

УДК 50.51.17

Д.И. Черемисинов

ОТОБРАЖЕНИЕ ЛОГИЧЕСКИХ СЕТЕЙ В ЗАДАННЫЙ ТЕХНОЛОГИЧЕСКИЙ БАЗИС

Рассматривается задача синтеза многоуровневых логических сетей в базе библиотечных элементов КМОП СБИС. Приводится описание структуры программы для решения этой задачи и форматов исходных данных. Обсуждается влияние формы исходных данных на результат решения. Программа отображения логических сетей в заданный технологический базис включена как проектная операция в программный комплекс энергосберегающего логического синтеза, предназначенного для автоматизации проектирования многоуровневых логических схем из библиотечных элементов заказных сверхбольших интегральных схем (СБИС), выполненных по КМОП-технологии.

Введение

Операция логического синтеза по заданному описанию поведения синтезируемой схемы в виде системы булевых функций строит структурное представление в виде логической сети из элементов заданной библиотеки [1]. Синтез представляет собой задачу нахождения структурной модели дискретного устройства по заданной функциональной модели.

Отображение логической сети в технологический базис (technology mapping) – это реализация комбинационной схемы с использованием элементов библиотеки, выполненных по конкретной (предоставляемой изготовителем микросхем) технологии СБИС. Библиотека, как правило, состоит из элементов различных размеров и быстродействия, реализующих примитивные булевы (AND и OR) и более сложные (исключающее ИЛИ, мультиплексор) функции. Так, исходное представление булевой функции (системы булевых функций) отображается в оптимальную схему, максимально использующую элементы библиотеки. Цель состоит в том, чтобы найти новую схему с той же функциональностью, но меньшей площади, задержки, потребляемой мощности и т. д., построенную из элементов библиотеки, а также в том, чтобы новая схема после размещения элементов и трассировки связей могла быть реализована в качестве аппаратного устройства высокого качества. Операция является неотъемлемой составляющей любого автоматизированного процесса проектирования СБИС.

Элементы пространства решений задачи технологического отображения являются некоторыми представлениями булевых функций. Установлено [2], что не существует метода построения пространства решений, содержащего все возможные представления заданной булевой функции. Все распространенные методы решения задачи синтеза используют пространство решений, построенное на основе исходного представления булевой функции. Таким образом, структура исходного представления булевой функции диктует в значительной степени структуру построенной схемы. Это и есть структурная чувствительность (structural bias [3]) инструментов синтеза, она неустранима. Ни один из известных методов синтеза не способен открыть схемы новой нестандартной структуры.

При синтезе строится многоуровневая логическая схема для данной булевой функции. Первоначально функция задается в виде таблицы истинности, ДНФ, системы уравнений или, например, в виде схемы сомнительного качества. Для нахождения лучшего решения желательно выполнять синтез несколько раз, используя различные эквивалентные формы задания исходной булевой функции [3]. Одним из способов такого синтеза является варьирование нескольких форматов задания исходной булевой функции.

В работе описывается программа технологического отображения, которая детально использует метод синтеза, основанный на структурном покрытии многоуровневой логической сети из вентилях элементами КМОП-библиотеки [1]. Эта программа отличается от известных инструментов синтеза тем, что допускает возможность применения нескольких форматов задания булевых функций. Из-за структурной чувствительности выбор формата одного из эквивалентных представлений исходной булевой функции влияет на результат синтеза. Предметом

настоящей работы являются представления булевой функции, используемые в операции технологического отображения, а не ее алгоритм. Основное внимание уделяется описанию подготовки исходных данных для программы и влиянию их формы на результат решения.

1. Структурная чувствительность технологического отображения

Элементом называется некоторое простейшее в каком-то смысле техническое устройство. Логический элемент – это такое устройство, которое описывается совокупностью двоичных физических величин, связанных функциональной зависимостью, представимой булевой функцией или системой булевых функций. Реализуемая элементом система булевых функций показывает его функциональные свойства и может рассматриваться как логическая или функциональная модель элемента.

