



УДК 681.32
<https://doi.org/10.37661/1816-0301-2022-19-3-25-39>

Оригинальная статья
Original Paper

Канонизация графов при декомпиляции транзисторных схем

Д. И. Черемисинов, Л. Д. Черемисинова[✉]

*Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
✉E-mail: cld@newman.bas-net.by*

Аннотация

Цели. Разрабатываются средства распознавания (экстракции) высокоуровневой структуры в транзисторной схеме, которые позволяют получить представление на уровне логических элементов, эквивалентное исходному плоскому описанию на транзисторном уровне. Получение такого представления существенно снижает время выполнения проверки топологии и служит основой для перепроектирования интегральных схем и обратного инжиниринга для обнаружения несанкционированных вложений.

Методы. Предлагаются графовые методы и программные средства распознавания топологически эквивалентных транзисторных схем, позволяющие разбить множество подсхем на классы. Задача сводится к проверке изоморфизма помеченных графов, задающих схемы на транзисторном уровне, путем их канонизации и сравнения канонических маркировок. Исходная плоская и полученная двухуровневая транзисторные схемы представляются в формате SPICE.

Результаты. Предложенные методы реализованы на языке C++ как часть программы декомпиляции транзисторных схем для случая, когда искомая библиотека логических элементов заранее неизвестна. Предложенный метод канонизации помеченных графов используется при распознавании топологически эквивалентных подсхем среди функционально эквивалентных подсхем, реализующих логические элементы; разбиении множества подсхем, не распознанных как логические элементы, на классы топологически эквивалентных; верификации результатов экстракции иерархической схемы на транзисторно-логическом уровне относительно плоской схемы на транзисторном уровне.

Заключение. Программа декомпиляции была протестирована на практических схемах транзисторного уровня. Показано, что она имеет достаточное быстродействие, чтобы обрабатывать схемы более чем со 100 тыс. транзисторов за несколько минут на ПЭВМ.

Ключевые слова: экстракция транзисторных подсхем, КМОП-схемы, верификация, распознавание логических вентилях, изоморфизм графов, формат SPICE

Для цитирования. Черемисинов, Д. И. Канонизация графов при декомпиляции транзисторных схем / Д. И. Черемисинов, Л. Д. Черемисинова // Информатика. – 2022. – Т. 19, № 3. – С. 25–39.
<https://doi.org/10.37661/1816-0301-2022-19-3-25-39>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 07.06.2022
Подписана в печать | Accepted 15.08.2022
Опубликована | Published 29.09.2022

Canonization of graphs during transistor circuits decompilation

Dmitry I. Cheremisinov, Ljudmila D. Cheremisinova[✉]

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus
✉E-mail: cld@newman.bas-net.by*

Abstract

Objectives. The objective of the work is to develop the means for recognition (extraction) of high-level structures in circuits on transistor level. This allows to obtain a representation on logical level, equivalent to original flat description on transistor level. Obtaining such a representation significantly reduces the time to perform VLSI topology check, but also provides the basis for reengineering of integrated circuits and reverse engineering for detecting unauthorized attachments.

Methods. Graph based methods and software tools are proposed for recognizing topologically equivalent transistor circuits, which makes it possible to divide the set of subcircuits into topologically equivalent classes. The problem is reduced to checking the isomorphism of labeled graphs defining circuits on transistor level by canonizing them and comparing canonical labeling. The original flat and resulting two-level transistor circuits are presented in SPICE format.

Results. The proposed methods are implemented in C++ as a part of a transistor circuit decompilation program for the case without predetermined cell library. The proposed method of canonization of labeled graphs is used: to recognize topologically equivalent subcircuits among functionally equivalent subcircuits that implement logical elements; to split the set of subcircuits not recognized as logical elements into classes of topologically equivalent ones; to verify the results of extraction of the hierarchical circuit at the transistor-logic level relative to the flat circuit at the transistor level.

Conclusion. The decompilation program has been tested on practical transistor-level circuits. Experiments indicate that this tool is fast enough to process the circuits with more than one hundred thousand transistors in a few minutes on a personal computer.

Keywords: transistor subcircuit extraction, CMOS circuits, VLSI layout verification, logical gates recognition, graph isomorphism, SPICE format

For citation. Cheremisinov D. I., Cheremisinova L. D. *Canonization of graphs during transistor circuits decompilation*. Informatika [Informatics], 2022, vol. 19, no. 3, pp. 25–39 (In Russ.).
<https://doi.org/10.37661/1816-0301-2022-19-3-25-39>

Conflict of interest. The authors declare of no conflict of interest.

Введение. Все более ответственным этапом проектирования современных СБИС, которые содержат до миллиарда транзисторов, становится их тестирование, позволяющее установить, соответствует ли описание проекта на уровне транзисторов спецификации на проектирование [1]. Важным инструментом, позволяющим снизить трудоемкость тестирования электрических схем, служат средства распознавания (экстракции) высокоуровневых структур в схемах на транзисторном уровне. Целью данного преобразования является замена представления схемы на транзисторном (низком) уровне функционально эквивалентным представлением на уровне логических элементов (более высокоуровневым) [2, 3]. Для построения такого описания нужно выделить в схеме наборы взаимосвязанных транзисторов в качестве отдельных компонентов, т. е. найти подсхемы, представляющие собой логические элементы. После замены подсхем из транзисторов логическими элементами описание схемы на уровне транзисторов становится двухуровневым. По нему можно построить логическую схему, функционально эквивалентную исходной транзисторной схеме, и тем самым облегчить возможность менее затратного логического моделирования всего описания СБИС или части его, оставляя трудоемкое схемотехническое моделирование только для критически важных участков СБИС.

Операция, в результате которой из плоской транзисторной схемы строится иерархическая транзисторная, инверсна задаче проектирования в смысле направления процесса преобразова-

ния от спецификации до реализации интегральной схемой. Ее цель заключается в восстановлении спецификации устройства на логическом уровне путем анализа плоского описания реализующей его транзисторной схемы. По аналогии с программированием этот процесс можно назвать *декомпиляцией плоской транзисторной схемы* [4].

Декомпиляция транзисторной схемы является мощным инструментом верификации топологии (позволяет существенно снизить время ее выполнения) [5–7], а также основой логического перепроектирования (англ. reengineering) интегральных схем [8]. Перепроектирование, в отличие от оригинального проектирования, предполагает проектирование схемы на новой элементной основе для замены схемы существующего устройства. В последнее время растет количество публикаций, посвященных применению средств декомпиляции транзисторных схем для обратного инжиниринга СБИС (англ. hardware reverse engineering) с целью обнаружения несанкционированных вложений (закладок, троянов) при проектировании и производстве СБИС [9–11].

