Верификация электрических и геометрических моделей с помощью алгоритмов проверки графов на изоморфизм

А.С.Варфоломеев

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

asvarfolomeev@stud.etu.ru

А.В.Лепов

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

avlepov@stud.etu.ru

Аннотация. Работа посвящена представлению результатов исследования авторов в области схемотопологической верификации ячеек КМОП БИС. Исследовались и разрабатывались методы и средства верификации на основе различных алгоритмов анализа изоморфизма графов. Приводятся результаты оценки эффективности программного обеспечения на основе разных алгоритмов.

Ключевые слова: электрическая схема; топология; верификация; изоморфизм

I. Введение

Верификация ячеек БИС (Больших Интегральных Схем) является критически важным этапом разработки, поскольку без неё невозможно гарантировать корректность логического функционирования И соответствие физической реализации заданным требованиям. Отсутствие своевременной верификации может привести к серьёзным функциональным сбоям, дефектам производства и значительным затратам на переработку проекта. Особенно это актуально на этапе интеграции ячеек в более крупные функциональные блоки, где даже незначительные ошибки могут масштабироваться и влиять на работоспособность всей системы.

Выделяют два ключевых направления верификации ячеек БИС:

- технологическая верификация это проверка соответствия топологии (геометрических моделей) ячеек технологическим правилам, установленным для выбранного производственного процесса. Она включает, например, контроль ширины и длины каналов транзисторов и расстояний между элементами ячейки БИС и других параметров, критичных для успешного изготовления кристалла;
- схемо-топологическая верификация это процесс сопоставления электрической геометрической моделей ячейки (электрической схемы и топологии). В ходе верификации проверяется, соответствует ли топология заданной схеме, и выявляются возможные расхождения, которые

С. Э. Миронов

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

semironovspb@yandex.ru

могут привести к функциональным сбоям или снижению производительности. Цель – убедиться, что все соединения, элементы и их взаимосвязи реализованы корректно.

Учитывая постоянный рост количества проектируемых БИС, увеличение их размеров и сложности, а также сжатые сроки выхода на рынок, автоматизация процессов верификации становится не просто желательной, а необходимой. Метолы автоматизированной верификации позволяют существенно снизить вероятность ошибок, ускорить цикл проектирования и обеспечить более высокую надёжность конечного изделия [1].

Существуют различные подходы к автоматизации схемо-топологической верификации. В последнее время появились публикации о использовании для этого нейросетей. А именно графовых нейронных сетей (GNN) [2]–[3] и гиперграфовых сверточных сетей (HGCN) [4]. Эти методы показывают высокую скорость работы, но при этом не способны гарантировать точное соответствие схем и топологии, так как работают на основе обучающих выборок и вероятностных моделей.

Также одним из перспективных направлений автоматизации схемо-топологической верификации остаётся применение алгоритмов поиска изоморфизма графов [5]. В отличии от подхода на основе нейросетей алгоритмы поиска изоморфизма графов позволяют точно и формально сопоставить топологическую и схемную реализации.

В рамках этого подхода процесс проверки корректности топологии заключается в сравнении графа, построенного на основе извлечённой из топологии схемы ячейки, с графом, соответствующим её электрической схеме. Таким образом, задача верификации может быть формализована и разделена на три ключевых этапа: построение графа топологической ячейки, построение графа электрической схемы и применение алгоритмов поиска изоморфизма для выявления соответствий между графами.

Такой подход обеспечивает формальное масштабируемое решение задачи соответствия схемной и

топологической реализаций, что особенно важно при работе с большими ячейками, содержащими десятки и сотни компонентов.

Статья посвящена изложению описанию этапов проектирования и изложению результатов и применения для сравнительного анализа разработанных средств.

II. Этапы схемо-топологической верефикации

А. Построение графа топологии ячейки

Исходными данными о топологии ячейки служат её описание на одном из языков описания топологии (ЯОТ). Авторы в своих исследованиях использовали текстовый язык описания топологии CIF (Caltech Intermediate Form) [6]. Формат CIF описывает топологические элементы в виде примитивов: многоугольниками (как правило ортогональных), задаваемые координатами их вершин) и трассами, задаваемые ширине и угловыми точками их осевой линии.

Каждой фигуре сопоставлен слой, соответствующий технологическому уровню полупроводниковой структуры. Описания в разработанной авторами САПР использует следующие обозначения слоёв: М1 – первый (нижний) слой металлизации, SI – слой поликремния (затвор транзистора), РА – активная область р-типа, NA – активная область п-типа, а также дополнительные слои, используемые для контактов и других конструктивных элементов.

