

**А.Г. АРУТЮНЯН, А.Р. МАРТИРОСЯН**

## **ТРАНСФОРМАЦИЯ VERILOG ОПИСАНИЯ ЛОГИЧЕСКОЙ СХЕМЫ В ГРАФ ПОТОКА СИГНАЛОВ**

Предложен метод преобразования Verilog описания логической схемы в граф потока сигналов, что дает возможность автоматизированного размещения элементов на основе ранжирования заданной схемы. Предложенный метод основан на представлении Verilog описания логической схемы в виде двоичного дерева данных. Реализовано программное обеспечение предложенного метода.

**Ключевые слова:** граф потока сигналов, ранжирование элементов, двоичное дерево.

**Введение.** Характерные к субмикронным технологиям реалии выдвигают новые требования к методам и инструментальным средствам проектирования интегральных схем (ИС). Эти требования в первую очередь относятся к усилению взаимосвязи между схмотехническим и физическим этапами проектирования ИС. Если учесть, что в современных цифровых ИС задержка сигнала в межсоединениях в несколько сотен раз превышает задержку сигнала в логических вентилях и в целом 80...90% задержек ИС приходится на межсоединения, то можно сказать, что учет схмотехнических особенностей на этапе автоматизированного физического проектирования ИС является актуальной проблемой, и его важность растет вместе с развитием технологий [1,2].

С точки зрения учета схмотехнических особенностей работы ИС на этапе физического проектирования, в настоящее время одной из важнейших задач физического проектирования ИС считается учет задержек в цепях при размещении элементов. Эта задача названа задачей временем направленного размещения [3]. Существующие методы временем направленного размещения можно разделить на две группы: методы, основанные на пути, в которых учитывается задержка сигнала на путях от первичных входов к выходам схемы, и методы, основанные на цепи, в которых учитывается задержка сигнала в каждой цепи схемы. Оба подхода имеют свои преимущества и недостатки, которые подробно рассмотрены в [4]. Однако в них имеется один общий недостаток, который заключается в том, что они не учитывают логику работы цифровых схем, в смысле учета логической последовательности работы элементов схемы при их размещении.

С этой целью в данной работе предлагается новый подход учета логических особенностей работы схемы при физическом проектировании ИС. Пред-

лагаемый подход основан на принципе использования потоков сигналов в комбинационных логических схемах при размещении элементов. Согласно предлагаемому принципу, проводится предварительное ранжирование элементов, результаты которого в дальнейшем используются при размещении элементов.

**Постановка задачи и описание метода.** Предлагаемый подход учета функционально-логических особенностей схемы при физическом проектировании ИС основан на применении графа потока сигналов (ГПС) при размещении логических ячеек. Он предполагает последовательное решение следующих задач:

1. Преобразование заданного Verilog описания логической схемы в ГПС.
2. Ранжирование элементов схемы.
3. Размещение элементов согласно их ранжированию, т.е. получение “мягкого” блока.
4. Переразмещение элементов полученного “мягкого” блока с учетом конструктивных ограничений, т.е. получение “жесткого” блока.

Под “мягким” блоком подразумевается непараметризованный блок произвольной формы, а под “жестким” - параметризованный блок прямоугольной формы, который можно импортировать в базу данных физического проектирования в виде файла GDS2. Такой подход будет способствовать сокращению длины межсоединений и количества их пересечений.

ГПС представляет собой ориентированный граф  $G(V,E)$ , где  $E\{e_1, e_2, \dots, e_m\}$  - множество вершин графа, соответствующих элементам схемы;  $V\{v_1, v_2, \dots, v_m\}$  - множество ребер графа, соответствующих цепям схемы, а  $m$  и  $n$  - соответственно количество элементов и цепей.

Таким образом, с точки зрения проектировщика, ГПС отражает логику обработки потока данных в заданной логической схеме. Цель такого представления - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Предлагаемый метод трансформации Verilog описания логической схемы в ГПС основан на представлении схемы в виде двоичного дерева поиска, основным преимуществом которого перед другими структурами данных является высокая эффективность реализации основанных на нём алгоритмов поиска и сортировки.

Описание предлагаемого метода рассмотрим на примере построения ГПС простейшей тестовой схемы с17 серии Iscas85 [5], приведенной на рис1.

```

1 // c17 Verilog file
2
3 module c17(G1,G16,G17,G2,G3,G4,G5);
4 input G1,G2,G3,G4,G5;
5 output G16,G17;
6
7 wire G8,G9,G12,G15;
8
9 nand NAND2_0(G8,G1,G3);
10 nand NAND2_1(G9,G3,G4);
11 nand NAND2_2(G12,G2,G9);
12 nand NAND2_3(G15,G9,G5);
13 nand NAND2_4(G16,G8,G12);
14 nand NAND2_5(G17,G12,G15);
15
16 endmodule

```

*Рис. 1. Verilog описание тестовой схемы c17 серии Iscas85*

Предлагаемый метод реализует следующие шаги:

1. Синтаксический анализ Verilog описания схемы.
2. Создание идентифицированного массива логических элементов схемы.
3. Создание идентифицированных массивов входных, выходных и связывающих цепей схемы.
4. Создание двоичного (бинарного) дерева поиска.
5. Синтез графического отображения логической схемы на основе п.4.
6. Преобразование логической схемы в “мягкий” блок путем замены логических элементов их физическими изображениями и ранжирования.
7. Преобразование “мягкого” блока в “жесткий” с учетом заданных временных и физических ограничений.