Структурный анализ цифровых схем путем их декомпиляции широко исследовался в прошлом, обзор результатов решения этой задачи можно найти в работах [2, 3]. При декомпиляции топологии СБИС может быть известна библиотека логических элементов, использованная при компиляции топологии СБИС, т. е. набор подсхем логических элементов на уровне транзисторов. В данном случае задача декомпиляции сводится к поиску в транзисторной схеме этих подсхем и покрытию ими. В более общем (и более сложном в теоретическом плане) случае библиотека логических элементов неизвестна. Тогда задача сводится к распознаванию транзисторных подсхем, реализующих логические элементы, и формированию библиотеки логических элементов, встречающихся в схеме. В обоих случаях в результате декомпиляции плоское описание транзисторной схемы преобразуется в двухуровневое описание путем выделения уровня подсхем, представляющих логические элементы.

Одной из проблем при формировании двухуровневого описания анализируемой схемы является задача выделения и распознавания не только функционально, но и топологически эквивалентных транзисторных подсхем. Это позволяет обеспечить эквивалентность исходной и декомпилированной схем, которую можно верифицировать известными средствами LVS (Logic Versus Schematic check) проверки топологии СБИС. Для произвольных схем задача установления их топологической эквивалентности является NP-трудной, однако для транзисторных схем удается найти существенно более простое решение за счет учета особенностей графа, соответствующего структуре транзисторной схемы.

В настоящей работе рассматриваются задачи и графовые методы их решения, возникающие при необходимости:

- распознавания топологически эквивалентных подсхем среди функционально эквивалентных подсхем, реализующих логические элементы;
- разбиения множества подсхем, не распознанных как логические элементы, на классы топологически эквивалентных;
- верификации результатов экстракции иерархической схемы на транзисторно-логическом уровне относительно плоской схемы на транзисторном уровне.

Постановка задачи и метод ее решения. Исходным объектом при декомпиляции является плоское (одноуровневое) описание схемы исследуемой СБИС. Для этой схемы может иметься заданная априори библиотека логических элементов, использованная при ее проектировании. В таком случае обратный инжиниринг (с уровня плоского описания схемы транзисторов) состоит в решении двух ключевых проблем. Первая заключается в выделении правильных транзисторных подсхем, которые выглядят как логические элементы, и иерархии их соединений, вторая – в распознавании реализующих логические элементы подсхем путем сравнения их со схемами библиотечных элементов. Распознавание логических элементов осуществляется либо на функциональном уровне через сравнение реализуемых подсхемами функций, либо на структурном уровне через анализ изоморфизма графов соединений транзисторов.

Библиотека логических элементов может быть и неизвестной, тогда задача заключается прежде всего в построении библиотеки встречающихся в анализируемой схеме транзисторных подсхем, реализующих логические элементы или псевдоэлементы, для которых не распознаны

реализуемые ими логические функции. Формирование библиотеки вентилях происходит во время работы программы декомпиляции. Результатом является иерархическое SPICE-описание, в которое включены модели всех идентифицированных КМОП-вентилей. Эти модели составляют и извлеченную библиотеку вентилях, и второй уровень формируемого иерархического описания.

Очевидно, что оба встречающихся при декомпиляции случая требуют решения одних и тех же задач, вместе с тем сначала нужно выделить правильные подсхемы из транзисторов и попытаться определить реализуемые ими функции. Разница заключается только в выполнении заключительного этапа построения иерархического описания. В первом случае эталоны логических элементов заданы и есть с чем сравнивать анализируемые подсхемы транзисторов для их классификации, во втором – эталоны формируются на основе сравнения и классификации выделяемых подсхем.

Целью декомпиляции транзисторной схемы является распознавание в ней правильных подсхем, реализующих логические элементы. Если же это не получилось, то следует разбить схемы на достаточно большие подсхемы, которые выглядят как логические элементы. В схеме из МОП-транзисторов правильные подсхемы находятся среди групп транзисторов, связанных по постоянному току. Для их нахождения используется структурный подход к распознаванию подсхем логических вентилях, имеющий линейную сложность.

Структурный подход к декомпиляции транзисторных схем позволяет разбить транзисторную схему на непересекающиеся подсхемы, представляющие группы транзисторов, связанных по постоянному току. Такой группой является произвольная схема из МОП-транзисторов с тремя типами внешних соединений, согласно которой входы группы подаются только на затворы транзисторов группы, выходы – только на затворы транзисторов других групп, а также имеются связи транзисторов группы с шинами питания Vdd и земли Gnd. Например, на схеме, приведенной на рис. 1, а, показаны две подобные группы, выделенные пунктирной линией. Среди найденных групп транзисторов так выделяют правильные подсхемы, представляющие собой статические КМОП-вентили, и определяют реализуемые ими функции. Каждая из подсхем, не распознанных как КМОП-вентиль, объявляется *нераспознанным псевдоэлементом*, функциональное описание которого неизвестно.

В работе [12] рассматриваются задачи и графовые методы их решения, которые возникают при структурном поиске в транзисторной схеме групп связанных по постоянному току транзисторов и при распознавании подсхем, представляющих КМОП-вентили, и передаточных элементов – пар транзисторов *n*-МОП- и *p*-МОП-типов, соединенных параллельно выводами стока и истока. Предлагается также метод построения канонического представления функций, реализуемых распознанными вентилями, который позволяет разбить множество найденных КМОП-вентилей на подмножества функционально эквивалентных. Множество псевдоэлементов тоже разбивается на подмножества (для дальнейшего анализа), каждое из которых содержит элементы с одинаковыми числами транзисторов и их связей. В дальнейшем каждое такое подмножество необходимо разбить на группы топологически эквивалентных псевдоэлементов.

Таким образом, в результате применения структурного метода декомпиляции транзисторной схемы формируется множество связанных подсхем, встречающихся в транзисторной схеме. Затем это множество разбивается на подмножества функционально эквивалентных вентилях и псевдоэлементов с одинаковыми характеристиками. Каноническое описание представителя каждого класса эквивалентных подсхем вентилях порождает библиотечный элемент, который заменяет все подсхемы своего класса в плоском описании декомпилируемой схемы, продуцируя двухуровневое описание. Все шаги предлагаемого в работе [12] алгоритма декомпиляции схемы из транзисторов выполняются за линейное время исходя из размерности исходных данных.

Некоторые особенности топологической реализации схем на транзисторном уровне объясняются существованием подсхем, реализующих одну и ту же логическую функцию, но различающихся в топологическом плане. В простейшем случае существование топологически разных подсхем может быть обусловлено взаимозаменяемостью выводов стока и истока МОП-транзисторов. К примеру, есть четыре варианта топологически разных подсхем для КМОП-инвертора. Для более сложных вентилях причина может заключаться и в разной струк-

турной организации функционально эквивалентных схем, которая задает порядок на множестве переменных для логической функции схемы с симметричными входами. Эта задача частично решается также на этапе структурного распознавания. Если в декомпилированной схеме топологически различающиеся варианты подсхемы одного и того же логического элемента заменить одной из таких подсхем, то декомпилированная и исходная схемы будут функционально эквивалентными, но топологически не эквивалентными. Топологический аспект эквивалентности описаний требует разбивать классы функционально эквивалентных КМОП-вентилей на подклассы топологически эквивалентных.

