

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ / COMPUTER MODELING AND DESIGN AUTOMATION

DOI: <https://doi.org/10.23670/IRJ.2024.139.26>

ПОСТРОЕНИЕ R*-ДЕРЕВА НА ОСНОВЕ ГЕОМЕТРИЧЕСКИХ ДАННЫХ ИЗ GDSII ФАЙЛА

Научная статья

Штанюк А.А.^{1,*}, Семенов А.О.²

¹ORCID : 0000-0003-1809-7173;

²ORCID : 0000-0002-1556-7988;

^{1,2}Нижегородский государственный технический университет им. Р.Е. Алексеева, Нижний Новгород, Российская Федерация

* Корреспондирующий автор (ashtanyuk[at]nntu.ru)

Аннотация

В настоящее время топологии интегральных схем становятся все более сложными: уменьшаются размеры элементов, увеличивается количество элементов и связей между ними. Количество элементов в топологиях ИС может достигать миллиардов, что затрудняет их анализ на стадиях проектирования и разработки. Особое внимание уделяется проблеме производительности операций навигации по топологии ИС. В данной статье приведен алгоритм трансформации геометрических данных из GDSII файла в R*-дерево. Данный вид деревьев хорошо зарекомендовал себя в индексации многомерной информации, такой как прямоугольники, географические координаты или полигоны. Это позволяет просто и быстро выполнять различные навигационные задачи и получать необходимую информацию об элементах.

Ключевые слова: GDSII, топология интегральных схем, ИС, анализ, алгоритм, R*-дерево.

R*-TREE CONSTRUCTION BASED ON GEOMETRIC DATA FROM GDSII FILE

Research article

Shtanyuk A.A.^{1,*}, Semenov A.O.²

¹ORCID : 0000-0003-1809-7173;

²ORCID : 0000-0002-1556-7988;

^{1,2}Nizhny Novgorod State Technical University, Nizhny Novgorod, Russian Federation

* Corresponding author (ashtanyuk[at]nntu.ru)

Abstract

Nowadays, topologies of integrated circuits are becoming more and more complex: the size of elements is decreasing, the number of elements and connections between them is increasing. The number of elements in IC topologies can reach billions, which makes it difficult to analyse them at the design and development stages. Special attention is paid to the problem of performance of IC topology navigation operations. This article presents an algorithm for transforming geometric data from a GDSII file into an R*-tree. This type of tree is well established in indexing multidimensional information such as rectangles, geographic coordinates or polygons. It allows to perform various navigational tasks simply and quickly and to retrieve the necessary information about elements.

Keywords: GDSII, integrated circuit topology, IC, analysis, algorithm, R*-tree.

Введение

При проектировании интегральных схем наиболее популярным форматом является GDSII, разработанный Calma Company (США). В течение многих лет GDSII был единственным в своем роде форматом для хранения и передачи данных по интегральным схемам и их топологиям. Вследствие этого многие производители стали активно использовать его в своих системах. Хотя Calma обновляла формат по мере развития своих систем САПР, они сохранили обратную совместимость, так что ни один файл GDSII не устарел.

Основной недостаток формата GDSII заключается в отсутствии оптимизации для уменьшения размера файла, вследствие чего проекты часто превышают размер в 20 гигабайт. Более подробную информацию о проблеме оптимизации файла, размерах и других недостатках GDSII можно найти в статье [1].

Серьезные недостатки формата GDSII стали основной причиной возникновения альтернативного формата для хранения топологий ИС – Open Artwork System Interchange Standard (OASIS), который более подробно рассмотрен в статье [2]. Основным преимуществом формата является то, что при использовании OASIS появляется возможность уменьшения размера файлов, вследствие чего время их обработки и загрузки сокращается. Данное преимущество основывается на том, что в OASIS существует множество различных опций для представления одних и тех же данных, однако количество возможных ошибок в файле OASIS резко возрастает по сравнению с GDSII (как минимум в 4 раза).

