

Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний

П.Н. Бибило, В.И. Романов

Объединенный институт проблем информатики НАН Беларуси,

bibilo@newman.bas-net.by, rom@newman.bas-net.by

Аннотация — Описывается система FLC-2, предназначенная для оптимизации иерархически организованных функциональных и структурных описаний логических схем, представленных на языке SF, являющемся внутренним языком системы. Для связи с другими САПР (системами автоматизированного проектирования) цифровых устройств в FLC-2 имеются средства конвертации SF-описаний в VHDL-описания (и обратно). Логическая оптимизация осуществляется на основе применения комбинированных методов, реализующих различные базовые оптимизационные приемы, – минимизацию систем булевых функций в классе ДНФ (дизъюнктивных нормальных форм), оптимизацию многоуровневых представлений систем булевых функций на основе булевых сетей и BDD-представлений, поляризации и декомпозиции матричных и скобочных форм, выделения подсистем функций, связанных по областям определения. В системе FLC-2 реализован продукционный подход для управления последовательностями проектных процедур, предназначенных для их использования на этапе логического проектирования сложных функциональных блоков заказных цифровых СБИС (сверхбольших интегральных схем). Эксперименты показали целесообразность использования программ системы FLC-2 для предварительной оптимизации проектов цифровых устройств, реализуемых в промышленных САПР заказных цифровых СБИС и FPGA.

Ключевые слова — система булевых функций, дизъюнктивная нормальная форма (ДНФ), Binary Decision Diagram (BDD), булева сеть, синтез логической схемы, VHDL, СБИС, база знаний.

I. ВВЕДЕНИЕ

Система FLC логической оптимизации функционально-структурных описаний цифровых устройств была описана в работе [1]. В качестве исходных данных для этой системы выступает функциональное либо иерархическое функционально-структурное описание проектируемой логической схемы. Результатом является оптимизированное описание, на основе которого можно получить логическую схему меньшей сложности и большего быстродействия. В рамках системы FLC реализована интеллектуальная поддержка проектирования, основанная на применении продукционно-фреймовой модели представления знаний [1]. Полезные качества, обеспечиваемые применением данной

модели, оказались весьма продуктивными и это привело к сохранению общей архитектуры в некоторых других системах автоматизированного проектирования, например в [2, 3].

Вместе с тем, к настоящему времени появились новые методы логической оптимизации, существенно расширились возможности использования инструментария разработки, стили применяемого человеко-машинного интерфейса, базового функционального обеспечения. Это привело к тому, что базовый вариант программного обеспечения системы FLC подвергся существенной модернизации, действующий вариант которой получил название FLC-2. В данной работе приводится перечень основных модифицированных элементов системы.

II. ПРЕДСТАВЛЕНИЕ ДАННЫХ

В качестве языка описания логических схем в FLC-2 используется язык SF [1], который ориентирован на иерархическое описание логической схемы, при этом уровень вложенности компонентов не ограничивается. Схема на языке SF определяется последовательностью функционально-структурных описаний подсхем (блоков), из которых состоит схема. Иерархия описания схемы представляется в виде дерева, вершине дерева соответствует отдельный блок. Любой не листовый блок, соответствующий узлу дерева иерархии, выражается заданием связей входящих в него подсхем. Функциональные описания листовых блоков дерева иерархии описания представляют собой либо логические уравнения - скобочные выражения в булевом базисе И, ИЛИ, НЕ (LOG-формат), либо матричные формы представления систем булевых функций в виде ДНФ (SDF-формат). Система ДНФ булевых функций задается парой матриц: строки троичной матрицы T задают элементарные конъюнкции, входящие в систему ДНФ, в булевой матрице В единицы задают вхождения конъюнкций в ДНФ функций. В литературе [4] системы ДНФ называют также *двухуровневыми* И-ИЛИ функциональными описаниями, в отличие от многоуровневых скобочных алгебраических описаний в базисе И, ИЛИ, НЕ. При задании логических уравнений в LOG-формате символы операций имеют следующую интерпретацию: через * обозначено логическое И (конъюнкция), через + обозначено логическое ИЛИ

(дизъюнкция), через \wedge обозначено логическое НЕ (отрицание). Если головной блок описан на функциональном уровне, то в этом случае весь проект представляет собой один листовой блок.

Возможности использования матричных форм и иерархических описаний являются важными отличительными особенностями языка SF от RTL-описаний, получаемых командой uppar в известном синтезаторе LeonardoSpectrum [5] и используемых при перепроектировании логических схем в этом либо других промышленных синтезаторах.

