

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

LOGICAL DESIGN



УДК 519.714.5
DOI: 10.37661/1816-0301-2024-21-4-7-23

Оригинальная статья
Original Article

Алгоритмы выделения из многоуровневого представления системы булевых функций подсистем для совместной минимизации

П. Н. Бибило[✉], Н. А. Кириенко, 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

Цели. Целью экспериментальных исследований является выяснение эффективности новых алгоритмов выделения из формульных описаний исходной системы булевых функций так называемых связанных подсистем. При этом каждая из выделенных подсистем впоследствии минимизируется независимо от других, однако функции, составляющие каждую связанную подсистему, минимизируются совместно.

Методы. Минимизация подсистем выполняется в классе многоуровневых BDD-представлений (BDD – Binary Decision Diagram – бинарная диаграмма решений) либо булевых сетей. После получения минимизированных описаний схем, заданных в виде совокупности взаимосвязанных формул разложения Шеннона, которые соответствуют BDD, либо в виде двухоперандных логических уравнений, соответствующих булевым сетям, выполняется синтез логических схем в библиотеке проектирования заказных цифровых КМОП СБИС (сверхбольших интегральных схем, выполненных по комплементарной технологии металл-оксид-полупроводник). В булевых сетях функциями вершин могут быть логические операции «конъюнкция» либо «дизъюнкция» над литералами булевых переменных. Литерал – это булева переменная либо ее инверсия. Минимизация BDD-представлений осуществляется по числу формул разложения Шеннона, минимизация булевых сетей – по числу литералов в формулах, задающих сети.

Результаты. Полученные логические схемы сравнены по площади кристалла и быстродействию (временной задержке). Эксперименты проведены на 39 промышленных примерах схем. Показано преимущество (в 29 случаях) применения предлагаемых алгоритмов выделения подсистем по сравнению с совместной либо раздельной минимизацией исходной системы булевых функций, которая обычно выполняется в качестве первого этапа синтеза логических схем.

Заключение. Предложенные в работе новые алгоритмы выделения подсистем доказали свою эффективность при выполнении различных программ оптимизации многоуровневых представлений систем булевых функций. Разработанный комплекс программ позволяет улучшать результаты технологически независимой оптимизации, применяемой при реализации проектов цифровых систем в заказных цифровых КМОП СБИС.

Ключевые слова: система булевых функций, дизъюнктивная нормальная форма, бинарная диаграмма решений, булева сеть, синтез логической схемы, VHDL, заказная СБИС

Для цитирования. Бибило, П. Н. Алгоритмы выделения из многоуровневого представления системы булевых функций подсистем для совместной минимизации / П. Н. Бибило, Н. А. Кириенко, В. И. Романов // Информатика. – 2024. – Т. 21, № 4. – С. 7–23. – DOI: 10.37661/1816-0301-2024-21-4-7-23.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 31.07.2024

Подписана в печать | Accepted 30.08.2024

Опубликована | Published 30.12.2024

Algorithms for extracting subsystems from a multilevel representation of a system of Boolean functions for joint minimization

Petr N. Bibilo[✉], Natalia A. Kirienko, 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

Objectives. The purpose of experimental research is to determine the effectiveness of new algorithms for extracting the so-called connected subsystems from formula descriptions of the original system of Boolean functions. Subsequently each of the extracted subsystems is minimized independently of the others, but the functions that make up each connected subsystem are minimized jointly.

Methods. Minimization of subsystems is performed in the class of multilevel BDD representations (BDD – Binary Decision Diagram) or Boolean networks. After obtaining minimized descriptions of circuits, specified as a set of interconnected Shannon expansion formulas that correspond to BDD, or as two-operand logical equations corresponding to Boolean networks, synthesis of logic circuits is carried out in the design library of custom digital CMOS ASIC (Application-Specific Integrated Circuits made using complementary metal oxide semiconductor technology). In Boolean networks, node functions can be the logical operations “conjunction” or “disjunction” over literals of Boolean variables. A literal is a Boolean variable or its inversion. Minimization of BDD representations is carried out according to the number of Shannon decomposition formulas, minimization of Boolean networks – according to the number of literals in the formulas defining the networks.

