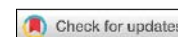


ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

LOGICAL DESIGN



УДК 519.714.5
<https://doi.org/10.37661/1816-0301-2021-18-1-7-24>

Оригинальная статья
Original Paper

Логическая минимизация при синтезе комбинационных структур в FPGA

П. Н. Бибило[✉], Ю. Ю. Ланкевич, В. И. Романов

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

Аннотация. Описываются результаты исследования эффективности применения программ минимизации функциональных описаний блоков комбинационной логики, входящих в проекты цифровых устройств, которые реализуются в FPGA (от англ. Field-Programmable Gate Array – программируемая пользователем вентильная матрица). Программы предназначены для отдельной и совместной минимизации функций в классе ДНФ (дизъюнктивных нормальных форм) и минимизации многоуровневых представлений систем полностью определенных булевых функций на основе разложения Шеннона с нахождением как равных, так и инверсных коэффициентов (кофакторов) разложения. Графические формы таких представлений широко известны в литературе как BDD (от англ. Binary Decision Diagram – бинарная диаграмма решений). Для технологического отображения применялась программа «укрупнения» полученных формул разложения Шеннона (логических уравнений) с условием, чтобы каждое уравнение зависело от ограниченного числа k входных переменных и могло быть реализовано на одном LUT- k – программируемом элементе FPGA, имеющем k входных переменных (LUT – Look-Up Table – таблица, реализующая логическую функцию). Показано, что предварительная логическая минимизация, выполняемая с помощью отечественных программ, позволяет улучшать результаты проектирования в зарубежных системах автоматизированного проектирования, таких как Leonardo Spectrum (корпорация Mentor Graphics), ISE (от англ. Integrated System Environment) Design Suite и Vivado (компания Xilinx). Эксперименты проводились для семейств FPGA Virtex-II PRO, Virtex-5, Artix-7 (компания Xilinx) на наборах стандартных промышленных примеров, задающих как системы дизъюнктивных нормальных форм булевых функций, так и системы булевых функций в виде взаимосвязанных логических уравнений.

Ключевые слова: булева функция, логическая минимизация, разложение Шеннона, BDD-представление, дизъюнктивная нормальная форма, синтез логических схем, VHDL, FPGA

Благодарности. Исследование выполнено при финансовой поддержке БРФФИ в рамках проекта № Ф19-023.

Для цитирования. Бибило, П. Н. Логическая минимизация при синтезе комбинационных структур в FPGA / П. Н. Бибило, Ю. Ю. Ланкевич, В. И. Романов // Информатика. – 2021. – Т. 18, № 1. – С. 7–24. <https://doi.org/10.37661/1816-0301-2021-18-1-7-24>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 27.08.2020
Подписана в печать | Accepted 08.10.2020
Опубликована | Published 26.03.2021

Logical minimization for combinatorial structure in FPGA

Petr N. Bibilo[✉], Yury Yu. Lankevich, Vladimir I. Romanov

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus*

[✉]E-mail: bibilo@newman.bas-net.by

Abstract. The paper describes the research results of application efficiency of minimization programs of functional descriptions of combinatorial logic blocks, which are included in digital devices projects that are implemented in FPGA. Programs are designed for shared and separated function minimization in a disjunctive normal form (DNF) class and minimization of multilevel representations of fully defined Boolean functions based on Shannon expansion with finding equal and inverse cofactors. The graphical form of such representations is widely known as binary decision diagrams (BDD). For technological mapping the program of "enlargement" of obtained Shannon expansion formulas was applied in a way that each of them depends on a limited number of k input variables and can be implemented on one LUT- k – a programmable unit of FPGA with k input variables. It is shown that a preliminary logic minimization, which is performed on the domestic programs, allows improving design results of foreign CAD systems such as Leonardo Spectrum (Mentor Graphics), ISE (Integrated System Environment) Design Suite and Vivado (Xilinx). The experiments were performed for FPGA families' Virtex-II PRO, Virtex-5 and Artix-7 (Xilinx) on standard threads of industrial examples, which define both DNF systems of Boolean functions and systems represented as interconnected logical equations.

Keywords: Boolean function, logical minimization, Shannon expansion, BDD representation, disjunctive normal form, logic synthesis, VHDL, FPGA

Acknowledgements. The study was carried out with the financial support of the BRFFR within the framework of the project no. F19-023.

For citation. Bibilo P. N., Lankevich Yu. Yu., Romanov V. I. Logical minimization for combinatorial structure in FPGA. *Informatics*, 2021, vol. 18, no. 1, pp. 7–24 (in Russian). <https://doi.org/10.37661/1816-0301-2021-18-1-7-24>

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

Введение. Среди программируемых логических интегральных схем (ПЛИС) центральное место занимают FPGA, которые имеют значительные преимущества перед другими ПЛИС как по техническим характеристикам, так и по удобству их проектирования с помощью свободно распространяемых САПР, обеспечивающих полный цикл проектирования: от моделирования исходных алгоритмических описаний на языках Verilog и VHDL (Very high speed integrated circuits Hardware Description Language – язык описания аппаратуры сверхскоростных интегральных схем) до получения файлов конфигураций FPGA [1]. В новых системах проектирования микросхем FPGA, таких как система Vivado [2], в качестве языков для описания проектов цифровых систем используются также языки программирования C, C++ и SystemC. Функциональные возможности FPGA постоянно совершенствуются. Они используются при создании специальных вычислителей, конкурирующих с суперкомпьютерами [3], позволяют решать задачи аппаратной реализации подсистем искусственного интеллекта [4], применяются в системах цифровой обработки сигналов, системах мультимедиа [5] и многих других областях. Расширение функциональных возможностей FPGA обусловлено тем, что они усложняются: увеличивается число CLB (Configurable Logic Block – конфигурируемый логический блок), в состав FPGA включаются макроблоки DSP (Digital Signal Processor – цифровой сигнальный процессор), умножители и блоки памяти [6].

Схемная реализация исходных описаний на языках C, C++ и SystemC осуществляется сначала путем получения синтезируемых VHDL-описаний, после этого каждая синтезируемая VHDL-конструкция заменяется соответствующим RTL-описанием (Register Transfer Level – уровень регистровых передач), которое состоит из элементов памяти и взаимосвязанных логических операторов, задающих блоки комбинационной логики. Число элементов памяти зависит от стиля исходного описания проекта и кодирования состояний проекта цифрового устройства, а минимизация сложности блоков комбинационной логики сводится к задаче нахождения логической сети, состоящей из наименьшего числа LUT- k , которые имеют k входных переменных

и могут при соответствующей настройке реализовать любую булеву функцию, зависящую от k переменных (обычно $k = 4, 6$). Следует заметить, что несколько LUT- k входят в состав конфигурационного блока CLB, который кроме LUT включает настраиваемые триггеры, мультиплексоры и другие элементы. Системы проектирования, как правило, оценивают сложность реализованных проектов в числе LUT либо CLB.

В настоящей статье рассматривается задача реализации блоков комбинационной логики LUT-структурами FPGA. Показывается, что уменьшение сложности комбинационных структур FPGA может быть во многих случаях достигнуто за счет предварительной логической минимизации исходных функциональных описаний.

Задача реализации комбинационной логики в FPGA. Исходное VHDL-описание комбинационного блока можно представить в виде алгоритмического описания с использованием различных типов данных и операторов, потока данных (dataflow), т. е. взаимосвязанных логических выражений, и т. д. Однако после этапа высокоуровневого синтеза реализации в FPGA подлжит система полностью определенных булевых функций, заданная либо «крупными» логическими уравнениями, либо в виде RTL-описания, представленного «мелкими» логическими уравнениями, когда каждое уравнение включает только один логический оператор, и т. п.

Задачу реализации комбинационной логики в FPGA сформулируем следующим образом: *задана система полностью определенных булевых функций. Требуется реализовать ее в виде суперпозиции (функционального разложения, логической сети) по возможности наименьшего числа булевых функций, зависящих не более чем от k переменных.*

Предполагается, что каждая из функций, входящих в суперпозицию, может быть реализована на одном LUT- k , поэтому минимизация числа булевых функций, входящих в полученную суперпозицию, приводит к минимизации числа программируемых LUT- k в схеме FPGA. Сформулированная в таком виде задача, возникающая на этапе логического проектирования FPGA, была известна в теории булевых функций и ранее (до появления FPGA) [7]. Вместо программируемых логических элементов LUT рассматривались универсальные логические модули либо мультиплексоры с числом k управляющих входов. Поэтому направление работ, связанное с развитием методов декомпозиции булевых функций, было привлечено в качестве теоретической базы синтеза структур FPGA [8]. Такой подход был близок и к синтезу схем на базе постоянных запоминающих устройств (ПЗУ) [9], так как LUT может рассматриваться как программируемое ПЗУ с k адресными двоичными входами и одним выходом. Таким ПЗУ может быть реализована любая булева функция, зависящая от k переменных.

