

ИЗБРАННЫЕ МАТЕРИАЛЫ КОНФЕРЕНЦИИ “ПРОБЛЕМЫ
— РАЗРАБОТКИ ПЕРСПЕКТИВНЫХ МИКРО- И НАНОЭЛЕКТРОННЫХ СИСТЕМ” —
ПРОЕКТИРОВАНИЕ ИНТЕГРАЛЬНЫХ СХЕМ

УДК 681.32

ИЗВЛЕЧЕНИЕ СЕТИ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ
ИЗ КМОП-СХЕМЫ ТРАНЗИСТОРНОГО УРОВНЯ

© 2019 г. Д. И. Черемисинов¹, *, Л. Д. Черемисинова¹

¹Объединенный институт проблем информатики НАН Беларуси
ул. Сурганова, 6, Минск, 220012 Республика Беларусь

*E-mail: cher@newman.bas-net.by

Поступила в редакцию 20.11.2018 г.

После доработки 03.12.2018 г.

Принята к публикации 03.12.2018 г.

Рассматривается задача преобразования плоской КМОП-схемы из транзисторов в формате SPICE в иерархическую схему из КМОП-вентилей в том же формате. Задача возникает при верификации лейаута СБИС путем сравнения исходного описания для синтеза транзисторной схемы со схемой, восстановленной из топологии, а также при перепроектировании (reengineering) схем. Описывается метод распознавания подсхем, являющихся КМОП-вентильями, реализованный в виде программы на языке C++. Метод распознает подсхемы, описываемые одинаковыми логическими функциями, но не изоморфные на уровне транзисторов, как различные. Это обеспечивает изоморфность исходной и декомпилированной схем.

DOI: 10.1134/S0544126919030037

ВВЕДЕНИЕ

В процессе проектирования цифровые схемы представляются на разных уровнях абстракции. На транзисторном уровне схемы описываются в терминах транзисторов и их взаимосвязей. Уровень логических элементов использует логические вентили как стандартные блоки, чтобы описать схемы. На функциональном уровне описание схемы составляют функциональные блоки типа триггеров, сумматоров и т.д., каждый из которых может содержать большое количество логических элементов.

Средства распознавания высокоуровневых структур в сетях транзисторов представляют собой важный инструмент автоматизированного проектирования СБИС. Построение иерархического структурного описания по плоскому структурному описанию является существенной операцией в процессе проектирования и верификации интегральных схем. Автоматическое распознавание структуры более высокого уровня абстракции в описании схемы транзисторного уровня может быть использовано для многих задачах проектирования интегральных микросхем. Современные интегральные схемы могут содержать более сотни млн. транзисторов, и примерно такое же количество соединений между ними. Процесс подготовки производства современной СБИС стоит очень дорого: только изготовление набора фотошаблонов требует затрат в несколько миллионов долла-

ров. Поэтому перед изготовлением фотошаблонов обязательно выполняется верификация топологии СБИС (LVS – layout versus schematic verification).

Задача верификации топологии интегральных схем заключается в сравнении исходной транзисторной схемы со схемой, восстановленной из топологии. Верификация выполняется в два этапа. На первом этапе восстанавливается описание электрической схемы СБИС из описания топологии. Электрическая схема цифровой СБИС, изготавливаемой по КМОП-технологии, содержит только униполярные транзисторы, т.е. является плоским структурным описанием. Затем, во время второго этапа, электрическая схема проверяется с помощью статической верификации или с помощью моделирования. Результаты моделирования и верификации используются проектировщиком, чтобы изменить описание топологии для устранения найденных ошибок. Для выполнения процедуры анализа за приемлемое время требуется построить по электрической схеме иерархическое структурное описание на логическом уровне.

Первоначально автоматические экстракторы логических подсхем из транзисторного описания использовались главным образом для функциональной проверки результата физического проектирования относительно результата логического проектирования. Позже, экстракторы, извлекая структуры более высокого уровня, стали

использовать для того, чтобы ускорить моделирование транзисторной схемы. Извлечение иерархического описания на уровне логических подсхем является также основой логического перепроектирования интегральных схем.

По аналогии с программированием преобразование иерархической схемы электронного устройства в схему, состоящую исключительно из примитивных элементов, естественно назвать компиляцией. Обратный процесс, в результате которого из плоской схемы строится иерархическая транзисторная схема называется декомпиляцией. Декомпилятор [1] является одним из инструментов верификации лейаута или перепроектирования (reengineering) схем [2]. Перепроектирование, в отличие от оригинального проектирования, предполагает проектирование новой схемы для замены схемы существующего устройства.

Так же как и при декомпиляции программ, целью декомпиляции схемы является замена представления схемы на низком (транзисторном) уровне более высокоуровневым ее представлением (на уровне логических элементов). В отличие от программного процесса декомпиляция схем не является языковой трансформацией, хотя декомпилятор также использует определенный формат (формальный язык) представления схем. В дальнейшем предполагается, что языком, используемым декомпилятором схем, является формат SPICE (Simulation Program with Integrated Circuit Emphasis) для обмена электрическими схемами [3]. Формат SPICE позволяет описывать как схемы транзисторного уровня, так и иерархические.

В настоящей работе рассматривается задача декомпиляции плоской КМОП-схемы из транзисторов в формате SPICE в иерархическую схему из КМОП-вентилей для наиболее общего и сложного в теоретическом плане случая, когда библиотека исходных логических элементов не известна.

1. РАСПОЗНАВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ В СХЕМЕ ИЗ КМОП-ТРАНЗИСТОРОВ

Проблема извлечения логических подсхем из сети транзисторов изучается в течение длительного времени. Обзор результатов решения этой задачи можно найти в работах [4, 5].

Известные алгоритмы извлечения логических вентилей из сети транзисторов можно разделить на две группы. Алгоритмы первой группы применяются в том случае, когда искомые подсхемы из транзисторов заданы, т.е. известна библиотека логических элементов. Основной операцией при выделении логических схем в этих алгоритмах является операция сравнения частей анализируе-

мой схемы с образцами, которые формируются из описания заданных библиотечных элементов. Задача декомпиляции в этом случае сводится к задаче поиска в исходном графе, задающем транзисторную схему, подграфов, изоморфных заданному [6].

