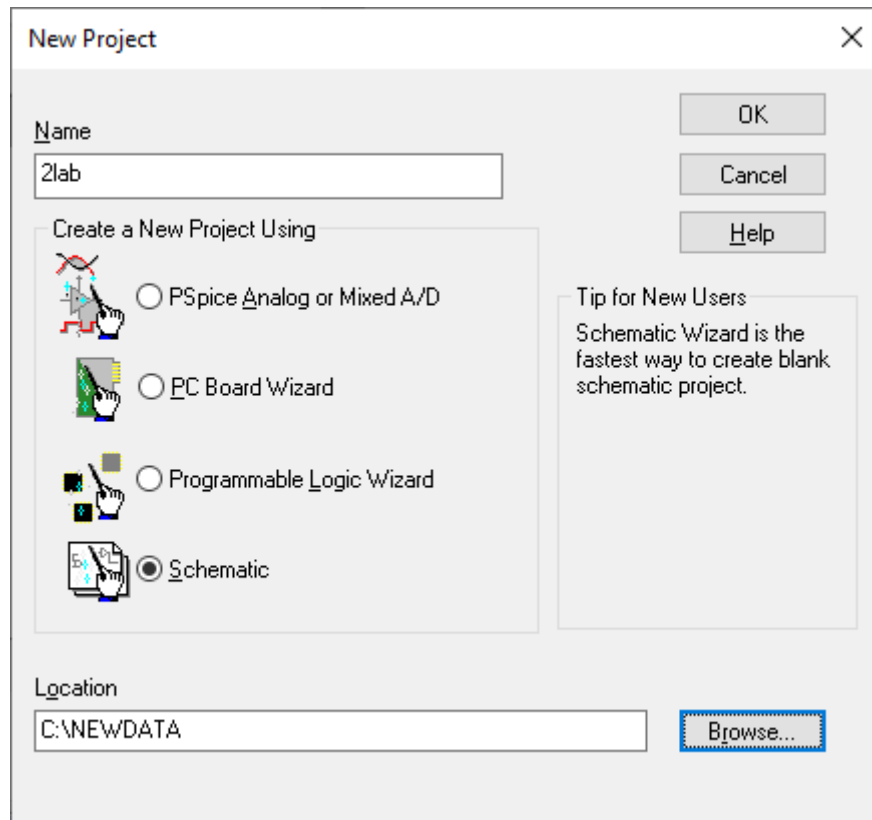


Рисунок 2 – Диалоговое окно создания нового проекта

Для проектирования принципиальной электронной схемы необходимо использование библиотек элементов, из которых она будет состоять. САПР имеет большое количество библиотек, а также позволяет создавать их самостоятельно.

Чтобы создать новую библиотеку необходимо воспользоваться командой *File/New/Library*. В менеджер проектов будет добавлена новая библиотека (рисунок 3). Далее необходимо выделить созданную библиотеку и выбрать опцию «SaveAs...».

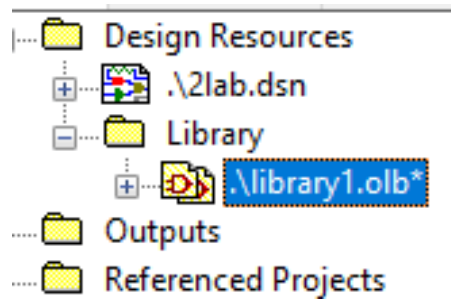


Рисунок 3 – Библиотека в менеджере проектов

В открывшемся диалоговом окне программа запросит имя новой библиотеки и место ее расположения (рисунок 4). Для создания нового элемента библиотеки необходимо выделить созданную библиотеку и выполнить команду *New/Part*.

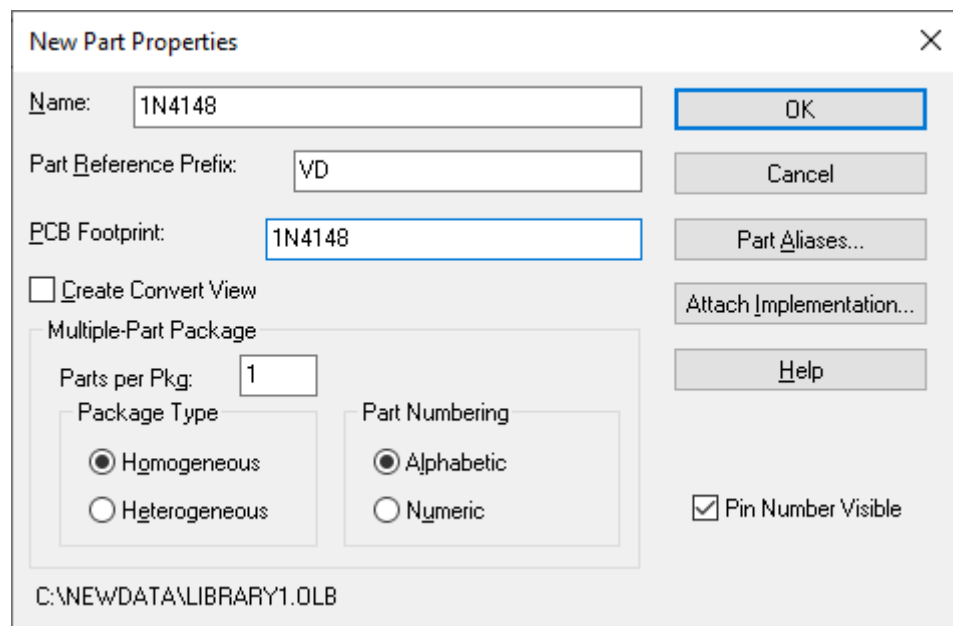


Рисунок 4 – Диалоговое окно создания нового компонента

В открывшемся диалоговом окне *New Part Properties* необходимо ввести следующие данные:

- Name – имя символа (согласно модели);
- Part Preference Prefix – префикс позиционного обозначения;
- PCB Footprint – название файла корпуса элемента;
- Parts per Package – количество всех секций в корпусе компонента;

- Homogeneous и Heterogeneous – выбор формирования компонента, имеющего одинаковые секции или секции разного типа;
- Alphabetic или Numeric – выбор между буквенной или цифровой нумерацией секций компонентов;
- Pin Number Visible – отображение номеров выводов элемента.

После заполнения окон и подтверждения операции откроется рабочее поле создания элемента, на котором штрихпунктирным прямоугольником ограничены габариты символа.

Перед началом создания элемента необходимо изменить шаг сетки, для этого необходимо вызвать диалоговое окно путем выполнения команд *Options/DesignTemplate/PageSize* (рисунок 5).

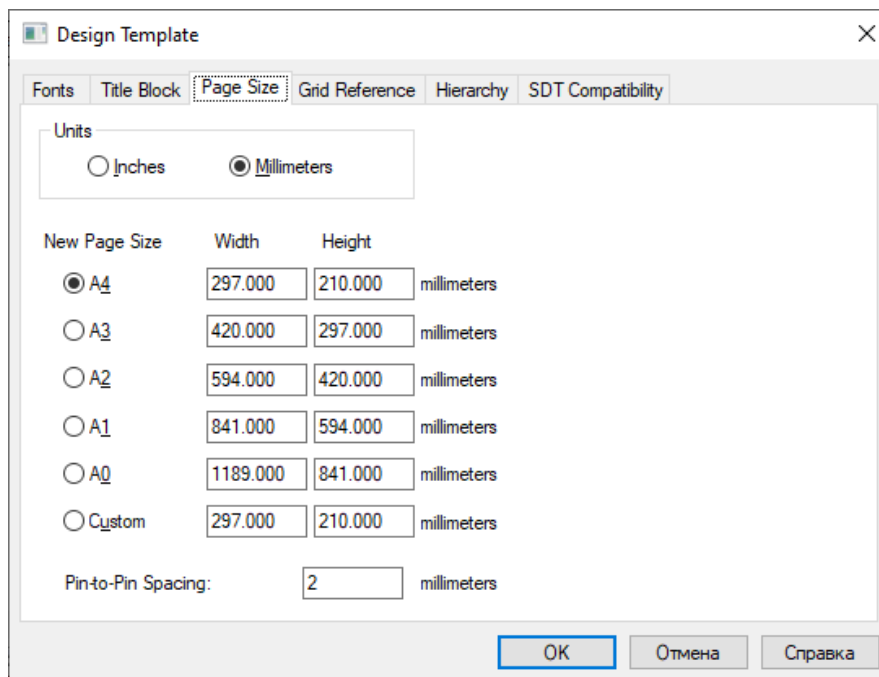


Рисунок 5 – Установка шага сетки

При помощи инструментов на боковой панели создается изображение элемента (рисунок 6).

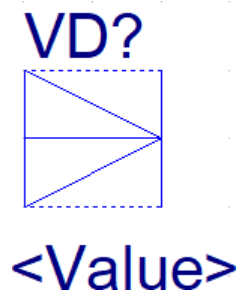


Рисунок 6 – Создание элемента библиотеки

Размеры элементов при их создании указываются в соответствии с единой системой конструкторской документации. Для добавления входов и выходов элемента используется команда Place Pin (рисунок 7).

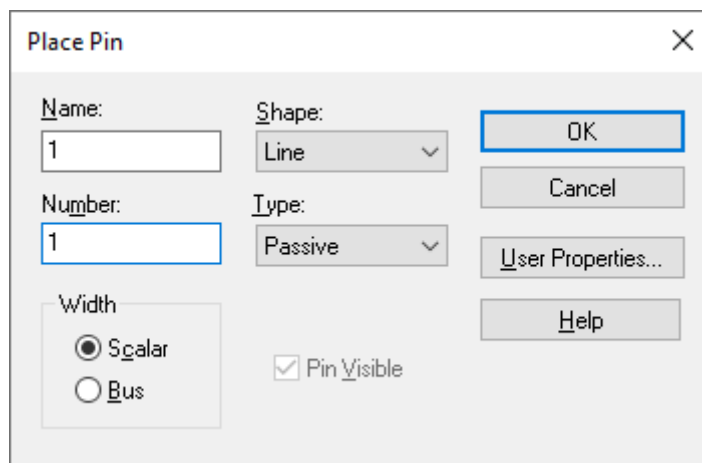


Рисунок 7 – Диалоговое окно добавления входов и выходов

В открывшемся диалоговом окне PlacePin требуется ввести следующие данные:

- Name – название вывода;
- Number – номер вывода;
- Shape – форма вывода;
- Type – тип вывода;
- Scalar или Bus – выбор между одиночным выводом и шиной;
- Pin Visible – отображение вывода;
- User Properties – диалоговое окно для редактирования дополнительных характеристик вывода компонента.

Окончание данного этапа можно считать последним в создании элемента (рисунок 8).

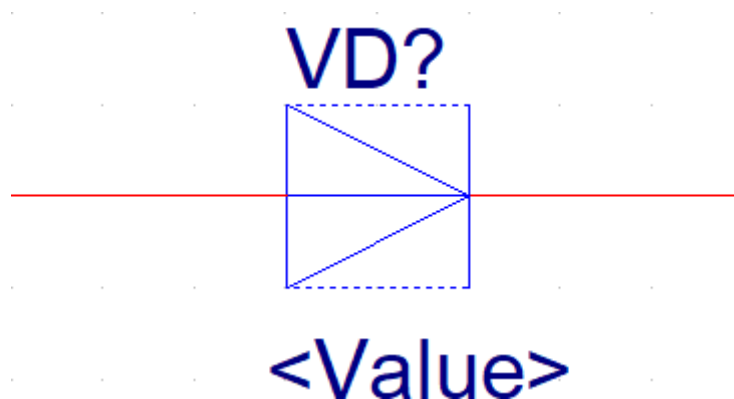


Рисунок 8 – Символ элемента диода

Когда все необходимые библиотеки подготовлены, можно переходить к созданию принципиальной схемы.

Для начала необходимо изменить шаг сетки orCAD Capture, аналогично тому, как требовалось ее изменить во время создания элемента библиотеки. С помощью команды *Place/Part* необходимо добавить библиотеки, которые были сформированы для данного проекта.

Теперь, чтобы добавить элемент на схему требуется выбрать нужную библиотеку из списка подключенных библиотек, и указать необходимый компонент (рисунок 9).

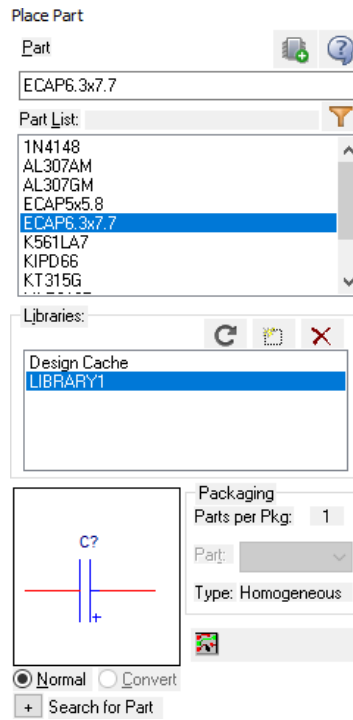


Рисунок 9 – Выбор символа из библиотеки

После размещения всех элементов необходимо соединить их между собой, по выполнению всех процедур будет получена принципиальная схема (рисунок 10).

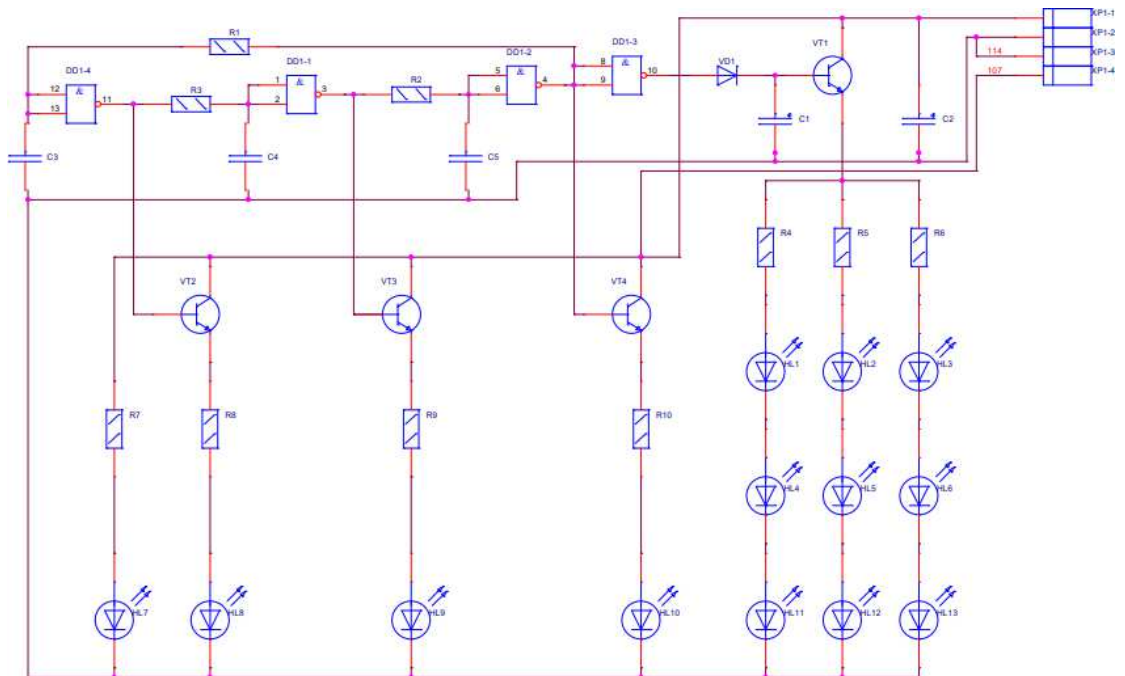


Рисунок 10 – Результат построения принципиальной электрической
СХЕМЫ

Формирование исходных данных для решения задачи размещения выполнено на примере формата *calay 9064*, на базе которого возможно провести анализ схмотехнического моделирования или начать конструирование (рисунок 11).