Логической сетью или схемой называют такую модель дискретного устройства, которая отражает его внутреннее строение с точностью до функций, выполняемых элементами [2]. Сеть образуется путем соединения элементов, т. е. указываются линии связи, по которым выходные сигналы одних элементов поступают на входные полюсы других элементов. Задать логическую сеть – означает указать ее входные и выходные полюсы, структуру связей между элементами и булевы функции, реализуемые этими элементами.

Любую логическую сеть можно полностью описать в алгебраической форме системой уравнений непосредственных связей, задающих функционирование ее элементов. Все полюсы схемы обозначаются некоторыми внутренними переменными. При этом отождествляемым полюсам (соединяемым между собой) приписывается одна и та же переменная. Система уравнений непосредственных связей является формой представления исходной булевой функции в виде суперпозиции функций, заданных выражениями внутри скобок, а логическая сеть в виде иерархической структуры – гибридным структурно-поведенческим представлением (рис. 1). Алгебраическое представление в виде ДНФ, матричное и табличное задание булевой функции имеют структуру двухуровневой логической сети.

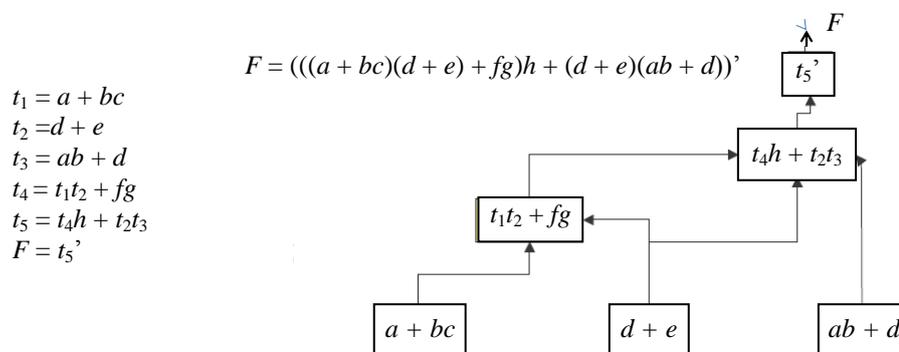


Рис. 1. Логическая сеть и ее алгебраические представления

В теории графов сетью называется помеченный ориентированный граф $G(V, E)$. В логической сети граф ациклический и множество вершин V разбито на три подмножества: первичных входов, первичных выходов, внутренних вершин. Каждая вершина помечена входной или выходной переменной либо функцией, реализуемой элементом. Для каждой вершины указаны два множества вершин, задающие отношение E : множество элементов, связанных со входами ($fanin$), и множество элементов, связанных с выходом ($fanout$). В каждой вершине множество вершин $fanin$ ($fanout$) задает входящие (исходящие) дуги соответственно. Вершину i связывает с вершиной j ребро, если функция в узле j зависит явно от переменной i . Множество дуг E графа $G(V, E)$ является объединением множеств $fanin$ и $fanout$ всех вершин графа.

Задача синтеза ставится следующим образом [2–5]. Дано множество логических элементов L , называемое библиотекой (пусть библиотека элементов – это совокупность логических сетей, каждая из которых представляет отдельный логический элемент), и логическая сеть G

(пусть M – это множество всех логических сетей, которые можно создать на основе библиотеки L , функционально эквивалентных G). Элементы M называются отображениями исходной сети G . Цель синтеза состоит в том, чтобы найти отображение $m \in M$, оптимальное для некоторого применения сети в качестве схемы дискретного устройства. В приведенной постановке проблема синтеза представляет алгоритмически неразрешимую задачу, так как невозможно перечислить явно или неявно все элементы M . Существенное упрощение проблемы синтеза можно получить, ограничивая множество M сетями, которые структурно подобны исходному заданию G . С этим ограничением задача синтеза алгоритмически разрешима, а метод синтеза называется комбинаторным структурным подходом.