Путём анализа комбинаций этих геометрических объектов осуществляется распознавание элементов топологии. Например: транзистор р-типа идентифицируется как пересечение слоя РА (р-область) и SI (поликремний), в месте их перекрытия образуется затвор – как показано на рис. 1*a*.

Соединение транзисторов с шинами, и соединения между шинами определяются наличием контакта в области пересечения соответствующих слоёв (например, SI с M1, PA с M1 и т. п.). На рис. 16 область PA соединена с шиной, проходящей в слое нижнего металла M1, через контакт СРА.



Рис. 1. Примеры основных типов выявляемых элементов топологии: а) транзистор р-типа; б) контакт между РА и М1.

Работу алгоритма преобразования топологии ячейки в граф рассмотрим на примере топологии ячейки логического «2И». На рис. 2 приведена её трёхмерная визуализация.



Рис. 2. Трехмерное представление топологии ячейки «2И». Цветами на рисунке обозначены: бирюзовый – шины верхнего металла, пурпурный – шины нижнего металла, зеленый – шины в слое поликремния, серый – затворы транзисторов, красный – области п-типа, синий - области р-типа.

Механизм преобразования топологии в граф начинается с разбора геометрических объектов в различных технологических слоях (металлы, поликремний, активные области транзисторов и др.) и извлечения функциональных элементов схемы – транзисторов (по пересечениям затворов и активных областей), контактов между слоями (по контактным окнам) и шин.

После анализа топологии выполняется логическая агрегация соединений:

- объединяются электрически связанные шины одного слоя,
- устанавливаются связи между транзисторами, контактами и шинами объединяются электрически связанные шины,
- формируется схемо-топологический граф.

Далее схемо-топологический граф преобразуется в граф электрической схемы. Для этого применяются процедуры объединения/устранения избыточных элементов графа, таких как электрически связанные шины в разных слоях и соединяющие их контакты.

Эти шаги позволяют упростить и нормализовать граф перед его экспортом в удобный для описания графов формат JSON (в этот же формат преобразуется описание электрической схемы).

Преобразованная в граф, топология ячейки «2И» представлена на рис. 3.



Рис. 3. Граф ячейки «2И», построенный по описанию топологии в формате CIF

На рис. 3 вершины N0, N1, N2 – это п-транзисторы; P3, P4, P5 – р-транзисторы; префикс M1 имеют шины нижнего слоя металла; M2 – верхнего; SI – соединения в слое поликремния; С – непосредственно соединённые стоковые и истоковые области транзисторов.

В процессе исследования для каждой тестовой ячейки проводилась проверка на соответствие её топологии и построенной по топологии электрической схемы.

В. Построение графа электрической схемы

Построение графа электрической схемы основывается на представлении транзисторов в виде вершин графа и соединений между ними в виде рёбер. Рассматривались три основных подхода к такому представлению (рис. 4).

В первом варианте тип транзистора не указывается явно (рис. 46), но канал представляется как отдельная вершина графа, соединённая с компонентом транзистора. При этом тип транзистора может быть определён по подключению канала: если он соединяется с линией питания (VCC), то это р-тип, а если с землёй (GND), то птип.

Во втором варианте транзисторы описываются без явного выделения канала, но с указанием их типа (n- или p-тип), как показано на puc. 4*в*. Это позволяет сократить число проверок при анализе. Кроме того, такой способ позволяет сделать описание графа схемы более компактным и пригодным для последующего сопоставления с топологическим графом.





Также существует вариант представления, где транзистор описывается тремя связанными вершинами: стоком, истоком и затвором [7]. В этом варианте транзистор моделируется как связанная тройка узлов, что может повысить точность описания физических соединений, однако приводит к усложнению графовой структуры и увеличению количества элементов, что затрудняет последующую обработку и сопоставление с топологией. Таким образом, второй вариант представления, где транзисторы задаются одной вершиной с указанием типа, является оптимальным для задач схемо-топологической верификации. Он обеспечивает баланс между информативностью и компактностью, снижает вычислительную сложность алгоритмов изоморфизма в автоматизированных системах анализа.

Для описания электрической схемы используется Netlist текстовое представление, в котором зафиксированы электрические соединения между компонентами схемы. Netlist содержит не только, какие компоненты присутствуют в схеме, к каким узлам они полключены. но И содержит электрические характеристики (длина и ширина канала и прочее). Это описание служит основой для построения графа электрической схемы, в котором компоненты отображаются как вершины, а соединения между ними как рёбра.