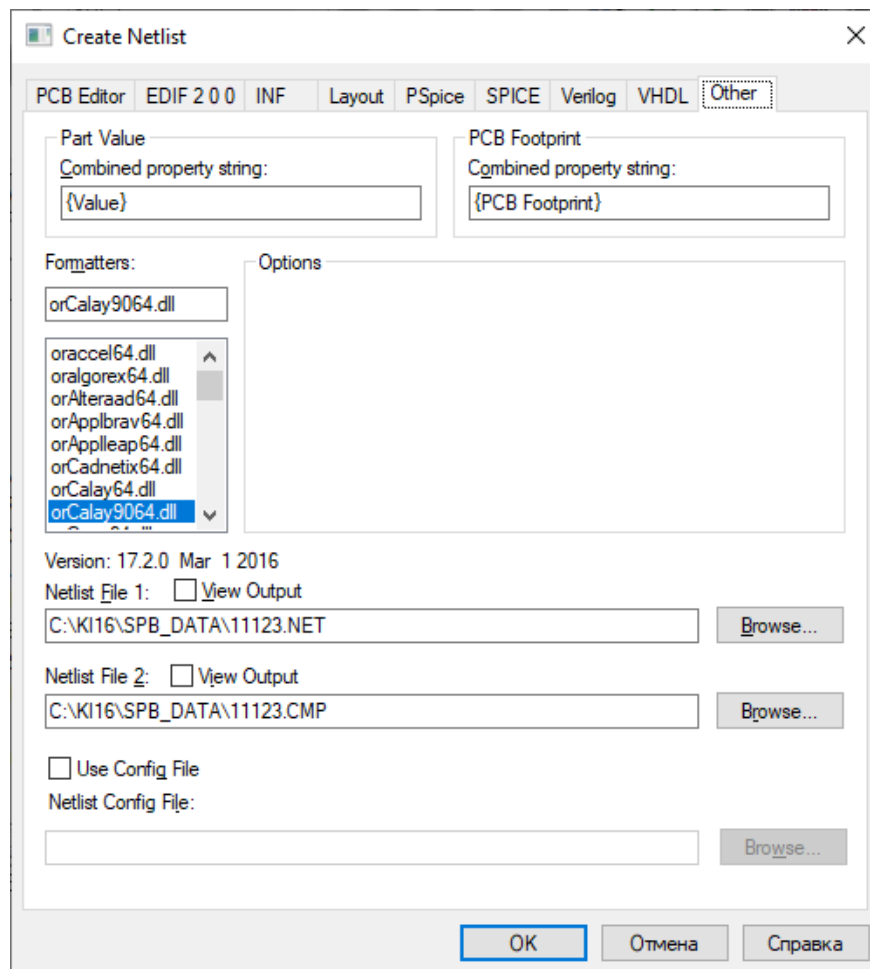


Рисунок 11 – Диалоговое окно *Create Netlist*

2.1.2 Интеграция с графическим редактором Capture

Интеграция с САПР OrCAD осуществляется через чтение конфигурационного файла списка соединений, который представляет собой текстовое представление информации о соединениях, связывающих компоненты электронного устройства (рисунок 12).

```

107      VT2('2) VT4('2) R7('2) VT3('2) VT1('2) C2('1) DD1('7);
114      HL8('2) C1('2) C3('1) HL10('2) HL9('2) C4('1) C5('1) HL7('2),
        HL11('2) HL12('2) HL13('2) C2('2) DD1('14);
N04319  C3('2) R1('1) DD1('12) DD1('13);
N04327  VT4('1) R1('2) DD1('4) DD1('8) DD1('9);
N04423  R2('2) C5('2) DD1('5) DD1('6);
N04431  VD1('1) DD1('10);
N04435  C1('1) VD1('2) VT1('1);
N04465  R3('2) C4('2) DD1('1) DD1('2);
N04517  R2('1) VT3('1) DD1('3);
N04541  VT2('1) R3('1) DD1('11);
N04715  VT2('3) R8('2);
N04719  VT3('3) R9('2);
N04723  VT4('3) R10('2);
N04787  R7('1) HL7('1);
N04791  HL8('1) R8('1);
N04795  HL9('1) R9('1);
N04799  HL10('1) R10('1);
N05036  HL1('1) R4('1);|
N05040  HL4('1) HL1('2);
N05048  HL4('2) HL11('1);
N05116  HL2('1) R5('1);
N05132  HL5('1) HL2('2);
N05148  HL5('2) HL12('1);
N05348  HL3('1) R6('1);
N05364  HL6('1) HL3('2);
N05380  HL6('2) HL13('1);
N05436  R4('2) VT1('3) R5('2) R6('2);

```

Рисунок 12 – Вид конфигурационного файла списка соединений

Каждое соединение начинается с названия узла, занимающее 6 символов, далее через запятую перечисляются все элементы, соединенные в этом узле, с учетом номера вывода компонента электронного устройства, указанного в круглых скобках. Описание узла коммутационной схемы в файле списка соединений заканчивается символом «;».

Каждый элемент можно представить, как экземпляр класса, содержащий в себе следующие свойства:

- 1) Название элемента;
- 2) Горизонтальное расположение элемента(X);
- 3) Вертикальное расположение элемента(Y);
- 4) Массив соединений с остальными элементами;
- 5) Массив расстояний с остальными элементами;
- 6) Сумма массива соединений;
- 7) Сумма массива расстояний.

Задачей программного блока является формирование из списка соединений массив объектов для дальнейшего использования в программном блоке реализованного метода обратного размещения.

Работа алгоритма преобразования данных работает следующим образом: программа начинает построчное считывание в буфер данных до того момента, пока последним символом строки не окажется «;», подразумевается, что строка начинается с названия узла, поэтому первые 6 символов пропускаются, после чего программа посимвольно записывает в буфер название элемента. Как только имя записано полностью, массив проверяется на существование идентичного объекта, если он не найден, то создается новый экземпляр, в противном случае программа его игнорирует (рисунок 13).

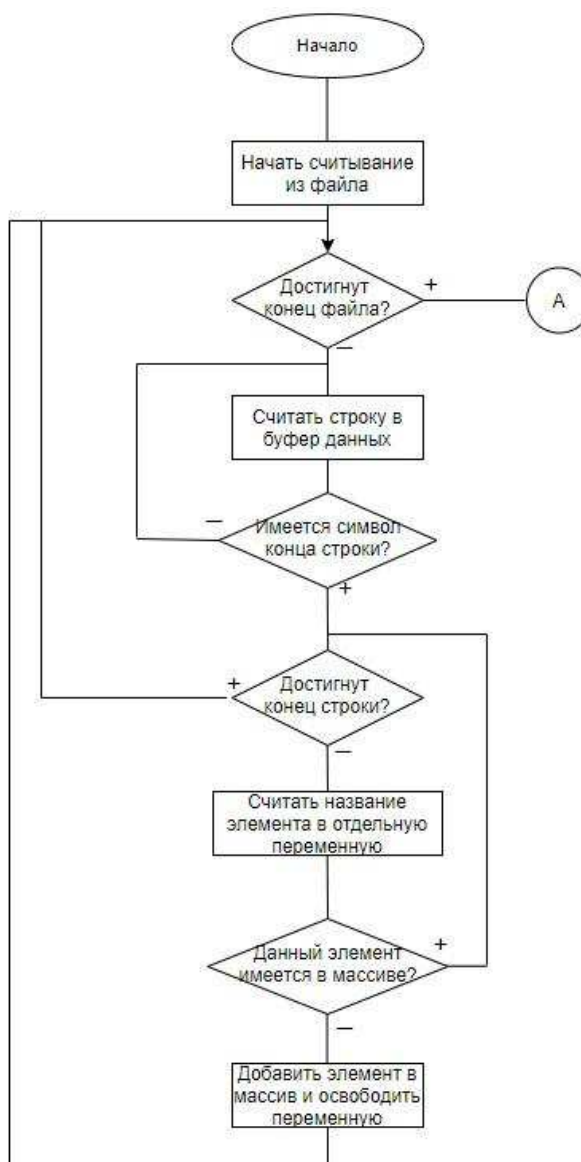


Рисунок 13 – Блок-схема чтения файла списка соединений (1 часть)

После формирования массива объектов, в которых записаны названия всех элементов необходимо получить матрицу соединений всех элементов.

Программа начинает повторное чтение из файла (рисунок 14). Начинается построчное считывание в буфер данных до того момента, пока последним символом строки не окажется «;». Программа поочередно пытается найти из строки каждый элемент массива, в успешном случае производится повторный поиск по элементам, в случае успеха запись связи между элементами осуществляется в массив в виде инкремента.

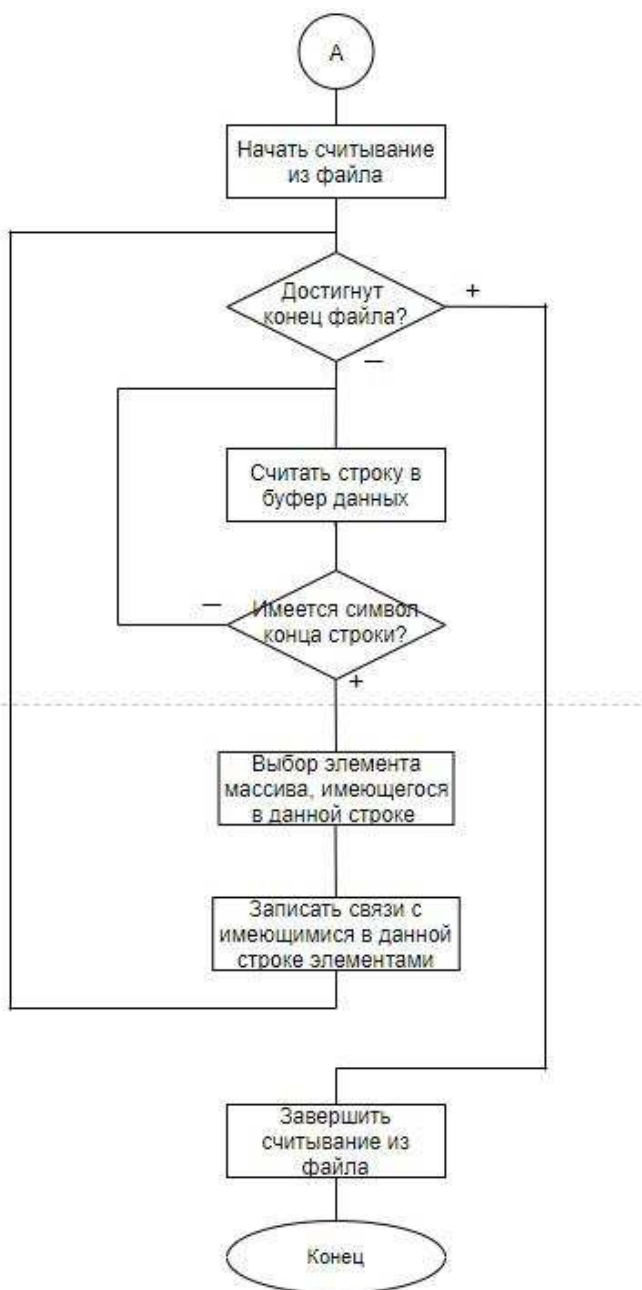


Рисунок 14 – Блок-схема чтения файла списка соединений (2 часть)

В конце повторного чтения в памяти системы хранится массив объектов элементов, каждый из которых имеет свое имя и массив соединений

с другими элементами. Таким образом исходные данные будут сформированных на базе рассмотренного алгоритма для принципиальной схемы, созданной в графическом редакторе Capture САПР OrCAD.

2.2 Технологии программной реализации

2.2.1 Метод обратного размещения

В методе обратного размещения осуществляется предварительная оценка каждого из размещаемых элементов e_1, e_2, \dots, e_n и каждой свободной позиции l_1, l_2, \dots, l_n после чего все элементы размещаются одновременно.

Пусть дана матрица соединений $R = \|r_{ij}\|_{n \times n}$ и матрица расстояний между позициями $D = \|d_{ij}\|_{n \times n}$.

Этапы метода обратного размещения:

1) Для каждого элемента e_i по матрице R найдем суммарное число соединений этого элемента с остальными элементами:

$$r_i = \sum_{j=1}^n r_{ij}, i = 1, 2, \dots, n, \quad (2.1)$$

2) Для каждой позиции l_i по матрице D найдем характеристику

$$d_i = \sum_{j=1}^n d_{ij}, i = 1, 2, \dots, n, \quad (2.2)$$

определяющую суммарное расстояние этой позиции до остальных позиций.

Позиции в центральной части коммутационного поля имеют меньшую характеристику d_i , чем позиции на периферии. Естественно, что центральные позиции наиболее благоприятны для размещения сильно связанных элементов, т.е. элементов, имеющих большее значение r_i (2.1). Рассматривая с этой точки зрения выражение (1.1) и учитывая условия минимальности скалярного произведения $r \cdot d$ получаем следующее:

3) Упорядочить элементы по возрастанию характеристики

$$r_i(r_{i_1} \leq r_{i_2} \leq \dots \leq r_{i_n})$$

4) Упорядочить позиции по убыванию характеристики

$$d_{ij}(d_{i_1} \geq d_{i_2} \geq \dots \geq d_{i_n})$$

5) Определить размещение $p(i_k) = j_k, k = 1, 2, \dots, n$.

На фоне других алгоритмов размещения данный отличается чрезвычайной простотой. Из-за этого существует возможность реализации метода при ручной методике разработки схем начального варианта размещения. Опыт использования данного алгоритма показывает, что он, как правило, дает решение, несколько уступающее по качеству размещению, выполненному опытным конструктором, тогда как последовательные алгоритмы дают сокращение длины соединений на 10-15%.

Для реализации программного решения, в котором используются объекты, более удачно подойдет любой объектно-ориентированный язык программирования.

В данной работе был применен язык программирования C#. Язык разработан компанией Майкрософт и является одним из самых популярных современных языков программирования. Он востребован на рынке разработки в различных странах, C# применяют при работе с программами для ПК, создании сложных веб-сервисов или мобильных приложений. Разработчик уделяет значительное внимание поддержке языка заинтересованы в популяризации инструмента и прилагают большое финансирование, а потому регулярно появляются обновления и дополнения, исправляются выявленные баги в компиляторе, расширяются библиотеки.