Некоторые компании начинают переходить на формат OASIS, однако большинство продолжает использовать GDSII. Проблема размера файлов GDSII в таких компаниях обычно решается с помощью увеличения емкости диска и оперативной памяти. Это считается более выгодной мерой по сравнению с заменой квалифицированного потока, основанного на GDSII, на новый, основанный на OASIS.

Файлы GDSII хранят данные в бинарном виде, эти данные содержат информацию о плоских геометрических фигурах (полигонах), различных текстовых метках и других данных касательно топологии интегральных схем в

иерархической форме. Эти данные также можно использовать для изменения всей архитектуры интегральной схемы или какой-то ее конкретной части, которая может быть использована для обмена данными между макетами или при конструировании фотошаблонов (фотомасок).

Файлы GDSII представляют из себя набор ячеек, которые содержат различные данные о полигонах. Также из-за того, что данные в GDSII могут быть представлены в иерархической форме, ячейки могут содержать ссылки на другие ячейки. Ячейки в GDSII представляют собой записи STRUCTURE и имеют буквенно-цифровые имена до 32 символов, которые однозначно их определяют.

Файл начинается с библиотеки этих структур, где запись заголовка библиотеки (BGNLIB) определяет начало, а запись конца библиотеки (ENDLIB) определяет конец файла. Между записями начала и конца библиотеки располагается последовательность структур, где каждая структура сама состоит из записи заголовка структуры (BGNSTR), последовательности элементов и записи конца структуры (ENDSTR).

Элементы в GDSII файлах разделяются на семь видов [3], [4]:

1) **граница** представляет собой заполненный многоугольник. Начинается с записи BOUNDARY и имеет обязательные атрибуты LAYER, DATATYPE и XY;

2) **путь** представляет собой незамкнутую фигуру с ненулевой шириной, которая обычно используется для размещения проводов. Этот элемент иницируется записью PATH и имеет обязательные атрибуты LAYER, DATATYPE и XY;

3) **ссылка** (SREF) на структуру представляет собой иерархический элемент, который вызывает все элементы из структуры, на которую он ссылается, с измененными координатами в зависимости от атрибутов;

4) **ссылка на массив** структур (AREF) представляет собой иерархический элемент, который вызывает массив структур и все элементы внутри каждой структуры с измененными координатами в зависимости от атрибутов;

5) **текст** (TEXT) предназначен для документации и имеет обязательные атрибуты LAYER, TEXTTYPE и XY;

6) **узел** (NODE) определяет электрический путь (сеть) и имеет обязательные атрибуты LAYER, NODETYPE и XY;

7) **прямоугольник** (BOX) представляет собой прямоугольную геометрию и имеет обязательные атрибуты LAYER, BOXTYPE и XY.

Как упоминалось ранее, данные топологий ИС содержат ячейки, сигналы, порты и другие элементы схемы в виде геометрических объектов, чаще всего полигонов. Увеличение сложности ИС увеличивает объем информации о топологии, из-за чего точность и скорость навигации для поиска целевого объекта становится серьезной проблемой.

Таким образом, навигация по макету проявляется как проблема пространственного поиска, которую можно решить с помощью доступных методов геометрических запросов. В вычислительной геометрии существует множество алгоритмов, которые имеют большое значение, особенно в автоматизированном проектировании. Так, в работе [5] задача эффективного вычисления пересечения нескольких наложенных друг на друга фигур решается с помощью двух алгоритмов плоской прогонки. Многие другие подходы предлагают методы доступа на основе индексов для быстрого извлечения элементов данных в соответствии с их пространственным расположением.

Наиболее популярным методом доступа является метод R-дерева Гутмана [6], основанный на оптимизации площади охватывающих прямоугольников в каждом внутреннем узле. Несколько других вариантов классического R-дерева также предложены для оптимизации алгоритмов R-дерева Гутмана. Упакованные R-деревья [7], [8], R*-дерево [9], R-дерево Грина [10] относятся к числу этих вариантов, предлагающих различные подходы к оптимизации загрузки дерева. Среди них наиболее примечательным является R*-дерево, в котором введены новые параметры настройки для устранения проблем, существующих в классических R-деревьях.

