

Также стоит обратить внимание на то, каким образом осуществляется селекция решений в новую популяцию. Здесь применяется, так называемый, *вариативный метод*, который позволяет сохранять достаточное генетическое разнообразие в популяции, т.е., в первую очередь, в новую популяцию попадают наиболее приспособленные особи, а также особи с отличными от оптимальных значениями целевых функций. Затем осуществляется уменьшение параметра дальнейшей работы эволюционного алгоритма K2 на единицу и проверка критерия останова. После чего проверяются значения динамических параметров K1 и K2. В случае обнуления обоих этих динамических параметров, происходит их повторная инициализация, и вся процедура повторяется, пока не пройдет заданное количество итераций или не будет найдено «оптимальное» решение.

Заключение. Предложена структурная схема интегрированного алгоритма поиска оптимальных решений и на её основе разработана программа в среде программирования Builder C++ 6.0.

Полученные в работе результаты экспериментальных исследований подтвердили эффективность предложенного метода. Временная сложность нового интегрированного алгоритма, в основном, совпадает со сложностью быстрых итерационных алгоритмов, т.е. интегрированные алгоритмы являются перспективным направлением для исследований. При этом, помимо определения локальных оптимумов, эволюционно-генетические алгоритмы дают возможность выделить и те решения, значения которых близки к глобальным.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Курейчик, В.В., Полупанов А.А. Эволюционные методы разбиения схем на основе адаптивных генетических процедур: монография. – Таганрог: Изд-во ТТИ ЮФУ, 2007. – 160 с.
2. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / Под ред. В.М. Курейчика. – М.: Физматлит, 2006. – 320 с.
3. Курейчик В.В., Мищенко М.Н. Бионический метод определения путей оптимальной длины в графовых моделях // III-й Международный научно-практический семинар «Интегрированные модели и мягкие вычисления в искусственном интеллекте». – М: Изд-во Физматлит, 2005. – С. 261-266.
4. Норенков И. П. Основы автоматизированного проектирования. – М.: Изд-во МГТУ имени Н.Э. Баумана, 2006. – 360 с.

УДК 658.512.2.011.5

Д.С. Кныш

АЛГОРИТМ ТРАССИРОВКИ В КОММУТАЦИОННОМ БЛОКЕ*

Постановка задачи трассировки цепей коммутационного блока в СБИС. В классической постановке коммутационный блок представляет собой область, ограниченную с четырех сторон линейками контактов (терминалов) различных цепей. Задача трассировки в блоке состоит в проведении участков трасс одноименных цепей при заданных конструкторских и технологических ограничениях. Основными ограничениями являются ширина проводников, минимально допустимый интервал между цепями и фиксированный размер коммутационного блока. Терми-

* Работа выполнена при поддержке РФФИ (грант № 07-01-00174).

налы в коммутационном блоке при двухслойной трассировки могут располагаться следующим образом [3]:

- ◆ терминалы доступны в обоих слоях;
- ◆ терминалы верхней и нижней линеек контактов доступны на одном слое, а терминалы правой и левой линеек контактов доступны на втором слое.

Разработанный алгоритм применяется для решения каналов СБИС со следующими критериями: количество выводов для одной цепи – от 2 до n (где n – количество терминалов в блоке), ширина проводников цепи различная и задается пользователем для каждой цепи отдельно, выводы цепей расположены на границах блока, сквозные переходные отверстия, двухслойная модель трассировки.

Модель трассировки может быть основана на назначении вертикальных и горизонтальных фрагментов в различные слои. Такие модели применимы только для многослойной трассировки. Если любой участок цепи может располагаться в любом слое, то такую модель называют *незарезервированной моделью*. Когда каждый тип фрагмента предназначен для определенного слоя, такая модель называется *зарезервированной моделью*. Большинство алгоритмов используют вторую модель. В двухслойной трассировке, если в первом слое располагаются вертикальные фрагменты, а во втором горизонтальные, то такая модель называется *VH моделью* (Vertical-Horizontal model).

