

Декомпиляция КМОП схемы из транзисторов в формате SPICE

Д.И. Черемисинов, Л.Д. Черемисинова

Объединений институт проблем информатики НАНБ, г. Минск {cher, cld}@newman.bas-net.by

Аннотация — 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, а также при перепроектировании (reengineering) схем. Описывается метод распознавания подсхем, являющихся КМОП-вентилем, 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.

Ключевые слова — перепроектирование СБИС, КМОП схемы из транзисторов, формат SPICE, КМОП-вентиль, распознавание подсхем, алгебраическая факторизация ДНФ, канонизация неориентированных графов.

1. ВВЕДЕНИЕ

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 схемы строится иерархическая транзисторная схема называется декомпиляцией. Декомпилятор [1] – программа осуществляющая декомпиляцию схем – 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 при декомпиляции программ, целью декомпиляции схемы является замена представления схемы на низком (транзисторном) уровне более высокоуровневым ее представлением. В отличие от программного процесса декомпиляция схем не является языковой трансформацией, хотя декомпилятор использует определенный формат (формальный язык представления данных) 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 формате SPICE электрические схемы состоят из элементов, которые соединены друг с другом цепями (рис. 1), и естественной формальной моделью описания

схемы является помеченный неориентированный двудольный граф. Одну долю составляют выходы элементов и порты схемы, а другую – соединения между выходами, т.е. цепи. На рисунках этот граф компактнее представляется в виде его реберного графа. 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 . Пометками элементов 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».

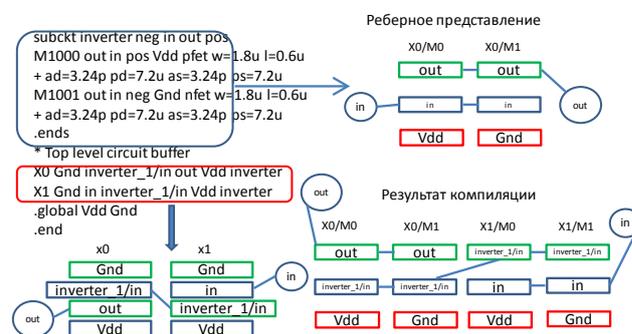


Рис. 1. Компиляция подсхем (circuit flattening) в формате SPICE

Для построения иерархического структурного описания нужно выделить наборы взаимосвязанных транзисторов в схеме в качестве отдельных компонент, т.е. найти подсхемы в исходной плоской схеме (рис. 2). В результате замены подсхем из транзисторов элементами строится схема из функциональных элементов (СФЭ) [4] – результат декомпиляции.

Технология КМОП позволяет реализовать функциональные элементы схемы различными способами. Существует множество «стилей» для реализации логических элементов: static, dynamic, Domino, CVSL, pass transistor logic [5, 6]. Не исключено, что в схеме используется способ реализации некоторых логических элементов, не относящийся ни к одному из этих стилей. Кроме того, схема может содержать кроме цифровой части аналоговые функциональные элементы. Частичная декомпиляция имеет смысл, хотя функции схемы в целом восстановить в этом случае нельзя, т.е. метод обработки СФЭ [4] неприменим. С учетом существования

множество «стилей» для реализации логических элементов в технологии КМОП, декомпиляция схемы из транзисторов – это не тривиальная задача.

II. РАСПОЗНАВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ В СХЕМЕ ИЗ КМОП-ТРАНЗИСТОРОВ

A. Задана библиотека элементов

Если искомые подсхемы библиотечных элементов заданы, т.е. известна библиотека подсхем, то образцы графов для поиска формируются из описания этих библиотечных элементов. Задача декомпиляции в этом случае сводится к задаче поиска в исходном графе подграфов, изоморфных заданному [7]. Декомпиляция успешна 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.