III. ПРОГРАММНО РЕАЛИЗОВАННЫЕ МЕТОДЫ И АЛГОРИТМЫ ЛОГИЧЕСКОЙ ОПТИМИЗАЦИИ

Синтез логических схем, реализующих функциональные комбинационные блоки цифровых заказных СБИС, остается одной из важных задач автоматизированного проектирования, так как возрастает размерность задач проектирования и, соответственно, растёт время выполнения этапов синтеза. Особенно трудоёмкой является *технологически независимая* оптимизация, являющаяся первым этапом синтеза схемы. Суть второго этапа синтеза – технологического отображения (technology mapping) заключается в «покрытии» оптимизированных логических представлений функций библиотечными логическими элементами [4]. Основные характеристики логической схемы, такие как площадь, часто выражаемая в числе транзисторов, быстродействие, энергопотребление зависят во многом от эффективности выполнения первого этапа – этапа глобальной логической оптимизации. При синтезе логических схем в библиотеках проектирования заказных СБИС реализуемые системы полностью определенных булевых функций чаще всего имеют алгебраические представления, основной оценкой сложности таких представлений является общее число литералов булевых переменных [4]. Кроме критерия «число литералов», применяемого при оптимизации листовых блоков иерархически описанной проектируемой схемы, при декомпозиции и разбиении сложных блоков используются критерии «число блоков», «число входных переменных блока», «число выходных переменных блока», «число связей между блоками». Перечислим программно реализованные в FLC-2 методы и алгоритмы логической оптимизации.

Логическая минимизация в классе ДНФ реализует различные методы и алгоритмы [6 - 9] совместной или отдельной минимизации по различным критериям (числу элементарных конъюнкций, суммарному числу литералов в конъюнкциях) как без учета поляризации (инверсирования) отдельных компонентных функций системы, так и с учетом поляризации.

Минимизация совместных либо отдельных BDD-представлений [10] систем булевых функций в классе многоуровневых представлений осуществляется на основе разложения Шеннона. Разложением Шеннона булевой функции $f(\mathbf{x}) = f(x_1, \dots, x_n)$ по переменной x_i называется представление $f(\mathbf{x})$ в виде

$$f(\mathbf{x}) = \bar{x}_i f_0 \vee x_i f_1.$$

Каждая из подфункций (кофакторов, cofactors) $f_0 = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f_1 = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ может быть разложена по одной из переменных из множества $\{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\}$. Процесс разложения кофакторов заканчивается, когда все n переменных будут использованы для разложения. На каждом шаге разложения выполняется поиск одинаковых и взаимно инверсных кофакторов, из такого множества оставляется один. Нахождение не только одинаковых, но и взаимно инверсных кофакторов обеспечивает более высокое качество логической оптимизации по сравнению с классическими сокращенными упорядоченными BDD. Соответствующий алгоритм подробно описан в [11]; при выборе очередной переменной разложения Шеннона алгоритм использует следующую эвристику: для очередной выбираемой переменной число различных взаимно инверсных кофакторов разложения Шеннона должно быть минимально. Критерием оптимизации является минимум числа формул разложения Шеннона для задания системы исходных функций, другими словами – минимальное число вершин в графе BDD, представляющем взаимосвязанные формулы разложения Шеннона.

Минимизация булевых сетей [12] осуществляется подобным образом: на основе разложения Шеннона выполняется поиск вершин сети, реализующих одинаковые булевы выражения (функции), а также поиск вершин, реализующих взаимно инверсные булевы выражения. При этом сложность булевой сети оценивается суммарным числом двухвходовых логических операторов, а инверсии переменных при подсчете сложности представления булевой сети не принимаются во внимание.

Выделение «связанных» подсистем булевых функций. Одним из направлений логической оптимизации многоуровневых представлений систем булевых функций являются методы, основанные на выделении подсистем функций, которые имеют одинаковые части в областях определения функций выделенных подсистем. Такие подсистемы называются связанными. Связанность функций приводит к появлению большого числа одинаковых структурных частей (конъюнкций, алгебраических выражений, подфункций и др.) в оптимизированных формах представления функций, по которым строятся в дальнейшем комбинационные логические схемы. Программно реализованы алгоритмы выделения подсистем связанных функций из заданий функций таблицами истинности либо из BDD-представления системы булевых функций на основе введенных численных оценок связанности BDD-представлений функций [13].

Выделение подсистем для дизъюнктивно-конъюнктивного разложения основывается также на связанности областей единичных либо нулевых значений компонентных функций системы и выполняется при задании функций таблицами истинности [14].

Факторизация скобочных представлений систем булевых функций (алгебраическая декомпозиция) сводится к поиску и выделению общих частей логических выражений: конъюнкций и дизъюнкций системы ДНФ как алгебраических выражений и к вынесению за скобки переменных. Цель этапа алгебраической декомпозиции заключается в построении такого варианта представления схемы, который мог бы служить хорошей отправной точкой для этапа технологического отображения в базис библиотечных элементов, выполненных по КМОП-технологии [15].

Синтез комбинационных схем в библиотеке КМОП-элементов реализует один из методов технологического отображения и подробно описан в [16].