Задача трассировки цепей коммутационного блока состоит в проведении электрических соединений между заданными терминалами сети внутри области трассировки, при соответствующих правилах проектирования, минимизируя длину соединений и интервал между ними.

Для оценки качества результатов трассировки используются три критерия оптимизации:

- ◆ *Межслойная емкость*. Между проводниками печатной платы, находящимися на разных слоях, возникает емкостная связь, когда они пересекаются. Проводники, находящиеся друг над другом на смежных слоях, создают длинный пленочный конденсатор. Поэтому важно уменьшить паразитную емкость:

$$F1 = \sum_{i=1}^{n_c} C_i \cdot \quad (1)$$

В формуле (1) $F1$ определяет суммарную паразитную емкость проводников расположенных на разных слоях. Где C_i – емкость, возникающая между парой пересекающихся проводников, расположенных на противоположных слоях и рассчитывается по формуле (2), а n – количество таких пар:

$$C = 0,0085 * E_R * \frac{A}{d}, \quad (2)$$

где E_R – диэлектрическая постоянная, A – площадь перекрытия, d – расстояние между слоями.

- ◆ *Число межслойных переходов*. Введение межслойного перехода между двумя слоями приводит к увеличению времени процесса и ухудшению качества трассировки. Следовательно, нужно добиться использования как можно меньшего количества переходов из слоя в слой:

$$F2 = \sum_{i=1}^{NumVia} Via_i \cdot \quad (3)$$

В формуле (3) значение функции $F2$ определяет количество межслойных переходов. Via_i – обозначает i -межслойный переход, $i=[1, NumVia]$.

- ◆ *Суммарная длина межсоединений.* Наименьшая длина цепи обеспечивает наименьшую задержку распространения сигнала. При этом необходимо учитывать ширину цепей, так как прохождение одной и той же цепи в различном месте может иметь различную суммарную длину:

$$F3 = \sum_{i=1}^{NumNet} LenghtNet_i, \quad (4)$$

где F3 равен сумме длин проводников всех цепей в канале (4); LenghtNet_i – определяет длину проводников цепи i, где i меняется в периоде от 1 до количества цепей NumNet.

Описание алгоритма трассировки. Для создания начальной популяции используется разработанный алгоритм, в основе которого лежит принцип поэтапной модификации и размещения цепей [2, 4]. Первоначально для каждой цепи формируется начальная трассировочная структура на основе следующего алгоритма:

1. Для каждой цепи формируется список занимаемых ею терминалов $T_i = \{t_1, t_2, \dots, t_j\}$.
2. Для каждой цепи формируется список промежуточных дополнительных узлов $Y_i = \{y_1, y_2, \dots, y_j\}$, которые являются точками пересечения продолжений координат x и y терминалов.
3. Выбирается случайным образом цепь.
4. Выбирается из множества дополнительных узлов Y_i наименее удаленный от всех терминалов узел y_j , заносится этот узел в список основных узлов $Y_{осн}$.
5. Случайным образом выбирается терминал t_j из множества T_i , и соединяется с узлом y_j прямым или L-образным участком цепи.
6. Исключается терминал t_j из списка не рассмотренных $T_i = T_i \setminus t_j$.
7. Случайным образом выбирается терминал t' , формируем временный список опорных узлов Y_{op} (точки пересечения продолжений координат x, y терминала с уже существующими трассами) и добавляем в него узлы из $Y_{осн}$.
8. Выбирается из Y_{op} ближайший узел y_k и соединяется с терминалом прямым или L-образным участком цепи, добавляются новые опорные узлы в $Y_{осн}$.
9. Если рассмотрены все терминалы, то переход к п. 10, в противном случае – к п. 7.
10. Если рассмотрены все цепи то переход к п. 11, в противном случае – к п. 3.
11. Конец работы алгоритма.

Для цепи №1 результат работы алгоритма показан на рис. 1.

Если цепь содержит терминалы расположенные только на противоположных сторонах, то такие терминалы соединяются методом вертикального/горизонтального столба Штейнера [5].

