

# Разработка алгоритмов визуализации цифровых интегральных схем на вентиляльном уровне

Г.И. Гоглев, И.В. Тиунов

Институт проблем проектирования в микроэлектронике РАН, г. Москва,

goglev\_g@ippm.ru, tiunov\_i@ippm.ru

**Аннотация** - Современные системы автоматизации проектирования разрабатываются с использованием графических интерфейсов пользователя. Применение таких интерфейсов позволяет упростить и ускорить разработку устройств. В данной работе рассматривается проблема разработки алгоритмов визуализации, необходимых для создания инструмента графического проектирования цифровых интегральных схем на вентиляльном уровне. Предложена графовая модель представления схемы как основа для разработки алгоритмов. Рассмотрены алгоритмы размещения и трассировки, необходимые для построения графического представления схемы. Упомянутые алгоритмы реализованы в виде программного модуля на основе разработанной графовой модели. Представлены результаты работы программного модуля для некоторых схем из набора ISCAS'89.

**Ключевые слова** — автоматизация проектирования, графическое проектирование, алгоритмы размещения, алгоритмы трассировки, визуализация, САПР.

## I. ВВЕДЕНИЕ

В области микроэлектроники конкуренция существует не только на рынках интегральных схем и компонентов, но и на рынке систем автоматизации проектирования (САПР) и, в частности, инструментов разработки, решающих задачи на различных этапах и входящих в упомянутые системы. Чем эффективнее инструменты разработки и, соответственно, системы автоматизации, тем более конкурентоспособен разработчик и разрабатываемое им устройство.

Графические интерфейсы являются неотъемлемой частью современных САПР. Их применение упрощает и ускоряет разработку, а инструменты некоторых этапов проектирования цифровых схем сложно представить без применения графики. В частности, существуют инструменты графического проектирования на различных этапах: логическом, схемотехническом и топологическом.

Одна из специализаций Института проблем проектирования в микроэлектронике РАН (ИППМ РАН) — разработка САПР цифровых схем для отечественных ПЛИС. На данный момент разработанный институтом САПР X-CAD обеспечивает проектирование по всему циклу: от создания логического описания на одном из языков проектирования до получения прошивки ПЛИС.

Система использует в своем составе как свободно распространяемые инструменты, так и собственные инструменты и их алгоритмы, как для разработки схем [1][2][3], так и для их оптимизации [4] и верификации [5]. X-CAD реализует полный цикл проектирования схем, но на данный момент в ней отсутствует возможность редактирования схем в графическом формате поэтому в случае необходимости редактирования в таком формате приходится использовать сторонние инструменты. Такие инструменты зачастую являются либо коммерческими, либо свободно распространяемыми. Но помимо проблемы лицензирования инструмент может также не содержать требуемого функционала или быть недостаточно гибким для внедрения в САПР.

Исходя из вышесказанного, а также беря во внимание текущую политику импортозамещения, актуальной задачей является разработка собственного инструмента графического проектирования, который будет удовлетворять всем требованиям.

Далее будет рассмотрена проблема разработки графовой модели представления данных и алгоритмов размещения и трассировки, необходимых для визуализации цифровых схем на вентиляльном уровне.

## II. ГРАФОВАЯ МОДЕЛЬ ДЛЯ ПРЕДСТАВЛЕНИЯ ЦИФРОВЫХ СХЕМ

Первый шаг на пути к созданию инструмента — разработка модели представления данных, на основе которой будут разрабатываться необходимые алгоритмы. Наиболее подходящий способ представления схем — ориентированные гиперграфы (оргиперграфы) [6]. Такие графы обладают всеми необходимыми параметрами для представления цифровой схемы: логические элементы можно представить в виде вершин графа, а межсоединения схемы, цепи — гипердугами графа, поскольку гипердуги, как и цепи, обладают направлением и могут быть инцидентны более чем двум вершинам.

На рис. 1 показана модель для представления схем в виде иерархического оргиперграфа.

Каждую схему можно представить как головной элемент (окружность с зеленым пунктирным контуром), который имеет совокупность элементов (окружности с зеленым сплошным контуром), их

портов, (красные контактные точки), и цепей (синие стрелки, выходящие из портов источника сигнала и приходящие в порты приемников сигнала). Иерархичность модели заключается в том, что каждый внутренний элемент может представлять из себя такой же головной элемент со своей схемой, содержащей элементы ещё более «низкого порядка».