Основная идея алгоритма комбинаторного структурного синтеза проста. Предполагается, что заданная логическая сеть G уже имеет «хорошую» структуру [1, 2], т. е. оптимизирована с учетом экономии энергопотребления. Исходная сеть преобразуется в сеть, у которой все элементы с функциональными метками принадлежат одной функции (обычно двухвходовой НЕ И). Логические сети элементов библиотеки преобразуются аналогично. Этап подготовки называется детализацией сети. Отображение m строится в процессе локальной переписи исходного G . При этом в G находится подсеть N с одним выходом, которая функционально эквивалентна некоторому элементу g из L , затем N заменяется в G элементом g . В процессе покрытия каждая функция из исходной системы булевых функций представлена двумя логическими сетями: G и G^- . Шаг покрытия состоит из трансформации сети G в сеть G^- .

Представлением графа логической сети в памяти программы является коллекция объектов, сопоставляемых с вершинами графа. Объект вершины кроме пометки содержит и уникальный цифровой идентификатор, а также две коллекции, задающие отношение E (fanin и fanout). Коллекция объектов графа представляет собой структуру хранения данных идентификаторов других вершин. В вершинах первичных входов пуста коллекция нагрузки, а в вершинах первичных выходов – коллекция питания.

После покрытия выполняется оптимизация нагрузки элементов схемы (fanout optimisation). Логические элементы обладают определенной нагрузочной способностью – числом входов других элементов, которые может питать выход этого элемента. Алгоритм покрытия строит схему, не учитывая ограниченную нагрузочную способность элементов и ее входов. В задаче оптимизации нагрузки элементов схемы просматриваются пути распространения сигнала с выхода элемента на входы, которые этот выход питает. При необходимости в цепь распространения сигнала выхода включаются инверторы так, чтобы не нарушались ограничения нагрузочной способности [3].

В настоящее время имеется ряд коммерческих программ в составе промышленных САПР и академические программы для решения задачи технологического отображения. Разница в качестве схем, построенных известными программами, называется в [6] «шумом алгоритмов синтеза». В работе [7] исследуется качество известных алгоритмов технологического отображения, в частности показывается, что шум алгоритмов синтеза приводит к колебаниям количества элементов в схеме примерно в два раза. При этом систематического преимущества какого-либо алгоритма не выявлено. Влияние структурной чувствительности на качество по косвенным оценкам в некоторых случаях может превышать шум алгоритмов синтеза. В [5] неполный перебор эквивалентных исходных представлений назван повторным синтезом и показано, что влияние структурной чувствительности на результаты синтеза с помощью синтезаторов промышленных САПР – того же порядка, что и шум алгоритмов синтеза. Для некоторых классов схем выбор подходящего исходного представления приводит к улучшению параметров схемы на порядок [6]. Различные схемы получаются даже для структурно изоморфных описаний. Несмотря на высокое качество инструментов автоматического синтеза, влияние квалификации разработчика на результаты синтеза остается значительным, так как качество схемы зависит от выбора подходящей формы представления исходных данных.

Структурная чувствительность инструментов синтеза логических схем давно и хорошо известна разработчикам инструментов синтеза. Пользователи академических программ синтеза учитывают эту особенность, так как она выступает очевидным логическим следствием поста-

новки задачи технологического отображения. Руководящие материалы для пользователей промышленных САПР не содержат описаний используемых алгоритмов, и их пользователи склонны преувеличивать возможности инструментов.

2. Унифицированное структурно-функциональное представление схемы

Структурная модель (netlist) представляет схему в терминах взаимосвязей составляющих компонентов и не содержит ничего, кроме компонентов и цепей с их атрибутами. В своем основном значении структурное описание есть описание внутреннего устройства чего-либо. Структурное описание устройства изнутри системы характеризует ее организацию через элементы (части) и их взаимозависимости. Функциональное описание системы – это взгляд внешнего наблюдателя, определяющий его взаимодействие с окружающей средой через соотношения между входами и выходами. Структурное описание выполнимо (может моделироваться), если известны функциональные модели компонентов; функциональное описание – это описание законов функционирования, эволюции системы, алгоритмов ее поведения и моделируемо само по себе по определению [8]. Таким образом, описания поведения при помощи понятий, структуры или функции содержат фундаментальные различия.

Для представления структурных моделей в современных САПР используются специальные текстовые языки описания данных, называемые форматами структурных описаний. Формальной моделью структурных описаний являются двудольные графы [9].