В этих алгоритмах в качестве модели плоской схемы и образца используется граф, в котором транзисторы являются вершинами, а провода, связывающие выводы транзисторов, — ребрами. Сложность сопоставления с образцом зависит от двух факторов. Первый из них связан с эффективностью алгоритма, который маркирует граф, задающий анализируемую сеть транзисторов. Если каждая вершина образца имеет уникальную метку, которая совпадает с меткой вхождения вершины в подсхему в анализируемом графе, то распознавание подсхемы — относительно простая задача (подсхема различима по структуре). К сожалению, графы образцов логических элементов на уровне транзисторов трудно различимы в графе анализируемой схемы, так как типы связи и типы транзисторов не помогают выделить подсхему в исходном графе.

Когда для некоторой транзисторной подструктуры найдено соответствие с одним из образцов, она заменяется соответствующим библиотечным элементом. Процессорное время, требуемое для сравнения, увеличивается очень быстро с ростом числа библиотечных элементов схемы. Кроме того декомпиляция успешна только в том случае, если схема состоит только из подсхем заданной библиотеки. Если же схема содержит также и подсхемы, не принадлежащие библиотеке, то они останутся не декомпилированными.

Библиотечная методика экстракции подсхем использует независимую от технологии технику распознавания, основанную на изоморфизме графов. Она может использоваться для перехода между любыми соседними уровнями абстракции от уровня транзисторов до уровня логических элементов и далее. Недостаток библиотечных методов экстракции логических подсхем состоит в том, что они требуют наличия целевой библиотеки элементов и в случаях неполных или неизвестных библиотек эти методы не работают.

Алгоритмы второй группы применяются в том случае, когда целевая библиотека транзисторных подсхем (элементов) не известна. Если ставится задача распознать библиотеку подсхем, то такая задача поставлена некорректно, так как любой фрагмент, содержащий целое число транзисторов, может считаться частью библиотечной подсхемы. Нужны дополнительные критерии, позволяющие выделять подсхемы и тем самым обеспечить решение задачи декомпиляции. В алгоритмах этого класса подсхемы, соответствующие логическим элементам, распознаются на основе набора

правил, позволяющих идентифицировать логические вентили через группы МОП транзисторов, связанных друг с другом через выводы стока или истока. Например, в работе [4], так же как и в настоящей статье, в качестве подсхем выделяются фрагменты транзисторной КМОП-схемы, для которых возможно определить логическую функцию.

Алгоритмы, идентифицирующие логические вентили группой МОП транзисторов, связанных друг с другом через выводы стока или истока, работают быстро, но они могут распознавать структуры ограниченного класса, как правило, это статические КМОП-схемы с комплементарными p - и n -структурами. Эти алгоритмы не распознают блоки с не комплементарными структурами, например, триггеры. Используемая методика экстракции подсхем существенно основана на комплементарности частей из NMOS- и PMOS-транзисторов в статических элементах логики на основе МОП-технологии. Такие алгоритмы экстракции работают быстрее, чем библиотечные методы сравнения. Кроме того, не требуется пользовательская библиотека. Основной недостаток метода состоит в том, что его приложение ограничено МОП-технологией. При переходе от уровня логических вентилях к более высокому уровню (триггеры, статическая оперативная память и т.д.), технологически зависимая методика распознавания неприменима.

Еще один рассматриваемый в литературе подход к решению задачи построения иерархического структурного описания, исходя из плоской схемы из транзисторов, формулируется как задача поиска часто встречающихся подграфов (frequent subgraph mining – FSM). Эта задача стала популярной областью исследований в последнее десятилетие, библиография по которой к настоящему времени насчитывает сотни публикаций (см. [7]). Литература на русском языке, посвященная этой проблеме практически отсутствует. За это время было предложено много алгоритмов поиска часто встречающихся подграфов. Однако трудоемкость задач FSM остается огромной в силу необходимости много раз решать задачу изоморфизма подграфов. Алгоритм наивного поиска часто встречающихся подграфов состоит из двух операций. Первая операция разыскивает все подграфы-кандидаты для данного графа G , а вторая подсчитывает частоту встречаемости каждого подграфа-кандидата. Из-за чрезвычайно большой трудоемкости второй операции известные практические алгоритмы FSM ограничиваются поиском подграфов небольшого размера (меньше 10 вершин) – графлетов. Для графов с миллионами вершин удается подсчитать оценку частоты встречаемости графлетов, размер которых ограничивается пятью вершинами.

2. ПРОГРАММА ДЕКОМПИЛЯЦИИ ПЛОСКОЙ ТРАНЗИСТОРНОЙ СХЕМЫ

В настоящей статье описывается метод декомпиляции КМОП-схемы из транзисторов, который реализован в виде программы на языке C++. Исходными данными для программы служит плоский нетлист КМОП-схемы в формате SPICE, имя головной схемы и имена цепей питания. Результатом является иерархическое SPICE-описание, в которое включены модели всех идентифицированных КМОП-вентилей. Для построения иерархического структурного описания в схеме выделяются наборы взаимосвязанных транзисторов в качестве отдельных компонентов. После замены подсхем из транзисторов элементами описание схемы становится двухуровневым (рис. 1, 2).

Декомпиляция выполняется следующей последовательностью шагов.

1. Анализ и преобразование исходного SPICE-описания плоской транзисторной схемы, в ходе которых строится графовая модель SPICE-описания (рис. 1) и иерархические хэш-таблицы для хранения синтаксических элементов анализируемой схемы.

2. Поиск передаточных элементов.

3. Разбиение множества оставшихся транзисторов на группы транзисторов, связанных по постоянному току (channel connected components – CCC).