Декомпозиция матричных форм систем ДНФ выполняется на основе метода «тождественных отображений в пространстве промежуточных переменных» [6] и на основе аппарата BDD [10]. При такой декомпозиции находится разбиение множества входных переменных на два непересекающиеся подмножества, одно из которых перекодируется новыми булевыми переменными. Реализованы также методы декомпозиции систем ДНФ в одноуровневую сеть ДНФ (если это возможно), каждая ДНФ которой зависит от ограниченного числа входных переменных, либо содержит ограниченное число элементарных конъюнкций [6].

Разбиение систем логических уравнений на подсистемы использует критерий оптимизации «суммарное число литералов» в задании системы уравнений. Литерал – это булева переменная либо ее инверсия. Каждое из уравнений в системе представляется в виде ДНФ, зависящей от своего (ограниченного по мощности) подмножества входных переменных для системы в целом, либо внутренних переменных функционального алгебраического описания. Методы решения задач разбиения на подсистемы (для задач разбиения употребляется также термин «кластеризация») преследуют различные цели, например, исходная система взаимосвязанных уравнений не может быть представлена в виде одной системы ДНФ, а каждый из полученных взаимосвязанных кластеров задается в виде системы ДНФ. Тогда для каждого из таких кластеров могут быть применены методы оптимизации в классе булевых сетей, либо методы минимизации в классе ДНФ и классе BDD, либо методы декомпозиции матричных форм систем ДНФ. При разбиении задаются ограничения на параметры кластеров – число входных переменных либо входных и выходных переменных. Основной целью кластеризации является получение кластеров, которые ограничены по числу входных или выходных переменных либо представляются более простыми уравнениями [17], содержащими меньшее число литералов. Для этого программно реализованы методы поляризации уравнений – т.е. выбора прямой либо инверсной формы ДНФ функции, задающей каждое уравнение. Логические уравнения, получаемые для перепроектирования комбинационной логики в синтезаторе LeonardoSpectrum, являются «мелкими» - в каждом уравнении имеется только один логический оператор. Такие уравнения, а также уравнения, задаю-

щие формулы разложения Шеннона, можно «укрупнять» по числу переменных, что благоприятно может сказаться на времени и качестве последующей кластеризации (каждое уравнение задает кластер в функциональном описании). При укрупнении задается число присутствующих в результирующем уравнении переменных, что позволяет получать функциональные описания, ориентированные на эффективную реализацию в FPGA. Соответствующие результаты экспериментального исследования данного подхода к синтезу комбинационных структур FPGA показали целесообразность применения программ предварительной логической BDD-оптимизации и укрупнения уравнений по числу переменных.

Изменение иерархии SF-описаний позволяет заменять в дереве иерархии проекта цифровой схемы выделенные структурные описания функциональными описаниями. Например, если некоторый блок реализован в виде одноуровневой структуры и имеет большую сложность, то переход к функциональному описанию и повторная оптимизация в классе многоуровневых представлений может привести к меньшей сложности функционально-структурного описания. Получение функциональных описаний может выполняться как без редукции связей, так с редукцией. В системе FLC-2 имеются средства получения описания схемы в виде структурного описания подсхем только верхнего уровня (в литературе такие описания называются netlist), а также средства объединения блоков на одном уровне иерархического описания, что позволяет в «ручном» режиме менять дерево иерархии описания. Так как в исходном описании могут быть булевы функции, являющиеся константами 0, 1, а константа 0 не имеет представления в виде ДНФ (это формат SDF языка SF), то в системе FLC-2 данная проблема решается изменением иерархии путем отделения констант 0,1 в отдельный блок, где они описываются в формате LOG (уравнениями). Для функций блока, не содержащего констант, могут применяться методы оптимизации в классе ДНФ, которые работают с матричными SDF-описаниями.

Форматные преобразования позволяют выполнять преобразование многоуровневых представлений (уравнений) из LOG-формата в матричную форму системы ДНФ (SDF-формат). При таком преобразовании устраняются (элиминируются) все промежуточные внутренние переменные. Заметим, что для некоторых примеров цифровых устройств это невозможно – число элементарных конъюнкций в системе ДНФ может быть огромным, примером может служить функциональное описание 16-разрядного умножителя. Поэтому в практике логического проектирования цифровых устройств используются многоуровневые алгебраические представления, для которых развиваются специальные методы логической оптимизации, основным является аппарат BDD и многочисленные его модификации [18 - 20].

Таблицы истинности систем булевых функций представляются в виде SDF-формата, в этом случае система функций, зависящая от n переменных, задает-

ся на 2^n полных элементарных конъюнкциях. Переход от SDF-формата к LOG-формату является тривиальным – каждая ДНФ системы записывается в виде отдельного логического уравнения.

Конвертирование SF-описаний в VHDL-описания (и обратно) используется для преобразования данных для логической оптимизации в системе FLC-2 и для преобразования минимизированных SF-описаний в VHDL-описания с целью их схемной реализации в зарубежных САПР [21].

Используемый аппарат продукций и стратегий для комбинированных методов логической оптимизации подробно описан в [1]. База знаний пополнена новыми стратегиями оптимизации листовых функциональных описаний, использующими новые программно реализованные методы на основе BDD и булевых сетей и кластеризации логических уравнений.