Вместе с тем методы декомпозиции систем булевых функций были развиты для матричных форм исходного задания – таблиц истинности, либо матричных форм систем ДНФ. Чтобы воспользоваться такими методами декомпозиции, нужен был переход от логических уравнений (скобочных форм) к матричным формам задания систем функций. Так как размерности задач были достаточно большими, то данный переход был не всегда возможен; поэтому выполнялась кластеризация (выделение блоков, подсхем) функционального описания. Например, в синтезаторе LeonardoSpectrum [10] был введен специальный параметр, задающий размеры кластеров (подсхем), для которых применяется независимая логическая оптимизация.

Заметим, что существующие методы технологически независимой оптимизации [11], основанные на факторизации алгебраических представлений булевых функций и применяемые при синтезе схем из библиотечных элементов, оказались недостаточно эффективными, поскольку последующий этап технологического отображения, связанный с укрупнением либо разбиением алгебраических уравнений (таким, чтобы каждое из них зависело не более чем от k переменных), был затруднен из-за нерегулярной структуры оптимизированных логических выражений. В это время технологически независимую оптимизацию при синтезе FPGA предложили вести на базе BDD (см., например, [12]), которые явились эффективным аппаратом для верификации цифровых систем. Логические уравнения, соответствующие BDD, описывают каскадную схему в базисе мультиплексоров с одним управляющим входом и могут эффективно покрываться базовыми LUT- k . Кроме того, много научных статей и монографий (см., например, [13, 14]) было посвящено нахождению порядка переменных, по которым ведется построение BDD, что позволило сокращать размер BDD, т. е. число формул разложения Шеннона, описывающих BDD. Применение BDD при синтезе FPGA рассмотрено в большом числе научных работ (см., в частности, [15]).

В 2006 г. был опубликован обзор [16], посвященный различным аспектам проектирования FPGA (трассировке соединений, сокращению энергопотребления, верификации и др.); в нем процитировано 219 работ, в том числе монографии и статьи по логическому синтезу структур FPGA.

Современные САПР (системы автоматизированного проектирования) FPGA содержат программы, реализующие эффективные методы решения задачи синтеза комбинационных структур FPGA. Синтезатор LeonardoSpectrum для каждого из семейств FPGA при синтезе опирается на свою технологическую библиотеку, состав которой доступен для пользователей. Подробности (методы, алгоритмы) логического синтеза в САПР ISE для пользователей скрыты, проектировщик может управлять синтезом с помощью задания определенных опций синтеза.

Разработка методов, алгоритмов и программ для решения задачи реализации комбинационной логики в FPGA не прекращается и в настоящее время. Например, в работах [17, 18] развивается подход, связанный с декомпозицией BDD векторной булевой функции. Корневой вершине такой BDD соответствует система функций, а листовые вершины задают значения системы функций на интервалах булева пространства – путях из корневой вершины к листовым. В работе [19] для случая FPGA описывается система BDS-pga, которая развивает известную систему BDS [20]. Эта система имеет средства дополнительной оптимизации BDD, использующие поиск в BDD подфункций, выражаемых через дизъюнкцию, конъюнкцию, сумму по модулю 2 и через простую (мультиплексорную) декомпозицию. В работе [21] развиваются идеи декомпозиции булевых функций на основе фундаментальной работы Ашенхерста [22] с применением подхода, связанного с решением проблемы выполнимости конъюнктивной нормальной формы булевой функции [23], и, в частности, утверждается, что такой подход позволяет обрабатывать функции, зависящие от нескольких сотен переменных.

В работе [24] исследован подход, связанный с повторным синтезом. Эксперименты показали, что смена технологических библиотек и сохранение RTL-описаний получаемых схем позволяют добиться некоторого улучшения результатов реализации комбинационных блоков в FPGA. Такой подход позволяет улучшить результаты, если размерность задачи проектирования логической схемы достаточно велика. Для небольших размерностей систем булевых функций изменение формы исходного описания мало влияет на результат синтеза, так как синтезаторы имеют собственные встроенные программы технологически независимой оптимизации, приводящие различные исходные формы к одной и той же внутренней форме, по которой и будет построена результирующая логическая схема. Более того, изменение формы исходного описания может привести и к ухудшению результатов синтеза. Вместе с тем лучших результатов можно добиться не повторным синтезом, а предварительной глобальной логической оптимизацией исходного описания комбинационной схемы. Оптимизированное описание и следует подавать на вход промышленного синтезатора [24]. Это было проверено экспериментально на большом потоке промышленных примеров и показало эффективность для синтеза схем из библиотечных элементов [25] в промышленном синтезаторе LeonardoSpectrum. Такой подход предлагается применять при синтезе комбинационных сетей из LUT- k при синтезе FPGA.

Программы технологически независимой логической оптимизации. Для уменьшения сложности комбинационных блоков FPGA перед выполнением синтеза в промышленных синтезаторах предлагается осуществлять логическую минимизацию реализуемых функциональных описаний комбинационных блоков в новом классе BDD-представлений (с нахождением не только равных, но и взаимно инверсных коэффициентов разложения Шеннона) и проводить укрупнение уравнений, выполняя этап технологического отображения в базис LUT- k . Реализация такого подхода опирается на перечисленные ниже программные модули модифицированной платформенно-независимой системы логической оптимизации FLC-2, первая версия которой (FLC) была описана в работе [24].

Минимизация булевых функций в классе ДНФ выполняется программным модулем MINIM, который реализует различные методы и алгоритмы минимизации (совместной или отдельной) [26] систем булевых функций в классе ДНФ по разным критериям (числу элементарных конъюнкций; суммарному числу литералов в конъюнкциях, входящих в минимизированную систему ДНФ).

Минимизация булевых функций в классе многоуровневых представлений (BDDI-минимизация на основе разложения Шеннона) выполняется модулем BDD_Builder.

Разложением Шеннона булевой функции $f(\mathbf{x}) = f(x_1, \dots, x_n)$ по переменной x_i называется представление $f(\mathbf{x})$ в виде выражения

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

Каждый из коэффициентов (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 разложен по одной из переменных множества $\{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\}$. Процесс разложения коэффициентов заканчивается, когда все n переменных будут использованы. В процессе разложения либо на последнем шаге некоторые коэффициенты могут вырождаться до констант 0, 1. На каждом шаге разложения выполняется поиск одинаковых и взаимно инверсных коэффициентов, из такого их множества оставляется один. Алгоритм, реализованный в программе BDD_Builder, подробно описан в статье [27]. При выборе очередной переменной разложения Шеннона алгоритм использует следующее правило: очередной переменной выбирается та, по которой вычисляется минимальное число различных взаимно инверсных подфункций (кофакторов) разложения Шеннона. Критерием оптимизации является минимум числа формул разложения Шеннона для задания системы исходных функций (минимальное число вершин в графе BDD, представляющем взаимосвязанные формулы разложения Шеннона).

Укрупнение уравнений выполняется программным модулем Presin, описанным в работе [28] и модифицированным для решения задач большей размерности. Программа, элиминируя некоторые промежуточные переменные в формулах разложения Шеннона, получает логические уравнения, каждое из которых содержит не более k различных булевых переменных. Критерием оптимизации является минимум числа таких уравнений.

Преобразование многоуровневых представлений булевых функций в двухуровневые матричные формы (системы ДНФ) осуществляется программным модулем Eliminate [24], который устраняет (элиминирует) все промежуточные внутренние переменные в исходных многоуровневых описаниях.

Конвертирование функциональных описаний систем булевых функций с языка SF в описания на VHDL и обратно выполняется программным модулем Kiscvt [29]. Этот модуль используется для преобразования входных данных с последующей логической оптимизацией в системе FLC-2 и для преобразования оптимизированных SF-описаний в VHDL-описания с целью их схемной реализации в зарубежных САПР.