В качестве примера на рис. 2 изображен фрагмент логической сети, а на рис. 3 – модель этого фрагмента в виде неориентированного двудольного графа. В данном фрагменте XORG – логический элемент сложения по модулю 2, GYMUX – трехвходовый элемент, DYMUX – двухвходовый элемент.

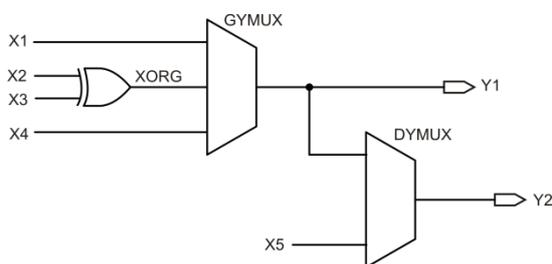


Рис. 2. Логическая сеть

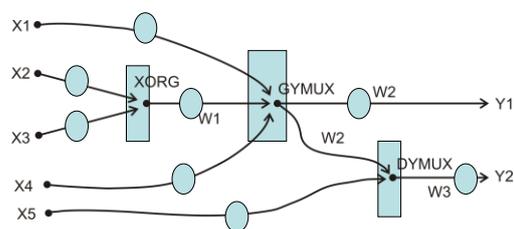


Рис. 3. Неориентированный двудольный граф соответствующей логической сети

Вершины, являющиеся выводами элемента (рис. 3), сгруппированы в прямоугольники (сами прямоугольники не являются элементами графа). Обозначения выводов предваряются именем элемента, показанным в фигуре элемента. Вершины цепей показаны кружками, обозначения (имена) выводов и некоторых цепей для упрощения опущены.

Моделью унифицированного структурно-функционального представления схемы выступает двудольный граф, одной долей которого являются порты (выводы) экземпляров элементов и порты самого устройства, а другой – цепи, соединяющие порты. Отношение графа задано в виде двух структур: списка экземпляров (instance based netlist) и списка цепей (net-based netlist).

3. Конвертация представлений схемы

Представлением графа логической сети является коллекция объектов, сопоставляемых с вершинами графа (рис. 4). Кроме пометки, объект вершины содержит уникальный цифровой идентификатор и две коллекции идентификаторов других вершин: нагрузку (fanin) и питание (fanout). Коллекция объектов графа представляет собой структуру хранения данных – двуправленный список. В вершинах первичных входов пуста коллекция нагрузки, а в вершинах первичных выходов – коллекция питания.

Вершина графа
логической сети

Рис. 4. Структура представления логической сети

Унифицированное структурно-функциональное описание в виде диаграммы, на которой отображены интерфейсы классов, т. е. связи объектов этих классов, показано на рис. 5.

Рис. 5. Диаграмма классов структурно-функционального описания цифрового устройства

Задача конвертации представлений схем состоит в трансформации представления унифицированного структурно-функционального описания в представление логической сети и наоборот. Первое преобразование выполняется при обработке исходного формата описания схемы. Обратная трансформация (представление логической сети в представление унифицированного структурно-функционального описания) выполняется по завершении операции покрытия логической сети (рис. 6).

4. Библиотека элементов синтеза

Библиотека элементов синтеза для программы задается в формате genlib [4]. Описание каждого библиотечного элемента содержит его булеву функцию, заданную в алгебраической форме в виде текстовой строки, и стоимость элемента, влияющую на выбор элемента при

покрытии. Стоимостью библиотечного элемента обычно является число транзисторов его микросхемы (или площадь топологических базовых ячеек, размеры которых зависят от технологии изготовления КМОП-элементов).