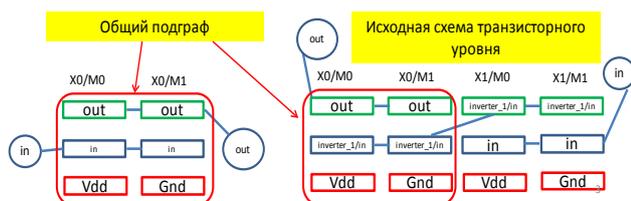


Рис. 2. Декомпиляция - поиск и извлечение подсхемы (subcircuit recognition – extraction)

B. Библиотека неизвестна

В дальнейшем задача декомпиляции рассматривается только для случая, когда библиотека подсхем не известна. Если нужно распознать библиотеку подсхем, то задача поставлена некорректно, так как любой фрагмент, содержащий целое число транзисторов, может считаться частью библиотечной подсхемы. Нужны дополнительные критерии, позволяющие выделять подсхемы. Так же как в работе [8], будем выделять в качестве подсхем фрагменты транзисторной КМОП схемы, для которых возможно определить логическую функцию. Нужно отметить, что предположение о классе распознаваемых подсхем совершенно необходимо, чтобы задача декомпиляции имела решение.

В цифровых КМОП схемах МОП-транзисторы могут рассматриваться как ключи, управляемые входными напряжениями на затворе. Простейшая цифровая схема – это передаточный элемент из одного МОП-транзистора, который обеспечивает управляемую передачу двоичного сигнала. Это пассивный элемент, поскольку он не обеспечивает усиление входного (коммутируемого) сигнала. Усиления двоичных сигналов обеспечивает симметричная схема (КМОП-вентиль), в которой n-МОП и p-МОП транзисторы включены последовательно между электродами напряжения питания. КМОП-вентили состоят из двух блоков, которые разделены выходной цепью (рис. 3).

Блок, содержащий n-МОП транзисторы (pull-down network), размещен между выходом и цепью нулевого потенциала. Блок, содержащий p-МОП транзисторы (pull-up network), размещен между цепью напряжения

питания и выходом. Причем затворы всех МОП транзисторов соединены с входами схемы. Блоки присоединяют выход Q либо к источнику питания VDD, либо к нулевому потенциалу (земле), если проводимости блоков комплементарны. Логическая функция КМОП-вентилей определяется отрицанием функции проводимости МОП транзисторов n-МОП блока (или функции проводимости p-МОП блока с инвертированием входных переменных) [6, 9].

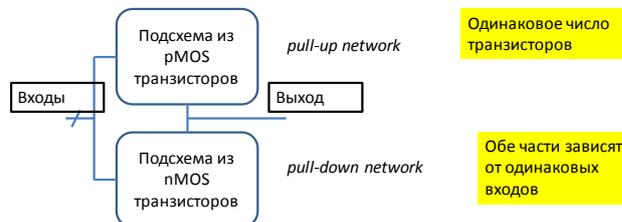


Рис. 3. КМОП-вентиль

Группой транзисторов, соединённых по постоянному току (channel-connected component – CCC) [10], является произвольная схема из МОП-транзисторов с тремя типами внешних соединений: (1) входы подаются только на затворы транзисторов; (2) выходы подаются только на затворы транзисторов в других компонентах и (3) имеются соединения с VDD и землей. КМОП-вентиль представляет собой группу транзисторов, соединённых по постоянному току, обратное верно не всегда.

В схеме с элементами из МОП-транзисторов группа транзисторов, соединённых по постоянному току, соответствует компоненте связности графа схемы, в которой удалены выводы затворов (и соответствующие цепи) и цепи питания, а выводы стока и истока каждого транзистора соединены ребром. Алгоритм поиска компонент связности в таком графе построен на основе алгоритма «поиск сначала в глубину» (DFS) [11]. Стоит отметить, что для поиска групп транзисторов, соединённых по постоянному току, необходимо знать, какие цепи в исходной схеме задают потенциалы VDD и земли. Множество групп транзисторов, соединённых по постоянному току, представляет собой разбиение множества транзисторов схемы.