Алгоритм построения R*-дерева на основе геометрических данных из GDSII файла

Благодаря тому, что данные GDSII имеют иерархическую структуру, можно описывать миллионы или миллиарды полигонов в ИС. Иерархия очень важна, потому что в дизайне ИС, как правило, повторно используются одни и те же геометрические фигуры. Благодаря этому можно создавать компактные файлы для очень больших конструкций [11].

Однако не существует эмпирических правил структурных организаций топологий интегральных схем. В связи с отсутствием точных правил проектирования, для извлечения иерархической информации из файла GDSII и построения сбалансированного дерева необходимо привести иерархическую топологию GDSII файлов в плоский вид.

Вычисление новых координат геометрических элементов GDSII

В формате GDSII иерархию определяют два элемента: SREF (элемент ссылки на структуру) для ссылки или размещения структуры и AREF (элемент ссылки на массив структуры) для ссылки на массив структур. Обязательными атрибутами данных элементов являются: SNAME, который хранит информацию структуры, на которую ссылается данный элемент, и XY, который хранит координаты в плоскости для смещения структуры. Для AREF также обязателен элемент COLROW, который хранит информацию про сетку ссылок на структуру. Необязательными атрибутами данных элементов являются STRANS, MAG, ANGLE, которые определяют следующие преобразования:

- Отражение: если флаг отражения установлен в атрибуте STRANS, структура отражается относительно оси x. Отражение осуществляется перед угловым поворотом. Отражение может быть достигнуто путем умножения матрицы отражения в (1),

$$R_f = \begin{pmatrix} 1 & 0 \\ 0 & r \end{pmatrix} \quad (1)$$

где r равно -1 в случае отражения. Значение r по умолчанию равно 1.

- Увеличение: если установлен атрибут MAG, структура увеличивается в соответствии с коэффициентом увеличения (значением данного атрибута). Это достигается умножением координат на матрицу увеличения, заданную в (2),

$$M = \begin{pmatrix} K & 0 \\ 0 & K \end{pmatrix} \quad (2)$$

где K — коэффициент увеличения. Значение K по умолчанию равно 1.

- Вращение: если установлен атрибут `ANGLE`, структура поворачивается в соответствии с углом поворота (значением данного атрибута). Это достигается предварительным умножением точек на матрицу вращения, заданную в (3),

$$R_o = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad (3)$$

где, α угол поворота. Его значение по умолчанию равно 0.

- Перемещение: Это самый важный шаг из всех наборов преобразований, именно поэтому атрибут `XU` является обязательным. Здесь каждая из точек перемещается с использованием значения атрибута `XU`. Перемещение может быть достигнуто добавлением вектором перемещения, указанного в (4),

$$t = \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (4)$$

где, $t_{x,y}$ является перемещением в плоскости. Значение по умолчанию — 0.

Все преобразования выполняются в определенной последовательности: отражение, увеличение, вращение и перемещение. Матрица преобразований, которая объединяет все модификации изменения местоположения точек на плоскости представлена в (5),

$$H = R_o * M * R_f + t = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} K & 0 \\ 0 & K \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & r \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (5)$$

Итоговые координаты точек на плоскости можно найти по (6),

$$\begin{pmatrix} \vec{x}' \\ \vec{y}' \end{pmatrix} = \begin{pmatrix} K \cos(\alpha) & -Kr \sin(\alpha) & t_x \\ K \sin(\alpha) & Kr \cos(\alpha) & t_y \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (6)$$

Применяя к координатам выражение (6), можно вычислить новые координаты геометрических объектов, которые были вызваны элементом `SREF`. Однако для вычисления новых координат геометрических элементов, которые были вызваны элементом `AREF`, необходимо помимо умножения точек на (6) добавлять трансляционный вектор решетки `AREF` согласно (7). Вектор смещения рассчитывается с использованием вектора строки и столбца, которые вычисляются с использованием трех опорных точек и количества строк и столбцов, информация которых храниться в атрибуте `COLROW` элемента `AREF` в файле `GDSII`.

$$\begin{pmatrix} \vec{x}'' \\ \vec{y}'' \end{pmatrix} = \begin{pmatrix} \vec{x}' \\ \vec{y}' \end{pmatrix} + \begin{pmatrix} \vec{t}_x \\ \vec{t}_y \end{pmatrix} \quad (7)$$

Благодаря (6) и (7) можно рассчитать новые координаты геометрических объектов, вызванных с помощью `SREF` и `AREF` соответственно.