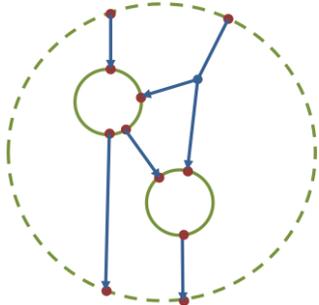


Рис. 1. Графовая модель схемы

Формально модель можно описать как граф  $G(E, P, W)$ , где  $E = (e_0, e_1 \dots e_n)$  – множество логических элементов схемы,  $P = (p_1, p_2 \dots p_k)$  – множество портов элементов,  $W = (w_1, w_2 \dots w_m)$  – множество межсоединений. При этом определены отношения инцидентности  $H(e_i)$  – список портов, относящихся к элементу  $e_i \in R(p_l, w_k)$  – отношение инцидентности, которое определяет наличие связи между портом  $p_l$  и цепью  $w_k$ : 1, если цепь выходит из порта, 0, если они не связаны, или -1 если цепь входит в порт.

На рис. 2 представлена в виде описанного выше оргиперграфа схема c17 из набора тестовых схем ISCAS'89 и входящая в состав сборника IWLS 2005 [7].

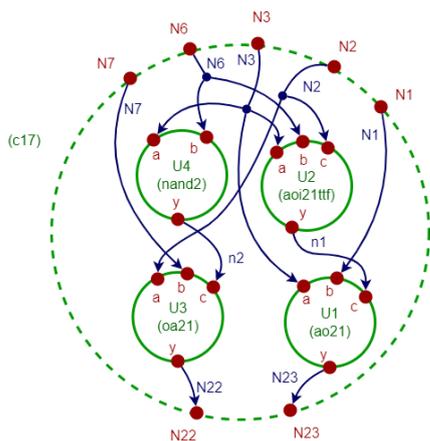


Рис. 2. Схема c17 в виде оргиперграфа

На рис. 3 показана иерархическая структура этой схемы в виде блок-схемы.

После разработки модели данных можно приступать к разработке алгоритмов визуализации схем. Первая задача, которую требуется решить – задача размещения элементов на дискретном рабочем поле (ДРП).

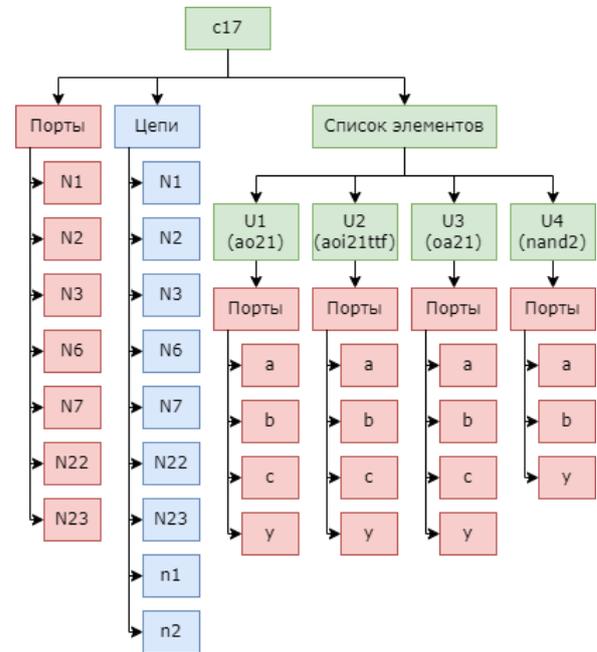


Рис. 3. Иерархическая структура схемы c17

### III. РАЗРАБОТКА АЛГОРИТМА РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ НА ДРП

Существует множество классов алгоритмов размещения на ДРП. Одними из таких классов являются силовые алгоритмы и ранжированные алгоритмы размещения. Каждый из этих классов обладает как своими преимуществами, так и недостатками [8]. Рассмотрим основные из них.

Силовые алгоритмы размещают элементы друг от друга на расстоянии, зависящем от количества связей между ними. Таким образом, после применения алгоритма наиболее связанные элементы схемы собираются в группы. Такие алгоритмы довольно просты в реализации и хорошо подходят для анализа связности элементов схемы и нахождения наиболее нагруженных ее участков. Однако они хороши лишь для малых графов с количеством элементов менее нескольких сотен [9].