C. Правильные группы транзисторов, соединённых по постоянному току

Не каждая группа транзисторов, соединённых по постоянному току, является КМОП-вентилем. Необходимыми условиями принадлежности группы к классу КМОП-вентилей являются следующие признаки, задающие правильные группы:

- единственной цепью, соединяющей p- и n- части, является выход элемента;
- группа содержит обе цепи питания;
- все пути из цепи выхода идут до цепей питания (и наоборот);
- внутренние цепи не должны содержать выводов посторонних транзисторов.

III. РАСПОЗНАВАНИЕ ФУНКЦИИ ЛОГИЧЕСКОГО ЭЛЕМЕНТА

Будем считать управление проводимостью каждого транзистора из группы независимым входом. Тогда число переменных логической функции проводимости из цепи выхода до цепи питания равно числу транзисторов в блоке pull-down network (pull-up network). Сама функция проводимости в виде дизъюнктивной нормальной формы (ДНФ) представляет собой список путей из цепи выхода до цепи питания (рис. 4). Каждый путь задает конъюнкцию переменных управления проводимостью соответствующих транзисторов (цепь, соединённая с затвором). Если функция проводимости блока pull-down network комплементарна (является отрицанием) функции проводимости блока pull-up network, то правильная группа транзисторов, соединённых по постоянному току, представляет КМОП-вентиль. Для классификации КМОП-вентилей схемы удобна скобочная форма представления функции проводимости блока pull-down network. Формула задания функции логического элемента строится путем алгебраической факторизации ДНФ [12] булевой функции, которая строит ее алгебраическое представление в скобочной форме (рис. 4).

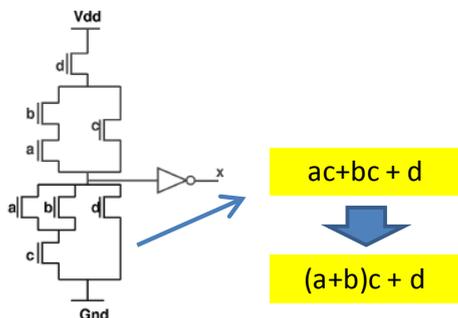


Рис. 4. Функция проводимости КМОП-вентилей в форме ДНФ и факторизованной ДНФ

Группы транзисторов, соединённых по постоянному току, являющиеся КМОП-вентилем, могут быть разбиты на классы по виду формулы логической функции. В один класс попадают все экземпляры каждого вентиля. Для классификации этих групп не требуется решать задачу изоморфизма графов.

A. Отсутствие симметрии входов у КМОП-вентилей, реализующих симметрические функции

Булева функция называется симметрической, если ее значение не зависит от перестановки её входных переменных. Однако, в КМОП-вентилей, реализующих симметрические функции, перестановка входов делает схему не изоморфной исходной (рис. 5). В правой схеме инвертор подключен к нижнему транзистору pull-down network, а в левой – к верхнему. Но логически эти схемы эквивалентны.

Так как входы КМОП-вентилей, реализующих симметрические функции, несимметричны, то, например, для вентиля И-НЕ возможны две неизоморфные схемы (рис. 6)

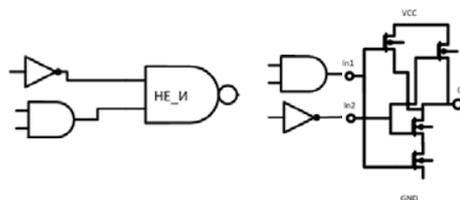


Рис. 5. Скомпилированные схемы не изоморфны после перестановки входов вентиля И-НЕ