Организация R^* -дерева с геометрическими элементами `GDSII`

R -дерево состоит из узлов, которые классифицируются как листовые и внутренние узлы. Конечные узлы соответствуют узлам нижнего уровня, которые содержат объекты пространственных данных. Эти объекты представляют собой геометрические объекты в многомерном пространстве, к которым будет применяться запрос. Каждый узел определен минимальным ограничивающим прямоугольником (`MBR`), который является минимальным охватывающим прямоугольником для пространственного объекта со сторонами, параллельными осям координат. `MBR` учитывается как в листовых узлах, так и во внутренних.

Минимальный ограничивающий прямоугольник в листовых узлах включает в себе геометрические объекты, а во внутренних узлах охватывает все `MBR` листов, на которые он ссылается. Структура дерева разработана с целью того, чтобы для пространственного поиска требовалось минимального количества посещений узлов.

В геометрических элементах `GDSII` необходимо помимо базовой информации, такой как номер слоя, координаты объекта, дополнительные атрибуты, хранить данные о вложенности объекта в иерархическую структуру в виде списка. Данная информация необходима для корректного определения данных в `GDSII` файле, так как в нем не используются уникальные идентификаторы элементов.

Если резюмировать все вышеописанные условия, то каждый листовой узел дерева будет состоять из элементов, имеющих вид (`ObjectGDSII`, `MBR`), где `ObjectGDSII` - это ссылка/указатель на геометрический элемент `GDSII`, который дополнительно хранит и данные о вложенности. Внутренние узлы дерева содержат элементы, имеющие похожую структуру: (`MBR`, `ChildPointer`), где `ChildPointer` ссылка/указатель на элемент последующего нижнего уровня. Минимальный ограничивающий прямоугольник внутреннего узла включает все `MBR` нижнего уровня, на который он указывает.

Блок-схема алгоритма построения R^* -дерева по данным `GDSII` файла представлена на рисунке 1.

Добавление данных осуществляется стандартным алгоритмом вставки в структуру R^* -дерева [12]. Поиск элемент в заданной площади находится с помощью стандартного алгоритма поиска по древовидной структуре [12]. Сложность

алгоритма поиска соответствует стандартной сложности поиска по R*-дереву и равна $O(\log n)$ в среднем и $O(n)$ худшем случае. На рисунке 3 проиллюстрирована базовая структура R*-дерева с включениями и перекрывающимися отношениями между MBR, построенная по топологии ИС из рисунка 2.