Results. The resulting logic circuits are compared in terms of chip area and speed (time delay). Experiments were carried out on 39 industrial circuit examples. The advantage (in 29 cases) of using the proposed subsystem extraction algorithms is shown compared to joint or separate minimization of the original system of Boolean functions, which is usually performed as the first stage of the synthesis of logic circuits.

Conclusion. The new algorithms for subsystem extraction proposed in the paper have proven their effectiveness in the execution of various programs for optimizing multilevel representations of systems of Boolean functions. The developed software package allows improving the results of technologically independent optimization used in the implementation of digital system projects in custom digital CMOS ASIC.

Keywords: system of Boolean functions, disjunctive normal form, binary decision diagram, Boolean network, logic synthesis, VHDL, ASIC

For citation. Bibilo P. N., Kirienko N. A., Romanov V. I. *Algorithms for extracting subsystems from a multilevel representation of a system of Boolean functions for joint minimization.* Informatika [Informatics], 2024, vol. 21, no. 4, pp. 7–23 (In Russ.). DOI: 10.37661/1816-0301-2024-21-4-7-23.

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

Введение. Синтез комбинационных логических схем в заданной библиотеке (базисе) проектирования всегда предваряется минимизацией представлений тех систем булевых функций, которые требуется реализовать в схеме. Для некоторых технологических базисов, например программируемых логических матриц (ПЛИМ), минимизация площади ПЛИМ сводится к совместной минимизации системы булевых функций в классе дизъюнктивных нормальных форм (ДНФ) [1, 2]. Если ПЛИМ имела выходной буфер с инверторами, то поляризация функций (выбор безынверсной либо инверсной формы) и последующая совместная минимизация позволяли дополнительно сокращать площадь ПЛИМ [3, 4], реализуемой в составе заказных СБИС. Минимизация числа вершин в графах BDD [5, 6] приводила к минимизации числа мультиплексоров (с одним управляющим входом) в схеме, так как каждой вершине BDD соответствует формула разложения Шеннона, которая реализуется одним мультиплексором. Если мультиплексоры имеют прямой и инверсный выходы, то минимизация схемы из таких мультиплексоров сводится к минимизации числа вершин в графе BDDI [7]. Приведенные примеры – это примеры технологически зависимой оптимизации, при которой ее результаты, по сути, и определяют логическую схему в однородном технологическом базисе – базисе одной ПЛИМ либо базисе одинаковых мультиплексоров.

Если технологическая библиотека проектирования состоит из различных логических элементов типа логических вентилях, то чрезвычайно трудно решить задачу предварительной логической оптимизации так, чтобы представление минимизированных функций однозначно определяло схему, поскольку для этого требуется найти такие суперпозиции для каждой из функций исходной системы, чтобы каждая функция, входящая в суперпозицию, могла быть реализована каким-либо элементом библиотеки. Поэтому этап технологически зависимой оптимизации заменяется этапом предварительной технологически независимой оптимизации, целью которого является нахождение минимизированного представления системы булевых функций по какому-то критерию. В качестве такого критерия чаще всего используется число литералов [8]. Минимизация числа литералов достигается за счет нахождения общих частей в алгебраических скобочных описаниях систем функций. Для *двухуровневых* (И-ИЛИ) представлений функций в виде ДНФ минимизация числа литералов приводит к задачам совместной либо раздельной минимизации ДНФ, после чего производится факторизация [8–11] – выделение одинаковых подвыражений в алгебраических формулах, задающих ДНФ. Это позволяет дополнительно сокращать число литералов.