После создания трассировочной структуры каждой цепи приступаем к этапу ее размещения в коммутационном блоке с соблюдением ограничений. Одним из ограничений является необходимость выдержать заданный интервал между цепя-

ми. В алгоритме используется зарезервированная модель (Vertical-Horizontal модель). Структурная схема алгоритма размещения представлена на рис. 2.

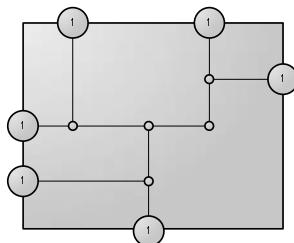


Рис. 1. Окончательно решение для цепи №1

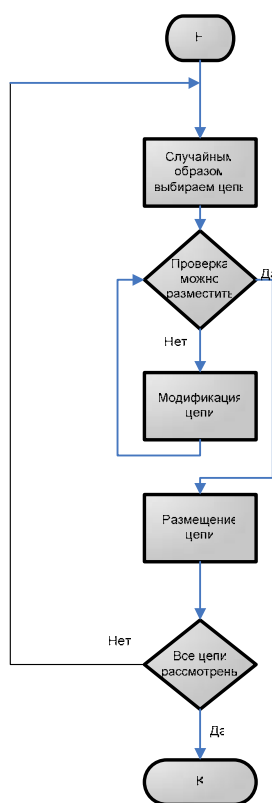


Рис. 2. Структурная схема алгоритма размещения цепей

Модификация цепи продолжается до тех пор, пока цепь нельзя будет разместить на коммутационном блоке. Модификация осуществляется следующим образом:

1. Определяется участок трассы, который не может быть размещен.
2. Определяется новая позиция для этого участка трассы.
3. Участок трассы перемещается на новую позицию.

После перемещения участка трассы могут возникать различные ситуации (циклы, накладывание цепей, появление «потерянных» участков) [2], которые надо учитывать. Преимущество этого метода в том, что перемещается не весь участок трассы, а только его часть. Алгоритм предусматривает *увеличение* размеров ком-

мутационного блока, если для перемещаемого участка трассы невозможно найти новой позиции.

Результаты исследования работы алгоритма. Для анализа качества решений, создаваемых данным алгоритмом они сравнивались с результатами работы генетического алгоритма на 30 итерации. Как видно из табл. 1 решения полученные алгоритмом довольно разнообразны. Но качество решений на 18-25% хуже, чем у генетического алгоритма (табл. 2). Целевая функция (ЦФ) рассчитывается как аддитивная свертка трех критериев, и измеряется в условных единицах. График зависимости средней ЦФ от числа выводов в коммутационном блоке для работы алгоритма представлен на рис. 3. Для сравнения на нем отображена степенная функция и средняя ЦФ для решений созданных генетическим алгоритмом. Анализируя затраченное время на генерацию решений (рис. 4), получим, что временная сложность алгоритма пропорциональна $O(N^2)$.

Таблица 1

Зависимость ЦФ решений, полученных разработанным алгоритмом, от числа выводов в коммутационном блоке

Серия опы- тов	Число выводов в коммутационном блоке									
	16		20		24		28		32	
	ЦФ	Время, с	ЦФ	Время, с	ЦФ	Время, с	ЦФ	Время, с	ЦФ	Время, с
1	18,76	0,19	30,92	0,63	29,45	0,56	43,9	1,12	53,55	0,39
2	19,09	0,27	24,9	0,36	30,23	0,55	71,45	3,01	56,34	0,64
3	15,13	0,16	36,32	0,79	30,45	0,50	38,13	1,31	49,37	0,50
4	16,08	0,24	40,95	0,95	31,28	0,63	83,14	3,90	72,19	1,09
5	17,18	0,25	26,46	0,55	34,58	0,55	45,84	1,72	54,43	0,67
min значе- ние	15,13	0,16	24,9	0,36	29,45	0,56	38,13	1,31	49,37	0,50