Майкрософт предоставляет подробную и развернутую документацию на своих официальных ресурсах. Существует множество учебников, курсов для новичков и других обучающих материалов.

Перечень возможностей разработки практически не имеет ограничений благодаря широчайшему набору инструментов и средств.

В языке принята общая система работы с типами, начиная от примитивов и заканчивая сложными, в том числе, пользовательскими

наборами. Применяется единый набор операций для обработки и хранения значений типизации. Также можно использовать ссылочные типы пользователя, что позволит динамически выделить память под объект или хранить упрощенную структуру в сети.

Язык программирования запрещает обращение к переменным, которые не были инициализированы, что исключает возможность выполнения бесконтрольного приведения типов или выхода за пределы определенного массива данных.

Хранение экземпляров класса основано на использовании коллекции List, это связано с удобством и простотой операций с массивом.

После получения коллекции экземпляров необходимо создать матрицу расстояний. Для этого каждому элементу присваиваются каждому две переменные, которые будут отображать вертикальное и горизонтальное расположение данного элемента. Каждый элемент массива расстояний определяется расстоянием до других элементов по следующей формуле

$$d_{ij} = d_{ji} = |(x_i - x_j) + (y_i - y_j)|, \quad (2.3)$$

где x_i, y_i - координаты i -того элемента;

x_j, y_j , -координаты j -того элемента.

Запись происходит таким образом, что индексы коллекции объектов совпадают с индексами массива. Доступно два закона формирования коммутационного поля: линейная и матричная. После выбора одного из них каждому элементу выдаются координаты по данному закону в том порядке, в котором они были записаны в коллекцию.

После формирования матрицы соединений программа сортирует объекты элементов в порядке возрастания сумм массива соединений, а после заменяет координаты объектов в порядке убывания сумм массива расстояний. По окончании сортировки производится подсчет суммарной длины соединений (СДС).

Далее программа проверяет, имеются ли одинаковые значения сумм, в успешном случае переставляет местами объекты и пересчитывает СДС, после

выполнения данного алгоритма сохранится комбинация объектов, имеющая минимальную СДС, непротиворечащая условию возрастания сумм массива соединений и убывания сумм массива расстояний.

2.2.2 Программная реализация задачи размещения

На рисунке 15 представлено основное окно программы, выбор расположения файла списка соединений происходит при нажатии кнопки «открыть». По завершению открытия формируется и выводится матрица соединений элементов, суммарная длина соединений до оптимизации, измеряемая в условных единицах.

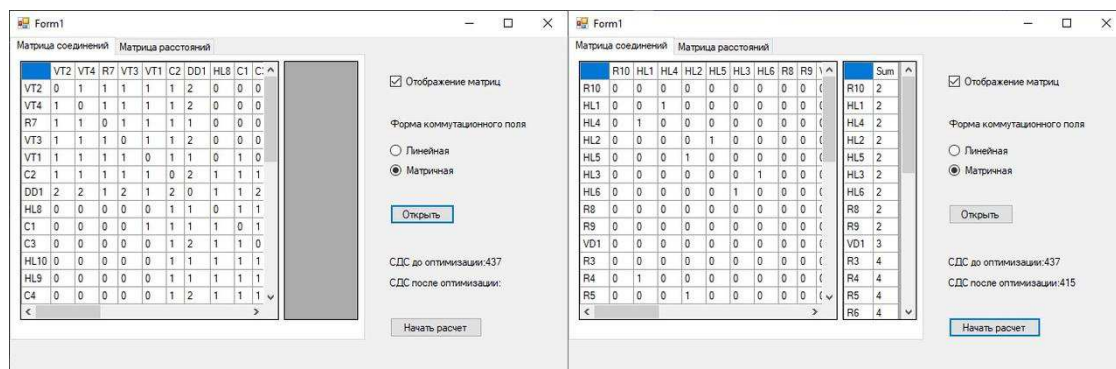


Рисунок 15 – Основное окно программы

Формируется окно пользовательского интерфейса, в котором представлено коммутационное поле в виде совокупности позиций с указанием позиционных обозначений, размещенных в них элементов (рисунок 16). Вид коммутационного поля зависит от контура печатной платы, которое определяется корпусом конструкции, методически ограничения конфигурации печатной платы отсутствуют.

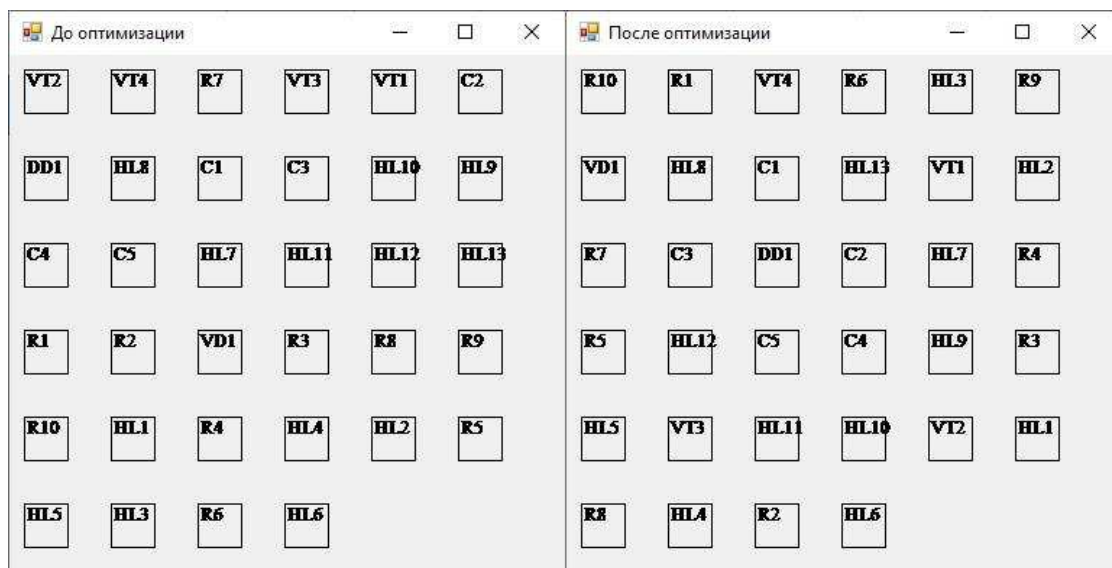


Рисунок 16 – Окно матрицы позиций

После работы алгоритма обратного размещения будут выведены информация о завершении расчета и отображены матрица соединений и пересчитанная матрица расстояний, значение СДС после оптимизации. Матрицы можно скопировать в табличной форме и использовать их в формате *.xls. В случае отсутствия необходимости в визуализации матриц, их прорисовку можно отключить, нажав на соответствующую опцию «Отображение матриц».

3 Применение метода обратного размещения при разработке печатных узлов

3.1 Разработка печатных узлов в САПР OrCAD

Создание библиотеки корпусов элементов происходит в графическом редакторе печатных плат orCAD PCB Editor. После открытия необходимо выбрать опции *File/New*.

В открывшемся диалоговом окне требуется ввести следующие данные (рисунок 17):

- Drawing Name –Имя корпуса элемента, записанное в PCB Footprint;
- Drawing Type – Тип создаваемого элемента (для создания корпуса элемента необходимо выбрать *Package symbol (wizard)*).

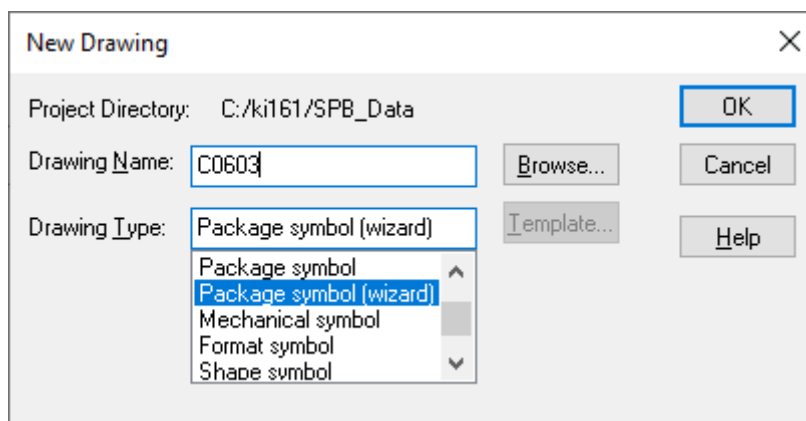


Рисунок 17 – Диалоговое окно создания элемента в графическом редакторе печатных плат

В появившемся диалоговом окне необходимо выбрать тип корпуса создаваемого элемента (рисунок 18):

- *DIP* – корпус с двумя рядами выводов по длинным сторонам микросхемы;
- *SIOC* – маленькая микросхема с выводами по длинным сторонам;
- *PLCC/QFP* – четырехугольный плоский корпус;
- *PGA/BGA* – матрица из штырьковых выводов;

- *TH DISCRETE* – компоненты выводного монтажа;
- *SMD* – корпуса компонентов для поверхностного монтажа;
- *SIP* – плоский корпус с выводами с одной стороны;
- *ZIP* – плоский корпус с выводами, расположенными зигзагообразно.

В рассматриваемом примере будет использован конденсатор типа SMD – корпус для поверхностного монтажа.

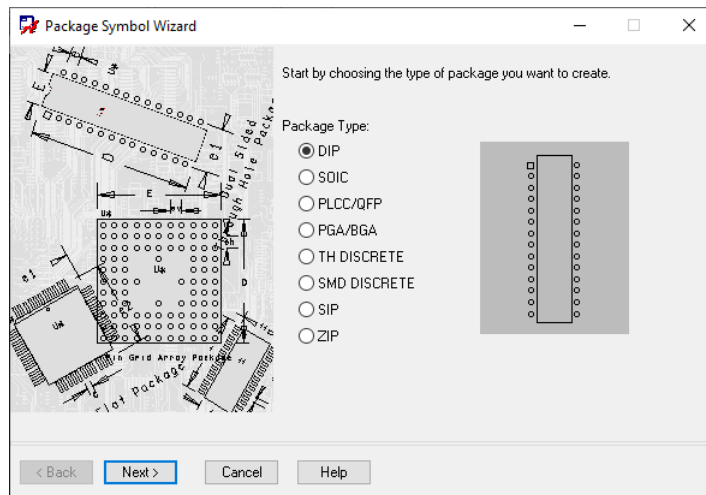


Рисунок 18 – Выбор типа корпуса элемента

Далее в открывшемся окне необходимо выбрать единицы измерения, в которых будет создан корпус, и их точность, префикс позиционного обозначения (рисунок 19).

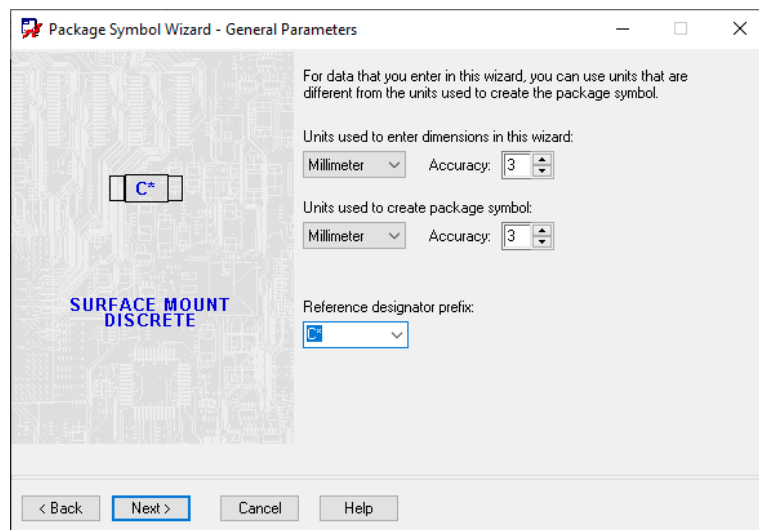


Рисунок 19 – Выбор единиц измерения

На следующем этапе необходимо указать требуемые размеры создаваемого корпуса и тип выводов (рисунок 20). Перечень требуемых размеров будет меняться в зависимости от выбранного типа корпуса. Размеры элементов указываются в соответствии с технической спецификацией элемента.

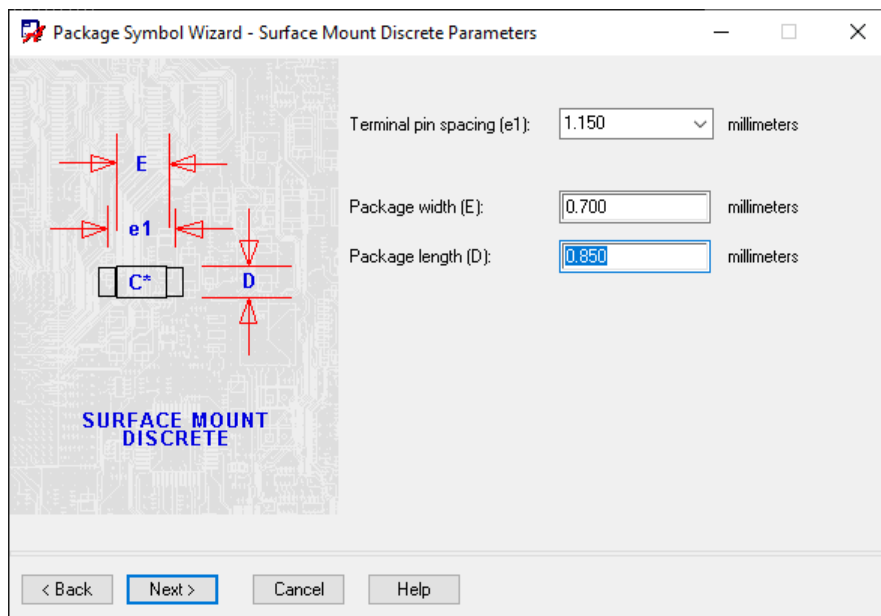


Рисунок 20 – Внесение размеров элемента

Остается выбрать необходимое отверстие или посадочное место для электронного компонента (рисунок 21).

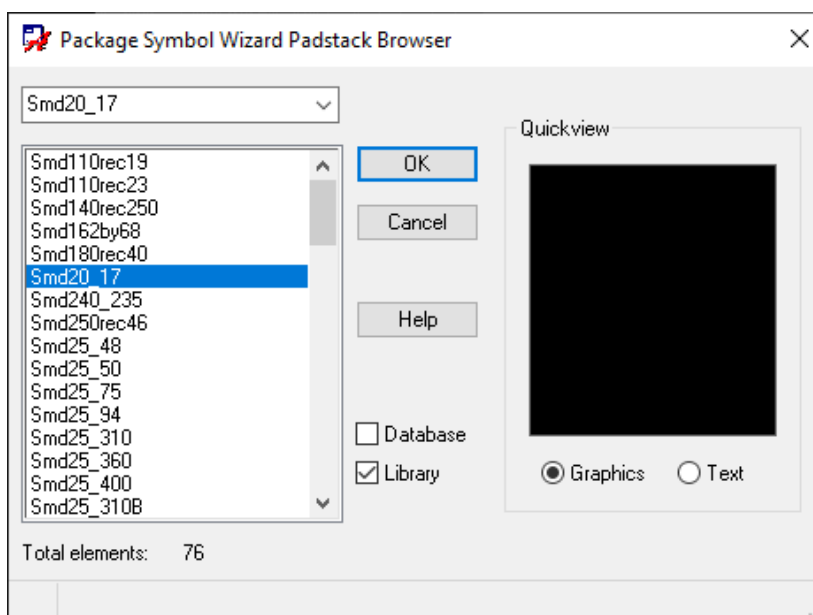


Рисунок 21 – Выбор типа выводов

По окончании всех этапов формируется корпус элемента выбранного компонента (рисунок 22).



