

Раздел II. Эволюционное моделирование, генетические и бионические алгоритмы

В.М. Курейчик

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ КОМПОНОВКИ НА ОСНОВЕ ДУБЛИРОВАНИЯ ЭЛЕМЕНТОВ*

Введение. При автоматизированном проектировании топологии СБИС возникают проблемы целостности сигналов при высоких скоростях передачи данных, несовместимость диэлектрических свойств и ширины проводников, и др. [1-6]. Все это усложняет проектирование для максимальной производительности СБИС. В этой связи требуются новые методы и подходы к проектированию. Одним из таких методов ухода от традиционных алгоритмов компоновки блоков СБИС являются бионические алгоритмы, использующие стратегии эволюционного моделирования. Они основаны на использовании генетических алгоритмов и генетического программирования [7,8]. Перспективны методы дублирования, когда некоторые элементы принципиальной или логической схемы СБИС могут принадлежать нескольким блокам разбиения. Такие способы могут быть экономически выгодными при современных технологиях производства СБИС [4]. Использование дублирования, троирования и «к-ирования» элементов позволяет во многих случаях уменьшить число внешних соединений блоков СБИС за счет увеличения числа логических элементов общей схемы. В работе предлагается гибридный генетический алгоритм дублирования элементов при компоновке СБИС, позволяющий повысить качество готовой компоновки, и получать набор квазиоптимальных решений.

Описание методики. Подграфы $G_i = (X_i, U_i)$ графа $G = (X, U)$, $X_i \subseteq X$, $U_i \subseteq U$, называются компонентами связности (компонентами) графа, если они не имеют друг с другом общих вершин и общих ребер. Тогда для компонент $G_i = (X_i, U_i)$, $G_j = (X_j, U_j)$, ..., $G_k = (X_k, U_k)$ имеем

$$X_i \cap X_j \cap \dots \cap X_k = \emptyset, U_i \cap U_j \cap \dots \cap U_k = \emptyset.$$

Причем

$$\bigcup_{i=1}^k X_i = X, \bigcup_{i=1}^k U_i = U, i \neq j \Rightarrow U_i \cap U_j \wedge X_i \cap X_j = \emptyset.$$

Граф можно разбить на компоненты, внутри которых вершины не связаны между собой. Ребро $u_i \in U$ графа $G = (X, U)$, удаление которого приводит к увеличению компонент связности, называется мостом.

Теорема Харари [9]. Пусть $u_i \in U$ ребро связного графа G . Следующие утверждения эквивалентны:

- ♦ u_i – мост графа G ;

* Работа выполнена при частичной финансовой поддержке целевой программы «Развитие научного потенциала высшей школы» (2006-2008) РНП 2.12.3193

- ◆ мост u_i не принадлежит ни одному простому циклу графа G ;
- ◆ в графе G существуют такие вершины x_k, x_l , что ребро u_i принадлежит любой простой цепи, соединяющей x_k с x_l ;
- ◆ существует разбиение множества вершин X графа G на такие подмножества X_q, X_p , что для любых вершин $x_k \in X_q$ и $x_s \in X_p$ ребро u_i принадлежит любой простой цепи, соединяющей x_k и x_s .

Точкой сочленения графа $G = (X, U), |X| = n, |U| = m$ называется вершина $x_i \in X$, удаление которой увеличивает число несвязных компонент графа. Следовательно, если $x_i \in X$ точка сочленения графа, то граф $G - x_i$ не связан [9].

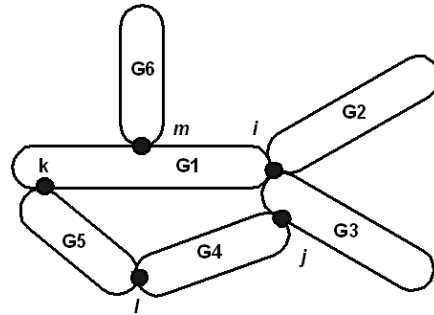


Рис.1. Граф из шести частей с точками касания i, j, k, l, m

Граф, не имеющий точек сочленения, называется неразделенным. Блоком графа называется его максимальный неразделенный подграф.

Приведем модифицированный алгоритм [10] определения точек сочленения и блоков:

1. Анализируем поочередно все вершины графа $G = (X, U), |X| = n, x_i \in X, i=1, 2, \dots, n$, удаляя одну за другой с инцидентными ребрами.
2. Просматриваем поочередно полученные подграфы и проверяем получение компонент связности. В результате получаем все точки сочленения G .
3. Образует систему строк. Каждая строка соответствует точке сочленения. Причем в эти строки добавлены индексы соответствующих точек сочленения.
4. Выполним операцию пересечения строк первой со второй, третью с результатом пересечения первой и второй строк и т. д. В результате получим все блоки графа и точки сочленения.
5. Конец работы алгоритма.