Другой подход к технологически независимой оптимизации основан на минимизации *многоуровневых* представлений систем функций и применении критериев сложности представлений в виде числа вершин графов. При этом вершина графа ассоциируется с трех- либо двухоперандной логической операцией над литералами булевых переменных. В качестве графовых структур используются BDD и их различные модификации, основанные на разложениях Шеннона, Давио и Амару [12]. В последнее время активно изучаются AIG (AND-Inverter-Graph) [13] и другие графы [14, 15].

Наибольшее распространение получили BDD, применяемые для синтеза схем в библиотеках проектирования как заказных СБИС, так и FPGA (Field-Programmable Gate Array – программируемая пользователем вентильная матрица) [7]. BDD-представления являются графической формой задания последовательных разложений Шеннона булевой функции по одной переменной. При этом для систем булевых функций наиболее часто употребляются ROBDD (Reduced Ordered Binary Decision Diagram – сокращенная упорядоченная BDD), в которых все функции системы разлагаются по одной и той же последовательности (перестановке) переменных. В литературе приведены примеры систем функций, таких, что для отдельных функций имеются свои оптимальные перестановки, а при построении BDD для системы функций ищется компромиссная перестановка, обычно доставляющая на одном шаге разложения Шеннона возможно меньшее число различных остаточных подфункций, называемых кофакторами разложения Шеннона. Таким образом, при минимизации многоуровневых BDD-представлений появляется задача выделения из системы функций таких подсистем, для которых целесообразно находить общую перестановку переменных, т. е. выполнять совместную BDD-минимизацию каждой выделенной подсистемы функций [16]. Подобный подход в литературе практически не изучался,

основное внимание уделялось совместной BDD-минимизации системы функций в целом либо частному случаю этой задачи – раздельной BDD-минимизации, когда система состоит из одной функции.

В настоящей статье в дополнение к работе [16] предлагаются и экспериментально исследуются новые алгоритмы выделения из формульных описаний исходной системы булевых функций так называемых связанных подсистем, при этом каждая из таких подсистем минимизируется отдельно. Однако функции, составляющие каждую связанную подсистему, целесообразно минимизировать совместно. В качестве исходных формульных описаний системы булевых функций выступают совместно минимизированные BDDI-представления (BDDI – Binary Decision Diagram with Inverse cofactors) либо формулы, соответствующие совместно минимизированным булевым сетям (Bool-представления). Минимизация выделенных подсистем выполняется также в классе таких представлений. После получения минимизированных функциональных описаний схем осуществляется синтез логических схем в библиотеке проектирования заказных цифровых КМОП СБИС в промышленном синтезаторе LeonardoSpectrum [17]. Схемы, полученные в результате выделения подсистем, сравнивались со схемами, полученными по совместно либо раздельно минимизированным представлениям исходной системы функций. Такой подход позволяет сравнивать эффективность технологически независимой оптимизации, выполненной различными маршрутами, а именно различными последовательностями оптимизационных процедур, применяемых для формульных заданий системы функций.

ДНФ-, BDDI- и Bool-представления системы булевых функций. ДНФ являются традиционной формой представления булевых функций [1, 2, 8, 9]. Матричное ДНФ-представление системы булевых функций [1] – это пара T^x, B^f матриц, где T^x – троичная матрица, каждая строка которой соответствует элементарной конъюнкции булевых переменных, а единичные значения в булевой матрице B^f отмечают вхождения конъюнкций в ДНФ функций. Именно матричные представления систем ДНФ были исходными данными для проведенных экспериментов. В некоторых экспериментах матричные представления переводились в функционально эквивалентные логические уравнения. В табл. 1 задан пример матричного представления системы ДНФ 10 булевых функций, зависящих от четырех переменных.