4. Распознавание правильных CCC и нахождение реализующей их логической функции в виде алгебраической формулы, а также построение хэш-таблицы для хранения данных о распознанных экземплярах КМОП-вентилей, в которой хешем служит формула вентиля.

5. Классификация экземпляров вентилях, реализующих одну и ту же функцию, но описываемых неизоморфными графами вследствие взаимозаменяемости стока и истока и несимметричности входов.

6. Генерация моделей в иерархическом SPICE-описании для каждого класса КМОП-вентилей, описываемых изоморфными графами.

7. Генерация экземпляров вентилях в искомом иерархическом SPICE-описании.

8. Вставка в искомое иерархическое SPICE-описание всех оставшихся нераспознанными элементами исходного описания.

Все шаги предлагаемого алгоритма декомпиляции КМОП-схемы из транзисторов выполняются за линейное время от размерности исходных данных, поэтому программа имеет достаточное быстродействие, чтобы обрабатывать схемы из 100 тыс. транзисторов за несколько минут на персональной ЭВМ.

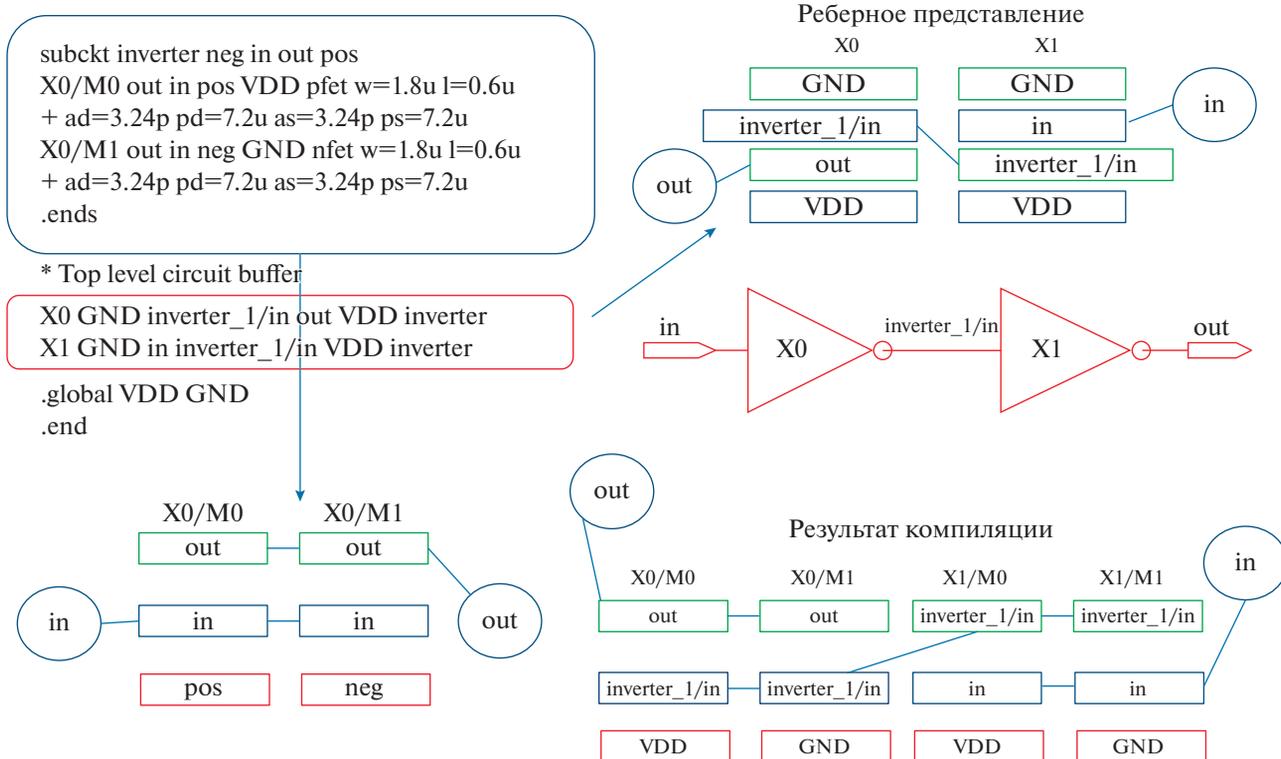


Рис. 1. Компиляция подсхем (circuit flattening) в формате SPICE.

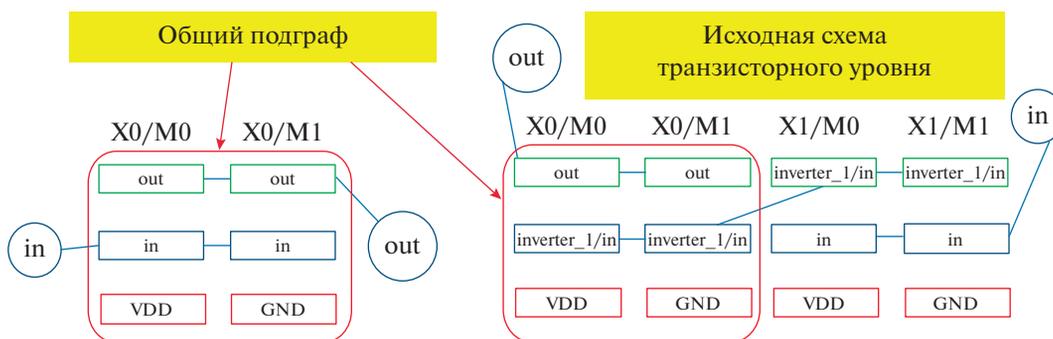


Рис. 2. Декомпиляция – поиск и извлечение подсхемы (subcircuit recognition – extraction).

Декомпилированную схему можно сравнить с исходной инструментами автоматизированного проектирования, решающими задачу сравнения схемы извлеченной из топологии, с принципиальной схемой устройства (Logic Versus Schematic check – LVS), например, используя Mentor Graphics Calibre nmLVS, Guardian LVS и другими. Для сравнения двух схем имеется и свободное программное обеспечение, например, netgen [8]. В большинстве таких программ обе схемы могут быть представлены в формате SPICE. Сравнение схем, выполняемое этими про-

граммами, заключается в решении задачи изоморфизма графов.