Ранжированный алгоритм размещает элементы основываясь на их ранге, который определяется по каким-либо правилам – например, по удалённости элемента от входного порта схемы. Такое размещение позволяет проследить пути распространения сигнала и оценить критический путь, поскольку фронт распространения будет направлен от входных портов к выходным, исключая цепи обратной связи. Кроме того, размещенная таким образом схема выстраивается по столбцам, соответствующим рангам элементов, что упрощает ее восприятие, в отличие от силовых алгоритмов. При этом результат работы такого алгоритма зачастую занимает больше пространства. Однако, учитывая специфику задачи, площадь, занимаемая схемой на рабочем поле, не является критическим фактором.

Исходя из вышесказанного, для решения задачи визуализации лучше подходят ранжированные алгоритмы размещения.

Для того, чтобы рассчитать ранг логического элемента необходимо рекурсивно обойти граф от каждого входа к выходам. Ранг элемента устанавливается равным количеству элементов, предостоящих текущему.

Алгоритм расчета ранга элемента можно описать следующим псевдокодом:

Для каждого входного порта  $p_i$  головного элемента (структуру которого нужно отобразить):

Для каждого порта  $p_j$ , в который входит провод из порта  $p_i$ :

Выполнить *setRank* для элемента порта  $p_j$ , считая ранг прошлого элемента равным 0

КОНЕЦ.

Функция “*setRank*” алгоритма расчета ранга элемента описывается следующим псевдокодом:

Если ранг данного элемента  $r$  не больше ранга прошлого элемента  $_r$ :

$r = _r + 1$ ;

Для каждого выходного порта  $p_i$  данного элемента:

Для каждого порта  $p_j$ , в который входит провод из порта  $p_i$ :

Выполнить *setRank* для элемента порта  $p_j$ , считая ранг прошлого элемента равным  $r$

В результате работы алгоритма каждому логическому элементу схемы ставится в соответствие ранг, который определяет его положение в пространстве относительно других элементов. Далее эту информацию необходимо использовать, чтобы рассчитать абсолютные координаты элемента на рабочем поле. Установив отступ между элементами, можно высчитать положение элементов на рабочем поле. Алгоритм расчета координат положения элементов можно описать псевдокодом, представленным ниже ( $X$  - горизонтальная координата для текущего ранга,  $Y$  - вертикальная координата,  $X'$  - горизонтальная координата для следующего ранга):

Задать  $X=10, Y=10, X'=X+3$ .

Для каждого входного порта головного элемента:

Разместить порт по координатам  $X$  и  $Y$ .

Увеличить  $X'$  на 1,  $Y$  на 3.

Задать  $X=X', Y=10, X'=X+4$ .

Для каждого ранга  $N$  от 1 до  $\max N-1$ :

Посчитать количество входных портов всех элементов с этим рангом и прибавить к  $X$ .

Для каждого элемента с этим рангом:

Разместить элемент по координатам  $X$  и  $Y$ .

Увеличить  $Y$  на (высота элемента+1),  $X'$  на количество выходных портов.

Задать  $X=X', Y=10, X'=X+4$ .

Посчитать количество выходных портов головного элемента и прибавить к  $X$ .

Для каждого выходного порта:

Разместить порт по координатам  $X$  и  $Y$ .

Увеличить  $Y$  на 3.

КОНЕЦ.

#### IV. РАЗРАБОТКА АЛГОРИТМА ТРАССИРОВКИ ЦЕПЕЙ НА ДРП

После размещения элементов необходимо провести между ними межсоединения, то есть решить задачу трассировки цепей схемы. Размещение и трассировка являются одними из наиболее сложных задач при проектировании схем [10]. Алгоритмы трассировки можно разделить на два семейства – лабиринтную и лучевую.

Одним из основных алгоритмов лабиринтной трассировки является алгоритм Ли [11] (он же «волновой алгоритм»). Суть алгоритма Ли в нумерации ячеек дискретного рабочего поля – на первой итерации пустым ячейкам вокруг начальной ячейки присваивается первый номер, на второй итерации пустым ячейкам вокруг ячеек с первым номером присваивается второй номер и т.д., пока на очередной итерации рядом с одной из ячеек не окажется конечная ячейка. Как только алгоритм достиг конечной ячейки, необходимо сформировать итоговый путь. Для этого достаточно пройти от конечной ячейки по нумерованным ячейкам в сторону уменьшения нумерации до начальной.