Для практических задач важно нахождение частей графа, имеющих точки касания (см. рис.1) между частями графа. На рис.1 показаны пять точек касания i, j, k, l, m которые соединяют шесть частей графа.

Алгоритм определения таких частей основан на эвристике жадного поиска и генетических алгоритмах [7,8]. Она эффективна, когда заранее заданы размеры частей G_i и количество возможных точек соприкосновения

Для решения NP – полных проблем компоновки предлагается анализировать структуру модели исходной схемы, чтобы "выращивать" полные решения, рекурсивно расширяя последовательные частичные решения. Основой жадного алгоритма является локально-оптимальный выбор на каждом шаге с прогнозом, что окончательное решение будет оптимальным. Очевидно, что в общем случае жад-

ный алгоритм может привести в локальный минимум, который далек от оптимального. Хотя для многих задач проектирования эти алгоритмы дают возможность получать оптимум искомой целевой функции.

Считается, что к оптимизационной задаче проектирования применим принцип жадного выбора, если последовательность локальных жадных выборов позволяет получить искомое решение. Жадный алгоритм на каждом шаге делает выбор частного оптимального решения. Затем алгоритм делает наилучший выбор среди оставшихся частных решений и т.д.

Приведем ряд эвристик жадного выбора:

Э1 – Жадный выбор на первом и последующих шагах не должен закрывать путь к оптимальному решению.

Э2 – Для упрощения жадного выбора и повышения скорости подзадача, вытекающая после жадного выбора на первом шаге, должна быть аналогична исходной.

Э3 – Задачи, решенные на основе жадных алгоритмов, должны обладать свойством оптимальности для подзадач.

Жадная стратегия такого типа является простейшей. Она позволяет одновременно определять блоки и точки сочленения графа. Данный алгоритм применялся для определения точек касания и дал хорошие результаты. Его временная сложность, т.е. зависимость времени работы от числа элементов схемы (вершин графа) составляет в среднем $O(n)$.

На рис.2 приведена схема параллельного генетического алгоритма с миграцией хромосом (альтернативных решений) для определения блоков и точек сочленения графов. В первом блоке формируется популяция альтернативных решений задачи. Во втором блоке происходит разбиение популяции на три части. Они представляют собой элитные (лучшие решения), худшие и среднего качества. В третьем блоке – вычисление целевых функций (ЦФ) задачи компоновки. В четвертом блоке выполняется анализ и выбор альтернативных решений для дальнейшего поиска локально-оптимальных результатов. Далее в каждой части реализуется гибридный генетический жадный алгоритм. В первой части в блоке генетических операторов (ГО) выполняется жадный алгоритм, во второй – генетический, а в третьей – комбинированный. Блоки миграции осуществляют обмен решениями для предотвращения преждевременной сходимости алгоритмов. В блоках репродукции определяются перспективные решения для дальнего анализа. В блоках ранжирования и селекции происходит упорядочение анализируемых решений [7,8]. Генетические алгоритмы (ГА) обычно используют представление фиксированной длины. Строки, обработанные обычными ГА, сохраняют постоянное число бит, каждый из которых соответствует конструкции фенотипа. В работе предлагается каждое альтернативное решение (хромосому) представлять в виде переменной длины. Вместо того, чтобы определять хромосому как последовательность битов, ее, подобно естественной копии будем определять как последовательность генов. Ген, в свою очередь, определен как последовательность позиций хромосомы, которые несут информацию о специфической особенности в фенотипе [11]. Эта модель, согласно [11] имеет вид: $G = \cup(g_i^{V_i})$. Здесь g_i - специфический ген, $i=1, 2, \dots, n$; $V_i \in \{0, 1\}$.

Ген g_i может быть представлен как целочисленный или вещественный вектор. n – общее количество генов в хромосоме. V - двоичный вектор, определяющий положение активации каждого гена g_i : если $V = 1$, то его связанный ген, g_i , активизируется; и наоборот. Тогда модель хромосомы - двойная строка, составленная из совокупности генов, и вспомогательной двоичной строки V . Первая строка - *главная строка*, вторая строка – *маска активации*. В этом случае ГА может представлять решения произвольных размеров. Оператор кроссинговера (ОК) - анализирует со-

держание двух строк, чтобы произвести потомков. Оператор мутации (ОМ) выполняется так же, как и в стандартных ГА, и действует только на главную строку хромосомы. В [11] вводится новый оператор проекции (ОП). Этот оператор дает возможность управлять способом, которым ГА охватывает различные области поиска. ОП применяется только к маске активации. Он подобен ОМ: каждый бит маски активации изменяется с низкой вероятностью, активизируя или деактивируя соответствующие гены. Существует 3 проекционные стратегии, изменяя способ применения этого оператора [11]: увеличивающейся длины генотипов, колеблющейся длины генотипов и равномерно распределенной начальной популяции [11]. Вводится еще две стратегии: стандартная и комбинированная. Когда не могут быть найдены лучшие решения, при увеличении числа компонентов, популяция будет сводиться к меньшим решениям.