Рисунок 22 – Результат создания корпуса конденсатора

После формирования библиотеки корпусов элементов можно начинать упаковку печатной платы. Передача изменений между схемой и платой осуществляется из схематехнического редактора путем использования диалогового окна *Create Netlist* (рисунок 23).

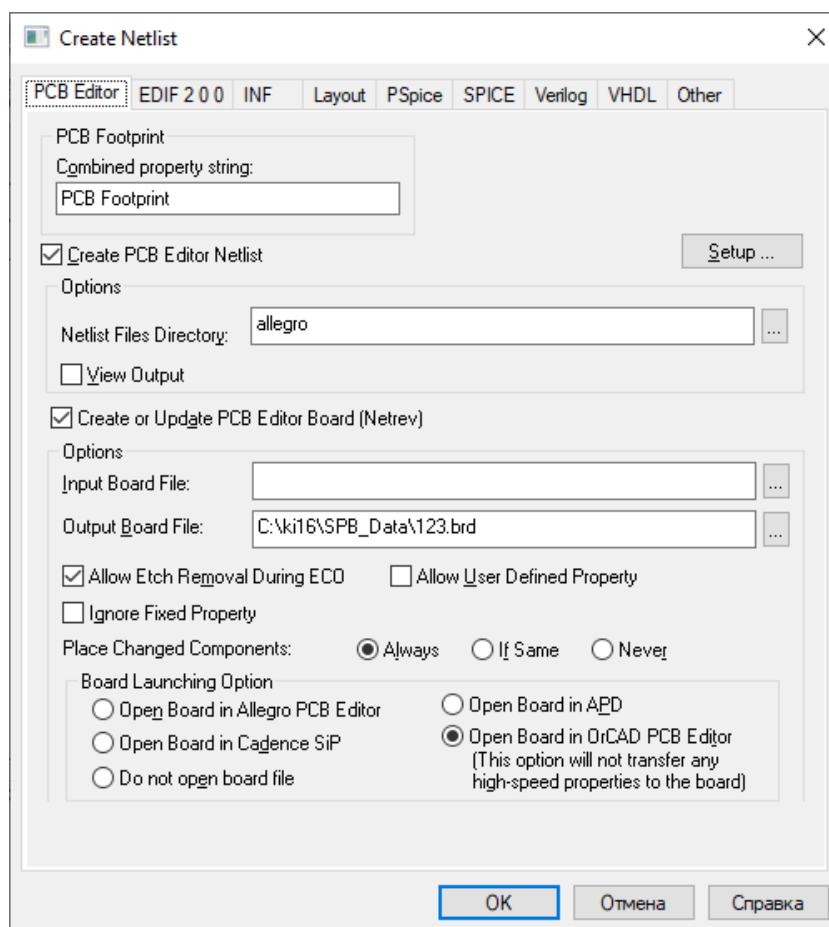


Рисунок 23 – Диалоговое окно *Create Netlist*

Для упаковки печатной платы необходимо перейти на дерево проектов и щелкнуть левой кнопкой мыши файл формата **dsn*. Выполнить команды *Tools/CreateNetlist /PCB Editor*.

В появившемся диалоговом окне *Create Netlist* необходимо выбрать опцию в графе создания платы *Createor Update PCB Editor Board* и в графе *Allow Etch Removal During ECO*.

В случае успеха, если при создании принципиальной электрической схемы не возникло никаких проблем, и она была создана правильно, откроется диалоговое окно со списком программ, в которых пользователь может открыть упакованную печатную плату (рисунок 24).

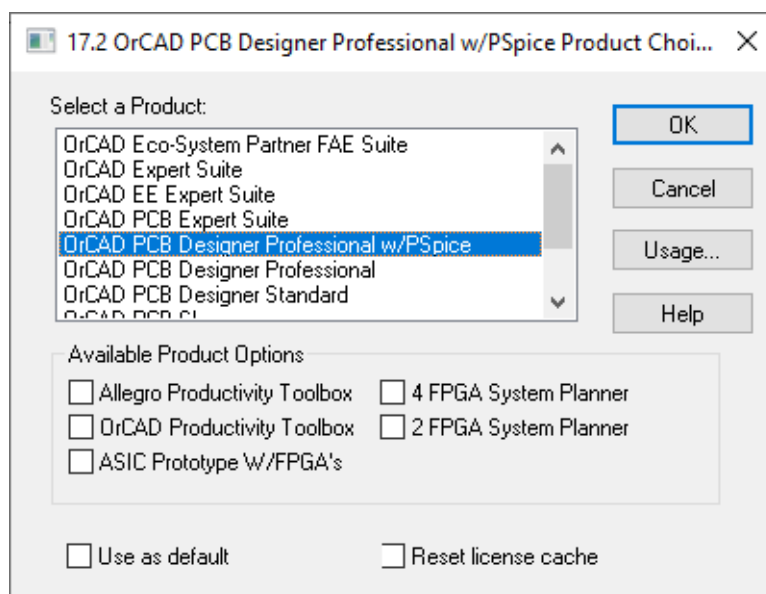


Рисунок 24 – Диалоговое окно с выбором программ

В открывшейся программе необходимо отобразить сетку и выбрать единицы измерения, в которых далее будет формироваться печатная плата. Для этого необходимо выполнить команды *Setup/DesignParameters/Design*. В качестве единиц измерения используются миллиметры. После выполняются команды *Setup/Grids*, чтобы настроить сетку. Шаг сетки был выбран на 1, 25 миллиметров. Для отображения сетки необходимо выбрать графе *Grids On*, чтобы отобразить сетку (рисунок 25).

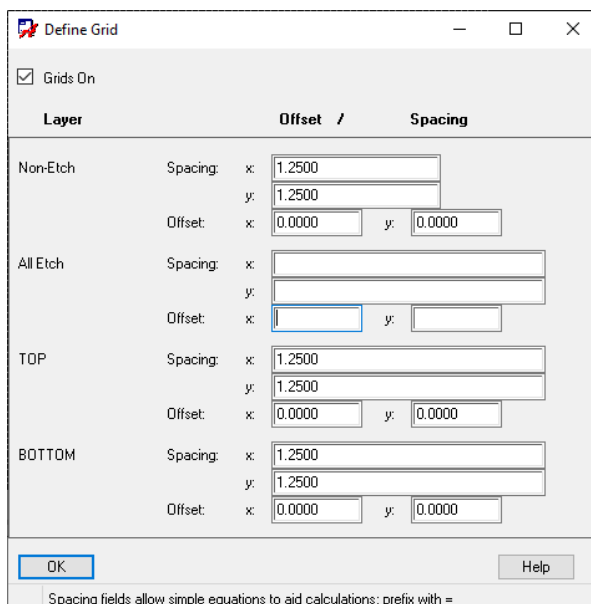


Рисунок 25 – Выбор шага сетки

Размеры платы подбираются вручную в соответствии с размерами элементов, формирующих данную плату, а также исходя из того, что элементы должны занимать от 80% всей поверхности платы. Для этого необходимо выбрать опцию *Design Parameter Editor*, и задать необходимые размеры в графе *Shape Rectangle* и разместить прямоугольник в рабочей области (рисунок 26).

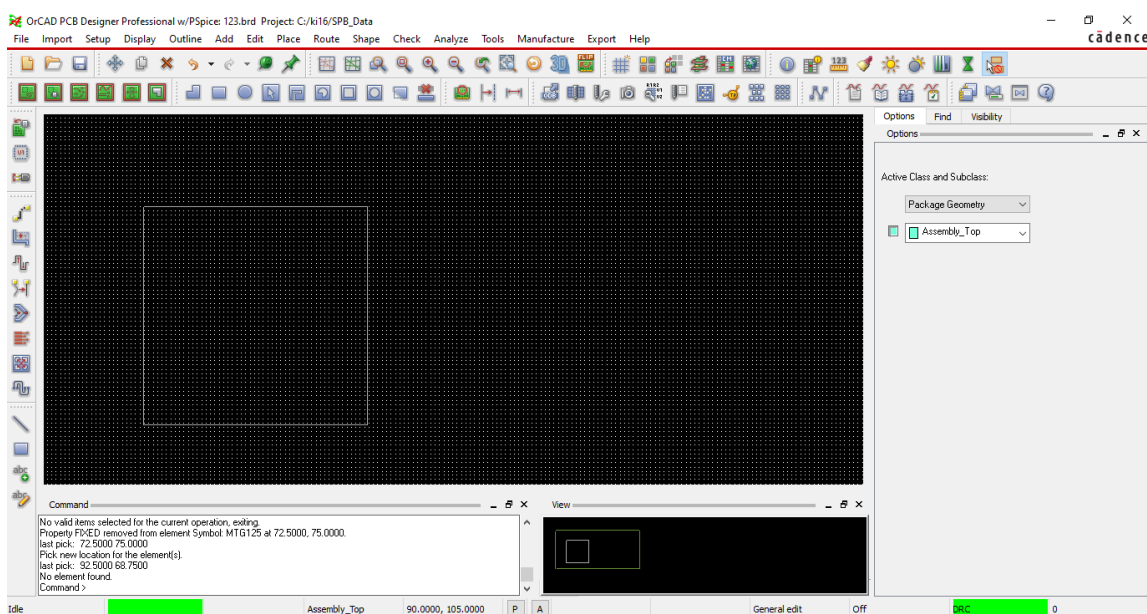


Рисунок 26 – Размещение платы

Выполнив команды *Place/MechanicalSymbol* производится размещение

монтажных отверстий на плате. После выбора нужного отверстия для размещения его на плате необходимо нажать на кнопку *Hide* (рисунок 27).

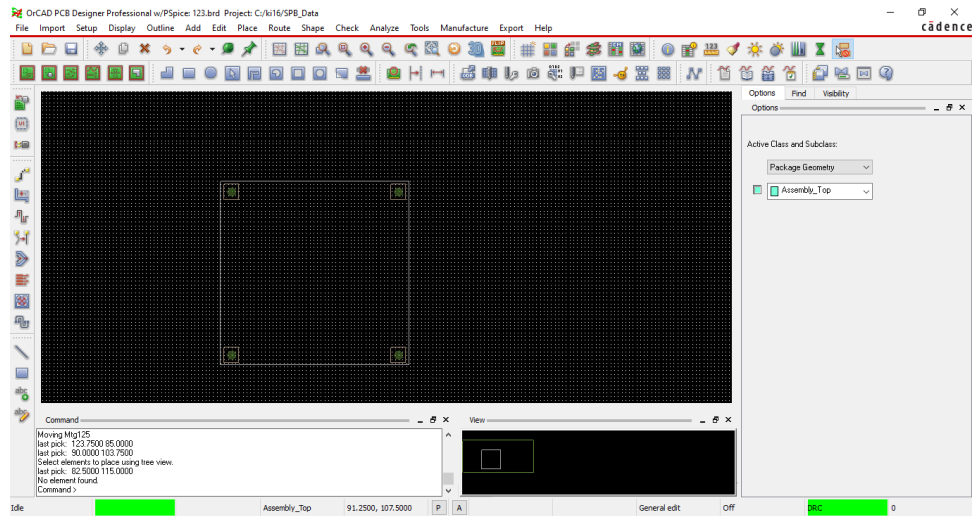


Рисунок 27 – Результат размещения на плате монтажных отверстий

Для размещения созданных ранее элементов требуется выполнить команды *Place/ComponentsManually*. В открывшемся диалоговом окне выбирается требуемый элемент и размещается на плате в нужном месте (рисунок 28).

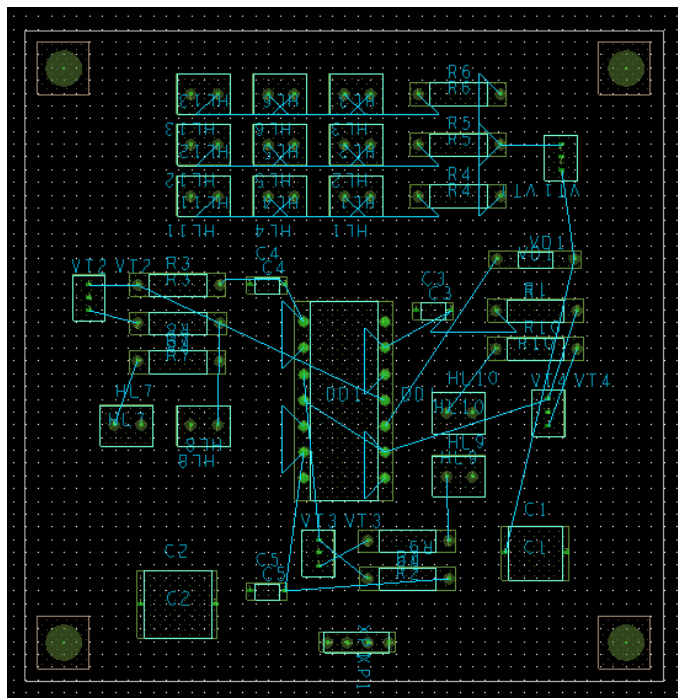


Рисунок 28 – Результат размещения элементов на плате

Для трассировки платы в *orCAD PCB Editor* выполняются команды *Setup/Constraints*. В открывшемся диалоговом окне устанавливается ширину цепей (рисунок 29). Для силовых – 0,75 миллиметров; для остальных цепей – 0,5 миллиметров.

Objects	Referenced Physical CSet	Line Width		Neck		Uncoupled Length
		Min mm	Max mm	Min Width mm	Max Length mm	
Type	S	Name				Gather Control
Net		N04319	0.5000	0.0000	0.1270	0.0000
Net		N04327	0.5000	0.0000	0.1270	0.0000
Net		N04423	0.5000	0.0000	0.1270	0.0000
Net		N04431	0.5000	0.0000	0.1270	0.0000
Net		N04435	0.5000	0.0000	0.1270	0.0000
Net		N04465	0.5000	0.0000	0.1270	0.0000
Net		N04517	0.5000	0.0000	0.1270	0.0000
Net		N04541	0.5000	0.0000	0.1270	0.0000
Net		N04715	0.5000	0.0000	0.1270	0.0000
Net		N04719	0.5000	0.0000	0.1270	0.0000
Net		N04723	0.5000	0.0000	0.1270	0.0000
Net		N04787	0.5000	0.0000	0.1270	0.0000
Net		N04791	0.5000	0.0000	0.1270	0.0000
Net		N04795	0.5000	0.0000	0.1270	0.0000
Net		N04799	0.5000	0.0000	0.1270	0.0000
Net		N05036	0.5000	0.0000	0.1270	0.0000
Net		N05040	0.5000	0.0000	0.1270	0.0000
Net		N05048	0.5000	0.0000	0.1270	0.0000
Net		N05116	0.5000	0.0000	0.1270	0.0000
Net		N05132	0.5000	0.0000	0.1270	0.0000
Net		N05148	0.5000	0.0000	0.1270	0.0000
Net		N05348	0.5000	0.0000	0.1270	0.0000
Net		N05364	0.5000	0.0000	0.1270	0.0000
Net		N05380	0.5000	0.0000	0.1270	0.0000
Net		N05436	0.5000	0.0000	0.1270	0.0000
Net		107	0.7500	0.0000	0.1270	0.0000
Net		114	0.7500	0.0000	0.1270	0.0000

Рисунок 29 – Выбор размера цепи

Выполним команды *Route/PCB Router/Route Automatic*. В появившемся окне задаются параметры, кратные размеру сетки, представленные на рисунке 30.