Верификация объектов проектирования осуществляется в отдельной системе [22], как на уровне SF-описаний, так и на уровне VHDL-описаний, в последнем случае используется система FormalPro (ф. Mentor Graphics) [23].

IV. ФОРМАТ БАЗЫ ЗНАНИЙ

В FLC-2 реализован производственно-фреймовый подход к обработке знаний, опирающийся на определение и использовании специализированной базы знаний, обслуживаемой при помощи автономного комплекса, ранняя версия которого описана в [1]. Главным отличием в этом направлении является переход от «доморощенного» формата описания базы к ее представлению в общепринятом формате JSON. Фактически в программе используется библиотека RapidJSON [24], при помощи которой обеспечивается сериализация хранимой базы знаний.

Как и ранее, верхний уровень в описании базы знаний представлен набором:

```
{
  "attributes": { ... },
  "strategies": { ...
    "productions": ... },
  "frames": { ...
    "slots": ... },
  "modules": { ... },
  "menu": { ... }.
}
```

Здесь "attributes" – перечень существенных свойств объектов, используемых при проектировании; "strategies" – множество заранее подготовленных сценариев (маршрутов) проектирования, определенных при помощи "productions", представляющих собой определения отдельных шагов процесса проектирования; "frames" определяют группирование объектов "slots", которые, в свою очередь, задают функциональное группирование объектов из "attributes"; "modules" представляют описания программных модулей, реализующих проектные процедуры, "menu" – описание ме-

ню таких процедур, доступное пользователю системы FLC-2 в процессе проектирования.

V. СПОСОБ ХРАНЕНИЯ ПРОЕКТОВ

Единицей обработки данных в рамках системы FLC-2 является проект. В первой версии системы [1] проект представлялся множеством специальным образом именованных файлов, каждый из которых содержал информацию об отдельном состоянии проекта, фиксируемом по завершению очередной проектной операции. Размещение в файловой системе (директорий проекта) определялось пользователем системы. Для FLC-2 выбор местоположения проекта остался за пользователем, а вот его внутреннее представление существенно изменилось. Если ранее отдельное состояние извне было недоступным (в силу специального представления в файле в виде бинарных данных), то в версии FLC-2 для определения всего проекта используется один файл, по сути являющийся архивом с плотно упакованными текстовыми данными. В этом архиве располагается множество директорий, имена которых соответствуют номерам хранимых состояний, а также файлы «limits» (для общего описания хранимых поколений) и «settings» (для представления используемого в проекте набора конфигурационных параметров). Каждая из хранимых директорий внутри себя содержит файл «attributes» – совокупность атрибутов (и их значений), представленных в базе знаний, и значения которых были изменены в процессе выполнения проектной процедуры, приведшей к созданию рассматриваемого состояния проекта; файл «operation» – имя выполненного программного модуля, реализовавшего проектную процедуру, и файл «source» - SF-описание обрабатываемой в проекте схемы. Как уже было сказано выше, все эти данные могут быть доступны извне как обычные текстовые файлы.

VI. ПРАВИЛА ОФОРМЛЕНИЯ МОДУЛЕЙ ПРОЕКТНЫХ ПРОЦЕДУР

Одним из основных подходов к разработке программного обеспечения в современных условиях является обеспечение кроссплатформенности создаваемого продукта. В этой связи в настоящее время все разрабатываемое программное обеспечение, как элементы ядра системы, так и ее интерфейсной оболочки, реализуется с применением кроссплатформенной библиотеки Qt [25].

Применение нового инструментария основывается на включении в состав значений атрибута «тип модуля» нового значения «терминальный», что соответствует определению модуля как консольного приложения, принимающего входные параметры на основе передачи ключевых параметров. Причем обязательными являются только параметры «путь доступа к файлу исходных данных» (используется ключ `--input` или в сокращенной форме `-i`) и «путь доступа к файлу результирующих данных» (ключ `--output` или `-o`). Прочие параметры могут быть использованы дополнительно, в частности, часто находят применение ключ `-r`, обеспечивающий передачу пути доступа к файлу configura-

ции, а тем самым, ко всему множеству параметров управления системы FLC-2 и ряда проектных процедур.

Как ранее подготовленные модули, разработанные по правилам модулей типа «Windows», так и новые, организованные по образцу консольных приложений Qt, могут использоваться одновременно в стратегиях и продукциях новой базы знаний при ее эксплуатации в рамках операционной системы Windows.

VII. ИЗМЕНЕНИЯ ВНЕШНЕГО ИНТЕРФЕЙСА

Внешний вид рабочего окна системы FLC-2 существенного изменения по сравнению с системой FLC не претерпел, незначительные отличия по форме можно сформулировать следующим образом:

- поле представления символьного отображения SF-описания в случае матричного формата SDF в новой версии представлено парой панелей – для каждой из матриц используется отдельная панель, навигация в которой по столбцам осуществляется теперь независимо, что расширяет имеющиеся возможности наблюдения обрабатываемых данных.