3. АНАЛИЗ И ПРЕОБРАЗОВАНИЕ SPICE-ОПИСАНИЯ ПЛОСКОЙ ТРАНЗИСТОРНОЙ СХЕМЫ

В качестве интерфейса между этапами верификации служит формат SPICE [3] для обмена электрическими схемами. В формате SPICE электрические схемы состоят из элементов, которые соединены друг с другом соединениями – цепями (рис. 1). Главной частью описания транзисторной

схемы является список транзисторов, в котором для каждого вывода транзистора указано имя цепи, соединяющей его с остальными частями схемы.

Например, общая форма описания униполярного транзистора в формате SPICE имеет следующий вид:

$$M \langle \text{name} \rangle \langle \text{nd} \rangle \langle \text{ng} \rangle \langle \text{ns} \rangle \langle \text{nb} \rangle \langle \text{model-name} \rangle [L = \text{value}] [W = \text{value}],$$

где сток d , затвор g , исток s и подложка b связаны с цепями nd , ng , ns и nb соответственно, а L длина и W ширина транзистора. Список транзисторов в SPICE-описании интерпретируется как задание бинарного отношения двудольного графа в виде списка пар. Например, $mp\ 2\ 1\ 3\ 3$ является сокращенной записью пар $(mp.d, 2)$, $(mp.g, 1)$, $(mp.s, 3)$, $(mp.b, 3)$, в которой имя транзистора вынесено, а имена его выводов опущены.

Электронные схемы могут быть представлены в виде помеченных графов. Простой моделью для электрических схем является гиперграф, в котором вершины соответствуют устройствам, а дуги — соединениям.

Для целей декомпиляции транзисторных схем более удобной моделью представления транзисторных схем является двудольный помеченный граф. Вершины двудольного графа разделены на два множества. Одно множество вершин составляют выводы транзисторов и выводы (порты) электрической схемы в целом, а другое множество — соединения между выводами, т.е. цепи. Цепь является оптимальным способом представления связи между произвольным количеством элементов. Представление электрической схемы двудольным графом в качестве структуры данных требует значительно меньшего объема памяти при программной реализации, чем представление в виде гиперграфа. Можно сказать, что по требованию к объему памяти сложность структуры гиперграфа оценивается как $O(n^2)$, в то время как сложность второй — как $O(n)$, где n — число элементов схемы. Кроме того, двудольный граф является естественной формальной моделью описания схемы, заданной в формате SPICE.

На рисунках этот двудольный граф компактнее представляется в виде соответствующего ему реберного графа (рис. 1, 2). Для графа G реберным называется граф $L(G)$, любая вершина которого представляет ребро графа G и две вершины графа $L(G)$ смежны тогда и только тогда, когда соответствующие им ребра смежны в G [9]. Пометками вершин, соответствующих элементам транзисторного уровня, являются названия выводов транзисторов. На рис. 1 и 2 в графе КМОП-схемы транзисторного уровня у транзисторов не показан вывод подложки, зеленым обозначен вывод “сток”, красным — “исток”, синим — “затвор”.

Задача анализатора формата SPICE заключается также в том, чтобы собрать всю информа-

цию, связанную с каждым синтаксическим элементом анализируемой схемы. Для решения этой задачи анализатор организует специальные хранилища данных, называемые таблицами идентификаторов. Таблица идентификаторов состоит из набора записей, каждая из которых соответствует одному элементу анализируемой схемы. Анализатор пополняет записи в таблице идентификаторов по мере анализа текста описания схемы. Поиск информации в таблице выполняется всякий раз, когда нужны сведения о том или ином элементе программы. Причем, поиск элемента в таблице выполняется существенно чаще, чем помещение в нее новых элементов. В анализаторе формата SPICE применяется обычно используемая реализация таблиц идентификаторов в виде хеш-таблиц.

Хеш-таблица содержит вектор (массив), элементы которого есть пары: ключ и значение (хеш-таблица с открытой адресацией) или списки пар (хеш-таблица с цепочками). Выполнение операции в хеш-таблице начинается с вычисления хеш-функции от ключа. При выполнении операции добавления или поиска используется элемент массива с номером, заданным хешем (хеш является индексом элемента в массиве). В реализованном анализаторе формата SPICE применяются хеш-таблицы с цепочками.

4. РАЗБИЕНИЕ МНОЖЕСТВА ТРАНЗИСТОРОВ НА ГРУППЫ СВЯЗАННЫХ ПО ПОСТОЯННОМУ ТОКУ

В цифровых КМОП-схемах МОП-транзисторы могут рассматриваться как ключи, управляемые входными напряжениями на затворе. Простейшая цифровая схема — это передаточный элемент, состоящий из одного МОП-транзистора, который обеспечивает управляемую передачу двоичного сигнала. Такой элемент является пассивным, поскольку он не усиливает входной (коммутируемый) сигнал. Усиление двоичных сигналов обеспечивает симметричная схема (КМОП-вентиль), в которой схемы соответственно из n -МОП и p -МОП транзисторов включены последовательно между электродами напряжения питания. КМОП-вентиль состоит из двух таких блоков, которые разделены выходной цепью (рис. 3).