На рис. 5 приведён пример электрической схемы ячейки «2И» и соответствующего ей Netlist описания.



Рис. 5. Представление ячейки «2И»: a) Netlist описание, б) электрическая схема

Алгоритм построение графа электрической схемы на основе Netlist-файла реализован следующим образом:

- каждый транзистор считывается из файла,
- на основе подключений к его истоку, стоку и затвору создаются шины – логические узлы схемы,
- структура схемы сохраняется в JSON-файл в формате графа, который можно использовать для визуализации и анализа.

Вершинами графа являются транзисторы и шины, а рёбра показывают, какие элементы соединены между собой.

Граф электрической схемы ячейки «2И» показан на рис. 6.



Рис. 6. Граф ячейки «2И», построенный по электрической схеме, где TP1, TP2, TP3 – р-транзисторы; TN4, TN5, TN6 – п-транзисторы; a, b, s – шины входных/выходных сигналов; p1, p2 – шины, соединяющие транзисторы; VCC – питание; GND – земля

С. Проверка изоморфизма графов

Существует множество алгоритмов анализа изоморфизма графов. В своей работе авторы ориентировались на четыре считающиеся одними из наиболее эффективных.

В 2015 году Ласло Бабай предложил алгоритм для проверки графов на изоморфизм с квазиполиномиальной сложностью [8]. Изначально в алгоритме были найдены ошибки, но позже они были исправлены, и к 2025 году он считается практически доказанным и признанным, хотя его исследование все еще продолжается. В данной работе рассматривается один из вариантов алгоритма Ласло Бабая.

Алгоритм Ласло Бабая строит дерево возможных изоморфизмов с использованием теории групп и рекурсивного деления графа на симметричные подструктуры. Он уменьшает пространство поиска, выделяя каноническое представление графа и анализируя его автоморфизмы.

Алгоритм Вейсфейлера–Лемана [9] использует итеративную раскраску вершин для различения графов. Он не всегда определяет изоморфизм точно, но эффективен на практике и применяется в машинном обучении на графах.

Алгоритм VF2 [10] реализован в библиотеке NetworkX и основан на рекурсивном поиске в глубину с поддержанием частичных отображений и проверкой ограничений. Он эффективно исключает неподходящие сопоставления, и особенно эффективен на малых и средних графах, особенно в реальных приложениях.

Алгоритм Nauty-Traces [11] разработан Бренданом Маккеем на основе алгоритма Nauty. Он находит канонические формы графов и их автоморфизмы, используя методы теории групп и оптимизированное упорядочивание. Тraces расширяет возможности, улучшая производительность на графах с высокой симметрией. Оба алгоритма считаются одними из самых мощных для точной проверки изоморфизма.

В табл. І и на рис. 7 представлены результаты исследований временной эффективности рассмотренных алгоритмов.

Количеств	Время выполнения алгоритма			
о вершин графа	Вейсфейлер а-Лемана	Nauty- Traces	VF2 (мс.)	Ласло Бабая
10	(MC.) 0.13	<u>(мс.)</u> 0.078	0.082	(MC.) 0.026
100	1,01	0,369	0,095	0,0654
1000	7,51	2,923	0,252	0,6389
10000	14,2389	5,4557	1,477	7,1271
100000	1803,192	624,283	18,472	89,257
1000000	19224,1	6320,8	776,35	1048,8

ТАБЛИЦА I.



Рис. 7. График сравнения времени выполнения алгоритмов

Как видно из графиков, реализованный алгоритм Ласло Бабая демонстрирует более высокую скорость обработки. Однако в диапазоне от 100 до 1000000 вершин графа его производительность становится ниже, чем у оптимизированного алгоритма VF2 из пакета Networkx. А алгоритмы Nauty-Traces и Вейсфейлера–Лемана, в свою очередь, уступают по быстродействию алгоритму Ласло Бабая.

III. САПР схемо-топологической верификации

В ходе проведенных исследований были разработаны программные средства, позволяющие:

- загрузить графы топологии и электрической схемы;
- выбрать алгоритм проверки изоморфизма графов;
- выполнить анализа графов на изоморфизм;
- визуализировать результаты анализа изоморфизма графов.

Использование алгоритмов поиска изоморфного подграфа в сочетании с процедурой обратного свёртывания топологического графа в промежуточное представление позволяет детектировать локальные несоответствия между схемами. Таким образом, можно определить конкретный элемент, например, отсутствующий контакт, вызвавший расхождение в графовых моделях.