Проиллюстрируем предлагаемый подход на примере системы ДНФ булевых функций

$$\begin{aligned} f^1 &= x_1 \bar{x}_2 \bar{x}_4 \bar{x}_5 \bar{x}_6 \vee \bar{x}_1 \bar{x}_4 \bar{x}_5 \bar{x}_6 \vee x_2 \bar{x}_3 \bar{x}_5, \\ f^2 &= \bar{x}_1 \bar{x}_4 \bar{x}_5 \bar{x}_6 \vee \bar{x}_1 \bar{x}_3 \bar{x}_5 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_5 \bar{x}_6 \vee x_1 \bar{x}_2 \bar{x}_4 \bar{x}_5 \bar{x}_6, \\ f^3 &= x_1 \bar{x}_2 \bar{x}_3 \bar{x}_6 \vee x_1 \bar{x}_2 \bar{x}_4 \bar{x}_6 \vee x_1 \bar{x}_3 \bar{x}_4 \bar{x}_6 \vee \bar{x}_1 \bar{x}_2 \bar{x}_4 \bar{x}_5 \bar{x}_6 \vee x_1 \bar{x}_2 \bar{x}_5 \vee x_2 \bar{x}_3 \bar{x}_5, \end{aligned}$$

заданных в матричной форме (табл. 1). Пусть требуется реализовать эту систему функций логической сетью в базисе LUT-4 ($k = 4$). Будем использовать для логической оптимизации системы ДНФ булевых функций (табл. 1) сначала программу BDD_Builder, а затем программу Presin.

Таблица 1
Пример системы ДНФ трех булевых функций

T^x	B^f
$x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6$	$f^1 \ f^2 \ f^3$
1 1 - 0 1 0	1 0 0
0 - - 1 0 1	1 0 0
0 - - 0 1 0	0 1 0
0 - 0 - 1 -	0 1 0
1 1 1 - 1 0	0 1 0
1 0 - 1 0 1	0 1 0

Окончание табл. 1

T^x						B^f		
x_1	x_2	x_3	x_4	x_5	x_6	f^1	f^2	f^3
1	0	0	-	-	1	0	0	1
1	0	-	1	-	1	0	0	1
1	-	0	1	-	1	0	0	1
0	1	-	0	1	0	0	0	1
1	0	-	-	1	-	0	0	1
-	1	0	-	1	-	1	0	1

В результате BDD-оптимизации были получены логические формулы многоуровневого представления

$$\begin{aligned}
 f^1 &= \bar{x}_1 \psi^1 \vee x_1 \psi^2; & f^2 &= \bar{x}_1 \varphi^3 \vee x_1 \psi^4; & f^3 &= \bar{x}_1 \psi^2 \vee x_1 \psi^6; \\
 \psi^1 &= \bar{x}_2 s^1 \vee x_2 \varphi^2; & \psi^2 &= x_2 \varphi^3; & \psi^4 &= \bar{x}_2 s^1 \vee x_2 \varphi^4; & \psi^6 &= \bar{x}_2 \varphi^5 \vee x_2 \varphi^6; \\
 \varphi^2 &= \bar{x}_3 s^2 \vee x_3 s^1; & \varphi^3 &= \bar{x}_3 \lambda^3 \vee x_3 s^4; & \varphi^4 &= x_3 \lambda^4; & \varphi^5 &= \bar{x}_3 \lambda^2 \vee x_3 s^2; & \varphi^6 &= \bar{x}_3 s^2; \\
 s^1 &= x_4 \lambda^1; & s^2 &= \bar{x}_4 \lambda^3 \vee x_4 \lambda^2; & s^4 &= \bar{x}_4 \lambda^4; \\
 \lambda^1 &= \bar{x}_5 \omega^1; & \lambda^2 &= \bar{x}_5 \omega^1 \vee x_5; & \lambda^3 &= x_5; & \lambda^4 &= x_5 \omega^2; & \omega^1 &= x_6; & \omega^2 &= \bar{x}_6,
 \end{aligned}
 \tag{1}$$

которому соответствует граф BDD на рис. 1.

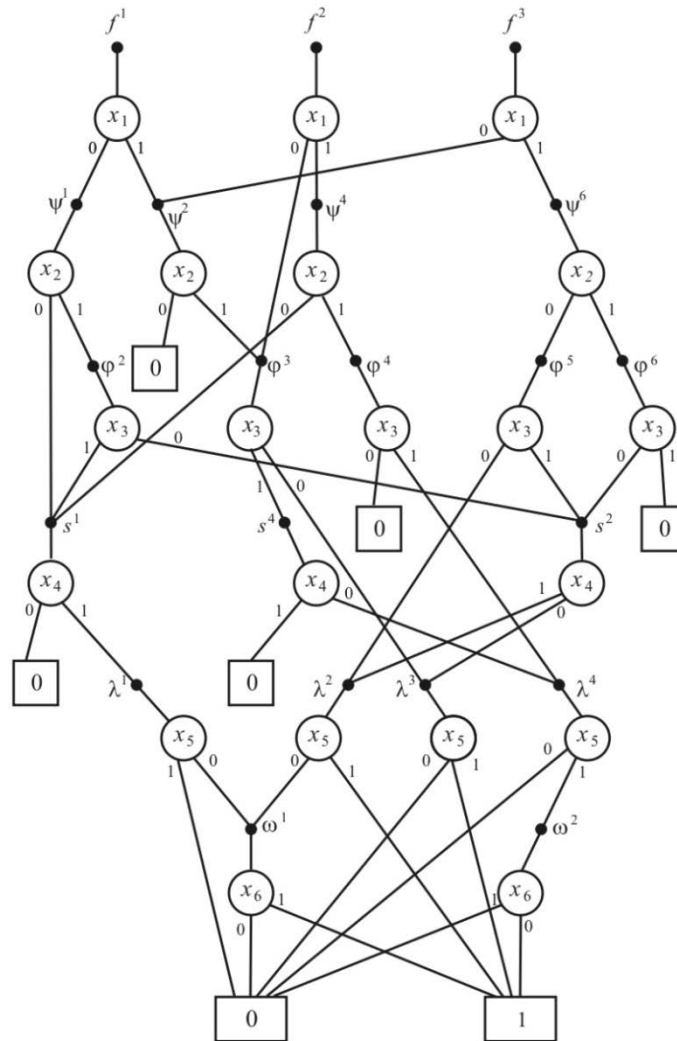


Рис. 1. Многоуровневые графические представления системы функций (1)

В результате укрупнения уравнений (промежуточные переменные $\varphi^2, \varphi^4, \varphi^5, \varphi^6, s^4, \lambda^1, \lambda^3, w^1, w^2$ были элиминированы) программа Presin для $k = 4$ получает 11 логических уравнений:

$$\begin{aligned} f^1 &= \bar{x}_1 \psi^1 \vee x_1 x_2 \varphi^3; & f^2 &= \bar{x}_1 \varphi^3 \vee x_1 \psi^4; & f^3 &= \bar{x}_1 \psi^6 \vee \bar{x}_1 x_2 \varphi^3; \\ \psi^1 &= \bar{s}_1 x_3 \vee x_2 \bar{x}_3 \bar{s}^2 \vee \bar{x}_2 \bar{s}^1; & \varphi^3 &= \bar{x}_3 x_5 \vee x_3 (\bar{x}_4 (x_5 \bar{x}_6)); & \psi^4 &= \bar{x}_2 s^1 \vee x_2 x_3 \lambda^4; \\ \psi^6 &= x_2 \bar{x}_3 \bar{s}^2 \vee \bar{x}_2 \bar{x}_3 \lambda^2 \vee x_2 x_3 s^2; & s^1 &= x_4 (\bar{x}_5 x_6); & \lambda^4 &= x_5 \bar{x}_6; & s^2 &= \bar{x}_4 x_5 \vee x_4 (\bar{x}_5 x_6 \vee x_5); & \lambda^2 &= \bar{x}_5 x_6 \vee x_5, \end{aligned} \quad (2)$$

каждое из которых может быть реализовано на одном LUT-4 (рис. 2).