Лучевые алгоритмы отличаются от лабиринтных тем, что трассировка проводится линиями, а не по ячейкам – путь, построенный таким образом, имеет наименьшее количество углов.

Основным алгоритмом лучевой трассировки является алгоритм Миками-Табучи [11]. Алгоритм начинает свою работу с проведения перпендикуляров первой итерации, т.е. через конечную и начальную точки. Далее к линиям первой итерации проводятся перпендикуляры – линии второй итерации, и т.д. до тех пор, пока не найдётся пересечение линий от конца и от начала. Поиск пути проходит от точки пересечения по линиям в сторону уменьшения их номера итерации.

На рис. 4 представлен пример работы алгоритмов Ли и Миками-Табучи, что наглядно демонстрирует разницу в подходах и результатах их работы.

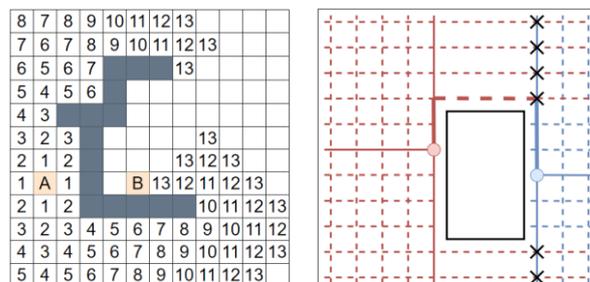


Рис. 4. Результат работы алгоритма Ли (слева) и алгоритма Миками-Табучи (справа)

Основное преимущество лабиринтных алгоритмов, по сравнению с лучевыми – простота их реализации.

Но в отношении задачи визуализации лучевые алгоритмы обладают рядом преимуществ перед волновыми. Во-первых, при хранении данных о цепях большой схемы выигрыш по памяти даёт трассировка, проведённая лучевым алгоритмом, так как её цепи представлены в виде линий (4 значения на линию – координаты её концов), а не трасс из ячеек, как в случае с лабиринтным алгоритмом (1 значение на ячейку – её маркировка). Во-вторых, цепи, построенные алгоритмом лучевой трассировки, имеют наименьшее количество углов, что значительно упрощает анализ схемы.

С учетом вышесказанного для трассировки цепей при решении задачи визуализации схем лучевая трассировка подходит лучше, чем лабиринтная. Поэтому для реализации был выбран именно этот алгоритм.

Реализация алгоритма лучевой трассировки на основе разработанной модели представлена следующим псевдокодом:

*Для каждой пары портов, которые нужно соединить:*

*Создать горизонтальные линии от начала (bl) и от конца (el).*

*Если созданные линии соединяют порты, то путь найден, КОНЕЦ.*

*Иначе*

*Выполнить trace(bl,el).*

*Если после выполнения получена пустая пара линий, то путь найти невозможно, КОНЕЦ.*

*Воссоздать по линиям из pathList путь, КОНЕЦ.*

Функция “trace” алгоритма трассировки описывается следующим псевдокодом:

*trace (список линий от начала bll1, список линий от конца ell1):*

*Построить перпендикулярные линии bll2 к линиям bll1.*

*Построить перпендикулярные линии ell2 к линиям ell1.*

*Если линии из начала или из конца не удалось создать, вернуть пустую пару линий, КОНЕЦ.*

*Если bll2 имеют пересечение с ell1, то:*

*найти такую линию midLine и записать в список линий пути pathList.*

*найти по 1 линии из bll1 и ell1, пересекающих midLine, записать в pathList и вернуть в качестве результата, КОНЕЦ.*

*Для bl2 и el2 выполнить trace(bll2, ell2).*

*Если линии не получены, вернуть пустую пару линий, КОНЕЦ.*

*Найти линию bl1 из bll1, пересекающую bl2, записать в pathList.*

*Найти линию e11 из ell1, пересекающую el2, записать в pathList.*

*Вернуть bl1 и e11 в качестве результата, КОНЕЦ.*

## V. РЕЗУЛЬТАТЫ РАБОТЫ АЛГОРИТМОВ

Разработанные алгоритмы были реализованы в программном модуле визуализации цифровых схем собственной разработки. Тестирование проводилось с использованием набора ISCAS’89. Ниже представлена работа алгоритмов для последовательностной схемы s27 с 8 элементами (рис. 5) и комбинационной схемы c418 с 93 элементами (рис. 6). Алгоритмы, также, успешно прошли тесты на схемах c17 и s344.