- в конце строки горизонтального меню располагается текстовое поле в начале сеанса работы, в котором представлена идентификация работающей версии, например «14022019_10». В процессе сеанса в этом поле размещается время выполнения последней проектной процедуры в микросекундах в форме «00:01:34.048»;

- расширен состав атрибутов, значения которых представлены на постоянно отображаемой панели атрибутов;

- древовидная структура вложенности блоков SF-описания проекта отображается при помощи общепринятых средств отображения и возможного управления им (например, свертыванием поддеревьев).

С точки зрения возможностей управления процессом проектирования, обеспечиваемого новой оболочкой в системе FLC-2, то следует отметить следующее:

- в роли входных данных для FLC-2 всегда выступает один файл, содержащий в себе все необходимые компоненты. Такой файл может быть определен как на языке SF, так и на языке VHDL [5]. Существовавшие ранее возможности компоновки проекта из отдельных элементов были признаны неудачными и избыточными;

- в новой версии системы отсутствуют встроенные средства верификации, как отдельных состояний проекта, так и целых проектов. В том виде, в каком они существовали ранее, их присутствие в рамках системы проектирования ориентировалось на подключение новых проектных процедур, а не на собственно проектирование.

- для унификации средств параметризации включаемых проектных процедур механизм приема параметров управления, основанный на подготовке специаль-

ных диалогов для каждого из параметров, в FLC-2 заменен на возможности прямого изменения содержимого конфигурационного файла проекта. Ранее определенные диалоги приема параметров сохранены, но, по сути, стали необязательными, и просто в другой форме дублируют возможности настройки опций проектных процедур;

- для удобства пользователей в FLC-2 добавлена возможность вызова из меню программного модуля непосредственно по его имени, а не только путем выбора в иерархическом меню.

VIII. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНОГО ИССЛЕДОВАНИЯ ПРОГРАММ ЛОГИЧЕСКОЙ ОПТИМИЗАЦИИ

Новые программы логической оптимизации были экспериментально исследованы на промышленных примерах из библиотек [26, 27]. Целью экспериментов было получение оценки влияния методов логической оптимизации на сложность (площадь) заказных логических КМОП схем и их энергопотребление [3, 10 - 13, 15], а также на сложность комбинационных структур, реализуемых в составе FPGA (ф. Xilinx). В работе [28] предложена технология проектирования цифровых устройств, позволяющая выполнять логическое моделирование VHDL-описаний комбинационной логики, формировать соответствующие системы булевых функций, проводить их логическую оптимизацию с помощью программ системы FLC-2 и синтезировать логические схемы в различных технологических библиотеках логических элементов, приведены результаты соответствующих экспериментов. Интеграция программных средств в рамках этой технологии основывается на использовании скриптов и BAT-файлов, которые поддерживаются современными САПР.

Эксперименты показали целесообразность использования программ системы FLC-2 для предварительной оптимизации проектов цифровых устройств, реализуемых в промышленных САПР.

IX. ЗАКЛЮЧЕНИЕ

Включенные в систему FLC-2 программы технологически независимой логической оптимизации прошли экспериментальную проверку на промышленных примерах схем практической размерности, многие из программ включены в отечественные системы автоматизированного проектирования цифровых устройств. Использование таких программ позволяет во многих случаях улучшать результаты синтеза логических заказных КМОП схем, получаемых в промышленном синтезаторе LeonardoSpectrum, и синтезаторе XST системы ISE Xilinx [29], предназначенном для проектирования цифровых устройств на FPGA.

ЛИТЕРАТУРА

- [1] Бибило П. Н., Романов В.И. Логическое проектирование дискретных устройств с использованием производственно-фреймовой модели представления знаний. – Минск : Беларус. навука, 2011. – 279 с.
- [2] Бибило П.Н., Черемисинова Л.Д., Кардаш С.Н., Кириенко Н.А., Романов В.И., Черемисинов Д.И.