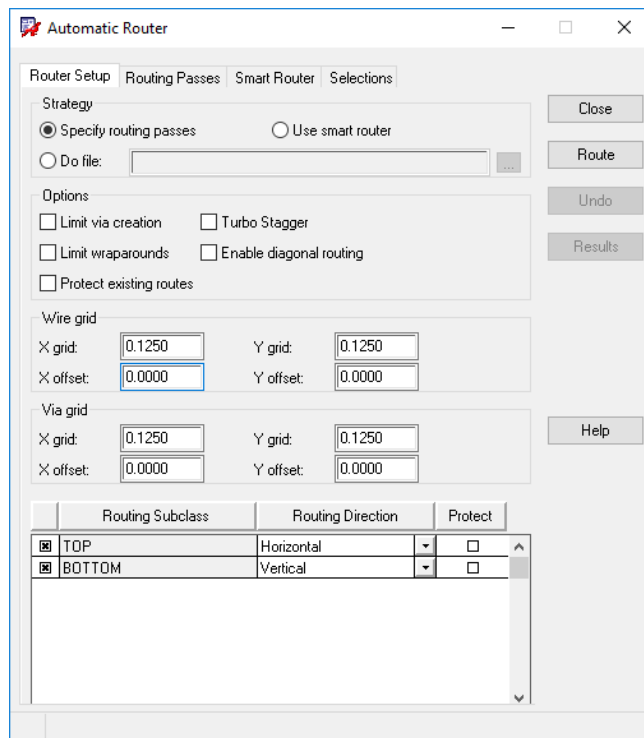


Рисунок 30 – Диалоговое окно *Route Automatic*

Для начала трассировки печатной платы необходимо выбрать опцию *Route* (рисунок 31).

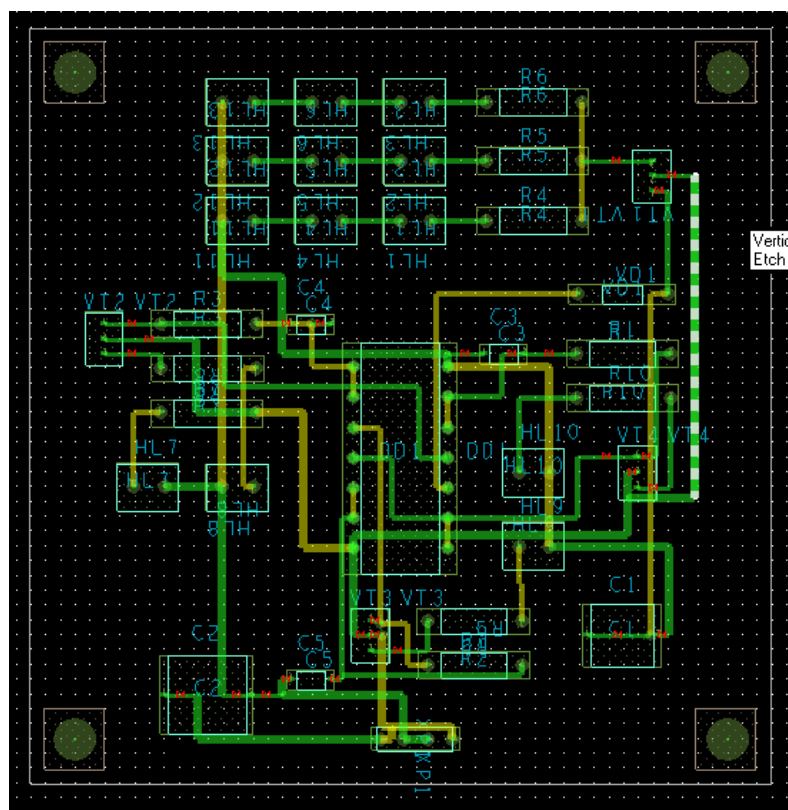


Рисунок 31 – Результат автотрассировки на плате

Для просмотра результатов в табличном виде выбирается команда *Result* (рисунок 32).

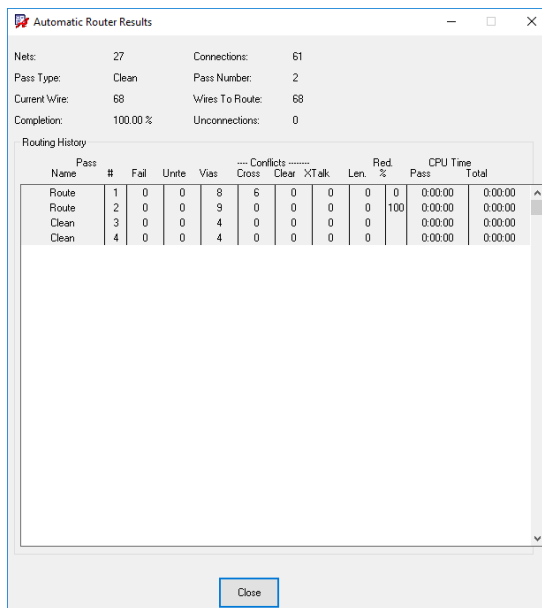


Рисунок 32 – Результат автотрассировки в табличном виде

Далее выполняются команды *Setup/User Preferences/Path/Library*. В данном окне указываются параметры *parampath*, *psmpath* и *steppath* указывается путь до директории с 3D - моделями расширения **.STEP*.

Для совмещения корпусов элементов и 3D-моделей необходимо выполнить команды *Setup/STEP Mapping*. В открывшемся диалоговом окне выберем корпус элемента и соответствующий ему файл (рисунок 33).

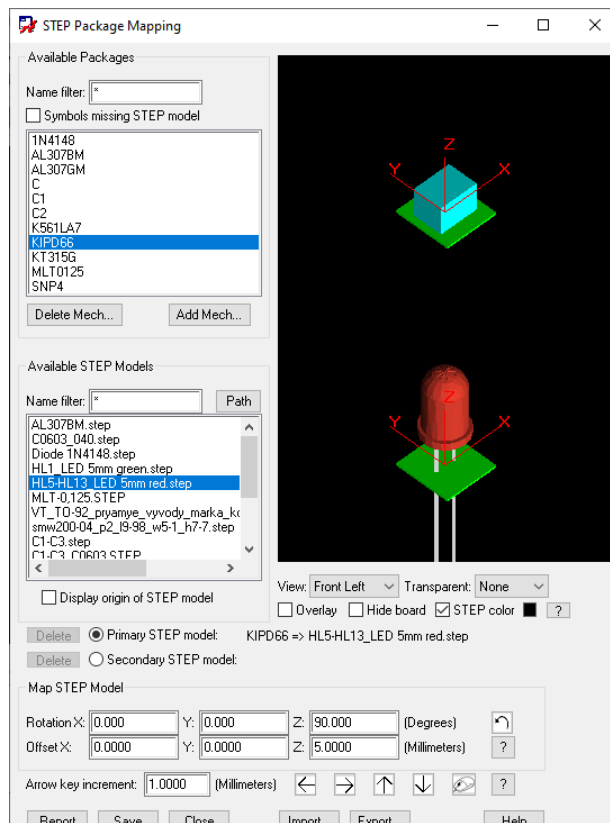


Рисунок 33 – Диалоговое окно *STEP Mapping*

По завершению совмещения корпусов с моделями необходимо выполнить команды – *Display/3D View* и сконвертировать 3D модель печатной платы (рисунок 34).

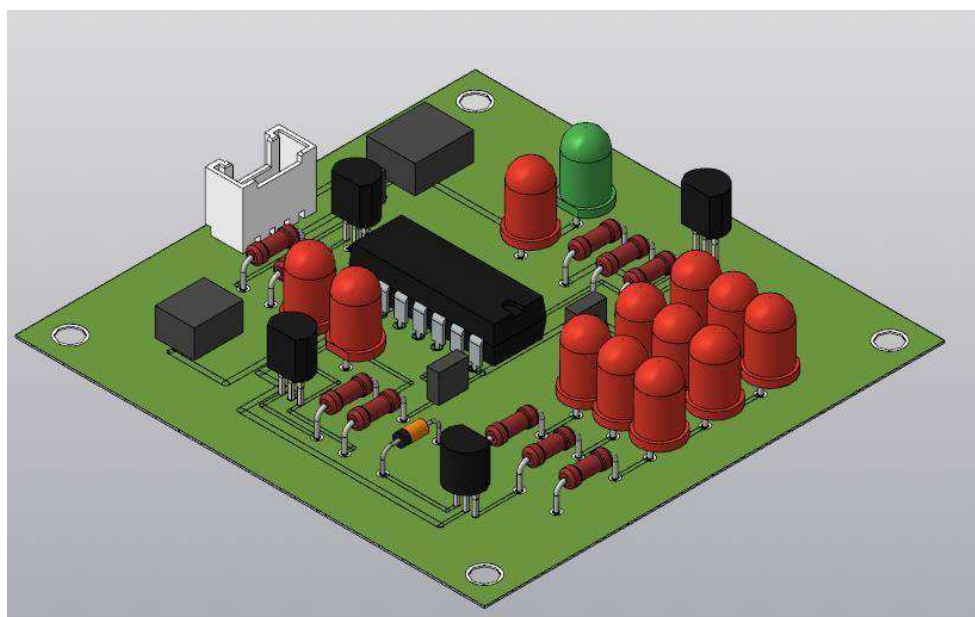


Рисунок 34 – Сконвертированная 3D модель

Разработанная 3D модель печатного узла является основой цифрового прототипа радиоэлектронного изделия при реализации цифровых технологий проектирования. Геометрическая модель печатного узла используется для решения задач полученных конструктивных решений: анализ тепловых характеристик, анализ механических характеристик.

3.2 Применение метода обратного размещения при разработке печатных узлов

Передача результатов размещения схемы на коммутационном поле в редактор печатных плат PCB Editor системы OrCAD осуществляется после упаковки печатной платы для заданного проекта.

Рассмотрим решение задачи размещения с использованием программно-реализованного метода обратного размещения для ряда заданных коммутационных схем. Первая коммутационная схема имеет 143 связи и 34 элемента, представленные рисунке 35.

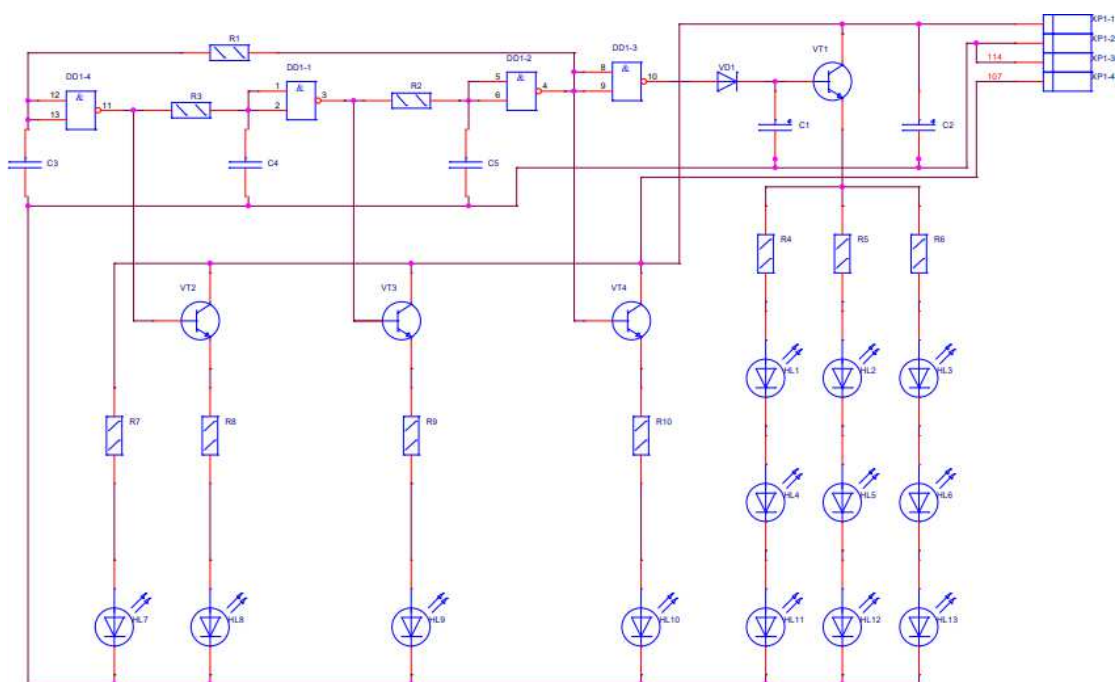


Рисунок 35 – Принципиальная электрическая схема

Файл списка соединений формата *calay 9064* представлен на рисунке 36.

```

107 VT2('2) VT4('2) R7('2) VT3('2) VT1('2) C2('1) DD1('7);
114 HL8('2) C1('2) C3('1) HL10('2) HL9('2) C4('1) C5('1) HL7('2),
    HL11('2) HL12('2) HL13('2) C2('2) DD1('14);
N04319 C3('2) R1('1) DD1('12) DD1('13);
N04327 VT4('1) R1('2) DD1('4) DD1('8) DD1('9);
N04423 R2('2) C5('2) DD1('5) DD1('6);
N04431 VD1('1) DD1('10);
N04435 C1('1) VD1('2) VT1('1);
N04465 R3('2) C4('2) DD1('1) DD1('2);
N04517 R2('1) VT3('1) DD1('3);
N04541 VT2('1) R3('1) DD1('11);
N04715 VT2('3) R8('2);
N04719 VT3('3) R9('2);
N04723 VT4('3) R10('2);
N04787 R7('1) HL7('1);
N04791 HL8('1) R8('1);
N04795 HL9('1) R9('1);
N04799 HL10('1) R10('1);
N05036 HL1('1) R4('1);
N05040 HL4('1) HL1('2);
N05048 HL4('2) HL11('1);
N05116 HL2('1) R5('1);
N05132 HL5('1) HL2('2);
N05148 HL5('2) HL12('1);
N05348 HL3('1) R6('1);
N05364 HL6('1) HL3('2);
N05380 HL6('2) HL13('1);
N05436 R4('2) VT1('3) R5('2) R6('2);

```

Рисунок 36 – Конфигурационный файл списка соединений

Вид матрицы соединений и матрицы расстояний в окне программы изображен на рисунке 37.