Таблица 1
Матричное задание системы ДНФ булевых функций

Table 1
The DNF system of Boolean functions

T^x					B^f									
x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	1
–	1	1	1	1	0	0	0	0	0	1	0	0	1	0
1	–	1	1	1	0	0	0	0	1	0	0	1	0	0
0	1	1	1	–	0	1	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	1	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0	1	1	0	0	1	0	1
1	0	1	–	1	0	1	1	0	0	0	0	0	0	0
1	1	1	–	1	0	0	0	0	0	1	0	0	1	0
1	1	1	–	–	1	0	0	0	0	0	0	0	0	0
0	0	0	1	–	0	0	0	0	1	0	0	1	0	0
1	0	0	1	1	0	1	0	1	0	0	1	0	0	1
1	0	0	0	–	0	1	0	0	1	0	0	1	0	0
1	1	1	1	–	0	0	0	0	1	0	0	1	0	0
–	0	0	–	1	0	0	0	0	0	1	0	0	1	0
0	1	0	1	1	0	0	1	1	0	0	1	0	0	1
0	1	1	0	1	0	0	1	0	0	0	1	1	0	1
1	1	0	0	1	0	1	1	1	0	0	1	0	0	1
1	1	0	1	1	0	0	0	1	1	0	1	1	0	1

Логическое уравнение для ДНФ функции y_9 задается формулой

$$y_9 = \bar{x}_1 * x_2 * \bar{x}_3 * \bar{x}_4 * \bar{x}_5 + x_2 * x_3 * x_4 * x_5 + x_1 * x_2 * x_3 * x_5 + \bar{x}_2 * \bar{x}_3 * x_5;$$

Формула записана на языке SF [18], в котором операция конъюнкции обозначается символом $*$, операция дизъюнкции – символом $+$, а операция инверсии (отрицания) – символом $\bar{}$. В данной формуле 16 литералов. Заметим, что ДНФ функций y_6 , y_9 одинаковы, о чем свидетельствуют одинаковые столбцы в матрице B^f .

BDDI-представления основываются на разложениях Шеннона. *Разложением Шеннона* булевой функции $f(x)$ по переменной x_i называется представление

$$f(x) = \bar{x}_i f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \vee x_i f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n). \quad (1)$$

Функции $f_0 = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$ и $f_1 = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ в правой части (1) называются кофакторами (*cofactors*) разложения по переменной x_i . Каждый из кофакторов $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ может быть разложен 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.

Под *BDDI-представлением* понимается ориентированный бесконтурный граф, задающий последовательные разложения Шеннона булевой функции $f(x) = f(x_1, \dots, x_n)$, $x = (x_1, \dots, x_n)$, либо системы $F(x) = \{f^1(x), \dots, f^m(x)\}$ булевых функций по всем переменным x_1, x_2, \dots, x_n при заданном порядке (перестановке) переменных, по которым проводятся разложения, и при условии нахождения пар взаимно инверсных кофакторов [7]. BDDI-представлению соответствует совокупность взаимосвязанных формул разложения Шеннона. Сравнение кофакторов на равенство и нахождение взаимно инверсных кофакторов осуществляется с использованием полиномов Жегалкина (канонических представлений булевых функций) либо ДНФ, задающих кофакторы. Минимизация сложности BDDI заключается в нахождении последовательности (*перестановки*) переменных разложений Шеннона, при которой число пар взаимно инверсных кофакторов является наименьшим [7, 19].

Bool-представление системы булевых функций соответствует булевой сети (ориентированному бесконтурному графу), функциями вершин которой могут быть логические операции «конъюнкция» либо «дизъюнкция» над литералами булевых переменных. Логическая минимизация булевых сетей на основе разложения Шеннона заключается в поиске такой перестановки переменных разложения, при которой число литералов в булевой сети является наименьшим [19]. После разложения Шеннона по очередной переменной минимизация булевой сети сводится к следующему: ищутся вершины булевой сети, опирающиеся на одинаковые подсети, после чего проводится сокращение сети и находятся уравнения, соответствующие редуцированной сети.

