УДК 519.711.3

ГРАФИЧЕСКОЕ ОТОБРАЖЕНИЕ ЛОГИЧЕСКИХ СЕТЕЙ, ПОЛУЧЕННЫХ ДЕКОМПИЛЯЦИЕЙ ОПИСАНИЙ СХЕМ ТРАНЗИСТОРНОГО УРОВНЯ

Д.И. Черемисинов
Ведущий научный сотрудник ОИПИ НАНБ
кандидат технических наук, доцент
cher@newman.bas-net.by

Л.Д. Черемисинова
Главный научный сотрудник ОИПИ НАНБ
доктор технических наук, профессор
cld}@newman.bas-net.by

Д.И. Черемисинов
Окончил радиофизический факультет Томского государственного университета. Область научных интересов: программирование, логическое проектирование и тестирование дискретных систем управления, реализация параллельных алгоритмов управления.

Л.Д. Черемисинова
Окончила радиофизический факультет Томского государственного университета. Область научных интересов: логико-комбинаторные задачи, логическое проектирование и тестирование дискретных систем управления, реализация параллельных алгоритмов управления.

Аннотация. В работе рассматривается задача генерации графических изображений для графов, представляющих логические сети. Анализируются известные алгоритмы рисования графов и предлагается алгоритм, ориентированный на диалоговую визуализацию логических сетей, построенных в результате работы САПР. Задача визуализации логической сети существенно отличается по критериям оптимизации от задач размещения и трассировки, возникающих на этапе технического проектирования дискретных устройств.

Ключевые слова: обратная инженерия, визуализация графов, графическое отображение схем, SPICE формат.

Введение. Обратная инженерия (*Reverse Engineering*) [1, 2] заключается в исследовании готового устройства (или программы) с целью понять принцип его работы. Обратная инженерия является инверсией разработки устройства в смысле направления процесса преобразований: её задача заключается в построении спецификации путем анализа продукта проектирования. Обратное проектирование в общем случае состоит из следующих стадий: 1) анализ продукта; 2) извлечение описания продукта промежуточного уровня; 3) анализ описания продукта интеллектом человека, для построения спецификации. Разработка нового продукта, используя построенную спецификацию, называется перепроектированием (reingineering).

В контексте обычной разработки аппаратных компонент обратное проектирование часто рассматривается как незаконное действие. Обоснованием запрещения является то, что намерением применения обратной инженерии служит определение функциональности устройства, состава и структуры связей его компонентов. В этом смысле обратное проектирование может привести к серьезным проблемам, которые касаются нарушения прав

интеллектуальной собственности, эффективности мер, связанных с безопасностью устройства, и даже возможностью для внедрения аппаратных троянов.

Однако обратная инженерия – это единственный надежный метод обнаружения злонамеренных изменений или подделок со стороны предприятий по производству полупроводников. Она обеспечивает возможность найти имеющиеся уязвимости в готовых коммерческих микросхемах, а также перепроектировать устаревшее (т.е. больше не производимое) оборудование. С помощью обратного проектирования на уровне чипа извлекается удобочитаемый список соединений уровня транзисторов из исследуемой интегральной схемы (*IC*) или *FPGA*. На этом этапе функциональный анализ списка соединений не проводится. Следующим этапом обратного инжиниринга является декомпиляция [3, 4] плоского описания транзисторной схемы, которая состоит в извлечении из него описания уровня логических элементов.

Результатом декомпиляции плоского описания транзисторной схемы является схема из логических элементов, представляющая собой текстовое описание, которое не совсем удобно для восприятия человеком в процессе анализа функционирования устройства. Следовательно, возникает нужда в автоматическом построении графического представления – рисунка функциональной схемы по ее текстовому описанию. Для представления структурных моделей в современных САПР используются специальные текстовые языки описания данных, называемые форматами структурных описаний.

В настоящей работе рассматриваются проблемы, возникающие при синтезе графических изображений схем из логических элементов, полученных в результате декомпиляции плоских описаний схем транзисторного уровня в формате *SPICE* [5], а также методы и программные средства автоматической визуализации таких схем.