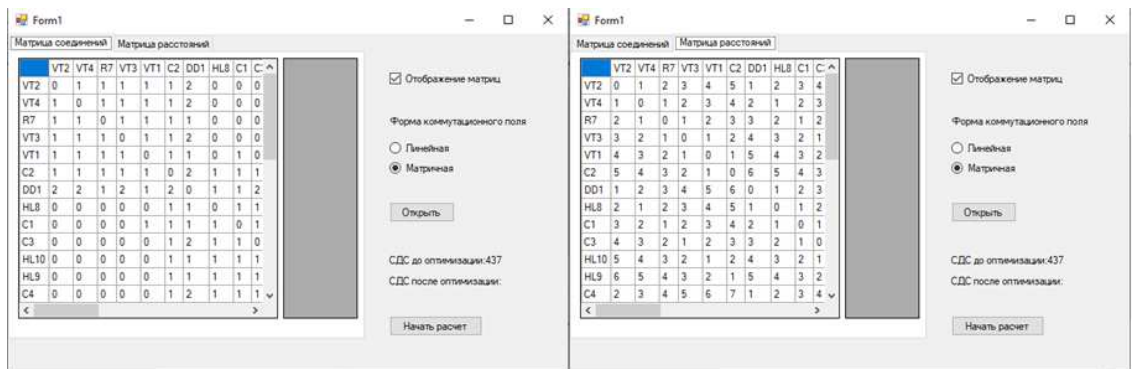


Рисунок 37 – Окно программы до оптимизации

Первичная расстановка выбранной формы коммутационного поля представлена на рисунке 38.

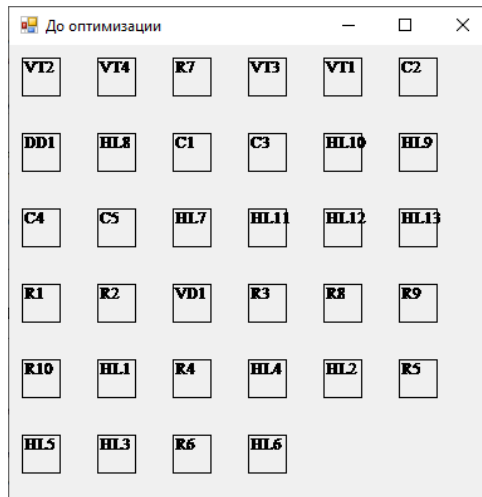


Рисунок 38 – Расстановка элементов на коммутационном поле

Преобразованные матрицы соединений и матрицы расстояний представлены на рисунке 39.

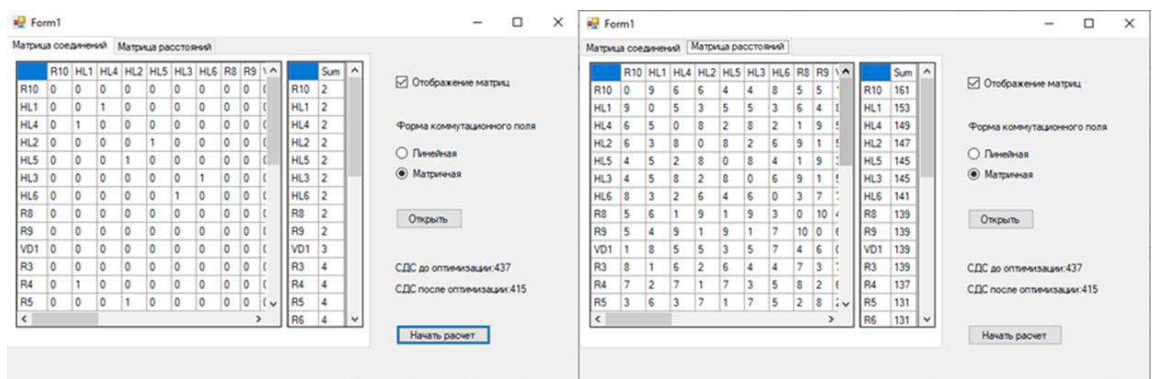


Рисунок 39 – Окно программы после оптимизации

Расстановка, выполненная по методу обратного размещения наглядно представлена на рисунке 40.

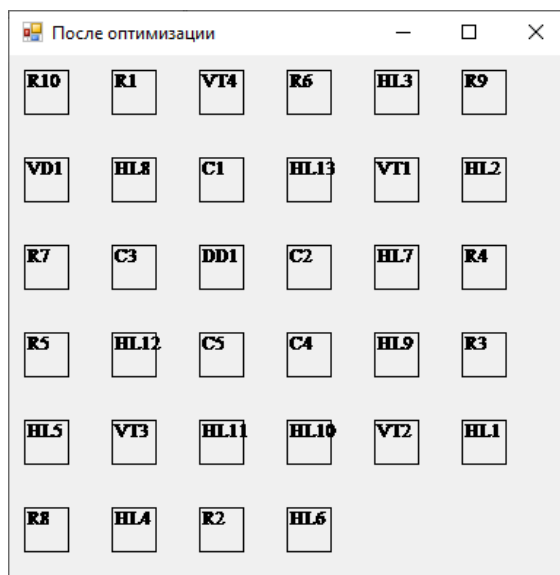


Рисунок 40 – Расстановка элементов на коммутационном поле

После получения вида коммутационного поля в PCB editor была построена соответствующая схема (рисунок 41).

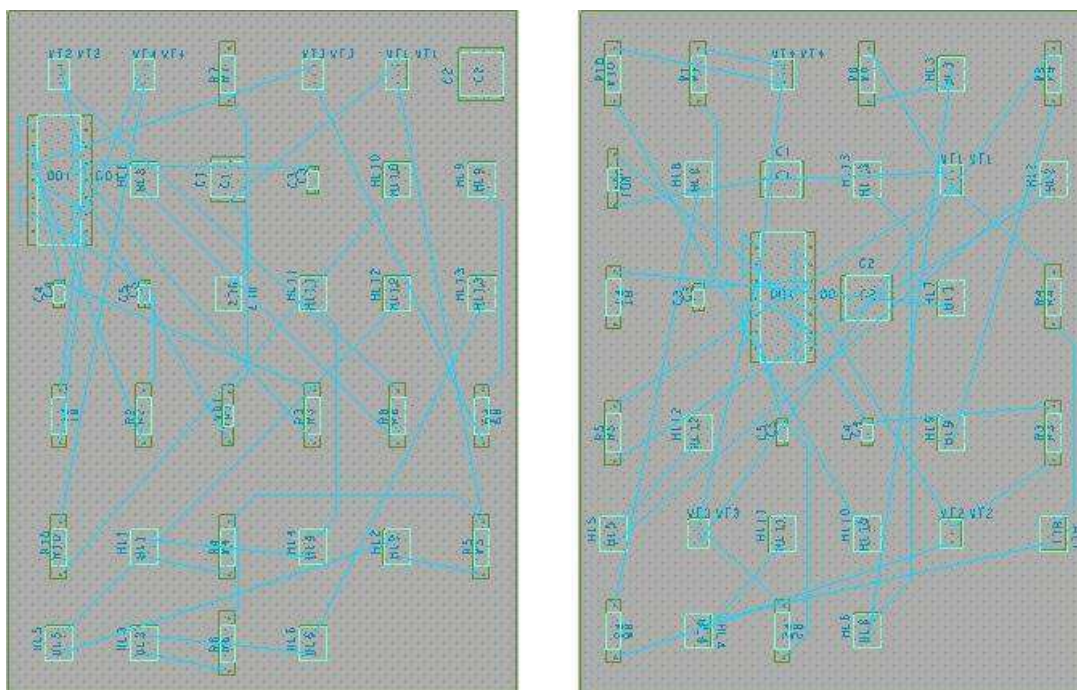


Рисунок 41 – Расстановка элементов до и после работы алгоритма

Конфигурация и длина проводников будет определена после решения задачи трассировки, выполненной с использованием автотрассировщика в составе данного редактора печатных плат (рисунок 42).

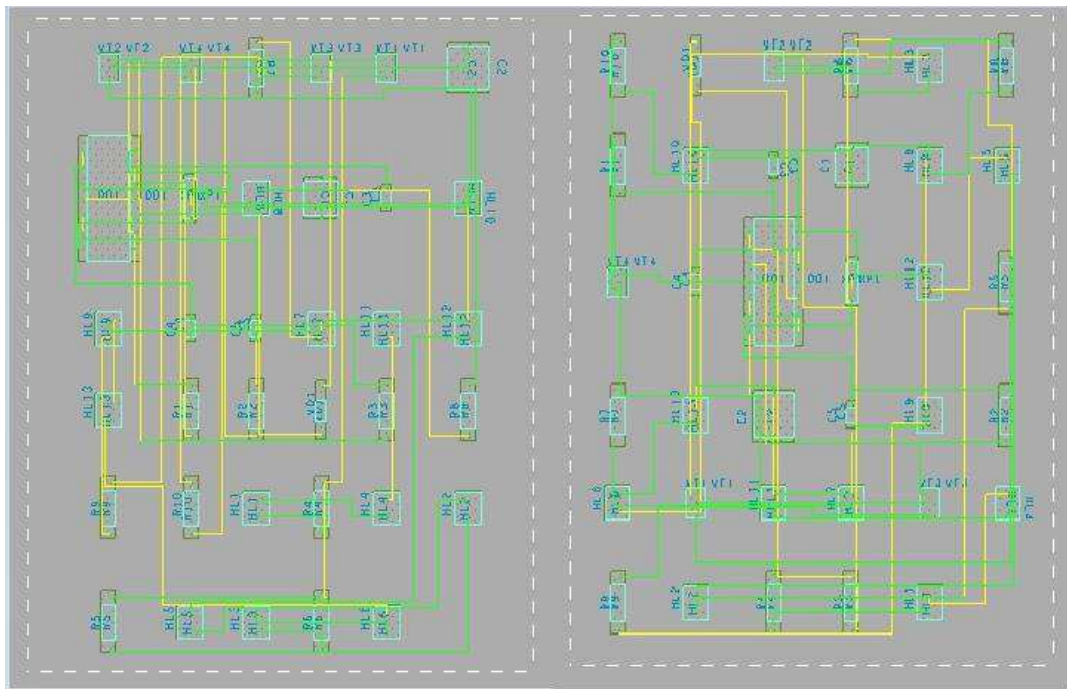


Рисунок 42 – Трассировка печатной платы

В результате работы алгоритма обратного размещения СДС уменьшилось на 7% (рисунок 43).

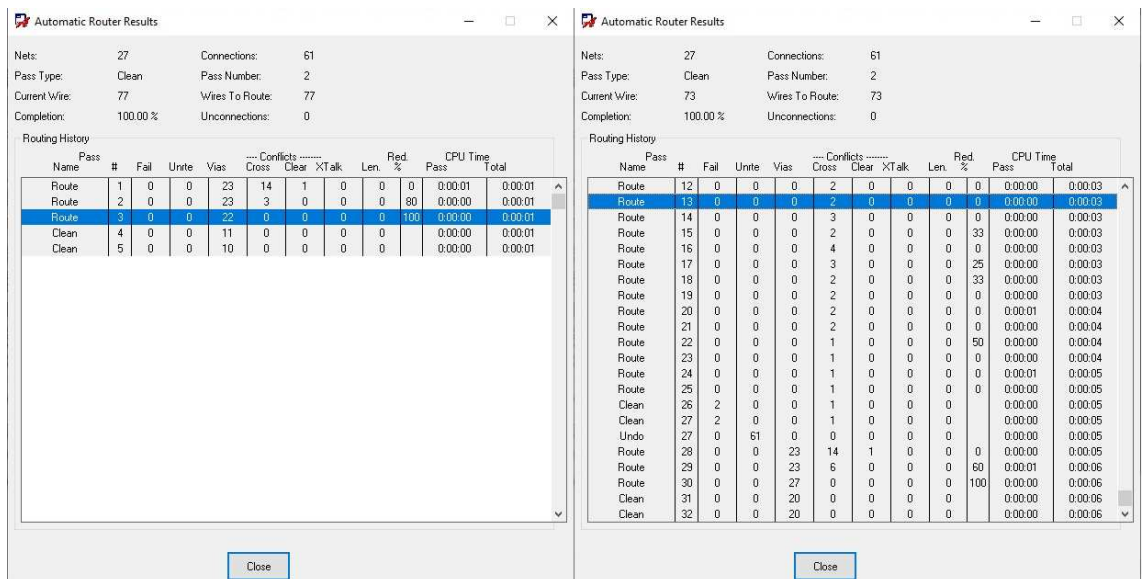


Рисунок 43 – Трассировка печатной платы

Вторая коммутационная схема имеет 83 связи и 23 элемента, представленные рисунке 44.

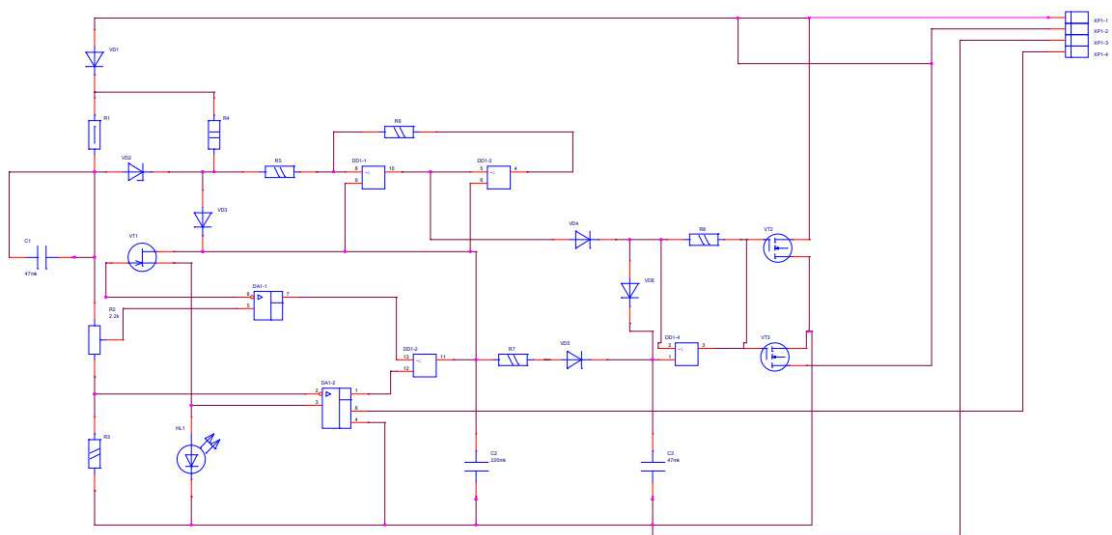


Рисунок 44 – Принципиальная электрическая схема

Файл списка соединений формата *calay 9064* представлен на рисунке 45.

```

107      DD1('7);
114      DD1('14);
N01439   C1('1) C1('2) R1('1) R2('2) VD2('1);
N01473   DA1('2) R2('1) R3('2);
N01835   R4('1) R5('1) VD2('2) VD3('1);
N01862   VT1('2) C2('2) DD1('9) DD1('11) DD1('6) R7('1) VD3('2);
N02161   VT1('1) DA1('6);
N02175   VT1('3) HL1('1) DA1('3);
N02373   DA1('5) R2('3);
N02718   DD1('8) R5('2) R6('1);
N02735   DD1('4) R6('2);
N02756   DD1('10) DD1('5) VD4('1);
N02895   R1('2) R4('2) VD1('2);
N02912   DA1('7) DD1('13);
N02930   DA1('1) DD1('12);
N03120   DD1('2) R8('1) VD4('2) VD6('1);
N03332   R7('2) VD5('1);
N03386   C3('2) DD1('1) VD5('2) VD6('2);
N03585   VT2('1) VT3('1) DD1('3) R8('2);
N03652   HL1('2) DA1('4) C2('1) VT2('3) VT3('2) C3('1) R3('1);
N04341   VT2('2) VT3('3) VD1('1);
N04626   DA1('8);

```

Рисунок 45 – Конфигурационный файл списка соединений

Вид матрицы соединений и матрицы расстояний в окне программы изображен на рисунке 46.