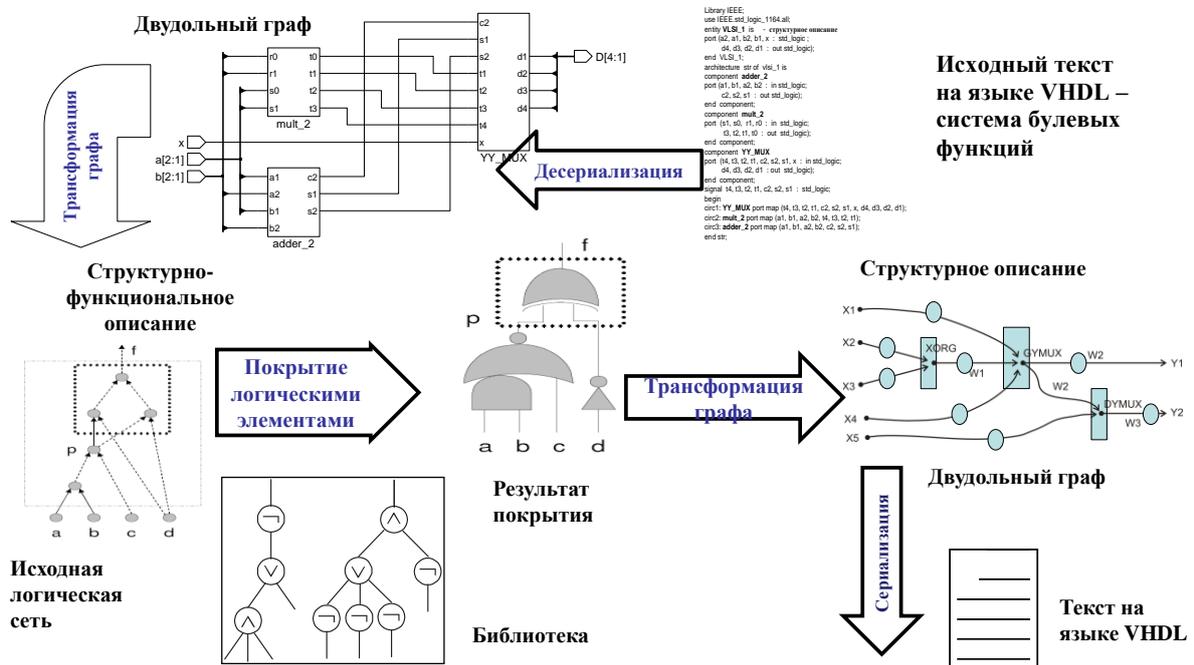


Рис. 6. Цепочка преобразований описания цифрового устройства

5. Структура программы технологического отображения

Входными данными для программы служит система ДНФ, заданная в одном из допустимых форматов. Анализаторы допустимых форматов строят унифицированное структурно-функциональное представление схемы [9]. Выбор анализатора происходит автоматически по расширению имени файла с исходным текстом (рис. 7).

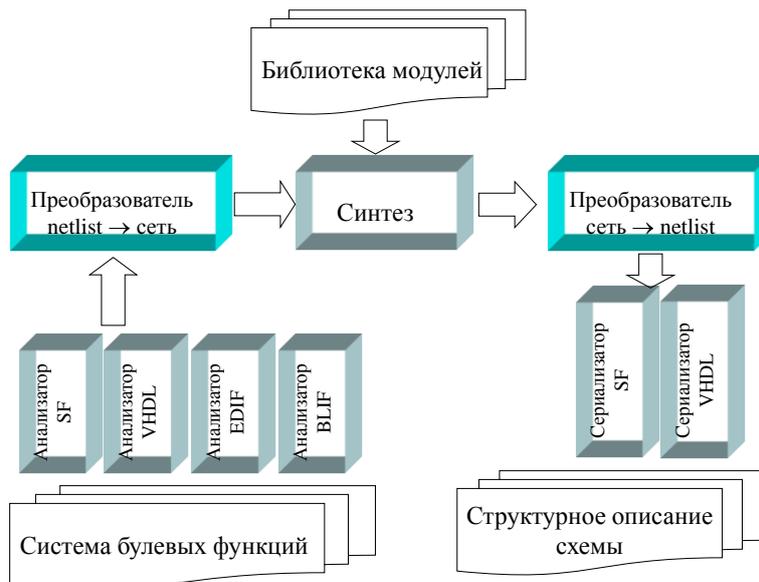


Рис. 7. Данные и операции программы технологического отображения схем из вентилей в заданный библиотечный базис

Затем это представление преобразуется в граф логической сети (см. разд. 1). Подграфу двудольного графа, составленному из вершин, которые являются портами элемента, в логической сети соответствует отдельная вершина. Анализатор формата genlib строится коллекция сетей элементов библиотеки так, что функцией каждой внутренней вершины является либо НЕ И2, либо инверсия.