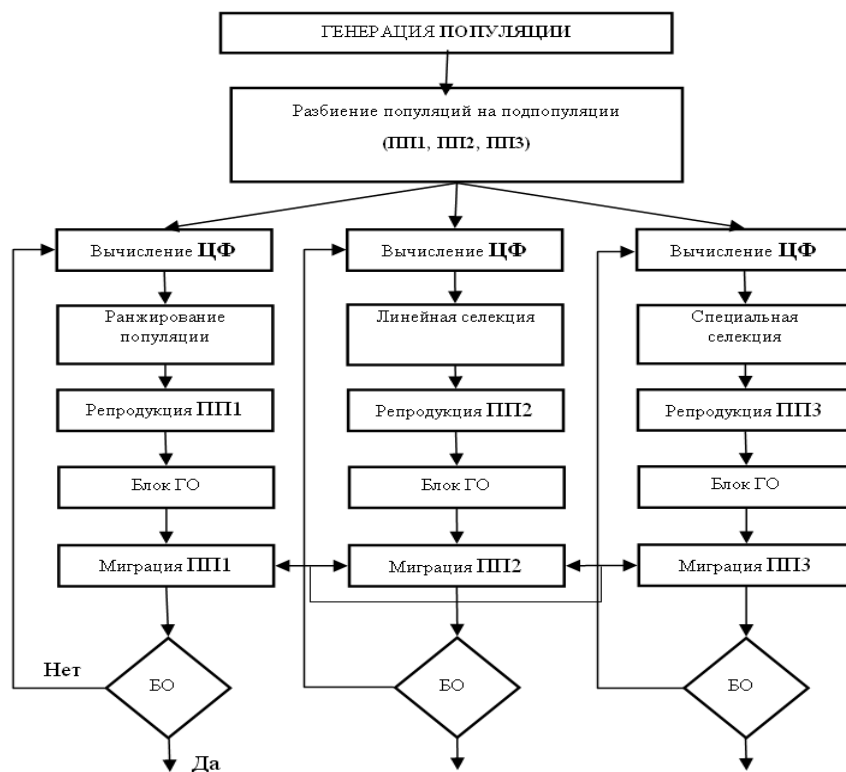


Рис.2. Схема параллельного ГА с миграцией

Наилучшее решение при проектировании состоит в том, чтобы заново проанализировать схему, производство и компоновку кристалла СБИС. Важная задача проектирования СБИС – переориентировать маршрутизацию высокоскоростных сигналов СБИС по оптимальным путям через управляемый канал. Для этого нужно установить маршруты сигнала в трехмерном измерении или применить генетические и эволюционные методы проектирования.

Модифицированные методы проектирования основаны на упаковке кристалла с разделением высоких и низких скоростей элементов и трассировки их в разных частях. Разделение высоких и низких скоростей сигналов упрощает межкомпонентное соединение более технологичных корпусов и подложек БИС.

Разбиение компоновки СБИС, разделение высоких и низких скоростей, заземление схемы и производительность могут устранить "дорожные заставы" (препятствия) при передаче от чипа к чипу [3-6].

Одним из способов решения указанных проблем является уменьшение числа выводов схемы путем введения дополнительных (необязательных) элементов, т.е. дублирования элементов. Причем дублированию предлагается подвергать элементы схемы, соответствующие точкам сочленения графа или специальным точкам i, j, k, l, m (см. рис.3). Это позволяет уменьшить число внешних связей. Если точка сочленения только одна, то число внешних связей равно единице. Здесь произведено дублирование элементов j, l, k, m и троирование элемента i .

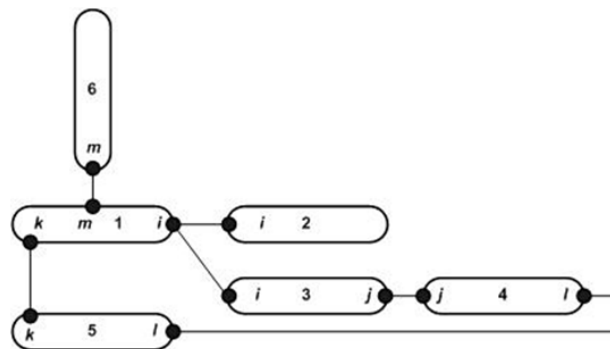


Рис.3. Пример дублирования и троирования элементов

В заключение отметим, что описанная процедура позволяет уменьшить число выводов за счет увеличения числа логических элементов схемы. Это особенно эффективно, когда схема может быть распределена на кристалле таким образом, что элементы, соответствующие точкам сочленения графа и специальным точкам могут принадлежать нескольким блокам.