- Автоматизация логического синтеза КМОП схем с пониженным энергопотреблением // Программная инженерия. – 2013. – № 8, – С. 35–41.
- [3] Бибило П.Н., Авдеев Н.А., Кардаш С.Н., Кириенко Н.А., Ланкевич Ю.Ю., Логинова И.П., Романов В.И., Черемисинов Д.И., Черемисинова Л.Д. Система логического проектирования функциональных блоков заказных КМОП СБИС с пониженным энергопотреблением // Микроэлектроника. – 2017. – Т. 46. – № 1. – С.72–88.
- [4] Брейтон Р.К., Хэчтел Г.Д., Санджованни-Винченцелли А. Л. Синтез многоуровневых комбинационных логических схем // ТИИЭР. – 1990. – Т. 78, № 2. – С. 38–83.
- [5] Бибило П.Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum. – М.: СОЛОН-Пресс, 2005. – 384 с.
- [6] Закревский А.Д. Логический синтез каскадных схем. М.: Наука, 1981. – 416 с.
- [7] Торопов Н. Р. Минимизация систем булевых функций в классе ДНФ // Логическое проектирование. – Минск: Ин-т техн. кибернетики НАН Беларуси, 1999. – Вып. 4. – С. 4–19.
- [8] Леончик П.В. Минимизация систем булевых функций в классе дизъюнктивных нормальных форм // Информатика – 2006. – № 1. - С. 88–96.
- [9] Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д. Логические основы проектирования дискретных устройств. – М.: Физматлит, 2007. – 589 с.
- [10] Бибило П.Н. Применение диаграмм двоичного выбора при синтезе логических схем. – Минск: Беларуская наука, 2014. – 231 с.
- [11] Бибило П. Н., Ланкевич Ю. Ю. Использование полиномов Жегалкина при минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона // Программная инженерия. – 2017. – № 3. – С. 369–384.
- [12] Бибило П.Н., Ланкевич Ю.Ю. Логическая минимизация булевых сетей с использованием разложения Шеннона // Информатика. – 2019. – Т. 16. – № 2. – С. 73 – 89.
- [13] Бибило П. Н., Позняк А. М. Выделение подсистем связанных функций из многоуровневого представления системы булевых функций // Информатика. – 2020. – Т. 17, № 1. – С. 63 – 77.
- [14] Кардаш С.Н. Экспериментальное исследование алгоритмов дизъюнктивно-конъюнктивного разложения систем полностью определенных булевых функций // Информационные технологии и системы 2018 (ITS 2018) – Information Technologies and Systems 2018 (ITS 2018) : материалы междунар. науч. конф. (Республика Беларусь, Минск, 25 октября 2018 года). – Минск : БГУИР, 2018. – С. 160–161.
- [15] Кириенко Н.А., Черемисинова Л.Д. Исследование эффективности технологически независимой оптимизации функциональных описаний КМОП схем // Информатика – 2016. – № 3 (51). - С. 59–66.
- [16] Черемисинов Д.И. Отображение логических сетей в заданный технологический базис // Информатика. – 2017. – № 1(53). С. 44 – 52.
- [17] Кириенко Н.А. Исследование оптимизационных преобразований логических схем путем разбиения на блоки // Информационные технологии в промышленности, логистике и социальной сфере (ITI*2019): тезисы докладов X Международной научно-технической конференции (23 – 24 мая 2019 года, Минск). – Минск : ОИПИ НАН Беларуси, 2019. – С. 94 – 96.
- [18] Карпов Ю.Г. MODEL CHECKING. Верификация параллельных и распределенных программных систем. СПб.: БХВ-Петербург, 2010. – 560 с.
- [19] Kubica M., Kania D. SMTBDD: New form of BDD for logic synthesis // International Journal of Electronics and Telecommunications. - 2016.- V. 62. N1. – P. 33–41.
- [20] Kubica M., Kania D. Decomposition of multi-output functions oriented to configurability of logic blocks // Bulletin of the Polish Academy of Sciences. Technical Sciences. – 2017. - Vol. 65.- N. 3. - P. 317 – 331.
- [21] Черемисинов Д.И. Анализ и преобразование структурных описаний СБИС. – Мн: Белорусская наука, 2006. – 275с.
- [22] Логинова И.П. Верификация с использованием средств formalpro в системе логического проектирования заказных КМОП СБИС // Новые информационные технологии в исследовании сложных структур : материалы Двенадцатой конференции с международным участием. 4–8 июня 2018 г. – Томск : Издательский Дом Томского государственного университета, 2018. – С. 72–73.
- [23] Лохов А. Функциональная верификация СБИС в свете решений Mentor Graphics // Электроника: наука, технология, бизнес. – 2004. – № 1. – С. 58 – 62.
- [24] URL: <http://www.rapidjson.org> (access date 22.04.2020).
- [25] Шлее М. Qt 5.10. Профессиональное программирование на C++. – СПб.: БХВ-Петербург, 2018. – 1072 с.
- [26] URL: <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex> (access date: 20.03.2018).
- [27] URL:<https://ddd.fit.cvut.cz/prj/Benchmarks> (access date: 12.04.2020).
- [28] Бибило П.Н., Авдеев Н.А., Романов В.И. Функциональная верификация VHDL-описаний синхронных цифровых устройств. – М.: ЛЕНАНД, 2020. – 328 с.
- [29] Зотов, Ю. В. Проектирование цифровых устройств на основе ПЛИС фирмы XILINX в САПР WebPack ISE. – М.: Горячая линия-Телеком, 2003. – 624 с.

The System of Logical Optimization of Digital Circuits Functional Structural Descriptions Based on Production-Frame Knowledge Representation Model

P.N. Bibilo, V.I. Romanov

United Institute of Informatics Problems of NAS of Belarus

bibilo@newman.bas-net.by, rom@newman.bas-net.by

Abstract — The FLC-2 system is described. The system is designed to optimize hierarchically organized functional and structural descriptions of logic circuits presented in the SF language, which is the internal language of the system. To communicate with other CAD systems of digital devices, FLC-2 has the means to convert SF descriptions to VHDL descriptions (and vice versa). Logical optimization is carried out through the use of combined methods that implement various basic optimization techniques: minimization of systems of Boolean functions in the class of DNF, optimization of multilevel representations of Boolean function systems based on Boolean networks and BDD representations, decomposition of matrix and bracket forms, allocation of subsystems of functions related by areas of definition.