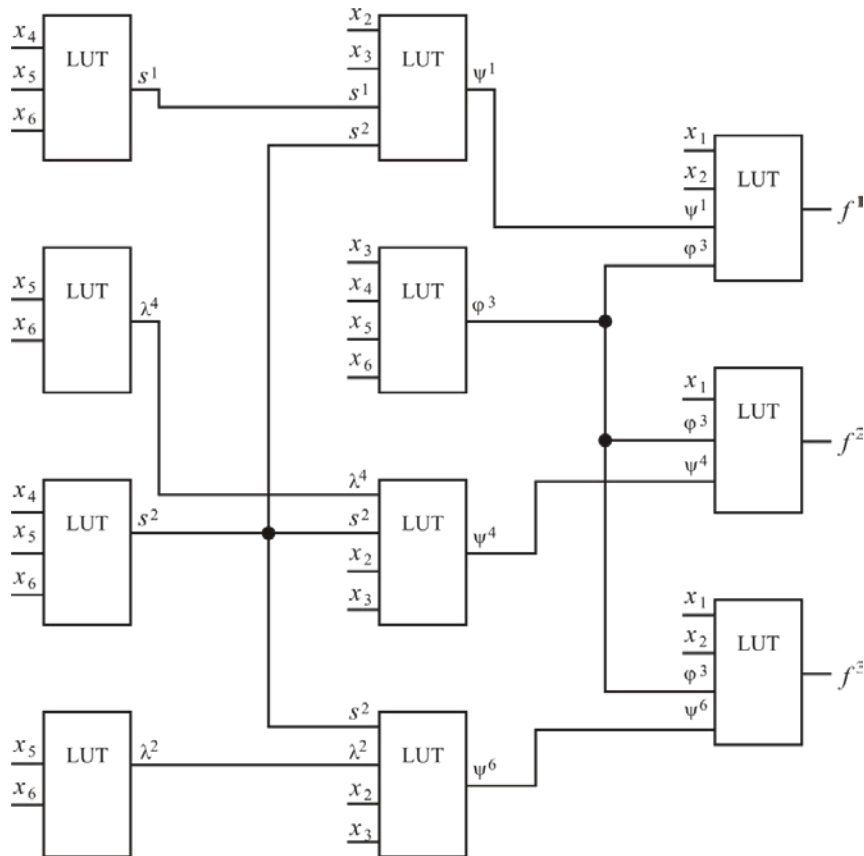


Рис. 2. Схемная реализация логических уравнений (2) базовыми LUT-4

Наборы примеров для экспериментов. Все исходные описания примеров блоков комбинационной логики для экспериментальных исследований задавали различные формы систем полностью определенных булевых функций. Для представления примеров были использованы языки VHDL [10] и SF [24]. VHDL является входным языком зарубежных систем проектирования, SF – языком представления входных и выходных данных в системе FLC-2.

Приняты следующие обозначения: n – число входных переменных, m – число выходных переменных, LUT- k – число элементов (сложность) FPGA, Delay – задержка схемы в наносекундах (нс).

Набор примеров 1 – матричные описания систем ДНФ булевых функций, взятые из библиотеки примеров [31].

Набор примеров 2 – многоуровневые описания блоков комбинационной логики в виде взаимосвязанных уравнений из библиотеки примеров LGSynth'89 (URL: <https://ddd.fit.cvut.cz/prj/Benchmarks>). Каждое из уравнений задавалось в виде ДНФ.

Набор примеров 3 – псевдослучайные системы ДНФ, матричные описания которых генерировались в системе FLC-2. Псевдослучайные примеры систем ДНФ характеризуются различ-

ными средними значениями числа конъюнкций, числа литералов в конъюнкциях и средними значениями числа вхождений конъюнкций в ДНФ функций системы (n_1 – среднее число литералов в элементарной конъюнкции; m_1 – среднее число вхождений конъюнкции в ДНФ функций системы). Для первого псевдослучайного примера GenP_1 число n столбцов троичной матрицы T^x , задающей конъюнкции, равно 15; число k строк матриц T^x , B^f – 664; среднее число n_1 определенных (0, 1) элементов в строке матрицы T^x – 10; среднее число m_1 единичных значений в строке матрицы B^f – 3 (табл. 2).

Таблица 2

Параметры примеров псевдослучайных ДНФ систем булевых функций

Пример	n	n_1	m	m_1	k
GenP_1	15	10	7	3	664
GenP_2	20	15	10	3	400
GenP_3	30	20	10	3	100
GenP_4	40	30	10	3	100
GenP_5	20	15	10	3	100
GenP_6	20	10	10	3	100
GenP_7	20	5	10	3	100
GenP_8	20	20	10	3	100

Экспериментальные исследования. Для проверки эффективности влияния алгоритмов логической минимизации на сложность (площадь) логических схем FPGA были проведены вычислительные эксперименты. Для каждого из трех наборов примеров был проведен эксперимент по схемной реализации в двух системах проектирования схем FPGA для двух семейств FPGA с архитектурами на основе программируемых элементов LUT-6. Синтез схем FPGA по VHDL-описаниям выполнялся в синтезаторе LeonardoSpectrum [10] версии 2010a.7 при одних и тех же режимах (опциях) синтеза и в синтезаторе ISE Xilinx (версии 13.1) при установках по умолчанию. Сложность реализации схемы подсчитывалась в числе LUT-6 для двух семейств FPGA: Virtex-II PRO и Virtex-5 [30]. Каждая из этих FPGA имеет LUT-6 в качестве базовых программируемых элементов. Число таких элементов, содержащихся в схеме, выдают синтезаторы LeonardoSpectrum и ISE Xilinx после выполнения этапа логического проектирования.

Эксперимент 1. Схемная реализация на FPGA матричных описаний систем ДНФ булевых функций. Этапы выполнения эксперимента 1 показаны на рис. 3, результаты представлены в табл. 3–5, где жирным шрифтом выделены лучшие по числу базовых LUT-6 решения.

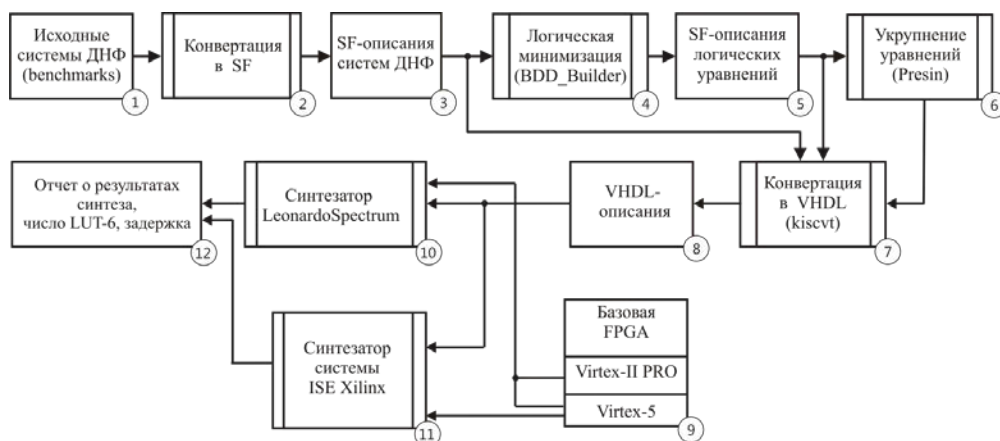


Рис. 3. Этапы эксперимента 1

Таблица 3

Схемная реализация систем ДНФ на FPGA Virtex-II PRO в синтезаторе LeonardoSpectrum

Пример	n	m	Исходное VHDL-описание		BDDI		BDDI и Presin, $k = 6$		Число уравнений, $k = 6$
			LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	
SQR6	6	12	137	15	24	8	38	11	30
SQN	7	3	96	16	20	9	49	13	20
rd73	7	3	88	13	21	10	30	14	15
Radd	8	5	99	14	13	8	18	9	14
Root	8	5	178	20	37	12	62	12	33
m2	8	16	199	17	56	11	97	14	57
m3	8	16	225	17	69	12	130	15	66
dc2	8	7	61	14	29	9	42	14	24
Dist	8	5	352	19	67	13	126	15	66
ADR4	8	5	367	17	9	8	19	11	12
z9sym	9	1	245	17	18	13	26	13	9
ADDM4	9	8	689	20	88	13	198	21	92
LIFE	9	1	77	18	19	11	48	17	24
MAX512	9	6	545	17	106	14	207	20	104
MAX1024	10	6	1006	24	193	14	345	17	190
SYM10	10	1	670	20	20	13	33	14	16
ADD6	12	7	1194	22	20	10	57	14	26
ALU1	12	8	8	5	8	5	8	5	8
BR1	12	8	63	13	57	15	70	15	47
BR2	12	8	66	16	37	12	55	14	35
T3	12	8	76	15	31	12	39	13	32
tial	14	8	678	23	527	27	657	21	352
B12	15	9	94	13	27	10	29	10	24
gary	15	11	224	16	190	15	225	19	118
M181	15	9	98	13	27	10	31	9	24
intb	15	7	921	23	550	24	775	28	431
b2	16	17	472	19	439	21	612	20	369
RYY6	16	1	7	9	6	9	6	8	4
IN2	19	10	287	17	168	21	278	23	146
Sist4	25	20	453	24	213	21	349	23	166
vtx1	27	6	52	12	49	14	93	19	48
x9dn	27	7	78	15	50	15	103	18	59
X1	51	35	254	21	217	18	418	25	216
soar	83	94	349	14	250	15	312	15	216