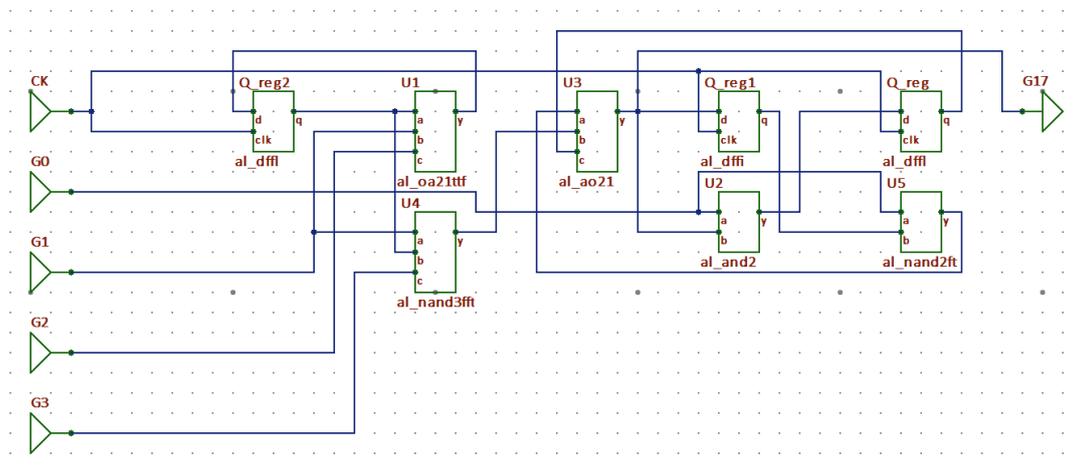


Рис. 5. Результат работы алгоритмов трассировки и размещения для схемы s27

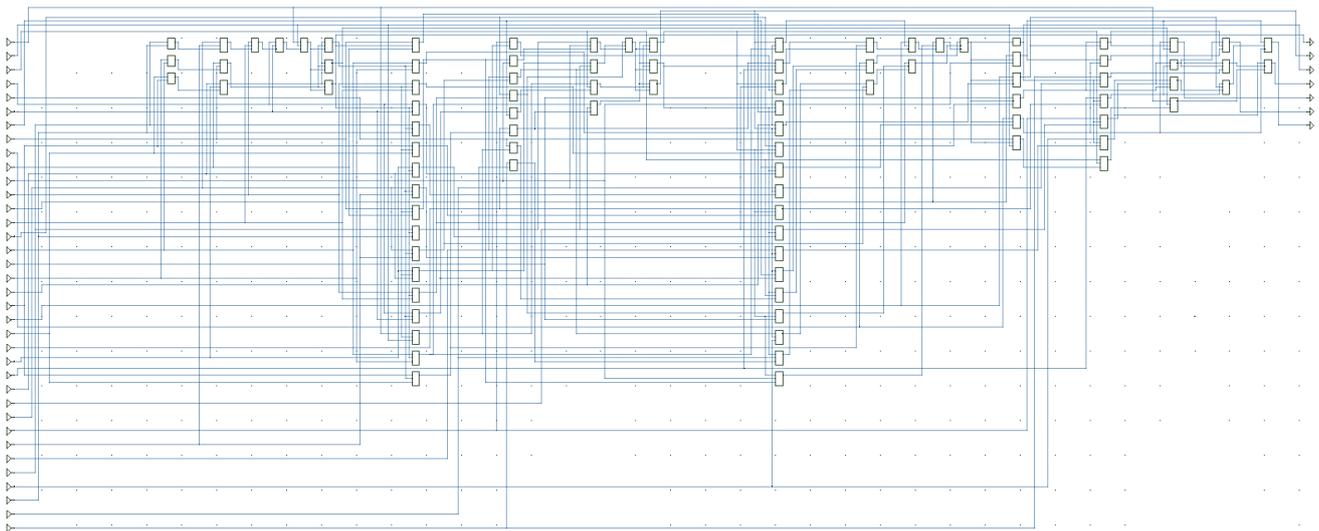


Рис. 6. Результат работы алгоритмов трассировки и размещения для схемы с418

## VI. ЗАКЛЮЧЕНИЕ

Как было сказано в начале работы, разработка инструментов и, в частности, выбор алгоритмов для решения определенных задач проектирования являются актуальными проблемами.