В настоящей работе рассмотрены задачи пп. 1 - 5, а также частично 6-й пункт.

Так как предлагаемый метод трансформации Verilog описания логической схемы в ГПС реализован на языке программирования C++, то целесообразно рассмотрение особенностей решения задач вышеприведенных пунктов в среде объектно-ориентированного программирования.

При синтаксическом анализе игнорируются пустые строки и строки, начинающиеся с символа “//” (в нашем примере - это строки 1, 2, 6, 8 и 15).

Далее формируется класс ГПС под названием “Graph”, который содержит следующие массивы:

- “Cell” - массив идентифицированных логических ячеек (в нашем примере шести ячейкам NAND2\_0, NAND2\_1, ..., NAND2\_5 будут присвоены идентификационные номера id=0, 1, ..., 5);

- “Wire” - массив идентифицированных цепей, в том числе “Input”-входных и “Output”-выходных (в нашем примере последовательности цепей G1, G16, ..., G15 будут присвоены идентификационные номера id=0, 1, ..., 10);

- “map”-двоичное сбалансированное дерево представления данных.

В массивах “Cell” и “Wire” формируется также информация об их привязанности. Эта информация представляется с помощью ребер ориентированного графа, соединяющих ячейки с инцидентными им цепями. Например, в рассматриваемой схеме цепь G9 идентифицируется ориентированным ребром с id=8, соединяющим выход элемента NAND2\_1, имеющий id=1, с входами элементов NAND2\_2 и NAND2\_3 с id=2 и id=3.

На основе этой информации в классе Graph формируется двоичное сбалансированное дерево представления данных с названием “map”. Любая вершина “map”-а представляет пару {key, value}, где key показывает id данной вершины, а value - ее значение.

Запись массивов “Cell” и “Wire” осуществляется таким образом, что каждому значению вводимой ячейки или цепи соответствует его id, что в дальнейшем обеспечит постоянство времени обращения к ячейке или цепи с заданным значением, т.е. эта процедура имеет алгоритмическую сложность  $O(1)$ .

Организация поиска на основе двоичного сбалансированного дерева представления данных позволяет снизить алгоритмическую сложность построения ГПС, по сравнению с традиционным алгоритмом поиска в глубину, имеющим сложность  $O(n^2)$ , до  $O(n \log n)$  [6].

**Результаты исследования.** На основе вышеприведенной информации, реализован графический интерфейс, который позволяет реализовать следующие процедуры:

- ввод Verilog файла заданной схемы;
- получение ГПС схемы;
- графическое отображение ГПС в виде случайного размещения логических ячеек и связей между ними (рис. 2а);
- геометрическое перемещение логических ячеек с сохранением правильности электрических соединений (рис. 2б).

В разработанной программе логические ячейки представлены в виде абстрактного представления их топологии.

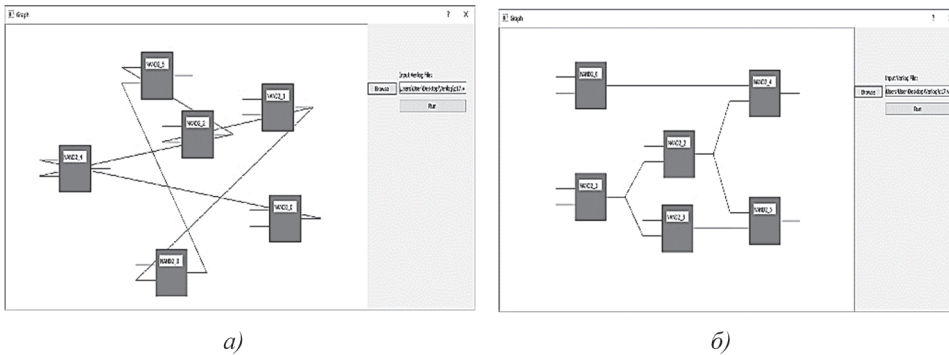


Рис. 2. Графическое отображение ГПС при случайном (а) и упорядоченном (б) размещении ячеек и цепей тестовой схемы с17

Реализация программы для ряда тестовых схем серии ISCAS85 показала зависимость времени работы программы от количества элементов (рис. 3), что подтверждает приведенную выше алгоритмическую сложность  $O(n \log n)$ .