Блок, состоящий из n -МОП транзисторов (pull-down network), размещен между выходом и цепью нулевого потенциала. Блок, содержащий p -МОП

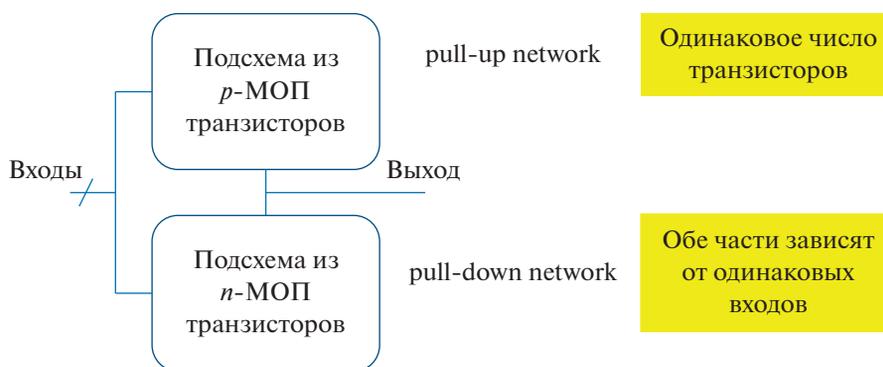


Рис. 3. КМОП-вентиль.

транзисторы (pull-up network), размещен между цепью напряжения питания и выходом. При этом затворы всех МОП-транзисторов связаны с входами схемы. Блоки обеспечивают связь выходного полюса схемы либо с источником питания VDD, либо с нулевым потенциалом GND (землей), так как проводимости блоков комплементарны. Логическая функция КМОП-вентиль определяется отрицанием функции проводимости транзисторов n -МОП-блока (или функции проводимости p -МОП-блока при инвертировании входных переменных) [11].

Группой транзисторов, соединенных по постоянному току (channel-connected component – CCC) [11], является произвольная схема из МОП-транзисторов с тремя типами внешних соединений:

- входы схемы подаются только на затворы транзисторов группы;
- выходы группы транзисторов подаются только на затворы транзисторов других групп;
- имеются соединения с цепями VDD и GND.

Множество групп транзисторов, соединенных по постоянному току, представляет собой разбиение множества транзисторов схемы. КМОП-вентиль представляет собой группу транзисторов, соединенных по постоянному току, обратное верно не всегда.

В схеме с элементами из МОП-транзисторов группа транзисторов, соединенных по постоянному току, соответствует компоненте связности графа схемы, в которой удалены выводы затворов (и соответствующие цепи), а также и цепи питания, а выводы стока и истока каждого транзистора соединены ребром. Алгоритм поиска компонента связности в определенном таким образом графе построен на основе алгоритма “поиск сначала в глубину” (DFS) [12]. Стоит отметить, что для поиска групп транзисторов, соединенных по постоянному току, необходимо знать, какие цепи в исходной схеме задают потенциалы VDD и GND.

5. РАСПОЗНАВАНИЕ ПРАВИЛЬНЫХ ССС ГРУПП И НАХОЖДЕНИЕ РЕАЛИЗУЮЩЕЙ ИХ ЛОГИЧЕСКОЙ ФУНКЦИИ

Как уже отмечалось выше, не всякая группа CCC транзисторов, связанных по постоянному току, является КМОП-вентилем. Необходимыми условиями принадлежности группы к классу КМОП-вентилей являются следующие признаки, которые должна иметь правильная группа:

- 1) выход является единственной цепью, соединяющей p - и n -части группы;
- 2) группа содержит обе цепи питания VDD и GND;
- 3) все пути из цепи выхода доходят до цепей питания (и наоборот);
- 4) внутренние цепи группы не содержат выводов транзисторов, ей не принадлежащих.

Будем считать управление проводимостью каждого транзистора из группы независимым входом. Тогда число переменных логической функции, задающей проводимость пути от цепи выхода до цепи питания, равно числу транзисторов в блоке pull-down network (или pull-up network). Сама функция проводимости представляется в виде дизъюнктивной нормальной формы (ДНФ), которая задает список путей от цепи выхода до цепи питания (рис. 4). Каждый путь представляется конъюнкцией входных переменных, управляющих проводимостью транзисторов (цепь, соединенная с затвором), входящих в этот путь.

Если функция проводимости блока pull-down network комплементарна (является отрицанием) функции проводимости блока pull-up network, то правильная группа транзисторов, соединенных по постоянному току, представляет собой КМОП-вентиль. Для классификации КМОП-вентилей схемы удобна не ДНФ, а скобочная форма представления функции проводимости блока pull-down network. Формула задания функции логического элемента строится путем алгебраической факто-

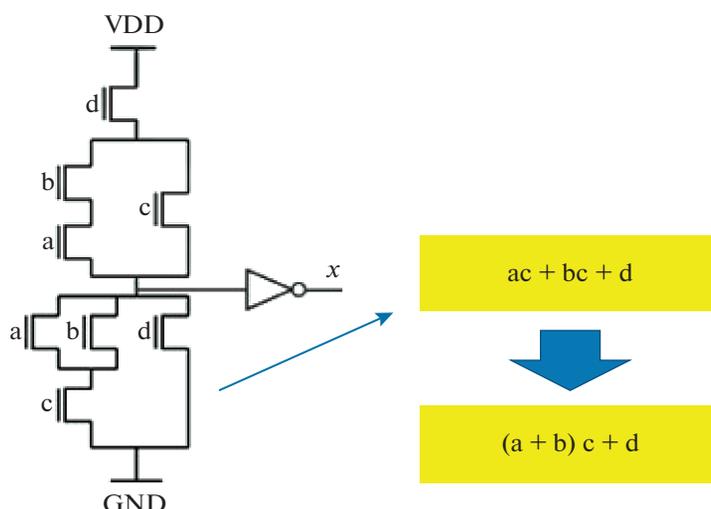


Рис. 4. Функция проводимости КМОП-вентиля в форме ДНФ и факторизованной ДНФ.