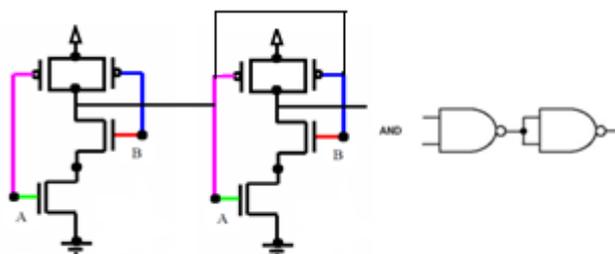
<pre>.SUBCKT G8_0 A B Y # (A AND B) M1 2 A 0 0 MODN M2 Y B 2 0 MODN M3 Y B VCC VCC MODP M4 Y A VCC VCC MODP .ENDS</pre>	<pre>.SUBCKT G8_1 A B Y # (A AND B) M1 Y A 1 0 MODN M2 1 B 0 0 MODN M3 Y A VCC VCC MODP M4 Y B VCC VCC MODP .ENDS</pre>
---	---

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

B. 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).

Рис. 7. Логический элемент с закороченными входами



C. 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 [13].

Взаимозаменяемость стока и истока МОП-транзистора приводит к тому, что существуют разные подходы, реализующие одну и ту же логическую функцию. Например, для КМОП-инвертора существует четыре варианта подходов. Если в декомпилированной схеме все варианты подходов логического элемента представить одним вариантом, то декомпилированная и оригинальная схема будут не изоморфны.

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, Mentor Graphics Calibre nmLVS, Guardian LVS и другими. Для сравнения двух схем имеется и свободное программное обеспечение, например, netgen [15]. В большинстве таких программ обе схемы могут быть представлены в формате SPICE. Сравнение схем, выполняемое этими программами, заключается в решении задачи изоморфизма графов.

Д. Передаточные элементы – Pass gates

Логические элементы могут быть построены на базе передаточных логических элементов (transmission gates) [14]. Передаточный элемент (ключ) составляют *n*-МОП- и *p*-МОП-транзисторы, соединённые параллельно выводами стока и истока соответственно. Соединённые выводы обозначаются через *A* и *B*, поскольку передача сигнала в таком логическом вентиле 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-транзистора соединен с отрицательным потенциалом питания. Выводы затворов обозначаются *EN* и \overline{EN} . Если *EN* равен 0, а \overline{EN} равен 1, то оба транзистора выключены. При этом вывод *A* не имеет связи с выводом *B*. В противном случае, передаточный логический вентиль включен, а *A* и *B* имеют одинаковый потенциал.

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

IV. КАНОНИЗАЦИЯ ПОДСХЕМ

Проверка изоморфности двух подсхем, реализующие одну и ту же логическую функцию, может быть сведена к задаче установления изоморфизма графов, в которые преобразуются сравниваемые подсхемы. Подсхема в формате SPICE эквивалентна помеченному неориентированному двудольному графу.

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 графов сводится к построению и сличению их канонических видов. Канонический вид графа определяется упорядочением вершин графа в соответствии с их свойствами, не зависящими от исход-

ной нумерации (инвариантами). В теории графов проблема построения канонической формы данного графа *G* называется канонизацией графа *G*. Канонизировав граф подсхемы и построив по нему представление в формате SPICE, проверку изоморфизма двух подсхем можно свести к проверке совпадения текстов описания подсхем. Проблема канонизации графа в вычислительном отношении так же трудна, как и проблема изоморфизма графов. Канонической формой графа может быть помеченный граф, поэтому задача канонизации сводится к изменению пометки (перемаркировке) его вершин.

Понятия, используемые для постановки задачи поиска канонической маркировки, взяты из теории графов и теории групп. Упорядоченным разбиением множества *V* является упорядоченная последовательностью непустых непересекающихся подмножеств ($W_1; W_2; \dots; W_k$), объединением которых является *V*. Подмножества *W_i* называются ячейками разбиения. Разбиение множества вершин графа можно рассматривать как раскраску вершин, которая присваивает один и тот же цвет двум вершинам тогда и только тогда, когда они принадлежат одной и той же ячейке. Разбиения, в которых каждая ячейка содержит один элемент, называются авторами метода в [15] дискретными. В русскоязычной литературе такое разбиение называется тривиальным. Для двух разбиений множества *V* первое лучше (<) чем второе, если первое может быть образовано путем расщепления нескольких ячеек второго разбиения. Отношение < является отношением частичного порядка, его минимальными элементами являются тривиальные разбиения.