Таблица 4

Схемная реализация систем ДНФ на FPGA Virtex-5 в синтезаторе LeonardoSpectrum

Пример	n	m	Исходное VHDL-описание		BDDI		BDDI и Presin, $k = 6$		Число уравнений, $k = 6$
			LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	
SQR6	6	12	118	6	22	5	22	5	30
SQN	7	3	126	6	22	5	56	6	20
rd73	7	3	212	6	21	5	28	5	15
radd	8	5	116	6	16	5	25	5	14
root	8	5	212	6	50	5	93	6	33
m2	8	16	243	6	61	5	130	5	57
m3	8	16	315	6	74	5	163	6	66
dc2	8	7	85	6	28	5	41	5	24
dist	8	5	452	7	80	5	162	6	66
ADR4	8	5	559	6	16	5	28	5	12
z9sym	9	1	304	6	19	5	23	5	9
ADDM4	9	8	1031	7	98	6	239	6	92
LIFE	9	1	75	6	25	5	56	6	24
MAX512	9	6	728	7	132	5	292	6	104
MAX1024	10	6	1388	7	247	5	529	6	190
SYM10	10	1	1045	7	21	5	37	6	16

Окончание табл. 4

Пример	n	m	Исходное VHDL-описание		BDDI		BDDI и Presin, $k = 6$		Число уравнений, $k = 6$
			LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	
ADD6	12	7	1705	7	23	5	76	6	26
ALU1	12	8	8	4	8	4	8	4	8
BR1	12	8	71	6	72	5	87	5	47
BR2	12	8	88	6	51	5	68	5	35
T3	12	8	91	6	47	5	61	6	32
tial	14	8	1072	7	800	7	981	6	352
B12	15	9	124	6	32	5	34	5	24
gary	15	11	285	6	214	6	336	6	118
M181	15	9	125	6	32	5	34	5	24
intb	15	7	1397	7	864	7	1181	7	431
b2	16	17	629	7	707	6	947	6	369
RYY6	16	1	13	5	7	6	7	5	4
IN2	19	10	401	6	225	6	370	7	146
Sist4	25	20	637	6	254	6	436	7	166
vtx1	27	6	74	5	62	6	124	6	48
x9dn	27	7	102	6	64	6	137	6	59
X1	51	35	340	6	340	6	590	7	216
soar	83	94	455	6	294	6	409	6	216

Таблица 5

Схемная реализация систем ДНФ на FPGA Virtex-5 в синтезаторе ISE Xilinx

Пример	n	m	Исходное VHDL-описание		BDDI		Минимизация (MINIM)		Число уравнений, $k = 6$
			LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	
m3	8	16	43	5,826	40	4,965	41	4,966	66
dist	8	5	29	6,008	43	6,006	36	6,154	66
ADR4	8	5	6	4,632	6	4,632	6	4,632	12
z9sym	9	1	9	5,790	9	5,790	9	5,790	9
ADDM4	9	8	47	6,126	43	5,658	31	5,491	92
MAX512	9	6	42	5,614	57	6,534	60	6,427	104
MAX1024	10	6	121	6,867	116	6,738	122	7,068	190
SYM10	10	1	15	5,962	15	5,962	15	6,044	16
ADD6	12	7	10	4,701	9	5,198	10	4,701	26
tial	14	8	290	8,077	276	9,046	299	8,270	352
gary	15	11	116	9,149	96	6,891	101	6,808	118
IN0	15	11	100	6,658	95	7,040	112	8,211	118
intb	15	7	292	8,167	348	8,817	297	8,112	431
b2	16	17	326	7,603	295	7,504	274	8,328	369
IN2	19	10	11	6,990	86	6,854	92	6,658	146
Sist4	25	20	95	7,595	85	7,379	77	7,495	166
X1	51	35	80	5,842	82	5,663	80	5,842	216
soar	83	94	150	6,600	159	7,023	152	6,482	216

Эксперимент 2. Схемная реализация на FPGA многоуровневых описаний систем булевых функций. Начальные этапы выполнения эксперимента 2 показаны на рис. 4, после них осуществляется переход на этап 3 либо 7 эксперимента 1 (рис. 3). Например, после выполнения элиминации промежуточных переменных получаются системы ДНФ булевых функций (см. блок 3 на рис. 3). Многоуровневая логика реализуется на FPGA без логической минимизации (см. переход на блок 7, рис. 3) либо сводится к представлению в виде системы ДНФ и реализуется далее по маршруту на рис. 3. Результаты приведены в табл. 6–8.



Рис. 4. Начальные этапы эксперимента 2

Таблица 6

Схемная реализация многоуровневых представлений систем булевых функций на FPGA Virtex-II PRO в синтезаторе LeonardoSpectrum

Пример	n	m	Исходное многоуровневое VHDL-описание		VHDL-описание системы ДНФ		BDDI		BDDI и Presin, $k = 6$		Число уравнений, $k = 6$
			LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	
B9	16	5	36	10	75	14	34	12	74	16	42
TTT2	24	21	105	15	178	14	81	14	126	15	65
UNREG	36	16	32	8	34	8	32	17	33	8	16
TOO_LARGE	38	3	1664	25	1361	24	983	32	1749	42	833
C880	60	26	47	15	38	11	53	22	92	21	78
X4	94	71	166	14	221	16	262	17	377	18	195
I8	133	81	426	23	251	18	385	21	531	20	340
X3	135	99	391	14	562	18	448	24	703	24	352
I7	199	67	32	8	32	8	32	8	32	8	67

Таблица 7

Схемная реализация многоуровневых представлений систем булевых функций на FPGA Virtex-5 в синтезаторе LeonardoSpectrum

Пример	n	m	Исходное многоуровневое VHDL-описание		VHDL-описание системы ДНФ		BDDI		BDDI и Presin, $k = 6$		Число уравнений, $k = 6$
			LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	
B9	16	5	48	5	114	6	38	5	97	6	42
TTT2r	24	21	137	6	261	6	113	5	174	6	65
UNREG	36	16	63	5	51	5	51	5	51	5	16
TOO_LARGE	38	3	2550	7	2056	7	1599	8	2808	9	833
C880	60	26	59	6	52	6	55	7	130	6	78
X4	94	71	288	5	321	5	333	6	507	6	195
I8r	133	81	709	6	362	6	376	6	709	6	340
X3	135	99	469	6	770	6	555	7	902	7	352
I7	199	67	31	4	31	4	31	4	31	4	67

Таблица 8

Схемная реализация многоуровневых представлений систем булевых функций на FPGA Virtex-5 в синтезаторе ISE Xilinx

Пример	n	m	Исходное многоуровневое VHDL-описание		VHDL-описание системы ДНФ		BDDI		Число уравнений, $k = 6$
			LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	
B9	16	5	25	5,517	20	5,948	15	5,775	42
TTT2	24	21	43	5,638	46	6,238	51	6,604	65
UNREG	36	16	16	4,180	16	4,180	16	4,180	16
TOO_LARGE	38	3	113	7,880	107	8,080	579	12,818	833
C880	60	26	34	6,590	33	6,680	34	7,110	78
X4	94	71	113	6,548	122	5,584	160	6,500	195
I8	133	81	166	7,690	167	8,045	227	7,366	340
X3	135	99	180	6,471	202	6,490	227	7,466	352
I7	199	67	31	4,136	31	4,136	31	4,136	67

Эксперимент 3. Схемная реализация на FPGA псевдослучайных систем ДНФ. Начальные этапы выполнения эксперимента показаны на рис. 5, после них осуществляется переход на этап 4 либо 7 эксперимента 1 (см. рис. 3). Результаты приведены в табл. 9–11.