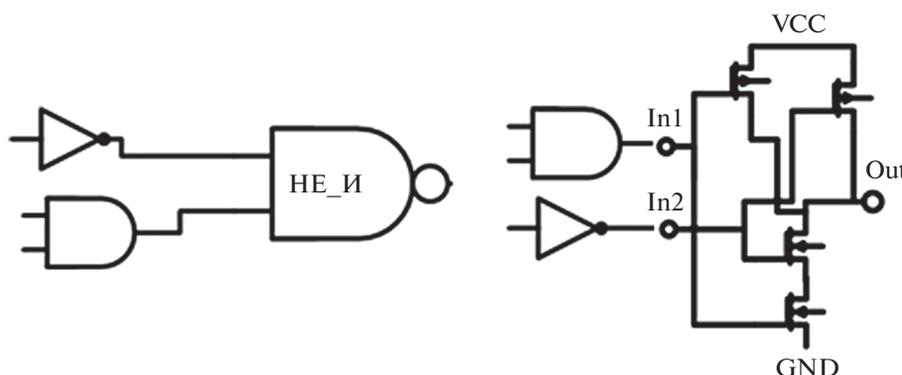


Рис. 5. Скомпилированные схемы не изоморфны после перестановки входов вентиля И-НЕ.

ризации ДНФ [13] булевой функции проводимости, результатом которой является алгебраическое представление функции элемента в скобочной форме (рис. 4).

6. КЛАССИФИКАЦИИ ЭКЗЕМПЛЯРОВ ВЕНТИЛЕЙ, РЕАЛИЗУЮЩИХ ОДНУ И ТУ ЖЕ ФУНКЦИЮ

Правильные группы транзисторов, соединенных по постоянному току и являющихся КМОП-вентилем, могут быть разбиты на классы по виду формул, представляющих их логические функции. Для классификации таких правильных групп не требуется решать задачу изоморфизма графов. Однако при таком определении класса КМОП-вентилей (порождающего одну библиотечную схему) в один класс попадают все экземпляры каждого вентиля, реализующие одну логическую функцию, но имеющие разные топологические структуры. Поэтому при разбиении правильных групп

ССС транзисторов на классы следует учитывать не только реализуемые ими функции, но и следующие особенности топологической реализации вентиля, которые требуют различать некоторые группы.

1) Отсутствие симметрии входов у КМОП-вентилей, реализующих симметрические функции

Булева функция называется симметрической, если ее значение не зависит от перестановки ее аргументов. Однако, в КМОП-вентилем, реализующих симметрические функции, перестановка входов делает схему не изоморфной исходной. К примеру, в правой схеме на рис. 5 инвертор подключен к нижнему транзистору pull-down network, а в левой – к верхнему. Но логически обе приведенные схемы эквивалентны. Так как входы КМОП-вентилей, реализующих симметрические функции, несимметричны, то, например, для вентиля И-НЕ на два входа возможны две неизоморфные схемы (рис. 6).

<pre>.SUBCKT G8_0 A B Y # (A AND B) M1 2 A 0 0 MODN M2 Y B 2 0 MODN M3 Y B VCC VCC MODP M4 Y A VCC VCC MODP .ENDS</pre>	<pre>.SUBCKT G8_1 A B Y # (A AND B) M1 Y A 1 0 MODN M2 1 B 0 0 MODN M3 Y A VCC VCC MODP M4 Y B VCC VCC MODP .ENDS</pre>
-------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------

Рис. 6. Задание схем вентиля И-НЕ, неизоморфных вследствие несимметричности его входов.

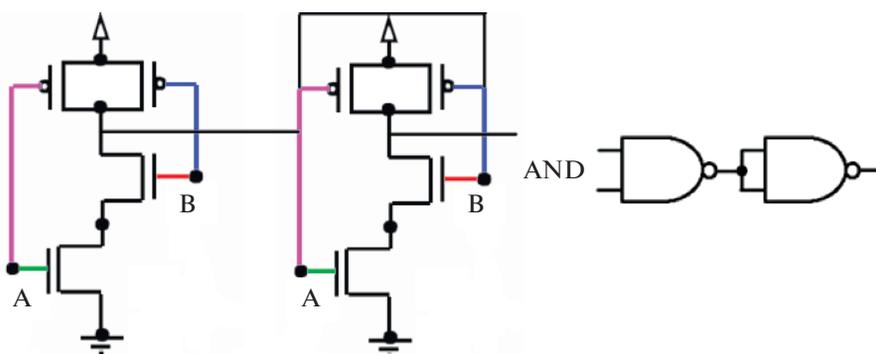


Рис. 7. Логический элемент с закороченными входами.

2) Закороченные входы

Соответствие входов блока pull-down network входам pull-up network устанавливается цепями, соединяющими выводы затворов транзисторов (левая схема на рис. 7). Предположение о независимости управления проводимостью каждого транзистора в блоке pull-down network (pull-up network) верно не всегда. В левой схеме на рис. 7 невозможно найти парный транзистор в pull-up network, потому что входы элемента закорочены (shorted inputs).

3) Взаимозаменяемость стока и истока

Работа МОП-транзистора основана на изменении концентрации свободных носителей заряда в канале под влиянием электрического поля, создаваемого напряжением, приложенным между затвором и истоком. Для этих приборов характерна взаимозаменяемость стока и истока, т.е. ток в канале может протекать в обоих направлениях в зависимости от полярности напряжения, приложенного к каналу [14]. Взаимозаменяемость стока и истока МОП-транзистора приводит к тому, что существуют разные подсхемы, реализующие одну и ту же логическую функцию. Например, для КМОП-инвертора существует четыре варианта подсхем. Если в декомпилированной

схеме все такие варианты подсхемы логического элемента представить одним вариантом, то декомпилированная и оригинальная схема будут не изоморфны.

7. ПРОВЕРКА ИЗОМОРФИЗМА ПОДСХЕМ

Проверка изоморфности двух подсхем, реализующих одну и ту же логическую функцию, может быть сведена к задаче установления изоморфизма графов, в которые преобразуются сравниваемые подсхемы. Как уже говорилось выше, подсхемы в формате SPICE эквивалентны помеченным неориентированным двудольным графам.