Алгоритм канонической маркировки выполняет поиск вначале в глубину дерева, где каждый узел представляет собой разбиение множества вершин. Каждый дочерний класс формируется отделив в новую ячейку некоторой вершины (индивидуализацией вершины) из родительского класса. Индивидуализация разбиает родительский класс, образуя для этой вершины собственный класс. 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 [15].

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

V. ПРОГРАММА ДЕКОМПИЛЯЦИИ

Описанный выше метод декомпиляции КМОП схем из транзисторов в формате SPICE реализован в виде программы на языке C++. Исходными данными 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 КМОП-вентилей. Декомпиляция выполняется следующей последовательностью шагов:

- 1) Анализ исходного SPICE-описания и построение хэш-таблицы для хранения двудольного помеченного графа схемы, в ходе которого группа параллельно (и последовательно) соединенных транзисторов заменяются одним транзистором.
- 2) Нахождение передаточных элементов.
- 3) Разбиение оставшихся транзисторов на группы транзисторов, соединённых по постоянному току (ССС).
- 4) Распознавание правильных СССР, нахождение их логической функции в виде алгебраической формулы и построение хэш-таблицы для хранения экземпляров КМОП-вентилей, в которой хешем служит формула вентиля.
- 5) Классификации экземпляров вентиля, неизоморфных вследствие взаимозаменяемости стока и истока и несимметричности входов.
- 6) Генерация моделей в иерархическом SPICE-описании для каждого класса изоморфности хэш-таблицы для хранения экземпляров КМОП-вентилей
- 7) Генерация экземпляров вентиля в иерархическом SPICE-описании
- 8) Вставка всех оставшихся нераспознанными элементами исходного SPICE-описания.

Все шаги программы 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.

VI. ИСПЫТАНИЯ ПРОГРАММЫ

С целью испытания разработанной программы выполнялось два эксперимента. Задачей первого являлась проверка правильности распознавания библиотеки вентиляей. Во втором эксперименте проверялась правильность распознавания вентиляей в практических схемах.

Для первого эксперимента использовались схемы, полученные автоматическим синтезом в САПР. В этом случае известна технологическая библиотека. 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 вентилями. Верификация правильности декомпиляции осуществлялась программой netgen [14]. Часть декомпилированных схем была верифицирована Mentor Graphics Calibre pmLVS. Во всех случаях декомпилированная схема успешно проходила проверку LVS.

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 (биполярные транзисторы и RC элементы). Здесь наблюдалось покрытие схемы транзисторного уровня логическими вентилями 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 вентиляи находились. Верификация правильности декомпиляции осуществлялась программой netgen. Декомпилированные схемы успешно проходили проверку LVS.

На компьютере с четырёхядерным процессором Intel i5-4460 3.20GHz с оперативной памятью 16,0 ГБ декомпиляция схемы с 50 тыс. транзисторов выполнялась за время меньше минуты.

VII. ЗАКЛЮЧЕНИЕ

Структурный анализ цифровых схем в прошлом широко исследовался [8]. 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 докладе метод позволяет 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.

ЛИТЕРАТУРА

- [1] Logic Gate Recognition in Guardian LVS - Silvaco // URL: https://www.silvaco.com/content/appNotes/iccad/2-003_LogicGates.pdf (дата обращения: 4.1.2018)
- [2] Hunt V.D. Reengineering: Leveraging the Power of Integrated Product Development. Wiley, 1993. – 283 p.
- [3] Baker R.J. CMOS Circuit Design, Layout, and Simulation, Third Edition – Wiley-IEEE Press, 2010. – 1214 p.
- [4] Брагин К.Р., Гаврилов С.В., Каграманян Э.Р. Методы логико-временного анализа для характеристики заказных блоков цифровых КМОП-схем // III Всероссийская научно-техническая конференция "Проблемы разработки перспективных

микроэлектронных систем – 2008": сб. научн. тр. / под общей ред. А.Л. Стемпковского. – М.: ИПИМ РАН, 2008. – С. 92-97.