The FLC-2 system implements a productive rule approach for managing sequences of design procedures intended for their use at the stage of logical design of complex functional blocks of custom digital VLSIs. The experiments demonstrated the expediency of using the FLC-2 system for preliminary optimization of digital circuits implemented in industrial CAD systems for custom digital VLSI and FPGA.

Keywords — system of Boolean functions, Disjunctive Normal Form (DNF), Binary Decision Diagram (BDD), digital logic synthesis, VHDL, VLSI, CMOS, knowledge base.

REFERENCES

- [1] Bibilo P.N., Romanov V.I. Logicheskoe proektirovanie diskretnykh ustroystv s ispol'zovaniem produkcionno-frejmovoj modeli predstavleniya znaniy (Logical design of discrete devices with use of productional and frame model of representation of knowledge). Minsk: Belarus. navuka, 2011. 279 p. (In Russian).
- [2] Bibilo P.N., ChRemisinova L.D., Kardash S.N., Kirienko N.A., Romanov V.I., ChRemisinov D.I. Avtomatizatsiya logicheskogo sinteza KMOP skhem s ponizhennym energopotrebleniem (Automation of logical synthesis of CMOS circuits with reduced power consumption) // Programmaya inzheneriya. 2013. № 8. P. 35–41. (In Russian).
- [3] Bibilo P. N., Avdeev N. A., Kardash S. N., Kirienko N. A., Lankevich Yu. Yu., ChRemisinova L. D. Sistema logicheskogo proektirovaniya funktsional'nykh blokov zakaznykh KMOP SBIS s ponizhennym energopotrebleniem (System for the logical design of functional blocks of custom CMOS VLSI with low power consumption). Mikroelektronika. 2017. Vol. 46, № 1, P. 72–88. (In Russian)
- [4] Brayton R. K., Hachtel G. D., Sangiovanni-Vincentelli A. L. Sintez mnogourovnevnykh kombinatsionnykh logicheskikh skhem (Multilevel Logic Synthesis). Trudy institute inzhenerov po jelektronike i radiotekhnike, 1990, Vol. 78. № 2. P. 38–83. (In Russian)
- [5] Bibilo P. N. Cistemy proektirovaniya integral'nykh skhem na osnove yazyka VHDL. StateCAD, ModelSim, LeonardoSpectrum (CAD of integrated circuits based on the VHDL. StateCAD, ModelSim, LeonardoSpectrum). Moscow, SOLON-Press Publ. 2005. 384 p. (In Russian)
- [6] Zakrevskij A.D. Logicheskij sintez kaskadnykh skhem (Logical synthesis of cascading circuits). M.: Nauka. 1981. 416 p. (In Russian)
- [7] Toropov N. R. Minimizatsiya sistem bulevykh funktsiy v klasse DNF (Minimization of Boolean function systems in the SOP class) // Logicheskoe proektirovanie. Minsk: In-t tekhn. kibernetiki NAN Belarusi. 1999. Vyp. 4. P. 4–19. (In Russian)
- [8] Leonchik P.V. Minimizatsiya sistem bulevykh funktsiy v klasse diz'yunktivnykh normal'nykh form (Minimization of Boolean function systems in the class of SOP) // Informatika. 2006. № 1. P. 88–96. (In Russian)
- [9] Zakrevskij A.D., Pottosin Ju.V., ChRemisinova L.D. Logicheskie osnovy proektirovaniya diskretnykh ustroystv (Logical bases of design of discrete devices). M.: Fizmatlit. 2007. 589 p. (In Russian)
- [10] Bibilo P.N. Primenenie diagramm dvoichnogo vybora pri sinteze logicheskikh shem (Application of Binary Decision Diagrams at synthesis of logical circuits) Minsk: Belarus. navuka, 2014. 231 p. (In Russian)
- [11] Bibilo P. N., Lankevich Yu. Yu. Ispol'zovanie polinomov Zhegalkina pri minimizatsii mnogourovnevnykh predstavlenij sistem bulevykh funktsiy na osnove razlozheniya Shennona (The use of Zhegalkin polynomials in minimizing multilevel representations of systems of Boolean functions based on the Shannon expansion). // Programmaya inzheneriya. 2017. № 3. P. 369–384. (In Russian)
- [12] Bibilo P.N., Lankevich Yu. Yu. Logicheskaya minimizatsiya bulevykh setej s ispol'zovaniem razlozheniya SHennona (Logical optimization of Boolean nets using Shannon expansion). // Informatika. 2019. Vol. 16. № 2. P. 3–89. (In Russian)
- [13] Bibilo P. N., Poznyak A. M. Vydelenie podsistem svyazannykh funktsiy iz mnogourovnevnogo predstavleniya sistemy bulevykh funktsiy (The search for subsystems of related functions from multilevel representation of systems of Boolean functions) // Informatika. 2020. Vol. 17. № 1. P. 63–77. (In Russian)
- [14] Kardash S.N. Eksperimental'noe issledovanie algoritmov diz'yunktivno-kon'yunktivnogo razlozheniya sistem polnost'yu opredelennykh bulevykh funktsiy (Experimental study of algorithms for disjunctive-conjunctive decomposition of systems of fully defined Boolean functions) // Informatsionnye tekhnologii i sistemy 2018 (ITS 2018) – Information Technologies and Systems 2018