Помеченные графы

$$G = (V, E, f) \text{ и } H = (W, F, g),$$

заданные множествами V и W вершин, множествами E и F ребер и пометками вершин

$$f : V \rightarrow P \text{ и } g : W \rightarrow P,$$

являются изоморфными, если между множествами их вершин существует взаимно однозначное соответствие, сохраняющее отношение смежности [15]. Другими словами, графы изоморфны, если существует такая биекция $\phi: V \leftrightarrow W$, что для любых вершин $v_i, v_j \in V$ их образы $\phi(v_i)$ и $\phi(v_j)$ смежны в H , если и только если они смежны в

G , и если $w = \varphi(v)$, то $f(v) = g(w)$. Тривиальный способ установления изоморфизма заключается в перестановке вершин одного из графов и проверке упомянутых условий, однако в этом случае для графа, имеющего n вершин, необходимо перебрать равно $n!$ вариантов, что неприемлемо в большинстве случаев.

Необходимым (но не достаточным) условием изоморфизма двух графов является равенство числовых характеристик графов, называемых инвариантами, которые совпадают у изоморфных графов. При установлении изоморфизма рассматриваются инварианты вершин, которыми являются их степени (полустепени), число вершин, отстоящих от данной на определенном расстоянии.

При массовом решении задачи об установлении изоморфизма двух графов удобно пользоваться следующей редукцией. В каждом классе попарно изоморфных графов выбирается один граф, называемый каноническим видом любого графа из этого класса. После этого вопрос об изоморфизме двух произвольных графов сводится к построению и сличению их канонических видов. Каноническое представление графа находится путем упорядочения вершин графа в соответствии с инвариантами его вершин, не зависящими от исходной нумерации вершин. В теории графов проблема построения канонической формы заданного графа G называется его канонизацией [15]. Канонизация графа подсхемы и построение по нему представления в формате SPICE, позволили свести проверку изоморфизма двух подсхем к проверке совпадения текстов описания подсхем. Однако проблема канонизации графа в вычислительном плане так же трудна, как и проблема изоморфизма графов.

Канонической формой графа в нашем случае является помеченный граф, поэтому задача канонизации сводится к перемаркировке его вершин, основная идея этого метода предложена в работе [16].

Понятия, используемые для постановки задачи поиска канонической маркировки, взяты из теории графов и теории групп. Упорядоченным разбиением множества V является упорядоченная последовательность непустых непересекающихся подмножеств V_i , объединением которых является V . Подмножества называются блоками разбиения. Разбиение множества вершин графа можно также рассматривать как раскраску вершин, которая присваивает один и тот же цвет двум вершинам тогда и только тогда, когда они принадлежат одному и тому же блоку. Разбиения, в которых каждый блок содержит один элемент, называются дискретными. Для двух разбиений множества V первое лучше ($<$) чем второе, если первое может быть образовано путем расщепления одного или нескольких блоков второго разбиения. От-

ношение $<$ является отношением частичного порядка, его минимальными элементами являются дискретные разбиения.

В качестве исходного разбиения для графа принимается разбиение множества его вершин по степеням и меткам (каждый блок разбиения содержит вершины с одинаковыми пометками и степенями). Каждой вершине графа приписывается вектор, компоненты которого соответствуют блокам разбиения, а значением компоненты является число вершин соответствующего блока, смежных с данной вершиной. Блоки графа упорядочиваются по не убыванию или по не возрастанию степеней его вершин.

Алгоритм канонической маркировки выполняет поиск вначале в глубину дерева, где каждый узел представляет собой разбиение множества вершин. Каждый дочерний класс формируется отделением в новый блок некоторой вершины (индивидуализацией вершины) из родительского класса, если ей приписан вектор, отличный от векторов других вершин блока. Индивидуализация разбивает родительский блок, образуя для этой вершины собственный блок. Затем результирующее разбиение минимизируется по отношению $<$. В конце концов, этот процесс приведет к дискретным разбиениям, которые являются листьями дерева поиска. Каждый лист соответствует возможной канонической маркировке. Лист с минимальным разбиением возвращается алгоритмом как каноническая маркировка [16].

Для упрощения задачи канонизации представляющий подсхему граф дополняется ребрами, связывающими все выводы каждого транзистора. Вместо задания канонизированного графа в формате SPICE для этого графа вычисляется хеш. Хеш подсхемы представляет собой преобразование канонической маркировки вектора степеней вершин графа как последовательности чисел в (выходную) битовую строку, имеющую длину слова. Графы изоморфны, если хеши их канонизированных форм совпадают.

8. ПЕРЕДАТОЧНЫЕ ЭЛЕМЕНТЫ – PASS GATES

Логические элементы могут быть построены на базе передаточных логических элементов (transmission gates) [17]. Передаточный элемент (ключ) состоит из пары транзисторов n -МОП и p -МОП-типа, соединенных параллельно выводами стока и истока соответственно. Соединенные выводы обозначаются через A и B , поскольку передача сигнала в таком логическом вентилю может идти в двух направлениях, ни одно из которых не является предпочтительным. Вывод подложки p -МОП-транзистора соединен с положительным потенциалом питания, а вывод подложки n -МОП-тран-

зистора соединен с отрицательным потенциалом питания. Выводы затворов обозначаются EN и 'EN. Если EN равен 0, а 'EN равен 1, то оба транзистора выключены. При этом вывод А не имеет связи с выводом В. В противном случае, передаточный логический вентиль включен, а А и В имеют одинаковый потенциал.

Передаточные элементы распознаются по n -МОП и p -МОП-транзисторам, соединенным параллельно выводами стока и истока. Поиск таких транзисторов производится путем попарного сравнения и имеет временную сложность, пропорциональную числу сочетаний из n (числу транзисторов в схеме) по 2 (биномиальному коэффициенту), т.е. имеет экспоненциальную сложность. Поиск с помощью хеш-таблицы транзисторов, в которой хешем служат имена цепей истока и стока, позволил получить линейную сложность поиска параллельно соединенных транзисторов.

9. ИСПЫТАНИЯ ПРОГРАММЫ

С целью испытания разработанной программы выполнялись две группы экспериментов. Задачей первых экспериментов являлась проверка правильности распознавания заданной библиотеки вентиляей. В экспериментах второй группы проверялась правильность распознавания вентиляей в практических схемах.