В рамках работы был проведен сравнительный анализ различных типов алгоритмов размещения и трассировки, необходимых для решения задачи визуализации цифровых схем на вентиляльном уровне. Оценив преимущества и недостатки разных семейств алгоритмов, были выбраны ранжированный алгоритм размещения и лучевой алгоритм трассировки Миками-Табучи.

Также в результате работы был разработан программный модуль, в котором были реализованы выбранные алгоритмы, основанные на предложенной графовой модели. Несмотря на то, что алгоритмы хорошо подходят для решения упомянутых задач, им требуются доработки, что будет являться темой будущих работ.

## ЛИТЕРАТУРА

- [1] Гаврилов С.В., Железников Д.А., Липатов И.А., Тиунов И.В. Маршрут проектирования для отечественных программируемых интегральных схем специального назначения: интеграция с существующими промышленными средствами автоматизированного проектирования и решение проблем импортозамещения // Электронная техника. Серия 3. Микроэлектроника. Москва. 2017. С. 5-11.
- [2] С. В. Гаврилов, Д. А. Железников, М. А. Заплетина, В. М. Хватов, Р. Ж. Чочаев, В. И. Эннс. Маршрут топологического синтеза для реконфигурируемых систем на кристалле специального назначения // Микроэлектроника. Москва. 2019. Т. 48. № 3. С. 211-223.
- [3] Д.Г. Савельев, С.В. Гаврилов, И.В. Тиунов. Разработка графического программного средства для проектирования цифровых схем на логическом уровне // Материалы V Международной научно-практической конференции «Научные исследования в современном мире: опыт, проблемы и перспективы развития». Ч. 2. Уфа, 2021. С. 56-65.
- [4] И.В. Тиунов. Методы ресинтеза схем для ПЛИС на основе ячеек с разделенными выходами и обратной связью // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2020. №2. С. 50-56.
- [5] А.Д. Иванников, А.Л. Стемповский. Математическая модель цифровых блоков для системы совместного моделирования технических средств и программно-микропрограммного обеспечения // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2021. №1. С. 2-7.
- [6] В.А. Овчинников. Операции над ультра- и гиперграфами для реализации процедур анализа и синтеза структур сложных систем. // Наука и образование: Эл. науч. Издание. 2009. №10.
- [7] URL: <http://iwls.org/iwls2005/benchmarks.html> (дата обращения: 26.08.2022)
- [8] П.И. Фролова, Р.Ж. Чочаев. Разработка и сравнительный анализ методов начального размещения на ПЛИС // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2021. С. 57-64.
- [9] Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003. – 1104 с.
- [10] М.А. Заплетина. Решение задачи трассировки межсоединений для реконфигурируемых систем на кристалле с различными типами коммутационных элементов // Электронная техника. Серия 3. Микроэлектроника. Москва. 2018. С. 31-36
- [11] Зыков А.Г., Поляков В.И. Алгоритмы конструкторского проектирования ЭВМ. СПб: Университет ИТМО, 2014. 136 с.

# Development of Digital Integrated Circuits Visualization Algorithms at Gate Level

G.I. Goglev, I.V. Tiunov

Institute for Design Problems in Microelectronics of Russian Academy of Sciences (IPPM RAS),  
Moscow, goglev\_g@ippm.ru, tiunov\_i@ippm.ru

**Abstract** — Modern computer aided design systems are developed using graphical user interfaces. The use of such interfaces makes it possible to simplify and speed up the development of devices. In this paper, the problem of developing a tool for graphic design of digital integrated circuits at the logical level is considered.

The introduction explains the relevance of the task of developing a graphic design tool based on economic considerations and domestic policy.