Основная операция программы преобразует многоуровневую схему из элементов технологически независимого базиса (называемую исходной) в многоуровневую схему (называемую объектной) из библиотечных элементов путем локальных замен подсхем исходной сети. Предварительно (до покрытия) исходная логическая сеть подвергается детализации. Процедура покрытия содержит проверку функциональной эквивалентности исходной и объектной логических сетей.

По завершении операции покрытия логическая сеть, функционально эквивалентная исходной системе ДНФ (результат работы этапа покрытия), преобразуется в унифицированное структурно-функциональное представление схемы, которое сериализуется в требуемый формат. Текст, являющийся результатом работы программы, строится в формате SF (CONNECT) или в форме VHDL-описания структурного стиля.

6. Форматы представления исходной системы булевых функций

Допустимыми входными форматами исходной системы булевых функций являются: SF (sf, SF), VHDL (vhdl, VHDL, vhd, VHD, v), EDIF (edf, EDF, edif, EDIF), BLIF (blf, BLF, blif, BLIF), PLA (pla, PLA). (В скобках указаны расширения имени файла.)

Обрабатываются описания в формате SF, содержащие и структурные, и функциональные описания схем, в том числе файлы с несколькими описаниями различных типов. Система ДНФ задается на языке SF структурно-функциональным описанием в виде схемы из подсхем (блоков) или на функциональном уровне в матричной либо алгебраической форме. Структурное описание головной подсхемы должно быть первым в тексте и формата CONNECT. Функциональные описания листовых блоков представляют собой либо логические уравнения – скобочные формы в базисе операций И, ИЛИ, НЕ (в формате LOG), либо матричную форму представления системы ДНФ булевых функций (в формате SDF). Если головной блок описан на функциональном уровне, то вся схема состоит из единственного блока.

В графе логической сети полностью сохранится структура иерархии описания блоков. В сети, построенной по формату LOG, сохранится структура уравнений, но будет потеряна структура вхождения скобок, т. е. описание уравнения будет приведено в форму ДНФ.

Анализатор VHDL, используемый в программе, построен по грамматике VHDL «Стандарт VHDL–93». Все конструкции языка допустимы, но программой обрабатывается только подмножество языка. Обрабатываемое подмножество позволяет описывать только цифровые синхронные схемы. Исходное описание должно соответствовать структурному стилю, допускается включение операторов сигнального присваивания с тривиальной правой частью. Описания всех элементов структуры должны содержаться в тексте в виде описаний потокового стиля, состоящих из операторов сигнального присваивания. Обрабатываемые описания должны принадлежать синтезируемому подмножеству VHDL по типам сигналов.

Структурное описание головной подсхемы должно быть первым в тексте. В графе логической сети полностью сохранится структура иерархии описания блоков. В описаниях потокового стиля в уравнениях сохранится структура вхождения скобок, т. е. описание уравнения будет приведено в форму логической сети из элементов И, ИЛИ, НЕ с двумя входами.

Анализатор EDIF версии 2.0.0, используемый в программе, построен трансляцией грамматики EDIF, свободно доступной через веб-сервер Манчестерского университета в Англии. Так как EDIF является языком структурных описаний, то программа может обработать их, только если по названиям блоков можно установить булеву функцию элемента.

В структурном описании должны использоваться следующие наименования примитивных элементов: T0, T1 – константы 0, 1; Buf, Inv – буфер, инвертор; And, Xor, Or, Nor, Xnor, Nand – логические элементы. На число аргументов функции логических элементов ограничений нет. В графе логической сети полностью сохранится структура соединений примитивных элементов.