Задачу установления топологической эквивалентности транзисторных подсхем приходится решать и в том случае, когда для декомпилируемой схемы априори известна библиотека подсхем логических элементов, использованная при ее проектировании, и необходимо распознавать их среди выделенных на этапе структурной декомпиляции групп транзисторов, связанных по току. Эта же задача встречается на этапе верификации декомпилированной схемы относительно исходной транзисторной схемы.

В настоящей работе рассматриваются графовая постановка и метод решения задачи распознавания топологически эквивалентных транзисторных подсхем. На этом этапе имеется множество связанных помеченных подграфов, соответствующих транзисторным подсхемам, найденным структурным методом. Задача сводится к разбиению множества таких подграфов на классы изоморфных графов, задающих топологически эквивалентные подсхемы логических элементов и псевдоэлементов. Представители этих классов будут задавать найденные библиотечные элементы, которые формируют второй уровень иерархического описания транзисторной схемы.

Задание транзисторных схем. Исходными данными для программы декомпиляции служат плоский нетлист транзисторной схемы в формате SPICE (Simulation Program with Integrated Circuit Emphasis) для обмена электрическими схемами [1], имя головной схемы и имена цепей питания. Результатом является двухуровневая транзисторная схема, представляемая также иерархическим SPICE-описанием, включающим модели всех идентифицированных КМОП-вентилей. Пример такой схемы показан на рис. 1, а.

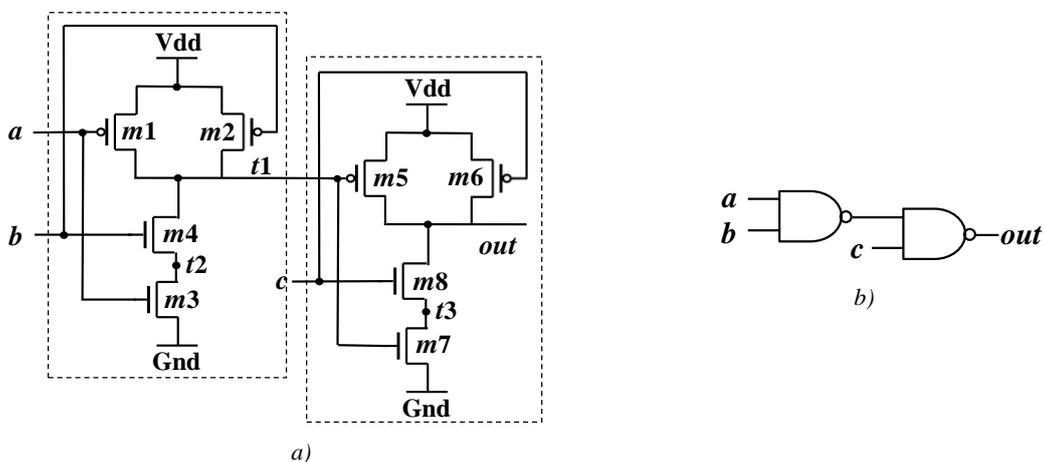


Рис. 1. Транзисторная схема (а) и реализующая ее логическая схема (b)
Fig. 1. Transistor circuit (a) and implemented logic network (b)

В SPICE-формате электрические схемы состоят из элементов, соединенных друг с другом цепями. Главной частью описания схемы в формате SPICE является список транзисторов, в котором для каждого вывода транзистора (сток, затвор, исток, подложка) указано имя цепи, соединяющей его с остальными частями схемы. Общая форма описания связей униполярного транзистора в формате SPICE имеет вид <name> <nd> <ng> <ns> <nb> <model-name>, где name – название транзистора; nd, ng, ns и nb – идентификаторы цепей, связанных с выводами стока (drain), затвора (gate), истока (source) и подложки (substrate) соответственно; model-name – тип транзистора *n*-МОП или *p*-МОП (nmos или pmos).

Например, в приведенном ниже SPICE-описании транзисторной схемы с рис. 1 строки описания связей выводов транзисторов начинаются с названий транзисторов и заканчиваются их типами (pmos и nmos). После названия транзистора перечисляются метки цепей, связанные с выводами стока, затвора, истока и подложки в заранее определенной последовательности.

Листинг 1. SPICE-описание транзисторной схемы

```
.GLOBAL Gnd Vdd
.subckt 2NAND a b c out
m1 t1 a Vdd Vdd pmos
m2 t1 b Vdd Vdd pmos
m3 Gnd a t2 Gnd N
m4 t2 b t1 Gnd N
m5 out t1 Vdd Vdd pmos
m6 Vdd c out Vdd pmos
m7 Gnd t1 t3 Gnd N
m8 t3 c out Gnd N
.ends.
```

Можно заметить, что в схеме на рис. 1, *a* *p*-МОП-транзистор *m6* включен нестандартно, т. е. к узлу питания *Vdd* он подключен выводом стока, а не истока (как остальные *p*-МОП-транзисторы). Соответствующая строка SPICE-описания выделена жирным шрифтом.

Графовая интерпретация задачи распознавания подсхем взаимосвязанных транзисторов. При решении задачи поиска и распознавания в транзисторной схеме подсхем, реализующих логические элементы, удобной и компактной моделью представления транзисторной схемы является помеченный неориентированный двудольный граф $G = (V_1, V_2, E)$, $V_1 \cap V_2 = \emptyset$. Одну долю V_1 графа составляют вершины, соответствующие выводам элементов и портам схемы (выводам всей электрической схемы), а другую долю V_2 – вершины, соответствующие цепям – соединениям между выводами транзисторов. Примерами таких соединений являются цепи питания и земли, связанные с большим числом элементов схемы. На рис. 2 изображен граф $G = (V_1, V_2, E)$, являющийся моделью транзисторной схемы (рис. 1, *a*). Здесь $V_1 = \{m1d, m1g, m1s, m1b, m2d, m2g, m2s, m2b, \dots, m8d, m8g, m8s, m8b\}$, $V_2 = \{a, b, out, Vdd, Gnd, t1, t2, t3\}$.