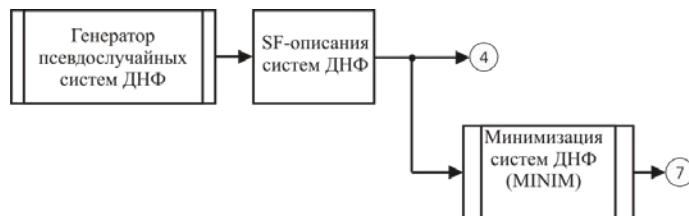


Рис. 5. Начальные этапы эксперимента 3

Таблица 9

Схемная реализация псевдослучайных систем ДНФ на FPGA Virtex-II PRO в синтезаторе LeonardoSpectrum

Пример	Синтез по исходным VHDL-описаниям		Синтез после отдельной минимизации функций в классе ДНФ, программа из работы [26]		Синтез после совместной минимизации функций в классе ДНФ, программа Espresso [32]	
	LUT-6	Delay	LUT-6	Delay	LUT-6	Delay
GenP_1	1624	21	1865	22	1430	21
GenP_2	1382	19	1505	18	1388	19
GenP_3	582	17	583	16	588	17
GenP_4	833	17	830	16	829	16
GenP_5	439	15	442	14	438	16
GenP_6	395	16	392	16	391	16
GenP_7	249	14	246	15	232	14
GenP_8	456	17	458	16	464	16

Таблица 10

Схемная реализация псевдослучайных систем ДНФ на FPGA Virtex-5 в синтезаторе LeonardoSpectrum

Пример	Исходное VHDL-описание		BDDI		BDDI и Presin, $k = 6$		Число уравнений, $k = 6$
	LUT-6	Delay	LUT-6	Delay	LUT-6	Delay	
GenP_1	2146	7	11730	6	15186	8	6153
GenP_2	1628	7	20225	7	24482	7	15286
GenP_3	675	6	4218	8	8183	8	2735
GenP_4	941	6	3023	8	6766	8	2118
GenP_5	499	6	3984	8	6393	8	3833
GenP_6	435	6	8186	8	12425	8	6248
GenP_7	282	6	14630	8	18957	8	521
GenP_8	543	7	649	7	968	7	3629

Таблица 11

Схемная реализация псевдослучайных систем ДНФ на FPGA Virtex-5 в синтезаторе ISE Xilinx

Пример	Исходное VHDL-описание		BDDI	
	LUT-6	Delay	LUT-6	Delay
GenP_1	988	9,299	6702	10,637
GenP_2	876	9,385	14278	12,325
GenP_3	319	8,171	1699	11,117
GenP_4	437	8,798	1471	11,677
GenP_5	249	8,277	1685	10,352
GenP_6	218	7,756	3307	11,172
GenP_7	148	7,022	5308	12,112
GenP_8	295	8,938	360	8,638

Эксперимент 4. Схемная реализация примеров из трех наборов в системе проектирования Vivado (компания Xilinx). Целью эксперимента является сравнение результатов синтеза по исходным описаниям с результатами синтеза по минимизированным многоуровневым BDDI-представлениям, полученным программой BDD_Builder. Результаты исследований для микросхемы xa712tсpg238-2I семейства Artix-7 приведены в табл. 12.

Таблица 12
Результаты эксперимента 4 на FPGA Artix-7
в синтезаторе Vivado Xilinx

Пример	Исходное VHDL-описание		BDDI	
	LUT-6	Delay	LUT-6	Delay
<i>Набор примеров 1 (системы ДНФ)</i>				
m3	36	7,744	25	7,574
dist	20	7,023	20	6,876
ADR4	7	7,716	4	7,118
z9sym	6	7,425	6	7,015
ADDM4	45	8,064	27	7,408
MAX512	37	8,294	38	7,765
MAX1024	72	8,633	73	8,260
SYM10	9	7,360	9	7,483
ADD6	9	7,780	6	7,526
tial	288	12,356	199	11,114
gary	109	9,799	72	8,945
IN0	105	9,628	72	8,945
intb	270	11,561	274	11,170
b2	191	11,083	267	11,329
IN2	87	9,741	73	9,315
Sist4	121	9,752	71	9,276
X1	73	10,730	72	10,419
soar	137	–	134	–
<i>Набор примеров 2 (многоуровневые описания)</i>				
B9	23	10,196	16	8,081
TTT2	37	9,783	37	8,105
UNREG	16	8,563	16	8,545
TOO_LARGE	98	9,887	547	16,460
C880	25	9,783	24	9,745
X4	84	–	101	–
I8	165	–	165	–
X3	152	–	178	–
I7	30	–	30	–
<i>Набор примеров 3 (псевдослучайные системы ДНФ)</i>				
GenP_1	1155	18,241	1690	18,981
GenP_2	915	16,886	10584	–
GenP_3	341	12,654	436	13,008
GenP_4	515	13,904	599	14,229
GenP_5	288	11,661	322	11,046
GenP_6	255	11,767	274	11,431
GenP_7	142	10,637	147	10,317
GenP_8	294	12,080	296	12,499

Для набора примеров 2 (многоуровневых описаний систем булевых функций) в эксперименте 4 осуществлялся переход к матричным представлениям, которые обрабатывались программой BDD_Builder. Микросхемы xa712tсpg238-2I содержат 112 информационных входных и выходных полюсов, поэтому в табл. 12 для проектов soar, X1, I8, X3, I7, для которых суммарное число входных и выходных переменных превышает 112, топологическая реализация (имплементация) была невозможна. В связи с этим задержки схем (Delay) не были посчитаны. Для проекта GenP_2 топологическая реализация также не была проведена из-за большого числа (10 584) LUT-6.

Таблица 13

Сравнение предложенного подхода с исследовательскими программами

Пример	n	m	Исходное VHDL-описание	BDDI	Результаты из работы [18]	Пример	n	m	BDDI и Presin, $k = 5$	Результаты из работы [19]
			LUT-6	LUT-6	LUT-6				LUT-5	LUT-5
B12	15	9	18	18	20	ALU2	10	6	35	41
F51M	8	8	11	13	10	ALU4	14	8	42	190
Pcle	19	9	11	12	12	B9	16	5	51	40
RD73	7	3	6	6	6	C880	60	26	87	103
SQN	7	3	6	6	9	Count	35	16	24	26
SQR6	6	12	10	10	10	Z4ml	7	4	8	5
X2	10	7	7	7	11					
Z5xp1	7	10	13	16	13					

В табл. 13 представлены результаты сравнения предложенного подхода с данными, полученными в работах [18, 19]. Сравнение приведено для табл. 2 (см. столбец MultiDec и ISE) из работы [18], где опубликованы результаты для промышленных примеров небольшой размерности. Приведенные в правой части табл. 13 результаты взяты из табл. 7 (см. столбец BDS-pga и FlowMap) работы [19], они даны для LUT-5. Однако, как показывает анализ современных архитектур FPGA, при создании новых семейств был совершен переход от архитектур с LUT-4 к архитектурам FPGA с LUT-6, поэтому результаты работы [19] представляют только научный интерес.

Заключение. Анализ результатов исследований позволяет сделать следующие выводы. Синтезатор системы ISE Xilinx (версии 13.1) получает лучшие результаты синтеза как для микросхем семейства Virtex-II PRO, так и для микросхем семейства Virtex-5 по сравнению с синтезатором LeonardoSpectrum (версии 2010a.7).

Применение предварительной BDDI-минимизации и синтез по минимизированным BDDI-представлениям практически всегда дают возможность улучшить результаты синтеза в синтезаторе LeonardoSpectrum.

Синтезатор ISE Xilinx является эффективным инструментом синтеза. Как показывают результаты эксперимента 1 (см. табл. 5), только в половине случаев предварительная логическая минимизация позволяет улучшить результаты синтеза на первом наборе примеров.

Синтез по укрупненным логическим уравнениям, полученным программой Presin из BDDI-представлений, не позволяет улучшить результаты синтеза в обоих синтезаторах. Число укрупненных уравнений может служить оценкой сложности схемы в базе LUT- k на этапе логического проектирования. Такая оценка возможна в условиях, когда не принимаются во внимание ни топологические аспекты (разводка соединений) проектирования, ни энергопотребление (нагрузочные способности элементов) проектируемых схем FPGA.

Реализация в LeonardoSpectrum многоуровневых описаний (эксперимент 2) может быть эффективной при замене исходного многоуровневого описания двухуровневым описанием (системой ДНФ) и применении как BDDI-минимизации, так и минимизации в классе ДНФ (см. табл. 7).

Для испытанных псевдослучайных примеров систем функций применение минимизации в классе ДНФ практически не дает эффекта, а применение BDDI-минимизации значительно ухудшает результаты синтеза.

Сравнение результатов синтеза схем в системах ISE и Vivado (компания Xilinx) показывает, что алгоритмы синтеза в системе Vivado являются более эффективными по сравнению с алгоритмами, реализованными в системе ISE. Это касается результатов синтеза как по исходным описаниям, так и по оптимизированным BDDI-представлениям систем функций. В системе Vivado улучшены алгоритмы трассировки для FPGA новых семейств [33]. Она поддерживает только новые семейства FPGA, на одном из которых (Artix-7) было проведено сравнение слож-

ности схем. Разница в результатах синтеза в системах ISE и Vivado особенно проявляется на примерах псевдослучайных систем ДНФ булевых функций, преимущества той или иной системы проектирования могут быть значительными. Здесь прослеживается общая закономерность, когда появление новых версий САПР приводит к изменению результатов проектирования: для одних проектов результаты в новых версиях САПР улучшаются, а для других могут быть ухудшены. Например, в работе [34] были изучены девять версий синтезатора LeonardoSpectrum и подтверждена аналогичная ситуация при оценке эффективности получаемых решений (проектов логических схем) по параметрам площади и быстродействия.