- [5] Myers D. J., Ivey P. A. A design style for VLSI CMOS // IEEE J. Solid-State Circuits, 1985, vol. SC-20, no. 3. – pp. 741-745.
- [6] Rabaey, J. M., Chandrakasan A., Nikolic B. Digital Integrated Circuits – Prentice Hall Press, 2008. – P. 702.
- [7] Krasilnikova L.V., Pottosin Yu.V. Partition of a transistor circuit into library modules from a given library // Proc. of the Second Intern. Conf. on Computer-Aided Design of Discrete Devices (CAD DD'97), Minsk, UIIP NASB. 1997. Vol. 1. P.94–97.
- [8] 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, V. 39. № 4. 2006. – P. 311–339.
- [9] Ракитин В.В. Интегральные схемы на комплементарных МОП-транзисторах: Учебное пособие. – М.: МФТИ, 2007. – 308 с.

- [10] Bushnell M., Agrawal Vishwani. Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI – Springer Science & Business Media, 2006. – 690 p.
- [11] Левитин А.В. Алгоритмы. Введение в разработку и анализ – М.: Вильямс, 2006. – 576 с.
- [12] Черемисинова Л.Д. Синтез и оптимизация комбинационных структур СБИС. – Мн: ОИПИ НАНБ, 2005. – 235 с.
- [13] Белоус А.И., Емельянов В.А., Турцевич А.С. Основы схемотехники микроэлектронных устройств. Москва: Техносфера, 2012. – 472 с.
- [14] Netgen version 1.5 Tutorial // URL: <http://opencircuitdesign.com/netgen/tutorial/tutorial.html> (access date: 16.02.2018)
- [15] Junttila T., Kaski P. Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs // 2007 Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX). 2007. P. 135–149.

Decompilation of Flat CMOS Circuits in SPICE Format

D.I. Chermisinov, L.D. Chermisinova

The United Institute of Informatics Problems of NAS of Belarus

cher@newman.bas-net.by, cld@newman.bas-net.by

Keywords — VLSI design, transistor-level netlist, CMOS circuits redesign, SPICE format, subcircuit recognition, DNF algebraic factorization, canonization of colored undirected graphs.

ABSTRACT

A problem of transformation of a flat CMOS circuit presented in SPICE format to a hierarchical network of CMOS gates in the same format is considered. The problem arises in VLSI layout verification as well as in the circuit reengineering. The paper presents a method for recognizing the CMOS gate subcircuits. The method is implemented as a C++ program. It recognizes subcircuits described by the same logical functions, but not isomorphic at the transistor level, as different. This ensures that the source and decompiled circuits are isomorphic.

Similarly the programming, the conversion process of the flat CMOS circuit presented in SPICE format to a hierarchical circuit in the same format is referred as decompilation. Automatic recognition of a high-level structure from the transistor level netlist of a circuit is important for many tasks in VLSI circuit design. In the SPICE format, electrical circuits consist of elements that are connected to each other by nets, and a colored undirected bipartite graph is the natural formal model for describing the circuit. One part is made up of the terminals of the elements and the ports of the circuit, and the other is the connection between the terminals, i.e. nets.

The decompilation consists in factoring the initial flat netlist. If the attempt to recognize a factor as subcircuit fails, then the decompilation problem is not tractable. Every

group of transistors can be refactored as subcircuit. We select as subcircuit the group of the transistors, for which it is possible to define a logical function.