Две подсхемы связанных по току транзисторов, на которые в процессе декомпиляции была разбита анализируемая транзисторная схема (см. рис. 1, *a*), были распознаны как вентили И-НЕ (см. рис. 1, *b*). Такие подсхемы находятся как компоненты связности графа, полученного из двудольного графа исходной транзисторной схемы (аналогичного приведенному на рис. 2) и преобразованного следующим образом: вначале удаляются выводы затворов и подложек транзисторов (и соответствующие цепи); затем глобальные порты питания *Vdd* и земли *Gnd* заменяются на локальные, а выводы стока и истока каждого транзистора соединяются ребром [12]. Ниже представлено двухуровневое описание транзисторной схемы, полученное в результате выполнения структурного этапа декомпиляции, причем первой в листинге представлена правая из двух подсхем на рис. 1, *a*. Обе найденные подсхемы функционально эквивалентны и реализуют функцию 2И-НЕ. Вместе с тем топологически они разнятся тем, что в первой подсхеме *p*-МОП-транзистор *m4* подключен к узлу питания *Vdd* вы-

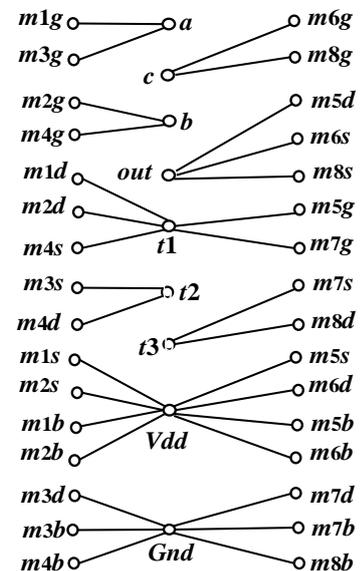


Рис. 2. Двудольный граф $G = (V_1, V_2, E)$, задающий структуру транзисторной схемы

Fig. 2. Bipartite graph $G = (V_1, V_2, E)$ defining the structure of transistor circuit

водом стока, а не истока (как p -МОП-транзистор $m4$ второй подсхемы). Соответствующие строки SPICE-описания выделены жирным шрифтом.

Листинг 2. Двухуровневое SPICE-описание транзисторной схемы

```
* SPICE deck for cell 2NAND_gen
.GLOBAL Vdd Gnd
.SUBCKT G0_0 A B Y
* (A AND B)
M1 Gnd A 2 Gnd N
M2 2 B Y Gnd N
M3 Y A Vdd Vdd pmos
M4 Vdd B Y Vdd pmos
.ENDS
.SUBCKT G0_1 A B Y
* (A AND B)
M1 Gnd A 2 Gnd N
M2 2 B Y Gnd N
M3 Y A Vdd Vdd pmos
M4 Y B Vdd Vdd pmos
.ENDS
.SUBCKT 2NAND_gen
XM0I1 t1 c out G0_0 Fets=Nn3+Nn4+pmsp3+pmsp4
XM0I2 a b t1 G0_1 Fets=Nn1+Nn2+pmsp1+pmsp2
.ENDS
```

Для современных СБИС число вершин в графе $G = (V_1, V_2, E)$ может достигать миллиардов. Доля V_1 выводов транзисторов двудольного графа, моделирующего схему из транзисторов, состоит из вершин степени 1. В доле V_2 цепей достаточно большое количество вершин имеют небольшие степени и только несколько вершин обладают большими степенями (например, цепи питания и земли). Таким образом, моделирующий транзисторную схему граф является разреженным.

Предположим, что граф $G = (V_1, V_2, E)$ является неориентированным и помеченным. Метки вершин зависят от типа соответствующих им полюсов схемы. Схемы, выполненные по КМОП-технологии, состоят из n -МОП- и p -МОП-транзисторов, каждый из которых имеет по четыре вывода (сток, затвор, исток и подложку). Поэтому вершины первой доли графа $G = (V_1, V_2, E)$, соответствующие выводам транзисторов, получают метки 1–8. Вершины второй доли, соответствующие цепям и портам схемы, – метки от 9 и больше, которыми помечаются входные и выходные полюсы схемы, шины питания и внутренние цепи.

Разбиение множества графов на классы изоморфных. На этапе разбиения транзисторных подсхем на подмножества изоморфных имеется множество неориентированных помеченных графов, задающих структуры этих подсхем. Проверка эквивалентности двух схем сводится к задаче установления изоморфизма графов, задающих структуры сравниваемых схем. Таким образом, ключевая операция, выполняемая при разбиении множества графов на классы изоморфных, состоит в анализе графов на изоморфизм.

При массовом решении задачи об установлении изоморфизма графов, когда требуется выполнить много таких проверок для множества графов, лучшим выходом является приведение сравниваемых графов к каноническому виду [13, 14], вычислению и сравнению их хешей. *Хеш* представляет собой уникальный код (последовательность битов) графа, который получается преобразованием задания множества ребер графа как последовательности чисел в битовую строку, имеющую длину слова, и является инвариантом графа относительно порядка его вершин и ребер.

Графы изоморфны, если хеши их канонизированных форм совпадают. После канонизации графов вопрос об изоморфизме любой пары графов сводится к сличению хешей их канонических форм. Более того, сравнивая эти канонические метки, можно разбить множество графов на классы попарно изоморфных графов. При этом в каждом классе выбирается один граф, называ-

емый каноническим видом любого графа из этого класса. Канонический вид графа находится путем переупорядочения вершин графа в соответствии с их свойствами, не зависящими от исходной нумерации (инвариантами).

В теории графов проблема построения канонической формы графа называется его *канонизацией* [14]. В процессе канонизации графа производится разбиение множества его вершин на подмножества. Стоит отметить, что проблема канонизации графа в вычислительном отношении так же трудна, как и проблема изоморфизма графов [15].

В настоящее время существует несколько успешных программ вычисления канонических форм, различающихся деталями редукции дерева поиска, формируемого в процессе уточнения разбиения на множестве вершин графа (первая такая программа предложена в работе [16]). Однако оценка вычислительной сложности самого быстрого из известных алгоритмов проверки изоморфизма n -вершинных графов (и канонизации графов) составляет $2^{O(\sqrt{n \log n})}$, а алгоритм полиномиальной сложности неизвестен.

В рассматриваемом случае задача упрощается за счет учета специфики канонизируемых графов: они являются помеченными и разреженными. В качестве прототипа алгоритма получения канонической формы графа используется алгоритм из работы [17].

В процедуре канонизации учитываются особенности топологии двудольных графов $G = (V_1, V_2, E)$, которые моделируют схемы из транзисторов. Во-первых, так как в схеме транзисторного уровня обычно всего два типа элементов (n -МОП- и p -МОП-транзисторы), то доля V_1 выводов элементов (кроме выводов самой схемы) состоит из вершин степени 1, раскрашенных в один из восьми цветов в соответствии с семантикой выводов этих транзисторов. Во-вторых, доля V_2 цепей состоит из большого количества вершин с небольшими степенями и нескольких вершин с большими степенями (например, цепи питания). Так как подсхемы, соответствующие изоморфным графам, топологически являются эквивалентными, то множество сравниваемых на изоморфизм графов ограничивается в соответствии с легко вычисляемыми параметрами подсхем. В качестве таких параметров кроме равенства вершин в графах приняты также количество транзисторов разных типов и количество цепей.