Таблица 2

Время работы разработанного алгоритма

Число выводов	ЦФ (разработанный алгоритм)	ЦФ(ГА)	Улучшение ЦФ, %	Время раб. (разработанный алгоритм), с
16	15,13	12,15	19,7	0,16
20	24,9	20,39	18,11	0,36
24	29,45	23,09	21,6	0,56
28	38,13	31,09	18,5	1,31
32	49,37	37,07	25	0,50

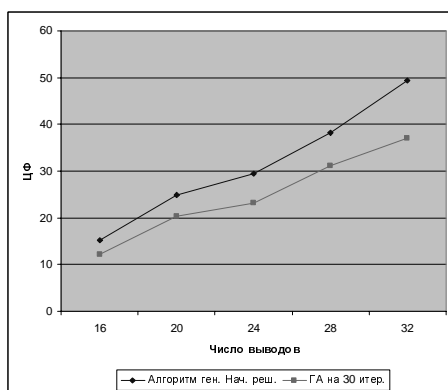


Рис. 3. Зависимость ЦФ для разработанного и генетического алгоритмов

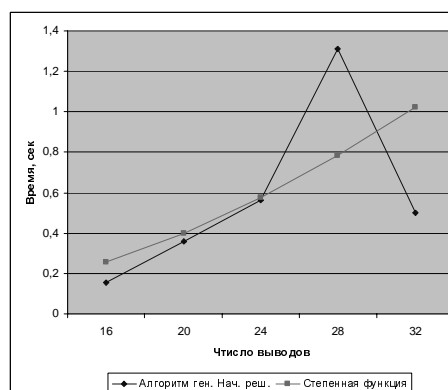


Рис. 4. Зависимость времени работы разработанного алгоритма от числа выводов

Заключение. Данный алгоритм за счет большого числа вариантов трансформации исходной топологии цепей, дает возможность генерировать разнообразные решения, что делает его очень удобным для использования в качестве алгоритма генерации начального решения в генетических алгоритмах. Временная сложность алгоритма составляет $O(n^2)$. Из преимуществ алгоритма можно выделить следующие:

- ◆ учет ширины цепей;
- ◆ соблюдение заданного интервала между цепями;
- ◆ возможность решения задачи трассировки, как в коммутационном блоке, так и в канале;
- ◆ бессеточная модель отражает действительные геометрические размеры соединений цепей.

При определенных модификациях данный алгоритм может учитывать различные физические эффекты, что при современных темпах развития технологий позволит сократить время проектирования электронных изделий и повысить их качество.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Bern M.W.* Two probabilistic results on rectilinear Steiner tree [текст] / in Proc. 18th Ann. Symp. Theory Computing, pp. 433-441, 1986.
2. *Gerez, S.H., Herrman, O.E.* Packer: a switch box routing based on conflict elimination by local transformation [текст] / Proc. Int. Conf. Circuits and Systems. 1989, pp. 961-964.
3. *Malgorzata Marek-Sadowska.* Electrical and Computer Engineering Department [текст] / University of California, Santa Barbara, CA 93106, USA.
4. *Marek-Sadowska,* Global router for gate array [текст] / Proc. Int. Conf. Computer Design, 1984, pp. 332-337.
5. *Shervani, N.* Algorithms for VLSI physical design automation [текст] / Kluwer Academy Publisher – USA, 1995. – 538 p.
6. *Курейчик, В.М.* Генетические алгоритмы [текст] / Монография. – Таганрог: ТРТУ, 1998. – 242 с.
7. *Гладков Л.А., Курейчик В.М., Курейчик В.В.* Генетические алгоритмы [текст] / Учебное пособие. – Ростов-на-Дону, РостИздаст, 2-е издание, 2006.
8. *Lienig J., Thulasiraman K.* A Genetic for Channel Routing in VLSI Circuits [текст] / Evolutionary Computation, vol. 1, no. 4, pp. 239-311, 1994.
9. *Adamidis P.* Review of Parallel Genetic Algorithms [текст] / Technical Report, Dept. of Electr. and Comp.Eng., Aristotle Univ. of Thessaloniki, 1994.