Данная методика компоновки элементов открывает пути совместного решения задач логического и конструкторского проектирования СБИС и электронных средств на их основе. Представляется перспективным использование методов дублирования элементов при проектировании систем-на-кристалле.

Проведенные серии тестов и экспериментов позволили уточнить теоретические оценки временной сложности алгоритмов компоновки с дублированием. Временная сложность алгоритма (ВСА) $\approx O(n) - O(n!)$, где n – число элементов схемы. ВСА и точность получения результата находятся как бы в противоречии, т.к. увеличение числа генераций просмотра приводит к качественным результатам за большее время.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Казеннов Г. Основы проектирования интегральных схем и систем. – М.: Бином. Лаборатория знаний, 2005. – 295 с.
2. Норенков Н.П. Основы автоматизированного проектирования. Учебник для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. – 360 с.
3. Немудров В., Мартин Г. Системы-на-кристалле. Проектирование и развитие. – М.: Техносфера, 2004. – 216 с.
4. Фридман А., Менон П. Теория и проектирование переключательных схем. Пер. с англ. – М.: Мир, 1978. – 580 с.
5. Shervani N., Algorithms for VLSI physical design automation. – USA. Kluwer Academic Publishers, 1995. – 538 p.

6. Проблемы разработки перспективных микроэлектронных систем. Сборник научных трудов под ред. А. Стемповского. – М.: ИППМРАН, 2005. – 537 с.
7. Емельянов В.В., Курейчик В.М., Курейчик В.В. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003. – 431 с.
8. Гладков Л.А., Курейчик В.М., Курейчик В.В. Генетические алгоритмы. Учебное пособие. Под ред. В.М. Курейчика. – Ростов-на-Дону: ООО «Ростиздат», 2004. – 400 с.
9. Харари Ф. Теория графов. – М.: Мир, 1973. – 300 с.
10. Зыков А.А. Теория конечных графов. – Новосибирск: Изд-во Наука, Сибирское отделение, 1969. – 543 с.
11. Zebulum R.S., Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms (CRC Press LLC, New York, 2002).

Б. К. Лебедев, А. Н. Дуккардт

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ РАЗБИЕНИЯ С АНТИМОНОПОЛЬНОЙ ПОЛИТИКОЙ РАЗВИТИЯ ОСОБИ ДЛЯ МИНИМИЗАЦИИ ЧИСЛА МЕЖУЗЛОВЫХ СОЕДИНЕНИЙ И ВРЕМЕННЫХ ЗАДЕРЖЕК*

Введение. В последнее десятилетие появилось много работ по решению задачи разбиения с учетом временных задержек [2,7,8,10]. В большинстве предыдущих подходов минимизация задержек осуществлялась путем внесения изменений в списки цепей с целью удовлетворения временных ограничений. При этом решалась задача минимизации числа разрезов цепей. В результате проведенного анализа предыдущих подходов к задаче разбиения, можно выделить несколько проблем:

- ◆ Используются нереалистичные модели задержек. В основном используется общепринятая модель задержек, в которой всем вентилям присваивается задержка 1, задержка 0 присваивается соединениям внутри одного узла и постоянная задержка присваивается соединениям между различными частями разбиения [2,7,10].
- ◆ Выполняются нереалистичные упрощения. Например, схемы преобразовываются в вентили с двумя входами [7].
- ◆ Время работы алгоритмов даже для схем умеренного размера слишком велико.

В работе целевая функция изменена таким образом, чтобы она учитывала число разрезов цепей и задержку сигнала. С целью достижения более реалистичной модели задержки модель Эльмора используется вместе с различными задержками проводников для каждой цепи.

Постановка задачи. В качестве модели схем при разбиении чаще всего используется гиперграф. На основе гиперграфовой модели задача разбиения формулируется следующим образом: дан гиперграф $G = (X, E)$, где $X = \{x_i \mid i = 1, 2, \dots, n\}$ – множество вершин, а $E = \{e_j \mid e_j \subset X, j = 1, 2, \dots, m\}$ – множество рёбер (каждое ребро – подмножество связываемых им вершин). Вес вершин задаётся множеством $\Phi = \{\varphi_i \mid i = 1, 2, \dots, n\}$, а вес рёбер – множеством $\Psi = \{\psi_j \mid j = 1, 2, \dots, m\}$. Необходимо сформировать K – узлов, т.е. множество X разбить на K непустых и непересекающихся подмножеств X_k :

$$X = \cup X_k \quad (\forall i, j), \quad X_i \cap X_j = \emptyset, \quad X_k \neq \emptyset.$$

* Работа выполнена при финансовой поддержке программы развития научного потенциала высшей школы РНП.2.1.2.2238