Канонизация графов транзисторных схем. На этапе канонизации имеется помеченный неориентированный граф. Для выполнения задачи канонизации граф, представляющий анализируемую подсхему, дополняется ребрами, связывающими четыре вывода каждого из его транзисторов. Так как канонизируется помеченный граф, то задача канонизации сводится к изменению пометки – перемаркировке его вершин.

Помеченные графы $G = (V, E, g)$ и $H = (W, F, h)$, заданные множествами V и W вершин, множествами E и F ребер и пометками вершин $g: V \rightarrow P$ и $h: W \rightarrow P$ (множество P составляют возможные метки вершин), являются изоморфными, если между множествами их вершин существует взаимно однозначное соответствие, сохраняющее отношение смежности. Другими словами, графы изоморфны, если существует такая биекция $\varphi: V \leftrightarrow W$, что для любых вершин $v_i, v_j \in V$ их образы $\varphi(v_i)$ и $\varphi(v_j)$ смежны в графе H , если и только если они смежны в G , а при $w = \varphi(v)$ их метки равны: $g(v) = h(w)$.

При установлении изоморфизма в качестве инвариантов графа рассматриваются такие числовые характеристики вершин, как их метки и степени, а также число вершин, отстоящих от данной на определенном расстоянии. Необходимым, но недостаточным условием изоморфизма двух графов является равенство их инвариантов.

В процессе канонизации графа производится разбиение множества его вершин на подмножества. Разбиение множества V представляет собой упорядоченную последовательность непустых непересекающихся подмножеств V_i , объединением которых является V . Подмножества называются блоками разбиения. Разбиение множества вершин графа можно также рассматривать как перемаркировку вершин, в процессе которой одна и та же метка присваивается двум вершинам тогда и только тогда, когда они принадлежат одному и тому же блоку. Разбиения, в которых каждый блок содержит одну вершину, называются дискретными. Канонизация графа называется полной, если ее результатом является получение дискретного разбиения, а значит, и полного упорядочения вершин.

В качестве исходного разбиения $\{V_1, V_2, \dots, V_k\}$ при канонизации графа принимается разбиение множества его вершин по их меткам $g(v_i)$ и степеням $d(v_i)$: каждый блок V_i разбиения содержит вершины с одинаковыми пометками и степенями. Блоки разбиения упорядочиваются по возрастанию меток входящих в него вершин, а при равенстве меток вершин – по возрастанию их степеней. Для каждой вершины графа $v \in V$ вычисляется вектор $C(v) = (n_1, n_2, \dots, n_k)$, i -я компонента n_i которого равна числу вершин i -го блока V_i , смежных с данной вершиной.

Если в текущем разбиении $\{V_1, V_2, \dots, V_k\}$ найдется блок V_i , который содержит вершины с разными векторами $C(v)$, то он разделяется на более мелкие блоки так, чтобы каждый из них содержал вершины с одинаковыми векторами. Вновь полученные блоки лексикографически упорядочиваются по возрастанию векторов входящих в них вершин и заменяют собой расщепляемый блок. После расщепления некоторого блока увеличивается длина векторов всех вершин и соответствующим образом пересчитываются те компоненты векторов вершин графа, которых касается произведенное расщепление блока. Процесс разбиения продолжается до тех пор, пока в каждом блоке ни останутся вершины только с одинаковыми векторами.

Если нерасщепляемое разбиение будет оставаться дискретным, т. е. состоящим из блоков, содержащих по одной вершине, то процесс канонизации заканчивается получением канонического представления графа. Заметим, что при канонизации подграфов, моделирующих структуру транзисторных схем, не встретилось ни одного случая, когда бы разбиение на множестве вершин ни заканчивалось получением дискретного разбиения. Это объясняется спецификой таких графов.

Применение канонизации графов при декомпиляции транзисторных схем. Рассмотрим выделенные при экстракции подсхемы (см. их описание в листинге 2). Эти подсхемы функционально эквивалентны, но топологически отличаются друг от друга тем, что во второй схеме p -МОП-транзистор $m4$ подключен к узлу питания Vdd полюсом стока, а не истока (как p -МОП-транзистор $m4$ в первой схеме). Покажем, к чему приводит такое отличие.

Канонизируем графы $G = (V, E, g)$ и $H = (W, F, h)$, которые соответствуют двум анализируемым схемам и находятся при экстракции как подграфы преобразованного графа на рис. 2. Для упрощения задачи канонизации дополним графы $G = (V, E, g)$ и $H = (W, F, h)$ ребрами, соединяющими вершины, которые соответствуют четырем выводам каждого транзистора. Полученные графы изображены на рис. 3, где под именем каждой вершины указаны ее метки (начальная и полученная в результате канонизации).

Метки 1–4 присвоены вершинам графа, соответствующим выводам стока, истока, подложки и затвора n -МОП-транзисторов, метки 5–8 – вершинам выводов стока, истока, подложки и затвора p -МОП-транзисторов. Затем последовательно выставлены метки вершинам, соответствующим входным полюсам схемы, полюсам Gnd, Vdd, выходным полюсам схемы и, наконец, вершинам цепей схемы.

Для графов анализируемых схем введены 14 меток, в соответствии с метками начальное разбиение множества вершин каждого из графов состоит из 14 блоков вершин с одинаковыми метками и степенями. Блоки разбиения упорядочиваются по возрастанию меток (и степеней вершин при равенстве меток). Оба графа имеют одну и ту же начальную маркировку вершин, она приведена в первом столбце таблицы. В результате канонизации каждого из графов получены дискретные разбиения на множестве их вершин, разбиения показаны во втором (для графа $G = (V, E, g)$) и четвертом (для графа $H = (W, F, h)$) столбцах таблицы. Эти разбиения различаются порядком нумерации блоков 9 и 10 (эти блоки помечены жирным шрифтом), что породило различие в списках связей вершин графов. Списки вершин, смежных каждой из вершин графа, приведены в третьем (для графа $G = (V, E, g)$) и пятом (для графа $H = (W, F, h)$) столбцах таблицы. Различающиеся связи помечены в таблице жирным шрифтом. Хэши связей канонизированных графов равны соответственно 1 538 850 246 и 1 583 347 427. Это говорит о том, что графы неизоморфны, а соответствующие им схемы топологически различаются.