- (ITS 2018): materialy mezhdunar. nauch. konf. (Respublika Belarus', Minsk, 25 oktyabrya 2018 goda). Minsk : BGUIR. 2018. P. 160–161. (In Russian)
- [15] Kirienko N.A., Cheremisinova L.D. Issledovanie effektivnosti tekhnologicheski nezavisimoy optimizacii funkcional'nyh opisaniy KMOP skhem (Analysis of effectiveness of technology independent optimization of CMOS circuit functional descriptions) // Informatika. 2016. № 3 (51). P. 59–66. (In Russian)
- [16] Cheremisinov D.I. Otbrazhenie logicheskikh setej v zadannyj tekhnologicheskij bazis (Technology Mapping Tool for VLSI CAD) // Informatika. 2017. № 1(53). P. 44–52. (In Russian)
- [17] Kirienko N.A. Issledovanie optimizacionnyh preobrazovanij logicheskikh skhem putem razbieniya na bloki (Research of optimization transformations of logic circuits by splitting into blocks) // Informacionnye tekhnologii v promyshlennosti, logistike i social'noj sfere (ITI*2019): tezisy dokladov X Mezhdunarodnoj nauchno-tekhnicheskoy konferencii (23–24 maya 2019 goda, Minsk). Minsk: OIPI NAN Belarusi. 2019. P. 94–96. (In Russian)
- [18] Karpov Yu.G. MODEL CHECKING. Verifikaciya paralel'nyh i raspredelennyh programmnyh system (MODEL CHECKING. Verification of parallel and distributed software systems). SPb.: BHV-Peterburg. 2010. 560 p. (In Russian)
- [19] Kubica M., Kania D. SMTBDD: New form of BDD for logic synthesis // International Journal of Electronics and Telecommunications. 2016. V. 62. N1. P. 33–41.
- [20] Kubica M., Kania D. Decomposition of multi-output functions oriented to configurability of logic blocks // Bulletin of the Polish Academy of Sciences. Technical Sciences. 2017. Vol. 65. N. 3. P. 317–331.
- [21] Cheremisinov D.I. Analiz i preobrazovanie strukturnykh opisaniy SBIS (Analysis and transformation of VLSI structural descriptions). – Mn: Belorusskaya nauka, 2006. 275 p. (In Russian)
- [22] Loginova I.P. Verifikaciya s ispol'zovaniem sredstv FormalPro v sisteme logicheskogo proektirovaniya zakaznyh KMOP SBIS (Verification using FormalPro tools in the system of logical design of custom CMOS VLSI) // Novye informacionnye tekhnologii v issledovanii slozhnyh struktur: materialy Dvenadcatoy konferencii s mezhdunarodnym uchastiem. 4–8 iyunya 2018 g. Tomsk: Izdatel'skij Dom Tomskogo gosudarstvennogo universiteta, 2018. P. 72–73. (In Russian)
- [23] Lohov A. Funkcional'naya verifikaciya SBIS v svete reshenij Mentor Graphics (Functional verification of VLSI in the light of Mentor Graphics solutions) // Elektronika: nauka, tekhnologiya, biznes. 2004. № 1. P. 58–62. (In Russian)
- [24] URL: <http://www.rapidjson.org> (access date 22.04.2020).
- [25] SHlee M. Qt 5.10. Professional'noe programmirovaniye na S++. (Professional C ++ Programming). Saint Petersburg, BHV-Peterburg. 2018. 1072 p. (In Russian)
- [26] URL: <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex> (access date: 20.03.2018).
- [27] URL: <https://ddd.fit.cvut.cz/prj/Benchmarks> (access date: 12.04.2020).
- [28] Bibilo P.N., Avdeev N.A., Romanov V.I. Funkcional'naya verifikaciya VHDL-opisaniy sinhronnykh cifrovyyh ustrojstv (Functional verification of VHDL-descriptions of synchronous digital devices). – M.: LENAND. 2020. 328 p (In Russian)
- [29] Zotov Yu. V. Proektirovaniye cifrovyyh ustrojstv na osnove PLIS firmy XILINX v SAPR WebPack ISE (The design of digital devices in CAD WebPack XILINX). M.: Goryachaya liniya-Telekom. 2003. 624 p. (In Russian)