Автоматический синтез изображений функциональных схем электронных устройств. До появления больших интегральных схем графическое изображение функциональной схемы устройства использовалась для прослеживания распространения сигналов между его элементами, и поэтому оно было важнейшим средством разработки и документирования. Стандарты построения таких диаграмм приняты как на государственном, так и международном уровне.

Автоматическое построение изображения функциональной схемы можно рассматривать как задачу визуализации информации (графа), определяющего структуру соединений компонентов устройства. Целью визуализации является обеспечение быстрого и эффективного восприятия пользователем информации о структуре устройства за счет формирования наглядных рисунков абстрактных структурных данных. Проблема визуализации информации состоит в том, чтобы создать алгоритмы и программы, которые генерируют такие рисунки.

Ключевой проблемой визуализации графа является его размер. Известно, что понимание и детальный анализ структуры данных, представленных рисунком графа, возможны, когда размер визуализируемого графа не велик. Обратное налагает жесткие ограничения на быстродействие программы, и в алгоритме рисования необходим учет пределов возможностей аппаратуры визуализации. Даже если аппаратура позволяет показать все элементы, возникает проблема «читабельности» или удобства и простоты восприятия данных, из-за невозможности различить отдельные вершины и ребра. Фактически, удобство и простота восприятия становится проблемой даже прежде, чем достигнута неразличимость деталей.

Большие графы чрезвычайно трудно нарисовать в визуально воспринимаемом виде, который позволял бы понимать структуру этого изображения. На первый взгляд кажется бессмысленным визуализировать большие и сложные графы. Особенности человеческого восприятия таковы, что эффективным может быть только восприятие информации с небольшой сложностью структуры. Когда граф используется как модель некоторой

предметной области, то его большие размеры говорят о плохой ее структуризации при формализации задачи. Следовательно, для улучшения восприятия нужно изменять способ формализации задачи с целью получения графов визуально обозримых размеров. Однако при визуализации результатов работы САПР приходиться добиваться воспринимаемости не «в общем», а со специальными целями, и для графов произвольных размеров.

Так как любой граф может быть представлен с помощью неограниченного числа рисунков, то существует бесконечное множество различных алгоритмов рисования графов. Применимость того или иного алгоритма для визуализации данных зависит от набора комбинаторно-логических задач, процедуры решения которых составляют его алгоритмический базис. Этот базис определяет и эстетическую предпочтительность рисунка, и эффективность самого алгоритма.

Задача рисования графа может быть сформулирована следующим образом: даны множество вершин и множество ребер (отношений), требуется вычислить положение вершин и кривых, которые изображают ребра. К настоящему времени известно множество способов построения хорошего рисунка графа [6].

Требуемое качество рисунка графа прямо зависит от прикладной области, в которой этот объект используется. Поэтому в алгоритме рисования графа нужно принять во внимание эстетику: критерии качества визуального отображения существенных характеристик графа. Оценки удобочитаемости и «существенность характеристик» субъективны и зависят от цели, для которой генерируется рисунок. В литературе известны формальные критерии, заменяющие эстетический аспект при автоматическом построении изображения графа. Например, сформулированы следующие принципы организации рисунка, улучшающие его восприятие [7, 8]:

Формально набор критериев, улучшающих эстетику рисунка, состоит из следующих требований:

– минимум пересечений ребер;

– линии ребер проводятся настолько прямо, насколько это возможно;

– вершины графа должны быть равномерно распределены на плоскости рисунка;

– большинство дуг должны быть нарисованы в одном направлении;

– в ломаных линиях число изгибов должно быть минимально;

– минимум площади рисунка.

Визуализация схем, полученных декомпиляцией *SPICE* описаний. Для оперативного построения изображения при визуализации схем использована следующая методика. Данные в формате *SPICE* [5], задающие элементы и логические связи элементов проектируемой схемы преобразуется в формат EDIF (обменный формат описания электронных схем). Затем данные в формате *EDIF* дополняются графической компонентой (перечнем линий, представляющих изображение принципиальной схемы). Вывод изображения, заданного в таком виде, может осуществляться программами промышленных САПР СБИС. Например, наиболее распространенными инструментами САПР, поддерживающими формат *EDIF* 3 0 0, являются *Mentor Graphics DesignArchitect, Mentor Graphics Viewdraw, Cadence ConceptHDL, Cadence OrCAD Capture*.