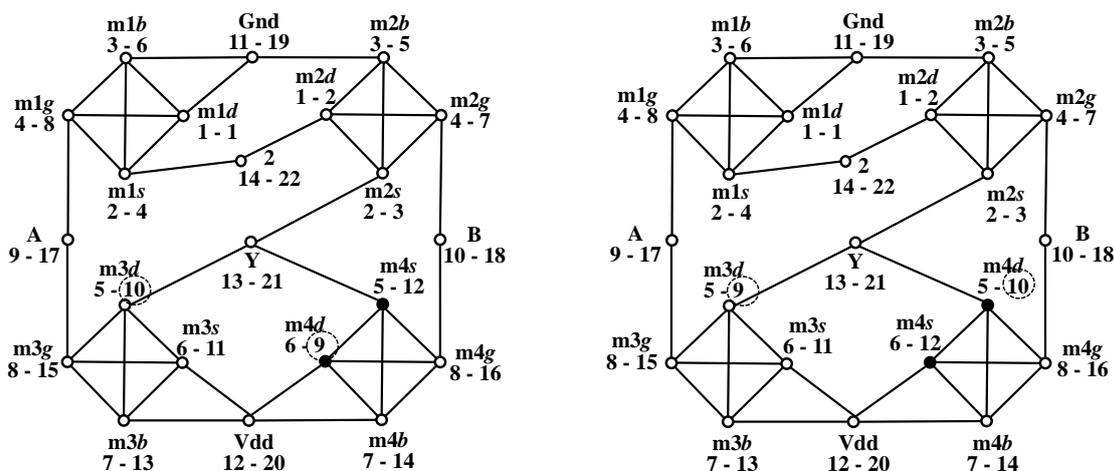


Рис. 3. Графы $G = (V_1, V_2, E)$, задающие структуры выделенных подсхем (см. листинг 2)
 Fig. 3. Graphs $G = (V_1, V_2, E)$ that define the structures of the selected subcircuits (see Listing 2)

Результаты канонизации графов $G = (V, E, g)$ и $H = (W, F, h)$

Results of graph canonization $G = (V, E, g)$ и $H = (W, F, h)$

Начальная маркировка Initial labeling	Граф $G = (V, E, g)$ Graph $G = (V, E, g)$		Граф $H = (W, F, h)$ Graph $G = (V, E, g)$	
	Маркировка Labeling	Связи Connections	Маркировка Labeling	Связи Connections
1. {m1d, m2d}	1. m1d	4, 6, 8, 19	1. m1d	4, 6, 8, 19
2. {m1s, m2s}	2. m2d	3, 5, 7, 22	2. m2d	3, 5, 7, 22
3. {m1b, m2b}	3. m2s	2, 5, 7, 21	3. m2s	2, 5, 7, 21
4. {m1g, m2g}	4. m1s	1, 6, 8, 22	4. m1s	1, 6, 8, 22
5. {m3d, m4d}	5. m2b	2, 3, 7, 19	5. m2b	2, 3, 7, 19
6. {m3s, m4s}	6. m1b	1, 4, 8, 19	6. m1b	1, 4, 8, 19
7. {m3b, m4b}	7. m2g	2, 3, 5, 18	7. m2g	2, 3, 5, 18
8. {m3g, m4g}	8. m1g	1, 4, 6, 17	8. m1g	1, 4, 6, 17
9. {a}	9. m4d	12, 14, 16, 20	9. m3d	11, 13, 15, 21
10. {b}	10. m3d	11, 13, 15, 21	10. m4d	12, 14, 16, 21
11. {Gmd}	11. m3s	10, 13, 15, 20	11. m3s	9, 13, 15, 20
12. {Vcc}	12. m4s	9, 14, 16, 21	12. m4s	10, 14, 16, 20
13. {y}	13. m3b	10, 11, 15, 20	13. m3b	9, 11, 15, 20
14. {2}	14. m4b	9, 12, 16, 20	14. m4b	10, 12, 16, 20
	15. m3g	10, 11, 13, 17	15. m3g	9, 11, 13, 17
	16. m4g	9, 12, 14, 18	16. m4g	10, 12, 14, 18
	17. a	8, 15	17. a	8, 15
	18. b	7, 16	18. b	7, 16
	19. Gmd	1, 5, 6	19. Gmd	1, 5, 6
	20. Vcc	9, 11, 13, 14	20. Vcc	11, 12, 13, 14
	21. y	3, 10, 12	21. y	3, 9, 10
	22. 2	2, 4	22. 2	23. 2, 4

Экспериментальные исследования. Предложенный графовый метод установления топологической эквивалентности транзисторных схем реализован на языке C++ как часть программы декомпиляции плоской транзисторной схемы. Канонизация графов использовалась не только при разбиении множества транзисторных подсхем на классы топологически эквивалентных, но и при верификации результатов экстракции иерархической схемы на транзисторно-логическом уровне относительно плоской схемы на транзисторном уровне.

В первом эксперименте декомпилировались схемы транзисторного уровня, полученные с помощью САПР, исходя из функциональных описаний на языке VHDL, во втором – практические, извлеченные из лейаута, о которых ничего не известно кроме полученного SPICE-описания.

Схемы, полученные с помощью САПР, были полностью покрыты логическими вентилями. (Нераспознанных подсхем не обнаружено.) Кроме того, все функционально эквивалентные подсхемы оказались и топологически эквивалентными.

Для извлеченных из лейаута схем было известно только иерархическое описание в формате SPICE, кроме описания на уровне транзисторов никакой дополнительной информации не имелось. Здесь наблюдалось покрытие схемы транзисторного уровня логическими вентилями на уровне 60–70 %. Анализ декомпилированных схем, для которых было известно иерархическое SPICE-описание, показал, что программа декомпиляции находила все логические вентили, выделялись псевдоэлементы (подсхемы из связанных транзисторов, похожие на элементы) и одиночные транзисторы.

Для схем, извлеченных из лейаута, распознанные множества функционально эквивалентных транзисторных подсхем, как правило, содержали топологически неэквивалентные подсхемы. Соответственно, эти множества разбивались на подмножества из топологически эквивалентных подсхем. Аналогично множества псевдоэлементов с одинаковыми характеристиками также разбивались на подмножества из топологически эквивалентных схем псевдоэлементов. Из приведенного ниже примера декомпиляции одной из транзисторных схем видно, насколько часто функционально эквивалентные транзисторные подсхемы (и псевдоэлементы с одинаковыми характеристиками) оказываются топологически неэквивалентными.