Для промышленных примеров схем как для системы ISE, так и для Vivado переход от систем ДНФ к многоуровневым оптимизированным BDDI-представлениям чаще всего позволяет уменьшить сложность схем, оцениваемых в числе LUT-6.

Предложенный в настоящей работе подход является более конкурентоспособным по сравнению с зарубежными исследовательскими программами, результаты сравнения на примерах из работ [18, 19] приведены в табл. 13.

Разработанные программы технологически независимой оптимизации являются эффективными, прошли экспериментальную проверку на примерах схем практической размерности и включены в отечественную систему логической оптимизации функционально-структурных описаний цифровых устройств. Использование отечественных программ логической оптимизации позволяет во многих случаях улучшить результаты синтеза комбинационных структур в зарубежных САПР реализации проектов цифровых устройств на FPGA.

Список использованных источников

1. Зотов, Ю. В. Проектирование цифровых устройств на основе ПЛИС фирмы XILINX в САПР WebPack ISE / Ю. В. Зотов. – М. : Горячая линия – Телеком, 2003. – 624 с.
2. Designing with Xilinx® FPGAs: Using Vivado / ed. S. Churiwala. – Springer, 2017. – 260 p.
3. Реконфигурируемые мультимониторные вычислительные структуры / И. А. Каляев [и др.] ; под общ. ред. И. А. Каляева. – Ростов н/Д : Изд-во ЮНЦ РАН, 2008. – 320 с.
4. Nakahara, H. A deep convolutional neural network based on nested residue number system / H. Nakahara, T. Sasao // 25th Intern. Conf. on Field Programmable Logic and Applications (FPL), Lausanne, 2–4 September 2015. – Lausanne, 2015. – P. 1–6.
5. Петровский, Ал. А. Быстрое проектирование систем мультимедиа от прототипа / Ал. А. Петровский, А. В. Станкевич, А. А. Петровский. – Минск : Бестпринт, 2011. – 410 с.
6. Соловьев, В. В. Архитектуры ПЛИС фирмы Xilinx: FPGA и CPLD 7-й серии / В. В. Соловьев. – М. : Горячая линия – Телеком, 2016. – 392 с.
7. Бибило, П. Н. Синтез комбинационных схем методами функциональной декомпозиции / П. Н. Бибило, С. В. Енин. – Минск : Наука и техника, 1987. – 189 с.
8. Sasao, T. FPGA design by generalized functional decomposition / T. Sasao // Representations of Discrete Functions / eds.: T. Sasao, M. Fujita. – Kluwer Academic Publishers, 1996. – P. 233–258.
9. Sasao, T. Memory-Based Logic Synthesis / T. Sasao. – N. Y. : Springer, 2011. – 189 p.
10. Бибило, П. Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum / П. Н. Бибило. – М. : СОЛОН-Пресс, 2005. – 384 с.
11. MIS: a multiple-level logic optimization systems / R. K. Brayton [et al.] // IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems. – 1987. – Vol. 6, iss. 6. – P. 1062–1081.
12. Chang, S.-C. Technology mapping for TLU FPGA's based on decomposition of binary decision diagrams / S.-C. Chang, M. Marek-Sadowska, T. Hwang // IEEE Transactions Computer-Aided Design of Integrated Circuits and Systems. – 1996. – Vol. 15, no. 10. – P. 1226–1235.
13. Meinel, C. Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications / C. Meinel, T. Theobald. – Berlin, Heidelberg : Springer-Verlag, 1998. – 267 p.
14. Ebdndt, R. Advanced BDD Optimization / R. Ebdndt, G. Fey, R. Drechsler. – Springer, 2005. – 222 p.
15. Scholl, C. Functional Decomposition with Applications to FPGA Synthesis / C. Scholl. – Boston : Kluwer Academic Publishers, 2001. – 288 p.
16. Chen, D. FPGA design automation: a survey / D. Chen, J. Cong, P. Pan // Foundations and Trends in Electronic Design Automation. – 2006. – Vol. 1, no. 3. – P. 195–330.
17. Kubica, M. SMTBDD: New form of BDD for logic synthesis / M. Kubica, D. Kania // Intern. J. of Electronics and Telecommunications. – 2016. – Vol. 62, no. 1. – P. 33–41.

18. Kubica, M. Decomposition of multi-output functions oriented to configurability of logic blocks / M. Kubica, D. Kania // *Bulletin of the Polish Academy of Sciences. Technical Sciences.* – 2017. – Vol. 65, no. 3. – P. 317–331.
19. Vemuri, N. BDD-based logic synthesis for LUT-6-based FPGAs / N. Vemuri, P. Kalla, R. Tessier // *ACM Transactions on Design Automation of Electronic Systems.* – 2002. – Vol. 7, no. 4. – P. 501–525.
20. Yang, S. BDS: a BDD-based logic optimization system / S. Yang, M. Ciesielski // *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* – 2002. – Vol. 21, no. 7. – P. 866–876.
21. Lin, H.-P. Ashenurst decomposition using SAT and interpolation / H.-P. Lin, J.-H. R. Jiang, R.-R. Lee ; eds.: S. P. Khatri, K. Gulati // *Advanced Techniques in Logic Synthesis, Optimizations and Applications.* – Springer, 2010. – P. 67–85.
22. Ashenurst, R. L. The decomposition of switching functions / R. L. Ashenurst // *Annals of Computation Laboratory of Harvard University.* – Cambridge, Mass., 1959. – Vol. 29. – P. 74–116.
23. *Handbook of Satisfiability* / ed. A. Biere [et al.]. – IOS Press, 2009. – 980 p.
24. Бибило, П. Н. Логическое проектирование дискретных устройств с использованием производственно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов. – Минск : Беларус. навука, 2011. – 279 с.
25. Авдеев, Н. А. Эффективность логической оптимизации при синтезе комбинационных схем из библиотечных элементов / Н. А. Авдеев, П. Н. Бибило // *Микроэлектроника.* – 2015. – Т. 44, № 5. – С. 383–399.
26. Торопов, Н. Р. Минимизация систем булевых функций в классе ДНФ / Н. Р. Торопов // *Логическое проектирование.* – Минск : Ин-т техн. кибернетики НАН Беларуси, 1999. – Вып. 4. – С. 4–19.
27. Бибило, П. Н. Использование полиномов Жегалкина при минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона / П. Н. Бибило, Ю. Ю. Ланкевич // *Программная инженерия.* – 2017. – № 8. – С. 369–384.
28. Бибило, П. Н. Оптимизация многоуровневых представлений систем булевых функций при перепроектировании логических схем / П. Н. Бибило, В. И. Романов // *Управляющие системы и машины.* – 2006. – № 5. – С. 20–29.
29. Черемисинов, Д. И. Анализ и преобразование структурных описаний СБИС / Д. И. Черемисинов. – Минск : Беларус. навука, 2006. – 275 с.
30. Кузелин, О. М. Современные семейства ПЛИС фирмы Xilinx : справ. пособие / О. М. Кузелин, Д. А. Кнышев, Ю. В. Зотов. – М. : Горячая линия – Телеком, 2004. – 440 с.
31. Jeong, C. Computer-aided design of digital systems / C. Jeong // *Department of Computer Science [Electronic resource].* – Mode of access: <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex>. – Date of access: 20.03.2018.
32. *Logic Minimization Algorithm for VLSI Synthesis* / K. R. Brayton [et al.]. – Boston : Kluwer Academic Publishers, 1984. – 193 p.
33. Тарасов, И. Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования / И. Е. Тарасов. – М. : Горячая линия – Телеком, 2020. – 538 с.
34. Авдеев, Н. А. Эффективность проектирования заказных схем в синтезаторе LeonardoSpectrum / Н. А. Авдеев, П. Н. Бибило // *Современная электроника.* – 2015. – № 1. – С. 58–61.

References

1. Zotov Yu. V. *Proektirovanie cifrovyyh ustrojstv na osnove PLIS firmy XILINX v SAPR WebPack ISE. The Design of Digital Devices Based on FPGA in XILINX ISE WebPack CAD.* Moscow, Goryachaya liniya – Telekom, 2003, 624 p. (in Russian).
2. *Designing with Xilinx FPGAs. Using Vivado.* In Churiwala S. (ed.). Springer, 2017, 260 p.
3. Kalyaev I. A., Levin I. I., Semernikov E. A., Shmojlov V. I. Rekonfiguriruemye mul'tikonvejernye vychislitel'nye struktury. *Multiconference Reconfigurable Computing Structures.* In Kalyaev I. A. (ed.). Rostov-on-Don, Izdatel'stvo Juzhnogo nauchnogo centra Rossijskoj akademii nauk, 2008, 320 p. (in Russian).
4. Nakahara H., Sasao T. A deep convolutional neural network based on nested residue number system. *25th International Conference on Field Programmable Logic and Applications (FPL), Lausanne, 2–4 September 2015.* Lausanne, 2015, pp. 1–6.
5. Petrovskij Al. A., Stankevich A. V., Petrovskij A. A. Bystroe proektirovanie sistem mul'timedia ot prototipa. *Rapid Design of Multimedia Systems from a Prototype.* Minsk, Bestprint, 2011, 410 p. (in Russian).