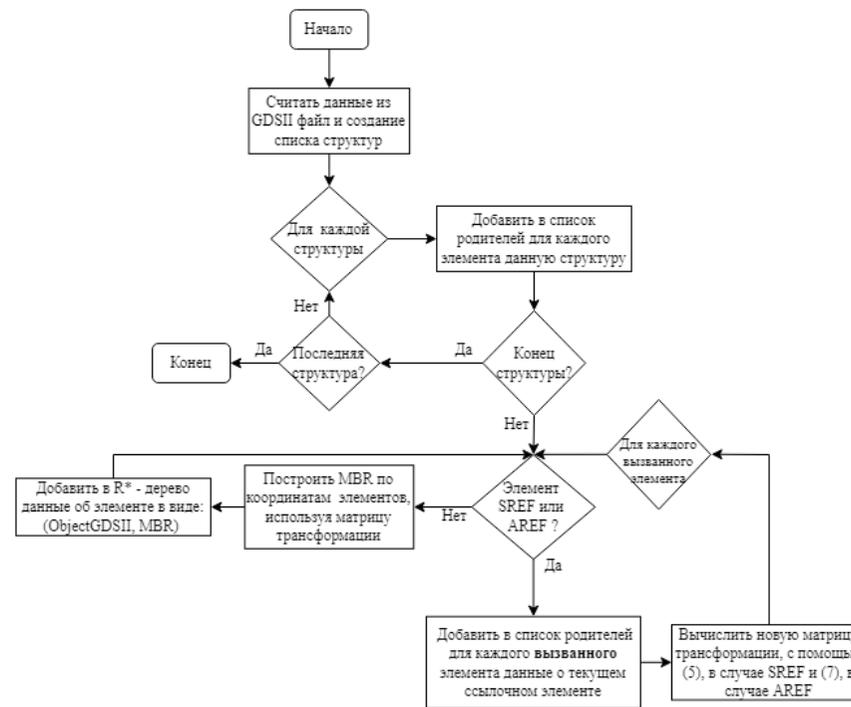


Рисунок 1 - Алгоритм построения R*-дерева по данным GDSII файла
 DOI: <https://doi.org/10.23670/IRJ.2024.139.26.1>

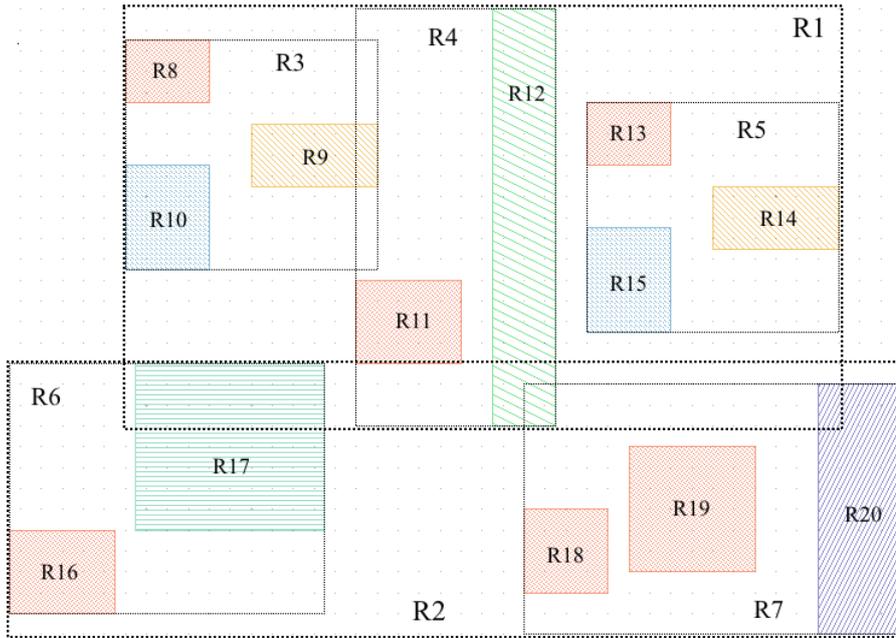


Рисунок 2 - Пример топологии ИС с МВР
DOI: <https://doi.org/10.23670/IRJ.2024.139.26.2>

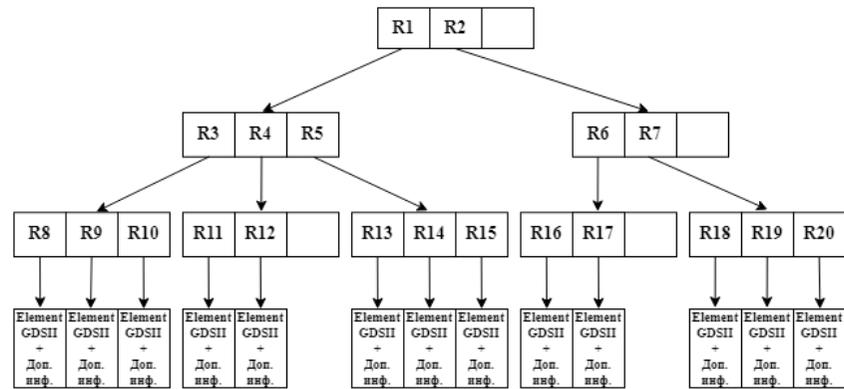


Рисунок 3 - Структура R*-дерева построенная по топологии ИС на рисунке 2
DOI: <https://doi.org/10.23670/IRJ.2024.139.26.3>