В качестве примера на рис. 8 и рис. 9 представлены окна программы проверки изоморфизма для изоморфных и неизоморфных графов соответственно на примере рассматривавшейся выше ячейки «2И».

Сравнение графа, извлечённого из топологии ячейки «2И» (рис. 3) и графа электрической схемы (рис. 6) с использованием алгоритма VF2 показало их изоморфизм, что свидетельствует об идентичности логических функций, реализуемых соответствующими схемами (рис. 8).

Однако при внесении намеренного дефекта в топологию, например, удалении контакта между транзистором Р5 и шиной нижнего слоя металлизации М1_26 (рис. 3), нарушается структура соединений, что приводит к утрате изоморфизма между графами. Результат подобного изменения представлен на рис. 9.



Рис. 8. Пример сравнения графа электрической схемы и топологии



Рис. 9. Пример сравнения графа электрической схемы и топологии с одним удаленным ребром

IV. ЗАКЛЮЧЕНИЕ

В работе рассмотрен подход к автоматизированной схемо-топологической верификации ячеек БИС, основанный на графовом представлении электрических и топологических структур. Построены алгоритмы формирования графов по данным форматов CIF и Netlist, обеспечивающие единое формализованное представление схемы на разных уровнях абстракции.

В ходе исследования были изучены и реализованы различные алгоритмы проверки графов на изоморфизм, включая алгоритмы Вейсфейлера–Лемана, Nauty-Traces, VF2 и упрощённый вариант алгоритма Ласло Бабая. Для каждого из них приведены характеристики и результаты сравнения по скорости работы на графах различного размера. Это позволило провести всестороннюю оценку их применимости в задачах схемо-топологической верификации.

Показано, что применение алгоритмов поиска изоморфизма графов обеспечивает эффективное средство для формальной проверки соответствия топологии и электрической схемы, а также может быть расширено до выявления локальных ошибок с помощью алгоритмов поиска изоморфного подграфа и методов восстановления промежуточного представления топологического графа.

Список литературы

[1] Миронов С. Э., Андреев Л. Е. Схемная верификация топологии ячеек БИС // Известия СПбГЭТУ «ЛЭТИ». 2013. № 2. С. 38–40.

- [2] Alrahis L., et al. GNN-RE: Graph Neural Networks for Reverse Engineering of Gate-Level Netlists // IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 2022. Vol. 41, № 8. P. 2435–2448.
- [3] Teliti A. Graph neural networks for topology recognition of ams integrated circuits: Ph.D. dissertation. Torino: Politecnico di Torino, 2023.
- [4] Li B., Wang S., Chen T., Sun Q., Zhuo C. Efficient Subgraph Matching Framework for Fast Subcircuit Identification // Proc. 6th ACM/IEEE Symp. Machine Learning for CAD (MLCAD). Salt Lake City (Snowbird), USA, 2024. P. 1–7.
- [5] Ohlrich M., Ebeling C., Ginting E., Sather L. SubGemini: Identifying Subcircuits Using a Fast Subgraph Isomorphism Algorithm // Proc. 30th Int. Design Automation Conf. (DAC'93). New York: ACM, 1993. P. 31–37.
- [6] Li C., Wang J., Xu D., Gao Y. A Study on Optimized Layout Transformation Algorithm // Proc. Int. Conf. Anti-Counterfeiting, Security and Identification (ASID). Shanghai, China, 2013. P. 1–4.
- [7] Dong G., Zheng Y., He S., Guo D., Li L. Subcircuit Identification Method Based on Subgraph Isomorphism // Proc. 15th IEEE Int. Conf. Anti-counterfeiting, Security and Identification (ASID). Xiamen, China, 2021. P. 97–101.
- [8] Babai L. Graph Isomorphism in Quasipolynomial Time: Extended Abstract // Proc. 48th Annu. ACM Symp. Theory of Computing (STOC'16). New York: ACM, 2016. P. 684–697.
- [9] Зыков А. А. Темы в теории графов. М.: [издательство], 1990. С. 191–192.
- [10] NetworkX Developers, "VF2 Algorithm NetworkX 1.11 Documentation." [Online]. Available: https://networkx.org/documentation/networkx-1.11/reference/algorithms.isomorphism.vf2.html. [Accessed: Apr. 7, 2025].
- [11] McKay B. D., Piperno A. Nauty and Traces User's Guide (Version 2.5). Canberra: Computer Science Department, Australian National University, 2013.