Формат BLIF (Berkeley Logic Interchange Format) предназначен для описания иерархических схем на логическом уровне в текстовой форме. Схема – это произвольная комбинационная или последовательная логическая сеть. Для каждого комбинационного элемента задана булева функция, описывающая вычисление значения единственного вывода этого элемента [4]. Программа не может обрабатывать описания, содержащие последовательностные элементы. Не обрабатываются также и иерархические описания.

Формат PLA является подмножеством формата BLIF и служит средством задания систем булевых функций в форме ДНФ.

Заключение

Программа отображения логических сетей в заданный технологический базис включена как проектная операция в программный комплекс энергосберегающего логического синтеза [10], предназначенного для автоматизации проектирования многоуровневых логических схем из библиотечных элементов заказных СБИС, выполненных по КМОП-технологии. Система энергосберегающего логического синтеза имеет развитые средства технологически независимой оптимизации, что позволяет использовать ее при синтезе логических схем не только в базисе КМОП-элементов, но и в других технологических базисах. Программный комплекс данного синтеза предназначен для проектирования схем комбинационной логики, имеющих сотни входных-выходных переменных и тысячи элементов. Исходное функциональное описание проектируемой логической схемы в переводе на эквивалентное представление в виде системы булевых функций в ДНФ может иметь до нескольких десятков переменных и функций и нескольких тысяч дизъюнкций.

Список литературы

1. Черемисинов, Д.И. Синтез комбинационных схем в базисе библиотечных элементов КМОП СБИС с учетом энергопотребления / Д.И. Черемисинов, Л.Д. Черемисинова // Информатика. – 2013. – № 4(40). – С. 91–102.
2. Закревский, А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.
3. Reducing Structural Bias in Technology Mapping / S. Chatterjee [et al.] // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2006. – Vol. 25, no. 12. – P. 2894–2903.
4. Sis: A system for sequential circuit synthesis : technical report UCB/ERL M92/41, EECS Department / E.M. Sentovich [et al.] [Electronic resource]. – University of California, Berkeley, 1992. – Mode of access : <https://www.eecs.berkeley.edu/Pubs/TechRpts/1992/2010.html>. – Date of access : 11.11.2016.
5. Бибило, П.Н. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний / П.Н. Бибило, В.И. Романов. – Минск : Беларуская навука, 2011. – 279 с.
6. Fišer, P. Sources of Bias in EDA Tools and Its Influence / P. Fišer, J. Schmidt, J. Balcárek // Proc. of the 2014 IEEE 17th Intern. Symp. on Design and Diagnostics of Electronic Circuits & Systems. – Piscataway : IEEE, 2014. – P. 258–261.
7. Бибило, П.Н. Покрытие булевой сети библиотечными элементами / П.Н. Бибило, В.Г. Лицкевич // Управляющие системы и машины. – 1999. – № 6. – С. 16–24.
8. Перегудов, Ф.П. Введение в системный анализ / Ф.П. Перегудов, Ф.П. Тарасенко. – М. : Высшая школа, 1989. – 360 с.
9. Черемисинов, Д.И. Обработка графов в программе перепроектирования FPGA / Д.И. Черемисинов // Танаевские чтения : доклады Шестой Междунар. конф. – Минск : ОИПИ НАН Беларуси, 2014. – С. 151–155.

10. Автоматизация логического синтеза КМОП-схем с пониженным энергопотреблением / П.Н. Бибило [и др.] // Программная инженерия. – 2013. – № 8. – С. 35–41.

Поступила 22.12.2016

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: cher@newman.bas-net.by*

D.I. Cheremisinov

TECHNOLOGY MAPPING TOOL FOR VLSI CAD

Technology mapping program implements a sequential circuit using the gates of a particular technology library. It is an integral component of any automated VLSI circuit design flow. The structure of the program for solving the technology mapping problem and formats of the source and result data are presented. Models of intermediate representations of the sequential circuit and their conversions are described. Technology mapping is a stage of logic synthesis and it is viewed as the transformation of a functional (i.e., algebraic) circuit specification into a gate (i.e., netlist) specification. The program is included as project operations in the VLSI CAD system for energy-saving logical synthesis developed in the United Institute of Informatics Problems of NAS of Belarus.