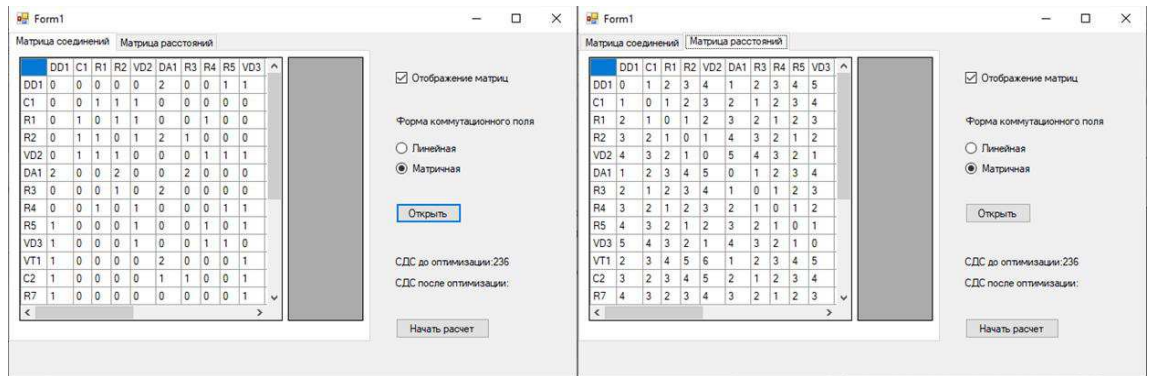


Рисунок 46 – Окно программы до оптимизации

Первичная расстановка выбранной формы коммутационного поля представлена на рисунке 47.

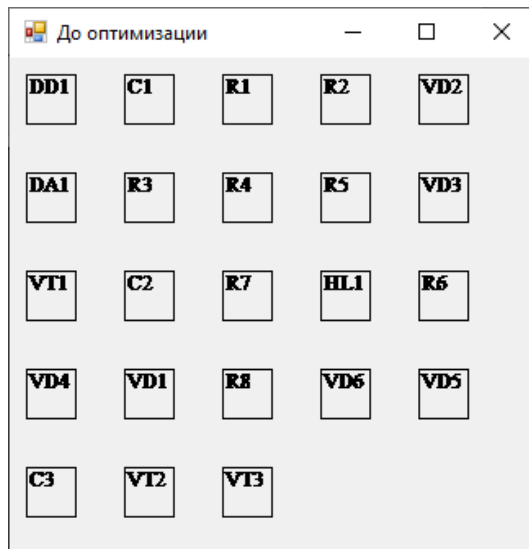


Рисунок 47 – Расстановка элементов на коммутационном поле

Преобразованные матрицы соединений и матрицы расстояний представлены на рисунке 48.

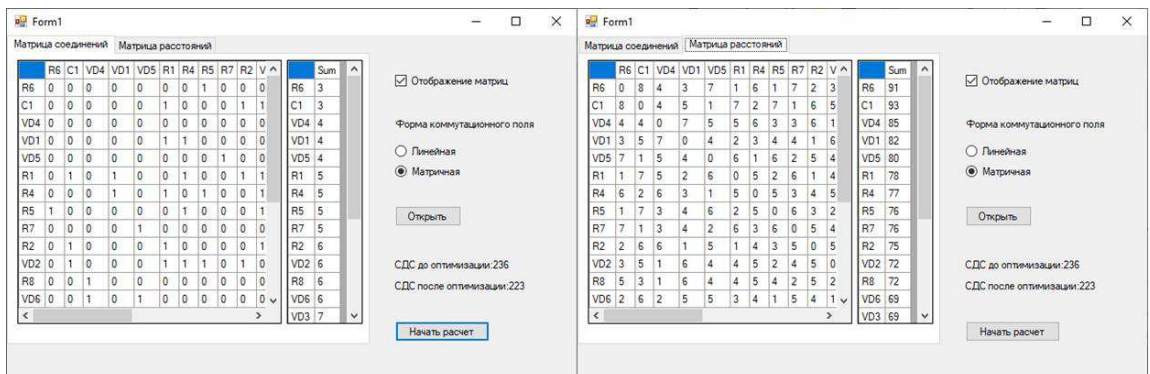


Рисунок 48 – Окно программы после оптимизации

Расстановка, выполненная по методу обратного размещения наглядно представлена на рисунке 49.

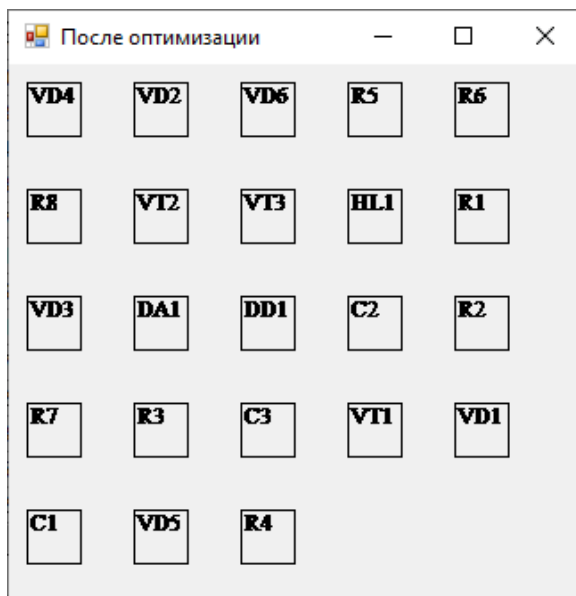


Рисунок 49 – Расстановка элементов на коммутационном поле

После получения вида коммутационного поля в PCB editor была построена соответствующая схема (рисунок 50).

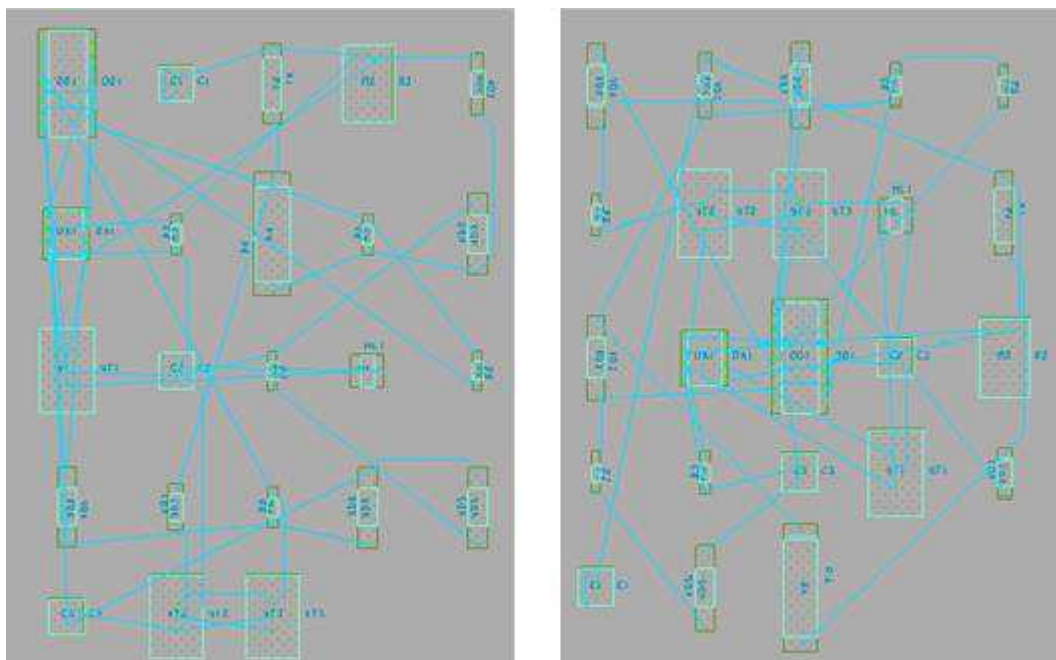


Рисунок 50 – Расстановка элементов до и после работы алгоритма

Конфигурация и длина проводников будет определена после решения задачи трассировки, выполненной с использованием автотрассировщика в

составе данного редактора печатных плат (рисунок 51).

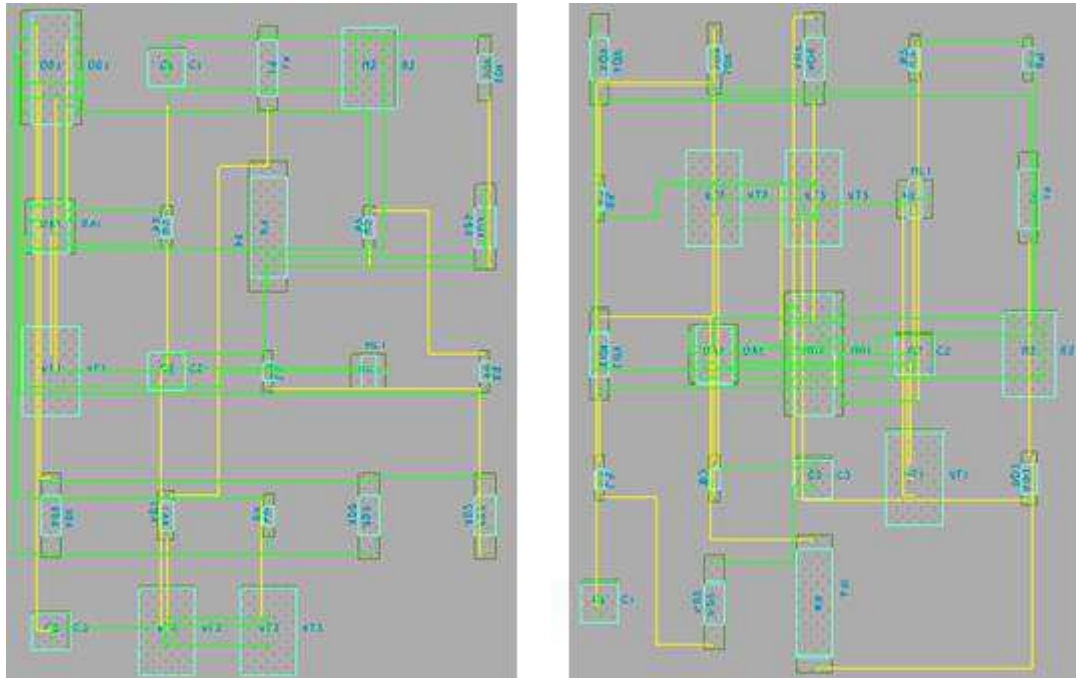


Рисунок 51 – Трассировка печатной платы

В результате работы алгоритма обратного размещения СДС уменьшилось на 6% (рисунок 52).

Automatic Router Results										
Nets:		22		Connections:		46				
Pass Type:		Clean		Pass Number:		2				
Current Wire:		63		Wires To Route:		63				
Completion:		100.00 %		Unconnections:		0				
Routing History										
Name	Pass #	Fail	Unrte	Vias	Conflicts Cross	Clear	XTalk	Len.	Red. %	CPU Time Pass Total
Route	1	0	0	18	17	0	0	0	0	0:00:00 0:00:00
Route	2	0	0	22	3	0	0	0	82	0:00:00 0:00:00
Route	3	0	0	24	0	0	0	0	100	0:00:01 0:00:01
Clean	4	0	0	15	0	0	0	0	0	0:00:00 0:00:01
Clean	5	0	0	16	0	0	0	0	0	0:00:01 0:00:02
Undo	5	0	46	0	0	0	0	0	0	0:00:00 0:00:02
Route	6	0	0	18	17	0	0	0	0	0:00:00 0:00:02
Route	7	0	0	19	6	0	0	0	64	0:00:01 0:00:03
Route	8	0	0	20	0	0	0	0	100	0:00:00 0:00:03
Clean	9	0	0	15	0	0	0	0	0	0:00:01 0:00:04
Clean	10	0	0	15	0	0	0	0	0	0:00:00 0:00:04

Automatic Router Results										
Nets:		22		Connections:		46				
Pass Type:		Clean		Pass Number:		2				
Current Wire:		59		Wires To Route:		59				
Completion:		100.00 %		Unconnections:		0				
Routing History										
Name	Pass #	Fail	Unrte	Vias	Conflicts Cross	Clear	XTalk	Len.	Red. %	CPU Time Pass Total
Route	28	0	0	20	2	0	0	0	84	0:00:01 0:00:12
Route	29	0	0	23	0	0	0	0	100	0:00:00 0:00:12
Clean	30	0	0	13	0	0	0	0	0	0:00:01 0:00:13
Clean	31	0	0	12	0	0	0	0	0	0:00:00 0:00:13
Undo	31	0	46	0	0	0	0	0	0	0:00:00 0:00:13
Route	32	0	0	17	18	0	0	0	0	0:00:00 0:00:13
Route	33	0	0	21	5	0	0	0	72	0:00:00 0:00:13
Route	34	0	0	24	0	0	0	0	100	0:00:00 0:00:13
Clean	35	0	0	14	0	0	0	0	0	0:00:01 0:00:14
Clean	36	0	0	16	0	0	0	0	0	0:00:00 0:00:14
Undo	36	0	46	0	0	0	0	0	0	0:00:00 0:00:14
Route	37	0	0	20	15	1	0	0	0	0:00:00 0:00:14
Route	38	0	0	21	3	0	0	0	81	0:00:00 0:00:14
Route	39	0	0	21	0	0	0	0	100	0:00:00 0:00:14
Clean	40	0	0	14	0	0	0	0	0	0:00:00 0:00:14
Clean	41	0	0	12	0	0	0	0	0	0:00:01 0:00:15
Undo	41	0	46	0	0	0	0	0	0	0:00:00 0:00:15
Route	42	0	0	18	18	0	0	0	0	0:00:01 0:00:16
Route	43	0	0	22	5	0	0	0	72	0:00:00 0:00:16
Route	44	0	0	26	0	0	0	0	100	0:00:00 0:00:16
Clean	45	0	0	15	0	0	0	0	0	0:00:00 0:00:16
Clean	46	0	0	14	0	0	0	0	0	0:00:00 0:00:16

Рисунок 52 – Трассировка печатной платы

Третья коммутационная схема имеет 176 связи и 26 элементов, представленные рисунке 53.