The second chapter deals with the problem of developing a hierarchical graph model for the representation of digital circuits at the logical level. A graph model is proposed and its application for c17 circuit from ISCAS'89 benchmark is demonstrated.

The third chapter is devoted to the development of an algorithm for placing elements on a discrete area. The chapter briefly describes the features of power algorithms and ranking algorithms. It is concluded that ranking algorithms are better suited for visualizing circuits at the logical level, since they simplify the analysis of signal propagation. A block diagram of the ranking algorithm based on the proposed graph model is presented.

The fourth chapter is devoted to the development of an algorithm for tracing interconnections. The features of ray tracing algorithms and wave tracing algorithms are considered. It is concluded that ray tracing is better suited for solving the problem of circuit visualization, based on the features of the software implementation and the visual simplicity of the resulting circuit. Also, a block diagram of the ranking algorithm based on the proposed graph model is presented.

The fifth chapter presents the results of the work of the developed algorithms, which were implemented in a software module. Circuits from the ISCAS'89 test suite were taken for testing.

In conclusion, the results of the work are summarized.

**Keywords** — design automation, graphical design, placement algorithms, routing algorithms, visualization, CAD.

## REFERENCES

- [1] Gavrilov S.V., Zheleznikov D.A., Lipatov I.A., Tiunov I.V. *Marshrut proyektirovaniya dlya otechestvennykh programmiruyemykh integral'nykh skhem spetsial'nogo naznacheniya: integratsiya s sushchestvuyushchimi promyshlennymi sredstvami avtomatizirovannogo proyektirovaniya i resheniye problem importozameshcheniya (Design flow for domestic programmable integrated circuits for special purpose: integration with existing computer-aided design systems and solution for problems of import substitution)* // Elektronnyaya tekhnika. Seriya 3. Mikroelektronika. 2017. S. 5-11.
- [2] Gavrilov S.V., Zheleznikov D.A., Zapletina M.A., Khvatov V.M., Chochaev R.Z., Enns V.I. *Layout synthesis design flow for special-purpose reconfigurable systems-on-a-chip* // Russian Microelectronics. 2019. V. 48. № 3. P. 176-186.
- [3] Savelyev D.G., Tiunov I.V. *Razrabotka graficheskogo programmnoy sredstva dlya proyektirovaniya cifrovyykh skhem na logicheskom urovne* // Materialy V Mezhdunarodnoy nauchno-prakticheskoy konferentsii «Nauchnye issledovaniya v sovremennom mire: opyt, problemy i perspektivy razvitiya». CH. 2. Ufa, 2021. S. 56-65.
- [4] Tiunov I.V. *Resynthesis methods for FPGAs based on cells with separated outputs and built-in feedback* // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2020. Issue 2. P. 50-56. doi:10.31114/2078-7707-2020-2-50-56.
- [5] Ivannikov A.D., Stempkovsky A.L. *Digital Block Mathematical Model for the Joint Hardware and Software/firmware Simulation System* // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2021. Issue 1. P. 2-8. doi:10.31114/2078-7707-2021-1-2-8
- [6] Ovchinnikov V.A. *Operacii nad ul'tra- i gipergrafami dlya realizacii procedur analiza i sinteza struktur slozhnykh sistem.* // Nauka i obrazovanie: El. nauch. Izdanie. 2009. №10.
- [7] URL: <http://iwls.org/iwls2005/benchmarks.html> (access date: 26.08.2022)
- [8] Frolova P.I., Chochaev R. *Development and Comparative Analysis of Initial Placement Methods for FPGA* // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2021. Issue 3. P. 57-64. doi:10.31114/2078-7707-2021-3-57-64
- [9] Kas'yanov V.N., Evstigneev V.A. *Grafy v programmirovanii: obrabotka, vizualizatsiya i primenenie.* – SPb.: BHV-Peterburg, 2003. – 1104 S.
- [10] Zheleznikov D.A., Zapletina M.A., Khvatov V.M. *Solution of interconnect routing problem for reconfigurable systems-on-chip with different types of switching elements* // Elektronnyaya tekhnika. Seriya 3. Mikroelektronika. 2018. S. 31-36.
- [11] Zykov A.G., Polyakov V.I. *Algoritmy konstruktorskogo proyektirovaniya EVM.* SPb: Universitet ITMO, 2014. 136 s.