Заключение

В связи с усложнением топологий интегральных схем в данной статье был предложен способ преобразования информации из файлов формата GDSII в структуру R*-дерева. Данная древовидная структура является более усовершенствованной, чем стандартное R-дерево. R*-дерево вводит новые концепции, основанные на уменьшении площади, отступов и перекрытий минимальных ограничивающих прямоугольников. Сложность поиска элементов в заданной траектории в соответствии со структурой R*-дерева составляет $O(\log n)$ в среднем и $O(n)$ в худшем случае. Однако для практических целей производительность все еще является недостаточной, и здесь открыт простор для дальнейших исследований. Одним из возможных путей исследования является организация фильтра избыточных данных и извлечение только тех элементов, которые содержат ценную для данного запроса информацию.

Конфликт интересов

Не указан.

Рецензия

Гибадуллин Р.Ф., Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, Казань, Российская Федерация
DOI: <https://doi.org/10.23670/IRJ.2024.139.26.4>

Conflict of Interest

None declared.

Review

Gibadullin R.F., Kazan National Research Technical University named after A.N. Tupolev – KAI, Kazan, Russian Federation
DOI: <https://doi.org/10.23670/IRJ.2024.139.26.4>

Список литературы / References

1. Штанюк А.А. Проблема анализа топологии интегральных схем на основе GDSII-файлов / А.А. Штанюк, А.О. Семенов // Инженерные и информационные технологии, экономика и менеджмент в промышленности: сборник научных статей по итогам второй международной научной конференции. — Волгоград, 2020. — С. 334-336.
2. Штанюк А.А. Формат файлов OASIS для хранения топологии интегральных схем / А.А. Штанюк, А.О. Семенов // Приоритетные направления инновационной деятельности в промышленности: сборник научных статей по итогам двенадцатой международной научной конференции. — Казань, 2020. — С. 106-107.
3. GDS II Stream Format Manual (B97E060), Release 6.0 / Calma Corporation. — URL: [GDS_II_Stream_Format_Manual_6.0_Feb87.pdf](https://www.calma.com/Support/Manuals/GDSII/GDS_II_Stream_Format_Manual_6.0_Feb87.pdf) (accessed: 26.10.2020).
4. GDSII. — URL: <https://sites.google.com/site/esdcat/gdsii> (accessed: 13.11.2020)
5. Nievergelt J. Plane-sweep Algorithms for Intersecting Geometric Figures / J. Nievergelt, F.P. Preparata // Communications of the ACM. — 25. — 1982. — P. 739-747.
6. Guttman A. R-Trees a Dynamic Index Structure for Spatial Searching / A. Guttman // Proceedings ACM SIGMOD International Conference on Management of Data. — 1984. — P. 47-57.
7. Roussopoulos N. Direct Spatial Search on Pictorial Databases Using Packed R-trees / N. Roussopoulos, D. Leifker // Proceedings of the 1985 ACM SIGMOD International Conference on Management of Data. — 1985. — P. 17-31.
8. Kamel I. Hilbert R-Tree: an Improved R-Tree Using Fractals / I. Kamel, C. Faloutsos // Proc. of Int. Conference of Information and Knowledge Management CIKM. — 1993. — P. 490-499.
9. Beckmann N. The R*-tree: an Efficient and Robust Access Method for Points and Rectangles / N. Beckmann, H.-P. Kriegel, R. Schneider [et al.] // Proc. ACM SIGMOD. — 1990. — P. 322-331.
10. Greene D. An Implementation and Performance Analysis of Spatial Data Access Methods / D. Greene // Proc 5th Int. Conf on Data Engineering. — 1989. — P. 606-615.
11. Le C. Improved File Sizes and Cycle Times through Optimization of GDSII Stream / C. Le, D. Gariepy // SPIE Proceedings, International Society for Optics and Photonics. — 2005. — P. 351-358.
12. Гулаков В.К. Структуры и алгоритмы обработки многомерных данных: монография / В.К. Гулаков, А.О. Трубаков, Е.О. Трубаков. — Санкт-Петербург: Лань, 2021. — 356 с.