В качестве основы для организации изображения принципиальной схемы принята каскадная структура связей элементов. Каскады элементов размещаются по вертикальным рядам. Элементы любого каскада расположены в ряду таким образом, что входы элементов находятся слева, а выход – справа. Предполагается, что графические символы всех элементов имеют одинаковую ширину (размер по горизонтали). Размер графического символа по вертикали зависит от числа входов элемента. Межкаскадные связи между соседними каскадами расположены в вертикальных полосах между рядами элементов. Линии не соседних межкаскадных связей проходят выше символов элементов. Линии соединений

элементов, расположенных в соседних каскадах имеют форму вилок, состоящих из отрезков прямых. Ручка вилки соединена с выходом, а зубцы с входами.

При разработке предлагаемой методики организации изображения схемы за основу было принято самое простое решение проблемы – такая формализация эстетического критерия, которая требовала бы минимальных затрат труда программиста при ее воплощении в программу. В предлагаемой организации изображения трассировка связей распадается на проведение связей между элементами из соседних каскадов и проведение всех оставшихся связей. Размещение и трассировка выполняются так, что критерий минимума длины связей и числа пересечений выполняется только для связей элементов соседних каскадов. Это позволяет сильно упростить комбинаторную сложность размещения и трассировки связей, облегчив программирование.

Для построения графического образа принципиальной схемы используется методика, подобная приведенной в [8]. Сначала проводится размещение символов элементов по вертикальным рядам, затем трассируются межэлементные соединения.

Чтобы правильно построить каскадную структуру схемы, нужно иметь информацию о разбиении выводов всех элементов на входные и выходные. Эта информация в представлении схемы в *SPICE* формате не может быть задана и ее невозможно однозначно восстановить на основе анализа структуры связей схемы. В результате декомпиляции строится двухуровневая схема из логических вентилей [4]. Все извлеченные вентили одновыходовые и в *SPICE* представлении выход указан последним по порядку выводом вентиля. Вентили, заданные в *SPICE* описании таким образом, позволяют построить логические сети, у которых можно определить входы и выходы. Формат *SPICE* не содержит средств указания и функций выводов сети. Чтобы отобразить эту информацию, в описании на *SPICE* логическая сеть выделена как отдельная модель (схема *C*0), ее параметры, имена которых начинаются с «*P*», задают входы схемы, параметры с именами, начинающимися с «*O*», – выходы схемы.

Размещение элементов начинается с построения их графических символов. В описываемой программе символ любого вентиля логической сети строится в виде прямоугольника, на правой стороне которого размещены входы, а на левой показан выход. Ширина прямоугольника выбрана такой, чтобы внутри его помещались обозначения типа вентиля, его названия и названия входов и выходов, заданных в *SPICE* описании. Высота символа элемента зависит от числа входов. Точкой привязки символа элемента считается верхний левый угол прямоугольника. Первый вход и выход относительно точки привязки вентиля расположены одинаково для символов любых элементов. Символ выхода не перемещается по вертикальной линии прямоугольника (как это предполагается в [7]), так как это приводит к неодинаковости символа для заданного типа элементов.

Элементы логической сети размещаются по каскадам. Для этого сеть элементов топологически сортируется по отношению связности. Номера каскадов для каждого элемента определяются так, как описано в [8]. Эта операция называется каскадированием. После распределения элементов сети по каскадам, фиксируется положение символа элемента в полосе каскада: решается задача называемая в [8] вертикальным планированием.

По информации о принадлежности элементов к каскадам и размерам символов элементов можно определить размер полосы соответствующего каскада. Точкой привязки полосы каскада служит точка привязки первого (верхнего) элемента в каскаде. В дальнейшем полосы каскадов в плоскости проектирования перемещаются как единое целое.