Листинг 3. Пример декомпиляции транзисторной схемы

```
Circuit AP2009 contains 3168 device instances.
  Class: n                instances: 1997
  Class: p                instances: 1171
Circuit contains 1646 nets.

Class AP2009: Merged 189 devices.
Contents of circuit d2009_all_topCher.sp: Circuit: 'AP2009'
Circuit AP2009 contains 2979 device instances.
  Class: n                instances: 1879
  Class: p                instances: 1100
Circuit contains 1646 nets.

Connected Components = 518
Invalid comps
Valid Components = 620
Valid gates = 607
Psevdo Components = 20 nets =705
Unclassified fets = 18 nets = 15
Pass gate          127

G0: ((A AND B) OR (C AND D)) 2 (1 3)
G1: (A AND B AND C AND D AND E) 4 (13 1 2 1)
G2: (A AND B AND C AND D) 3 (2 1 2)
G3: (A AND B AND C) 2 (5 20)
G4: (A AND B) 5 (13 35 27 15 1)
G5: (A OR (B AND C AND D)) 1 (1)
G6: (A OR (B AND C)) 1 (1)
G7: (A OR B OR (C AND D)) 1 (1)
G8: (A OR B OR C OR D) 2 (4 9)
G9: (A OR B OR C) 2 (4 10)
G10: (A OR B) 5 (9 13 18 2 1)
G11: (C AND (A OR B)) 1 (2)
G12: A 2 (135 128)
Classes=13 TopologySum=31

Defining cell: AP2009_gen
  Defining global node: VCC1
```

```

Defining global node: gnd

G1_3 all instances have shorted pins
G2_1 all instances have shorted pins
G4_4 all instances have shorted pins
G10_1 all instances have shorted pins
G10_3 all instances have shorted pins

Psevdo
P0: (2) (3) 2 (1 1)
P1: (4) (4) 2 (1 1)
P2: (48) (32) 2 (1 1)
P3: (5) (5) 1 (1)
P4: (5) (6) 1 (1)
P5: (6) (5) 2 (1 1)
P6: (6) (6) 1 (6)
P7: (7) (4) 1 (1)
P8: (7) (5) 1 (1)
P9: (8) (7) 1 (1)
P10: (843) (554) 1 (1)
PsevdoClasses=11 TopologySum=15

Start of Computation: 15h43m24s 28/04/2022
End of Computation: 15h43m24s 28/04/2022
Computation Time (s): 0.1330

```

Исследуемая схема (листинг 3) содержит 3168 транзисторов (и 1646 цепей), среди которых 1997 *n*-МОП- и 1171 *p*-МОП-транзисторов. В этой схеме найдено 518 групп транзисторов, связанных по постоянному току. Среди них распознано 480 статических КМОП-вентилей и 20 псевдоэлементов, 18 групп вентилях (содержащих в сумме 15 цепей) не классифицированы. Также найдено 127 передаточных элементов.

Множество из 480 статических КМОП-вентилей разделено на 13 классов функционально эквивалентных вентилях. В свою очередь, множество вентилях каждого класса функционально эквивалентных вентилях разделено на подмножества, представляющие группы топологически эквивалентных вентилях. Всего получилась 31 группа топологически эквивалентных КМОП-вентилей. Данные о каждом классе G_i вентилях представляются в листинге строкой $G_i: (f_i) \ g \ (t_1 \ t_2 \dots t_g)$, где f_i – инверсия функции, реализуемой каждым вентиляем класса и представленной в виде булевой формулы канонического вида; g – число групп топологически эквивалентных вентилях из G_i ; $t_1 \ t_2 \dots t_g$ – число вентилях, принадлежащих каждой группе.

Кроме того, в исследуемой транзисторной схеме обнаружены КМОП-вентили, у которых некоторые входные полюсы закорочены [4]. Такие вентили в каждом классе G_i выделяются в виде отдельной группы топологически эквивалентных вентилях. Данные о вентилях с закороченными входами, обнаруженных в классе G_i , задаются в листинге строками вида $G_i_g_j \ all \ instances \ have \ shorted \ pins$, где g_j – номер группы топологически эквивалентных вентилях в классе G_i .

Множество из 20 выделенных схем псевдоэлементов разделено на 11 групп P_i элементов с одинаковым числом транзисторов и цепей. Каждая группа P_i псевдоэлементов в свою очередь разделена на подгруппы топологически эквивалентных элементов. Данные о каждой группе псевдоэлементов представляются в листинге строкой $P_i: (k_t) \ (k_n) \ g \ (t_1 \ t_2 \dots t_g)$, где k_t , k_n и g – число транзисторов, цепей и подгрупп топологически эквивалентных элементов, $t_1 \ t_2 \dots t_g$ – число псевдоэлементов, принадлежащих каждой подгруппе. Для приведенного примера всего получилось 15 классов топологически эквивалентных псевдоэлементов.

Эксперименты показали, что канонизация подграфов, соответствующих всем распознаваемым транзисторным подсхемам, всегда заканчивалась нахождением канонической маркировки вершин.

Заключение. Предложенные графовые методы распознавания и классификации подсхем, реализующих КМОП-вентили и часто встречающиеся фрагменты (псевдоэлементы), в транзисторной схеме реализованы как часть программ декомпиляции транзисторных схем и верификации результатов экстракции иерархического описания схемы на транзисторно-логическом уровне относительно исходного плоского описания схемы на транзисторном уровне. Программа декомпиляции протестирована на практических схемах транзисторного уровня и имеет достаточное быстродействие, чтобы обрабатывать схемы более чем с 100 тыс. транзисторов за несколько минут на ПЭВМ. Самая большая из исследованных схем содержала 345 301 транзистор и декомпилировалась за 163 с на компьютере с четырехъядерным процессором Intel i5-4460 3,20 ГГц и оперативной памятью 16,0 ГБ ЭВМ. Декомпилированные схемы прошли проверку на соответствие исходной топологии транзисторной схемы с помощью разработанных средств верификации [18], а также с помощью Mentor Graphics Calibre nmLVS. Во всех случаях декомпилированные схемы успешно проходили проверку LVS топологии СБИС.

Список использованных источников

1. Baker, R. J. CMOS Circuit Design, Layout, and Simulation / R. J. Baker. – Third ed. – Wiley-IEEE Press, 2010. – 1214 p.
2. Zhang, N. The subcircuit extraction problem / N. Zhang, D. C. Wunsch, F. Narary // Proc. IEEE Intern. Behavioral Modeling and Simulation Workshop. – 2005. – Vol. 33(3). – P. 22–25.
3. Yang, L. FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits / L. Yang, C.-J. R. Shi // Integration the VLSI J. – 2006. – Vol. 39, no 4. – P. 311–339.
4. Черемисинов, Д. И. Извлечение сети логических элементов из КМОП-схемы транзисторного уровня / Д. И. Черемисинов, Л. Д. Черемисинова // Микроэлектроника. – 2019. – Т. 48, № 3. – С. 224–234. <https://doi.org/10.1134/S0544126919030037>
5. Abadir, M. S. An improved layout verification algorithm (LAVA) / M. S. Abadir, J. Ferguson // Proc. of the European Design Automation Conf., Glasgow, UK, 12–15 Mar. 1990. – Glasgow, 1990. – P. 391–395.
6. Framework for simulation of the Verilog/SPICE mixed model: Interoperation of Verilog and SPICE simulators using HLA/RTI for model reusability / M. G. Seok [et al.] // 22nd Intern. Conf. on Very Large Scale Integration (VLSI-SoC), Playa del Garmen, Mexico, 6–8 Oct. 2014. – Playa del Garmen, 2014. – P. 1–6.
7. Kundu, S. GateMaker: A transistor to gate level model extractor for simulation, automatic test pattern generation and verification / S. Kundu // Proc. of the Intern. Test Conf., Washington, DC, USA, 18–23 Oct. 1998. – Washington, 1998. – P. 372–381.
8. Hunt, V. D. Reengineering: Leveraging the Power of Integrated Product Development / V. D. Hunt. – Wiley, 1993. – 283 p.
9. Rostami, M. A primer on hardware security: Models, methods, and metrics / M. Rostami, F. Koushanfar, R. Karri // Proceedings of the IEEE. – 2014. – Vol. 102, no. 8. – P. 1283–1295.
10. Tehranipoor, M. A survey of hardware trojan taxonomy and detection / M. Tehranipoor, F. Koushanfar // IEEE Design & Test of Computers. – 2010. – Vol. 27, no. 1. – P. 10–25.
11. Белоус, А. И. Основы кибербезопасности. Стандарты, концепции, методы и средства обеспечения / А. И. Белоус, В. А. Солодуха. – М. : Техносфера, 2021. – 482 с.
12. Черемисинов, Д. И. Распознавание логических вентилях в плоской транзисторной схеме / Д. И. Черемисинов, Л. Д. Черемисинова // Информатика. – 2021. – Т. 18, № 4. – С. 96–107. <https://doi.org/10.37661/1816-0301-2021-18-4-96-107>
13. Hartke, S. G. McKay's Canonical Graph Labeling Algorithm / S. G. Hartke, A. J. Radcliffe // Communicating Mathematics. – 2009. – Vol. 479. – P. 99–111.
14. Закревский, А. Д. Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. – М. : Физматлит, 2007. – 589 с.
15. Garey, M. R. Computers and Intractability: A Guide to the Theory of NP-Completeness / M. R. Garey, D. S. Johnson. – N. Y. : W. H. Freeman and Company, 1979. – 340 p.
16. McKay, B. D. Practical graph isomorphism / B. D. McKay // Congressus Numerantium. – 1981. – Vol. 30. – P. 45–87.