Список литературы на английском языке / References in English

1. Shtanjuk A.A. Problema analiza topologii integral'nyh shem na osnove GDSII-fajlov [The Problem of Topology Analysis of Integrated Circuits on the Basis of GDSII-Files] / A.A. Shtanjuk, A.O. Semenov // Inzhenernye i informacionnye tehnologii, jekonomika i menedzhment v promyshlennosti: sbornik nauchnyh statej po itogam vtoroj mezhdunarodnoj nauchnoj konferencii [Engineering and Information Technologies, Economics and Management in Industry: Collection of Scientific Articles on the Results of the Second International Scientific Conference]. — Volgograd, 2020. — P. 334-336. [in Russian]
2. Shtanjuk A.A. Format fajlov OASIS dlja hranenija topologii integral'nyh shem [OASIS File Format for Integrated Circuits Topology Storage] / A.A. Shtanjuk, A.O. Semenov // Prioritetnye napravlenija innovacionnoj dejatel'nosti v promyshlennosti: sbornik nauchnyh statej po itogam dvenadcatoj mezhdunarodnoj nauchnoj konferencii [Priority Directions of Innovation Activity in Industry: collection of scientific articles based on the results of the Twelfth International Scientific Conference]. — Kazan, 2020. — P. 106-107. [in Russian]
3. GDS II Stream Format Manual (B97E060), Release 6.0 / Calma Corporation. — URL: [GDS_II_Stream_Format_Manual_6.0_Feb87.pdf](https://www.calma.com/Support/Manuals/GDSII/GDS_II_Stream_Format_Manual_6.0_Feb87.pdf) (accessed: 26.10.2020).
4. GDSII. — URL: <https://sites.google.com/site/esdcat/gdsii> (accessed: 13.11.2020)

5. Nievergelt J. Plane-sweep Algorithms for Intersecting Geometric Figures / J. Nievergelt, F.P. Preparata // Communications of the ACM. — 25. — 1982. — P. 739-747.
6. Guttman A. R-Trees a Dynamic Index Structure for Spatial Searching / A. Guttman // Proceedings ACM SIGMOD International Conference on Management of Data. — 1984. — P. 47-57.
7. Roussopoulos N. Direct Spatial Search on Pictorial Databases Using Packed R-trees / N. Roussopoulos, D. Leifker // Proceedings of the 1985 ACM SIGMOD International Conference on Management of Data. — 1985. — P. 17-31.
8. Kamel I. Hilbert R-Tree: an Improved R-Tree Using Fractals / I. Kamel, C. Faloutsos // Proc. of Int. Conference of Information and Knowledge Management CIKM. — 1993. — P. 490-499.
9. Beckmann N. The R*-tree: an Efficient and Robust Access Method for Points and Rectangles / N. Beckmann, H.-P. Kriegel, R. Schneider [et al.] // Proc. ACM SIGMOD. — 1990. — P. 322-331.
10. Greene D. An Implementation and Performance Analysis of Spatial Data Access Methods / D. Greene // Proc 5th Int. Conf on Data Engineering. — 1989. — P. 606-615.
11. Le C. Improved File Sizes and Cycle Times through Optimization of GDSII Stream / C. Le, D. Gariepy // SPIE Proceedings, International Society for Optics and Photonics. — 2005. — P. 351-358.
12. Gulakov V.K. Структуры и алгоритмы обработки многомерных данных: монография [Structures and Algorithms of Multidimensional Data Processing: monograph] / V.K. Gulakov, A.O. Trubakov, E.O. Trubakov. — Saint Petersburg: Lan, 2021. — 356 p. [in Russian]