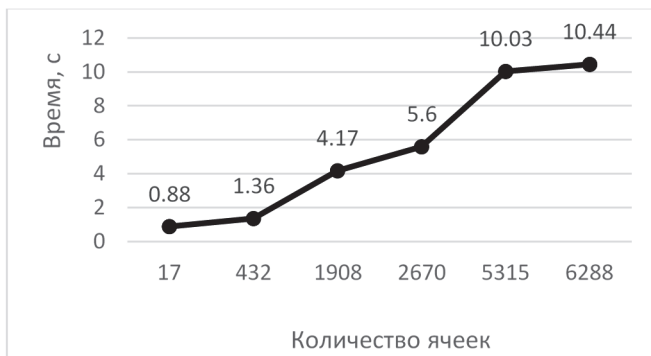


Рис. 3. Зависимость времени работы программы от количества элементов

В настоящее время ведутся работы по автоматизированному ранжированию логических ячеек и синтеза “мягкого” и “жесткого” блоков проектируемой схемы.

**Заключение.** Предложена концепция автоматизированного физического синтеза цифровых схем, основанная на ГПС. Разработаны быстродействующий метод и соответствующая программная реализация построения и применения ГПС, позволяющая снизить алгоритмическую сложность решения задачи с  $O(n^2)$ , до  $O(n \log n)$ .

## СПИСОК ЛИТЕРАТУРЫ

1. The International Technology Roadmap for Semiconductors (ITRS), 2015 Edition, More Moore,  
[https://www.semiconductors.org/clientuploads/Research\\_Technology/ITRS/2015/5\\_2015%20ITRS%202.0\\_More%20Moore.pdf](https://www.semiconductors.org/clientuploads/Research_Technology/ITRS/2015/5_2015%20ITRS%202.0_More%20Moore.pdf)
2. **Khang A., Carballo J.**, ITRS 2.0, System Integration Focus Team. Stanford. -2015. - 39 p.
3. **Naveed A. Sherwani.** Algorithms for VLSI Physical Design Automation. Intel Corporation.-Kluwer Academic Publishers, 2007. - 572 p.
4. **Մելիքյան Վ.Շ., Հարությունյան Ա.Գ., Գևորգյան Ա.Ա.** Միկրոէլեկտրոնային սխեմաների ֆիզիկական նախագծման մեթոդներ: Մենագրություն/ ՀՊՃՀ.-Եր.: Ճարտարագետ, 2015. - 240 էջ:
5. ISCAS85 Combinational Benchmark Circuits.  
<https://filebox.ece.vt.edu/~mhsiao/iscas85.html>
6. **Кормен Т., Лейзерсон Ч., Ривест Р.** Глава 22. Элементарные алгоритмы для работы с графами // Алгоритмы: построение и анализ (2-е издание). - М.: Вильямс, 2005. - С. 622-632.

### Ա.Գ. ՀԱՐՈՒԹՅՈՒՆՅԱՆ, Ա.Ռ. ՄԱՐՏԻՐՈՍՅԱՆ

#### ՏՐԱՄԱԲԱՆԱԿԱՆ ՍԽԵՄԱՅԻ VERILOG ՆԿԱՐԱԳՐԻ ՓՈԽԱՐԿՈՒՄԸ ԱԶԴԱՆՇԱՆՆԵՐԻ ՀՈՍՔԻ ԳՐԱՖԻ

Առաջարկված է տրամաբանական սխեմայի verilog նկարագրից ազդանշանների հոսքի գրաֆի փոխարկման մեթոդ, որը հնարավորություն է տալիս տրված սխեմայի ունգավորման հիման վրա կատարել նրա տարրերի ավտոմատացված տեղաբաշխում: Առաջարկված մեթոդը հիմնված է տրված սխեմայի Verilog նկարագիրը տվյալների ներկայացման երկուական ծառի տեսքով նկարագրման վրա: Իրագործված է առաջարկված մեթոդի ծրագրային ապահովումը:

**Առանցքային բառեր.** ազդանշանների հոսքի գրաֆ, տարրերի ունգավորում, երկուական ծառ:

### A.G. HARURYUNYAN, A.R. MARTIROSYAN

#### TRANSFORMING THE VERILOG DESCRIPTION OF THE LOGICAL CIRCUIT IN THE SIGNAL FLOW GRAPH

A method for converting the Verilog description of a logical circuit into a signal flow graph is proposed, which makes it possible to automate the placement of elements based on the ranking of a given circuit. The proposed method is based on the presentation of the Verilog description of the logical scheme in the form of a binary data tree. The software of the proposed method is implemented.

**Keywords:** signal flow graph, ranking of elements, the binary tree.