The paper presents a computer program for to extract the hierarchy of a large-scale digital CMOS circuit from its transistor-level netlist automatically. The program performs the following sequence of steps: 1) analyzing the original SPICE description and constructing a hash table to store the bipartite colored graph of the circuit; 2) dividing the bipartite colored graph into a net of channel-connected-components (CCC); 3) recognizing the correct CCCs, finding the appropriate logical functions for them in the form of an algebraic formula and constructing the hash table to store instances of CMOS gates;; 4) generating the hierarchical SPICE description.

The structural analysis of digital circuits was broadly covered with research in the past. For the recognition of a subcircuits, two classes of approaches exist: library based approaches and algorithmic approaches. Library based approaches use the predefined library circuits as templates for recognition of subcircuits and their recognition ability is limited to subcircuits contained in the library. Algorithmic approaches recognize subcircuits for which there are a possibility to compute the logic function of the gate. Our method allowing recognizing subcircuits with the same logic functions that are not isomorphic at the transistor level. Two kinds of experiments with the developed program have been performed. The purpose was to check the correctness of recognizing subcircuits in practical cases with and without predefined gate library.

REFERENCES

- [1] Logic Gate Recognition in Guardian LVS - Silvaco // URL: https://www.silvaco.com/content/appNotes/iccad/2-003_LogicGates.pdf (access date: 4.1.2018)
- [2] Hunt V.D. Reengineering: Leveraging the Power of Integrated Product Development. Wiley, 1993. – 283 p.
- [3] Baker R.J. CMOS Circuit Design, Layout, and Simulation, Third Edition – Wiley-IEEE Press, 2010. – 1214 p.
- [4] Bragin K.R., Gavrilov S.V., Kagramanyan E.H.R. Metody logiko-vremennogo analiza dlya kharakterizatsii zakaznykh blokov tsifrovyykh KMOP-skhem // III Vserossiyskaya nauchno- tekhnicheskaya konferentsiya "Problemy razrabotki perspektivnykh mikroelektronnykh sistem – 2008": sb. nauchn. tr. / pod obshhej red. A.L. Stempkovskogo. – M.: IPPM RAN, 2008. – S. 92-97.
- [5] Myers D. J., Ivey P. A. A design style for VLSI CMOS"// IEEE J. Solid-State Circuits, 1985, vol. SC-20, no. 3. – pp. 741-745.
- [6] Rabaey, J. M., Chandrakasan A., Nikolic B. Digital Integrated Circuits – Prentice Hall Press, 2008. – P. 702.
- [7] Krasilnikova L.V., Pottosin Yu.V. Partition of a transistor circuit into library modules from a given library // Proc. of the Second Intern. Conf. on Computer-Aided Design of Discrete Devices (CAD DD'97), Minsk, UIIP NASB. 1997. Vol. 1. P.94-97.
- [8] 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, V. 39. N 4. 2006. P. 311-339.
- [9] Rakitin V.V. Integralnyye skhemy na komplementarnykh MOP-tranzistorakh: Uchebnoye posobiye. (Integrated circuits on complementary MOSFETs: Textbook) – M.: MFTI. 2007. – 308 p.
- [10] Bushnell M., Agrawal Vishwani. Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI – Springer Science & Business Media, 2006. – 690 p.
- [11] Levitin A.V. Introduction to the Design and Analysis of Algorithms – Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2002. 593 p.
- [12] Cheremisinova L.D. Sintez i optimizatsiya kombiniatsionnykh struktur SBIS (Synthesis and optimization of combinational structures of VLSI) – Minsk: OIPI NANB. 2005. 235 p.
- [13] Belous A.I., Emelianov V.A., Turtsevich A.S. Osnovy skhemotekhniki mikroelektronnykh ustroystv (Fundamentals of circuit technology for microelectronic devices) – Moskva: Tekhnosfera. 2012. 472 p.
- [14] Netgen version 1.5 Tutorial – URL: <http://opencircuitdesign.com/netgen/tutorial/tutorial.html> (access date: 16.02.2018)
- [15] Junttila T., Kaski P. Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs // 2007 Proc. of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX). 2007. P. 135–149.