Для первого эксперимента использовались схемы, полученные автоматическим синтезом транзисторных схем по заданным функциональным описаниям на языке VHDL с помощью САПР. В этом случае технологическая библиотека считалась известной. В эксперименте использовалось около десятка схем сложностью от сотни до 10 тыс. транзисторов. Схемы представляли собой цифровые устройства, как комбинационные, так и последовательностные. Элементы памяти в технологических библиотеках целиком состояли из логических вентиляей. Наблюдалось стопроцентное покрытие схемы транзисторного уровня логическими вентиляями. Верификация правильности декомпиляции осуществлялась программой netgen [8]. Часть декомпилированных схем была верифицирована также с помощью Mentor Graphics Calibre nmLVS. Во всех случаях декомпилированная схема успешно проходила проверку LVS (layout versus schematic) топологии СБИС.

Во втором эксперименте использовалось тоже около десятка схем, извлеченных из лейаута. Для части устройств иерархическая модель схемы в формате SPICE была известна. Для других устройств никакой дополнительной информации кроме схемы на уровне транзисторов не имелось. Устройства содержали от 5 до 50 тыс. транзисторов. В некоторых схемах кроме МОП-транзисторов имелись и другие примитивные элементы (би-

полярные транзисторы и RC элементы). Здесь наблюдалось покрытие схемы транзисторного уровня логическими вентиляями на уровне 60–70 процентов. В устройствах, для которых была известна иерархическая модель, подсхемы этой модели представляли собой смесь логических вентиляей и транзисторов. Анализ декомпилированных схем показал, что все логические вентиляей находились. Верификация правильности декомпиляции осуществлялась программой netgen. Декомпилированные схемы также успешно проходили проверку LVS.

На компьютере с четырехядерным процессором Intel i5-4460 3.20GHz с оперативной памятью 16.0 ГБ декомпиляция схемы с 50 тыс. транзисторов выполнялась за время меньше минуты.

ЗАКЛЮЧЕНИЕ

Структурный анализ цифровых схем в прошлом широко исследовался [4]. Для распознавания подсхем в литературе описаны два класса подходов: библиотечные подходы и алгоритмические подходы. Подходы, основанные на библиотеке, используют схемы из библиотеки как шаблоны для распознавания подсхем, и их способность распознавания ограничена теми подсхемами, которые содержатся в библиотеке. В алгоритмических подходах фрагмент схемы распознается как подсхема, если имеется возможность вычислить его логическую функцию. Предлагаемый в работе метод, в отличие от известных в литературе, позволяет также распознавать как различные подсхемы, задаваемые одинаковыми логическими функциями, но не изоморфные на уровне транзисторов. Это обеспечивает эквивалентность исходной и декомпилированной схем, которую можно верифицировать проверкой LVS топологии СБИС.

СПИСОК ЛИТЕРАТУРЫ

1. Logic Gate Recognition in Guardian LVS – Silvaco // URL: https://www.silvaco.com/content/appNotes/ic-cad/2-003_LogicGates.pdf. Date of access: 4.1.2018.
2. Hunt V.D. Reengineering: Leveraging the Power of Integrated Product Development. N.Y.: John Wiley and Sons, Inc. 1993. 283 p.
3. Baker R.J. CMOS Circuit Design, Layout, and Simulation, Third Edition – Wiley-IEEE Press. 2010. 1214 p.
4. Yang L., Shi C.-J.R. FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits // Integration, the VLSI J. 2006. V. 39. № 4. P. 311–339.
5. Zhang N., Wunsch D.C., Harary F. The subcircuit extraction problem // Proc. IEEE Intern. Behavioral Modeling and Simulation Workshop. 2005. V. 33. № 3. P. 22–25.
6. Krasilnikova L.V., Pottosin Yu.V. Partition of a transistor circuit into library modules from a given library // Proc.

- of the Second International Conference on Computer-Aided Design of Discrete Devices (CAD DD'97), Minsk: Republic of Belarus, November 12–14, 1997. Minsk: National Academy of Sciences of Belarus Institute of Engineering Cybernetics. 1997. V. 1. P. 94–97.
7. Черемисинов Д.И., Черемисинова Л.Д. Поиск часто встречающихся подграфов // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня: сб. материалов IV Междунар. науч.-практ. конф. (Республика Беларусь, Минск, 3–4 мая 2018 г.) / Редкол.: М.П. Батура [и др.]. Минск, БГУИР. 2018. С. 171–176.
 8. Netgen version 1.5 Tutorial // URL: <http://opencircuitdesign.com/netgen/tutorial/tutorial.html/>. Date of access: 16.02.2018.
 9. Харари Ф. Теория графов. М.: Мир. 1973. 304 с. (Изд. 3, М.: Ком Книга. 2006, 296 с.)
 10. Ракитин В.В. Интегральные схемы на комплементарных МОП-транзисторах: Учебное пособие. М.: МФТИ. 2007. 308 с.
 11. Bushnell M., Agrawal Vishwani. Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI. Springer Science & Business Media. 2006. 690 p.
 12. Левитин А.В. Алгоритмы. Введение в разработку и анализ. М.: ООО “И.Д. Вильямс”. 2006. 576 с.
 13. Черемисинова Л.Д. Синтез и оптимизация комбинационных структур СБИС. Минск: ОИПИ НАН Беларуси. 2005. 235 с.
 14. Белоус А.И., Емельянов В.А., Турцевич А.С. Основы схемотехники микроэлектронных устройств. М.: Техносфера. 2012. 472 с.
 15. Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д. Логические основы проектирования дискретных устройств. М.: Физматлит. 2007. 589 с.
 16. Junttila T., Kaski P. Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs // Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX). 2007. P. 135–149.
 17. Рабаи Ж.М., Чандраксан А., Борижович Н. Цифровые интегральные схемы. Методология проектирования, 2-е издание: Пер с англ. М.: ООО “И.Д. Вильямс”. 2007. 912 с.