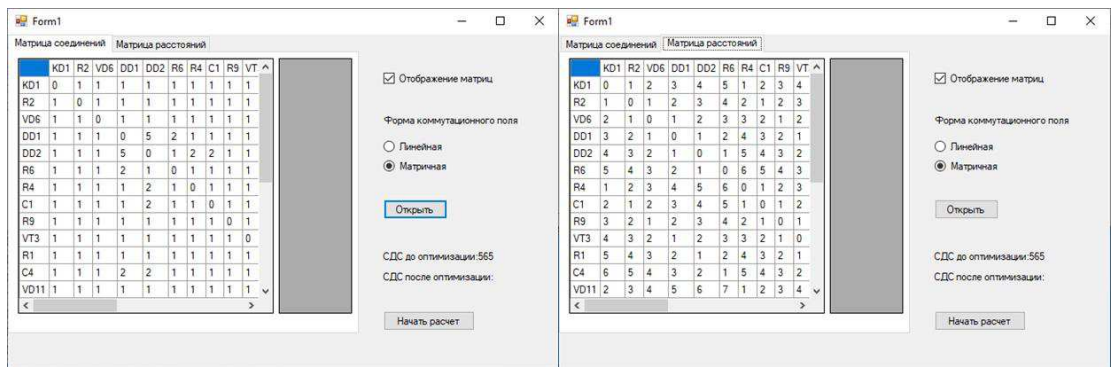


Рисунок 55 – Окно программы до оптимизации

Первичная расстановка выбранной формы коммутационного поля представлена на рисунке 56.



Рисунок 56 – Расстановка элементов на коммутационном поле

Преобразованные матрицы соединений и матрицы расстояний представлены на рисунке 57.

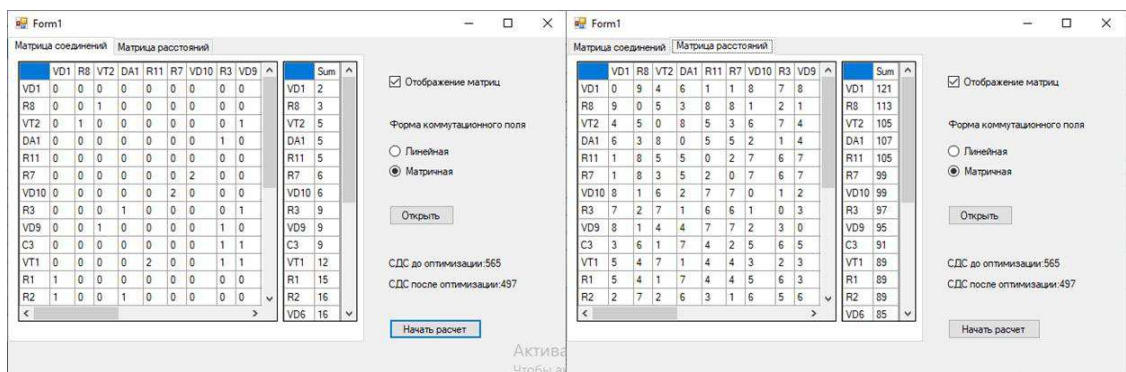


Рисунок 57 – Окно программы после оптимизации

Расстановка, выполненная по методу обратного размещения наглядно представлена на рисунке 58.

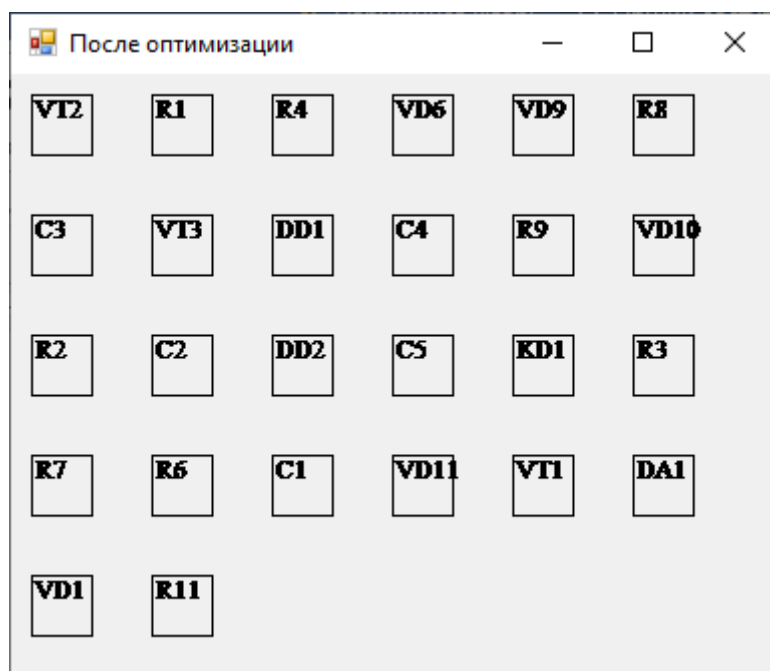


Рисунок 58 – Расстановка элементов на коммутационном поле

После получения вида коммутационного поля в PCB editor была построена соответствующая схема (рисунок 59).

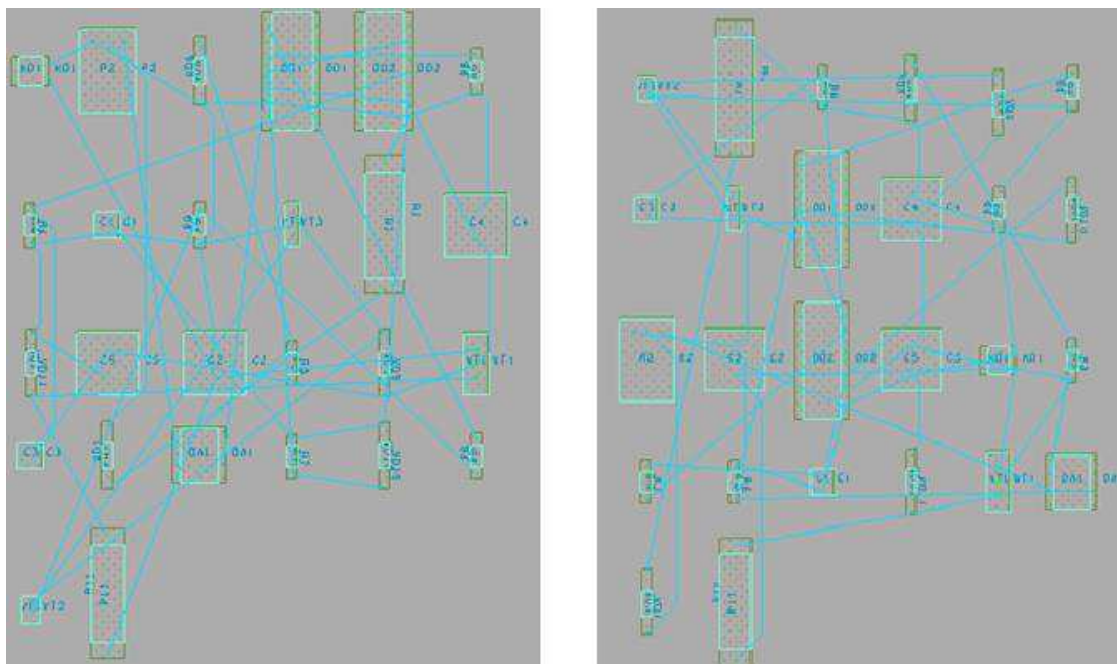


Рисунок 59 – Расстановка элементов до и после работы алгоритма

Конфигурация и длина проводников будет определена после решения

задачи трассировки, выполненной с использованием автотрассировщика в составе данного редактора печатных плат (рисунок 60).

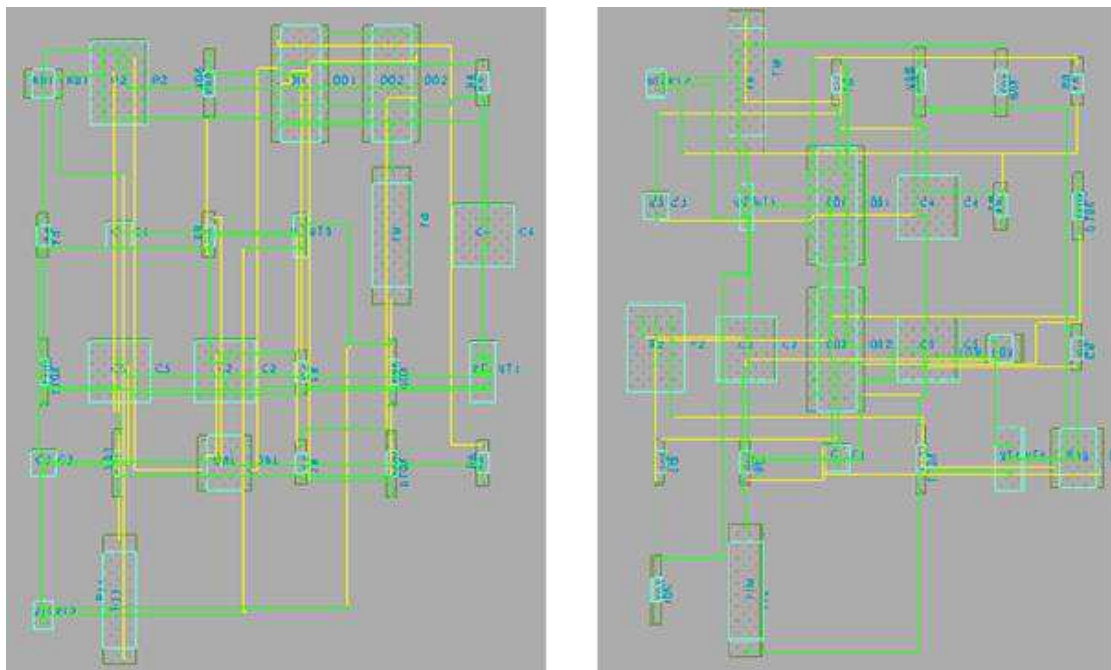


Рисунок 60 – Трассировка печатной платы

В результате работы алгоритма обратного размещения СДС уменьшилось на 5% (рисунок 61).

Automatic Router Results													
Nets:		20		Connections:		49							
Pass Type:		Clean		Pass Number:		2							
Current Wire:		69		Wires To Route:		69							
Completion:		100.00 %		Unconnections:		0							
Routing History													
Name	Pass #	Fail	Unrte	Vias	Conflicts			Len	Red %	CPU Time	Total		
					Cross	Clear	XTalk			Pass			
Route	1	0	0	22	18	2	0	0	0	0.0000	0.0000		
Route	2	0	0	26	5	0	0	0	75	0.0000	0.0000		
Route	3	0	0	28	0	0	0	0	100	0.0000	0.0000		
Clean	4	0	0	21	0	0	0	0	0	0.0000	0.0000		
Clean	5	0	0	20	0	0	0	0	0	0.0000	0.0000		
Undo	5	0	49	0	0	0	0	0	0	0.0000	0.0000		
Route	6	0	0	22	25	1	0	0	0	0.0001	0.0001		
Route	7	0	0	25	10	0	0	0	61	0.0000	0.0001		
Route	8	0	0	32	0	0	0	0	100	0.0000	0.0001		
Clean	9	0	0	20	0	0	0	0	0	0.0000	0.0001		
Clean	10	0	0	18	0	0	0	0	0	0.0001	0.0002		
Undo	10	0	49	0	0	0	0	0	0	0.0000	0.0002		
Route	11	0	0	24	21	0	0	0	0	0.0000	0.0002		
Route	12	0	0	27	8	0	0	0	61	0.0000	0.0002		
Route	13	0	0	30	0	0	0	0	100	0.0000	0.0002		
Clean	14	0	0	22	0	0	0	0	0	0.0000	0.0002		
Clean	15	0	0	20	0	0	0	0	0	0.0001	0.0003		

Рисунок 61 – Трассировка печатной платы

Таким образом интеграция реализованного метода обратного размещения с OrCAD PCB editor позволила получить выигрыш в СДС на 5-10% при

проектировании печатных узлов. На базе предложенного подхода возможна реализация других методов размещения, которые позволят проектировщикам радиоэлектронных систем и комплексов выбирать оптимальный метод и тем самым обеспечить автоматическое размещение компонентов коммутационной схемы. Разработанное программное решение позволяет положить начало формированию библиотеки методов размещения, интегрированной с редактором PCB editor.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был произведен анализ методов и алгоритмов задачи размещения, рассмотрены их критерии.

Пройдены этапы проектирования радиоэлектронного устройства в системе автоматизированного проектирования OrCAD от построения принципиальной схемы и до получения цифрового прототипа печатного узла.

Программно реализован метод обратного размещения с использованием объектно-ориентированного языка программирования C#.

Реализована интеграция метода обратного размещения с системой автоматизированного проектирования OrCAD через обработку файла списка соединений, генерируемым программным модулем Capture CIS, состоящим в пакете программ OrCAD. Программное решение для размещения позволяет визуализировать матрицу соединений, матрицу состояний и вид коммутационного поля.

Анализ результатов трассировки на коммутационном поле с размещением элементов до оптимизации и после показывает, что при использовании реализованного метода размещения для конструкций в виде печатного узла с заданными коммутационными схемами СДС улучшается на 5-10%.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Норенков, И.П. Основы автоматизированного проектирования / И.П. Норенков. – Москва: И.П МГТУ им. Н.Э. Баумана, 2002. – 336 с.
- 2 Кергроуч, С. Индустрия 4.0: новые вызовы и возможности для рынка труда / С. Кергроуч // Форсайт. – 2017. – № 4. – С. 6–8.
- 3 Муромцев, А.Ю. Математическое обеспечение САПР / А.Ю. Муромцев, И.В. Тюрин. – Москва: Лань, 2014. – 464 с.
- 4 Россохин, А. Виртуальное счастье или виртуальная зависимость / А. Россохин, В. Измагурова. – Москва: Смысл, 2004. – 255 с.
- 5 Свон, М. Блокчейн: схема новой экономики / М. Свон. – Москва: Олимп-бизнес, 2017. – 295 с.
- 6 Коатанея, Е. Печатная электроника, настоящее и будущее / Е. Коатанея, В. Кантола, Дж. Куловеси, Л. Лахти, Р. Лин, М. Заводчикова. – Хельсинки: Университет технологии Хельсинки, 2009. – 286 с.
- 7 СТО 4.2 07 2014. Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Взамен СТО 4.2 07 2012; дата введ.09.01.2014. – Красноярск, 2014. – 60с.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра «Системы автоматики, автоматизированное управление
и проектирование»




УТВЕРЖДАЮ
Заведующий кафедрой
С.В. Ченцов

« 26 » 06 2020 г.

БАКАЛАВРСКАЯ РАБОТА

15.03.04 – Автоматизация технологических процессов и производств

**РЕШЕНИЕ ЗАДАЧИ РАЗМЕЩЕНИЯ ПРИ ЦИФРОВИЗАЦИИ
ПРОЕКТИРОВАНИЯ РАДИОЭЛЕКТРОННЫХ ИЗДЕЛИЙ**

Руководитель		26.06.2020 г.	доцент, канд. техн. наук Е.Е. Носкова
Выпускник		26.06.2020 г.	И.В. Паршинцев
Нормоконтролер		26.06.2020 г.	Т.А. Грудинова

Красноярск 2020