6. Solov'ev V. V. Arhitektury PLIS firmy Xilinx: FPGA i CPLD 7-j serii. *XILINX FPGA Architectures: FPGA and CPLD 7-Series*. Moscow, Goryachaya liniya – Telekom, 2016, 392 p. (in Russian).
7. Bibilo P. N., Enin S. V. Sintez kombinacionnyh skhem metodami funkcional'noj dekompozicii. *Synthesis of Combinational Circuits by Methods of Functional Decomposition*. Minsk, Nauka i tekhnika, 1987, 189 p. (in Russian).
8. Sasao T. FPGA design by generalized functional decomposition. *Representations of Discrete Functions*. In Sasao T., Fujita M. (eds.). Kluwer Academic Publishers, 1996, pp. 233–258.
9. Sasao T. *Memory-Based Logic Synthesis*. New York, Springer, 2011, 189 p.
10. Bibilo P. N. Cistemy proektirovaniya integral'nyh skhem na osnove yazyka VHDL. StateCAD, ModelSim, LeonardoSpectrum. *Integrated Circuit Design Systems Based on the VHDL Language*. StateCAD, ModelSim, LeonardoSpectrum. Moscow, SOLON-Press, 2005, 384 p. (in Russian).
11. Brayton R. K., Rudell R., Sangiovanni-Vincentelli A. L., Wang A. R. MIS: A multiple-level logic optimization systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1987, vol. 6, iss. 6, pp. 1062–1081.
12. Chang S.-C., Marek-Sadowska M., Hwang T. Technology mapping for TLU FPGA's based on decomposition of binary decision diagrams. *IEEE Transactions Computer-Aided Design of Integrated Circuits and Systems*, 1996, vol. 15, no. 10, pp. 1226–1235.
13. Meinel C., Theobald T. *Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications*. Berlin, Heidelberg, Springer-Verlag, 1998, 267 p.
14. Ebdndt R., Fey G., Drechsler R. *Advanced BDD Optimization*. Springer, 2005, 222 p.
15. Scholl C. *Functional Decomposition with Application to FPGA Synthesis*. Boston, Kluwer Academic Publisher, 2001, 288 p.
16. Chen D., Cong J., Pan P. FPGA design automation: a survey. *Foundations and Trends in Electronic Design Automation*, 2006, vol. 1, no. 3, pp. 195–330.
17. Kubica M., Kania D. SMTBDD: New form of BDD for logic synthesis. *International Journal of Electronics and Telecommunications*, 2016, vol. 62, no. 1, pp. 33–41.
18. 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, no. 3, pp. 317–331.
19. Vemuri N., Kalla P., Tessier R. BDD-based logic synthesis for LUT-6-based FPGAs. *ACM Transactions on Design Automation of Electronic Systems*, 2002, vol. 7, no. 4, pp. 501–525.
20. Yang S., Ciesielski M. BDS: a BDD-based logic optimization system. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2002, vol. 21, no. 7, pp. 866–876.
21. Lin H.-P., Jiang J.-H. R., Lee R.-R. Ashenurst decomposition using SAT and interpolation. *Advanced Techniques in Logic Synthesis, Optimizations and Applications*. In Khatri S. P., Gulati K. (eds.). Springer, 2010, pp. 67–85.
22. Ashenurst R. L. The decomposition of switching functions. *Annals of Computation Laboratory of Harvard University*. Cambridge, Mass., 1959, vol. 29, pp. 74–116.
23. *Handbook of Satisfiability*. In Biere A., Heule M., Van Maaren H., Walsh T. (eds.). IOS Press, 2009, 980 p.
24. Bibilo P. N., Romanov V. I. Logicheskoe proektirovanie diskretnykh ustrojstv s ispol'zovaniem produkcionno-frejmovoj modeli predstavlenija znaniy. *Logical Design of Discrete Devices Using a Production-Frame Knowledge Representation Model*. Minsk, Belaruskaja navuka, 2011, 279 p. (in Russian).
25. Avdeev N. A., Bibilo P. N. Effektivnost' logicheskoy optimizacii pri sinteze kombinacionnyh skhem iz bibliotechnykh elementov [Logical optimization efficiency in the synthesis of combinational circuits]. *Mikroelektronika [Microelectronics]*, 2015, vol. 44, no. 5, pp. 383–399 (in Russian).
26. Toropov N. R. Minimizaciya sistem bulevykh funkcij v klasse DNF [Minimization of Boolean function systems in the DNF class]. *Logicheskoe proektirovanie [Logical Design]*, Minsk, Institut tehnichejskoj kibernetiki Nacional'noj akademii nauk Belarusi, 1999, iss. 4, pp. 4–19 (in Russian).
27. Bibilo P. N., Lankevich Yu. Yu. Ispol'zovanie polinomov Zhegalkina pri minimizacii mnogourovnevnyh predstavlenij sistem bulevykh funkcij na osnove razlozheniya Shennona [The use of Zhegalkin polynomials for minimization of multilevel representations of Boolean functions based on Shannon expansion]. *Programmnyaya inzheneriya [Software engineering]*, 2017, no. 8, pp. 369–384 (in Russian).
28. Bibilo P. N., Romanov V. I. Optimizaciya mnogourovnevnyh predstavlenij sistem bulevykh funkcij pri pereproektirovanii logicheskikh skhem [Optimization of multi-level representations of systems of Boolean functions in the re-design of logic circuits]. *Upravlyayushchie sistemy i mashiny [Control Systems and Machines]*, 2006, no. 5, pp. 20–29 (in Russian).
29. Cheremisinov D. I. Analiz i preobrazovanie strukturnykh opisaniy SBIS. *Analysis and Transformation of VLSI Structural Descriptions*. Minsk, Belorusskaya nauka, 2006, 275 p. (in Russian).

30. Kuzelin O. M., Knyshev D. A., Zotov Yu. V. Sovremennyye semejstva PLIS firmy Xilinx. *Modern XILINX FPGA Families*. Moscos, Goryachaya liniya – Telekom, 2004, 440 p. (in Russian).
31. Jeong C. Computer-aided design of digital systems. *Department of Computer Science*. Available at: <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex> (accessed 20.03.2018).
32. Brayton K. R., Hachtel G. D., McMullen C., Sangiovanni-Vincentelli A. *Logic Minimization Algorithm for VLSI Synthesis*. Boston, Kluwer Academic Publishers, 1984, 193 p.
33. Tarasov I. E. PLIS Xilinx. Yazyki opisaniya apparatury VHDL i Verilog, SAPR, priemy proektirovaniya. *XILINX FPGA. Hardware Description Languages VHDL and Verilog, CAD, Design Techniques*. Moscow, Goryachaya liniya – Telekom, 2020, 538 p. (in Russian).
34. Avdeev N. A., Bibilo P. N. Effektivnost' proektirovaniya zakaznyh skhem v sintezatore LeonardoSpectrum [Efficiency of custom circuit design in the Leonardo Spectrum synthesizer]. *Sovremennaya elektronika [Modern Electronics]*, 2015, no. 1, pp. 58–61 (in Russian).

Информация об авторах

Бибилу Петр Николаевич, доктор технических наук, профессор, Объединенный институт проблем информатики Национальной академии наук Беларуси. E-mail: bibilo@newman.bas-net.by

Ланкевич Юрий Юрьевич, младший научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси. E-mail: yurafreedom18@gmail.com

Романов Владимир Ильич, кандидат технических наук, доцент, Объединенный институт проблем информатики Национальной академии наук Беларуси. E-mail: rom@newman.bas-net.by

Information about the authors

Petr N. Bibilo, Dr. Sci. (Eng.), Professor, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus. E-mail: bibilo@newman.bas-net.by

Yury Yu. Lankevich, Junior Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus. E-mail: yurafreedom18@gmail.com

Vladimir I. Romanov, Cand. Sci. (Eng.), The United Institute of Informatics Problems of the National Academy of Sciences of Belarus. E-mail: rom@newman.bas-net.by