17. Junttila, T. Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs / T. Junttila, P. Kaski // Proc. Meeting on Algorithm Engineering & Experiments SIAM, New Orleans, LA, 6 Jan. 2007. – New Orleans, 2007. – P. 135–149.

18. Черемисинов, Д. И. Верификация логических схем из КМОП-транзисторов / Д. И. Черемисинов, Л. Д. Черемисинова // Новые информационные технологии в исследовании сложных структур : материалы 13-й Междунар. конф., 7–9 сент. 2020 г. – Томск : Изд. дом Томского гос. ун-та, 2020. – С. 150–151.

References

1. Baker R. J. *CMOS Circuit Design, Layout, and Simulation*. Third ed. Wiley-IEEE Press, 2010, 1214 p.
2. Zhang N., Wunsch D. C., Harary F. The subcircuit extraction problem. *Proceedings IEEE International Behavioral Modeling and Simulation Workshop*, 2005, vol. 33(3), pp. 22–25.
3. 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 Journal*, 2006, vol. 39, no 4, pp. 311–339.
4. Cheremisinov D. I., Cheremisinova L. D. *Extracting a logic gate network from a transistor-level CMOS circuit*. *Mikroelektronika [Russian Microelectronics]*, 2019, vol. 48, no. 3, pp. 224–234. <https://doi.org/10.1134/S0544126919030037> (In Russ.).
5. Abadir M. S., Ferguson J. An improved layout verification algorithm (LAVA). *Proceedings of the European Design Automation Conference, Glasgow, UK, 12–15 March 1990*. Glasgow, 1990, pp. 391–395.
6. Seok M. G., Park D. J., Cho G. R., Kim T. G. Framework for simulation of the Verilog/SPICE mixed model: Interoperation of Verilog and SPICE simulators using HLA/RTI for model reusability. *22nd International Conference on Very Large Scale Integration (VLSI-SoC), Playa del Garmen, Mexico, 6–8 October 2014*. Playa del Garmen, 2014, pp. 1–6.
7. Kundu S. GateMaker: A transistor to gate level model extractor for simulation, automatic test pattern generation and verification. *Proceedings of the International Test Conference, Washington, DC, USA, 18–23 October 1998*. Washington, 1998, pp. 372–381.
8. Hunt V. D. *Reengineering: Leveraging the Power of Integrated Product Development*. Wiley, 1993, 283 p.
9. Rostami M., Koushanfar F., Karri R. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, 2014, vol. 102, no. 8, pp. 1283–1295.
10. Tehranipoor M., Koushanfar F. A survey of hardware trojan taxonomy and detection. *IEEE Design & Test of Computers*, 2010, vol. 27, no. 1, pp. 10–25.
11. Belous A. I., Solodukha V. A. Osnovy kiberbezopasnosti. Standarty, kontseptsii, metody i sredstva obespecheniya. *Fundamentals of Cybersecurity. Standards, Concepts, Methods and Means of Support*. Moscow, Tekhnosfera, 2021, 482 p. (In Russ.).
12. Cheremisinov D. I., Cheremisinova L. D. *Logical gates recognition in a flat transistor circuit*. *Informatika [Informatics]*, 2021, vol. 18, no. 4, pp. 96–107. <https://doi.org/10.37661/1816-0301-2021-18-4-96-107> (In Russ.).
13. Hartke S. G., Radcliffe A. J. McKay's Canonical Graph Labeling Algorithm. *Communicating Mathematics*, 2009, vol. 479, pp. 99–111.
14. Zakrevskij A. D., Pottosin Yu. V., Cheremisinova L. D. Logicheskiye osnovy proyektirovaniya diskretnykh ustroystv. *Logical Basis for Designing Discrete Devices*. Moscow, Fizmatlit, 2007, 589 p. (In Russ.).
15. Garey M. R., Johnson D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, W. H. Freeman and Company, 1979, 340 p.
16. McKay B. D. Practical graph isomorphism. *Congressus Numerantium*, 1981, vol. 30, pp. 45–87.
17. Junttila T., Kaski P. Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs. *Proceedings Meeting on Algorithm Engineering & Experiments SIAM, New Orleans, LA, 6 January 2007*. New Orleans, 2007, pp. 135–149.
18. Cheremisinov D. I., Cheremisinova L. D. *Verification of logic circuits from CMOS transistors*. *Novyye informatsionnyye tekhnologii v issledovanii slozhnykh struktur : materialy 13-j Mezhdunarodnoj konferencii, 7–9 sentyabrya 2020 g. [New Information Technologies in the Study of Complex Structures: Proceedings of the 13th International Conference, 7–9 September 2020]*. Tomsk, Izdatel'skij dom Tomskogo gosudarstvennogo universiteta, 2020, pp. 150–151 (In Russ.).

Информация об авторах

Черемисинов Дмитрий Иванович, кандидат технических наук, доцент, ведущий научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.

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

Черемисинова Людмила Дмитриевна, доктор технических наук, профессор, главный научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.

E-mail: cld@newman.bas-net.by

Information about the authors

Dmitry I. Cheremisinov, Ph. D. (Eng.), Associate Professor, Leading Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

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

Ljudmila D. Cheremisinova, D. Sc. (Eng.), Professor, Chief Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

E-mail: cld@newman.bas-net.by