Трассировка связей начинается с выделения связей между элементами, расположенными в несоседних каскадах (несоседние межсоединения). В этот момент уже известны координаты по вертикали точек привязки каждого каскада.

Затем конструируются вилки, соответствующие сначала соседним, а затем и не соседним межсоединениям. В этот момент положение по вертикали ручек и зубцов каждой вилки определено, и задача конструирования вилки состоит в определении положения

вертикального сегмента. Координатой ручки вилки не соседнего соединения служит положение соответствующей соединительной линии в верхнем канале. Положение вертикальных сегментов вилок выбирается так, чтобы углы вилок не смыкались (и не накладывались). По окончании построения всех вилок очередного каскада определяется ширина полосы межсоединений. Это дает возможность зафиксировать координату по горизонтали точки привязки следующего каскада.

Заключение. Предлагаемый практичный алгоритм размещения элементов и трассировки связей между ними был использован при разработке программ, предназначенных для визуализации результатов работы программы декомпиляции плоских схем в формате *SPICE*, полученных в процессе обратного проектирования. Предложенный алгоритм предназначен для построения рисунка, представленного в обменном формате *EDIF*, используемом в качестве обменного в промышленных САПР СБИС. Потолок по размеру логической сети определяется возможностями средств визуализации. Практически строились изображения логических схем, содержащих до 10 тысяч элементов.

Список литературы

[1] Белоус А.И., Солодуха В. А. Основы кибербезопасности. Стандарты, концепции, методы и средства обеспечения. Москва: ТЕХНОСФЕРА, 2021. – 482 с. ISBN 978-5-94836-612-8

[2] Tehranipoor M., Koushanfar F.. A Survey of Hardware Trojan Taxonomy and Detection. IEEE Design & Test of Computers. 2009.

[3] Zhang N., Wunsch D.C., Harary F. The subcircuit extraction problem. Proceedings IEEE International Workshop on Behavioral Modeling and Simulation. 2003; 33(3): 22–25.

[4] Cheremisinov D., Cheremisinova L. Subcircuit Pattern Recognition in Transistor Level Circuits. Pattern Recognition and Image Analysis. 2020;30(2): 160–169.

[5] Baker R.J. CMOS Circuit Design, Layout, and Simulation, Third Edition. Wiley-IEEE Press, 2010. – 1214 p.

[6] Eades K., Sugiyama P. How to Draw a Directed Graph. Journal of Information Processing. 1990; 13(4): 424-437.

[7] Закревский А.Д. Графическое отображение комбинационных схем. Автоматика и вычислительная техника. 1990; 6: 59–65.

[8] Черемисинов Д.И. Автоматический синтез изображений функциональных схем электронных устройств. Вестник компьютерных и информационных технологий. Москва: Машиностроение, 2007; 2: 14–21.

Авторский вклад

Черемисинов Дмитрий Иванович – разработка программ визуализации результатов декомпиляции схем, являющихся логическими схемами формате SPICE.

Черемисинова Людмила Дмитриевна – постановка задачи исследования и обсуждение критериев качества графического изображения логических схем, полученных в процессе обратного проектирования транзисторных схем.

GRAPHICAL MAPPING OF LOGICAL NETWORKS OBTAINED BY DECOMPILING DESCRIPTIONS OF TRANSISTOR LEVEL CIRCUITS

D.I. Cheremisinov

Leading researcher of UIIP of NAS of Belarus, PhD of technical sciences, associate professor

L.D. Cheremisinova

Principal researcher of UIIP of NAS of Belarus, doctor of technical sciences, professor

Abstract. The paper considers the problem of generating graphical mapping for graphs representing logical networks. Known algorithms for drawing graphs are analyzed and an algorithm is proposed that is focused on interactive visualization of logical networks constructed as a result of CAD work. The task of visualizing a logical network differs significantly in terms of optimization criteria from the problems of placement and routing that arise at the stage of technical design of discrete devices.

Keywords: reverse engineering, graph visualization, graphical mapping of circuits, SPICE format.