Раздельные BDDI- и Bool-минимизации для выделенных подсистем системы булевых функций заключаются в нахождении своей перестановки переменных разложения для каждой из подсистем функций, в то время как при совместной BDDI- и Bool-минимизации исходной системы используется одна и та же перестановка переменных разложения для всех функций $f^1(x), \dots, f^m(x)$ системы $F(x)$. Примеры совместных многоуровневых BDDI- и Bool-представлений для матричного задания системы ДНФ функций (см. табл. 1) даны в табл. 2. Формулы записаны на языке SF. Жирным шрифтом в табл. 2 выделены формулы для равных функций y_6 , y_9 . Формулы получены с помощью программ [19] BDD_Builder BDDI-минимизации и программы BoolNetOpt Bool-минимизации, имеющихся в системе FLC-2 [18]. Эти же программы были использованы при проведении экспериментов.

Таблица 2
Формульные задания многоуровневых представлений системы булевых функций

Table 2
Formulas for multilevel representations of a system of Boolean functions

BDDI-представление <i>BDDI representation</i>	Bool-представление <i>Bool representation</i>
$y_2 = \bar{x}_5 s_0 + x_5 s_1;$ $y_7 = \bar{x}_5 s_{10} + x_5 s_{11};$ $y_3 = \bar{x}_5 s_2 + x_5 s_3;$ $y_6 = s_{72};$ $s_{72} = \bar{x}_5 s_2 + x_5 s_9;$ $y_9 = s_{72};$ $y_{10} = \bar{x}_5 s_4 + x_5 s_{11};$ $y_4 = \bar{x}_5 s_4 + x_5 s_5;$ $y_8 = \bar{x}_5 s_6 + x_5 s_{13};$ $y_5 = \bar{x}_5 s_6 + x_5 s_7;$ $s_9 = \bar{x}_3 \bar{x}_2 + x_3 s_{30};$ $s_2 = \bar{x}_3 s_{18};$ $s_0 = \bar{x}_3 s_{22} + x_3 s_{23};$ $s_6 = \bar{x}_3 s_{24} + x_3 s_{25};$ $s_1 = \bar{x}_3 s_{26} + x_3 s_{27};$ $s_5 = \bar{x}_3 s_{28};$ $s_{11} = \bar{x}_3 s_{28} + x_3 s_{18};$ $s_3 = \bar{x}_3 s_{31} + x_3 s_{32};$ $s_7 = \bar{x}_3 s_{35} + x_3 s_{36};$ $s_{13} = \bar{x}_3 s_{35} + x_3 s_{38};$ $s_{10} = x_3 s_{19};$ $s_4 = x_3 s_{20};$ $y_1 = x_3 s_{21};$ $s_{20} = \bar{x}_2 \bar{x}_1;$ $s_{38} = \bar{x}_2 s_{36} + x_2 s_{47};$ $s_{28} = \bar{x}_2 s_{36} + x_2 s_{57};$ $s_{19} = \bar{x}_2 s_{39};$ $s_{22} = \bar{x}_2 s_{40};$ $s_{25} = \bar{x}_2 s_{42} + x_2 s_{36};$ $s_{24} = \bar{x}_2 s_{45};$ $s_{35} = \bar{x}_2 s_{45} + x_2 s_{47};$ $s_{32} = \bar{x}_2 x_1 + x_2 s_{39};$ $s_{26} = \bar{x}_2 x_1 + x_2 s_{40};$ $s_{27} = \bar{x}_2 x_1 + x_2 s_{42};$ $s_{18} = x_2 s_{39};$ $s_{23} = x_2 s_{42};$ $s_{31} = x_2 s_{50};$ $s_{30} = x_2 s_{57};$ $s_{21} = x_2 x_1;$ $s_{39} = \bar{x}_1 \bar{x}_4;$ $s_{47} = \bar{x}_1 \bar{x}_4 + x_1 x_4;$ $s_{42} = \bar{x}_1 x_4;$ $s_{57} = \bar{x}_1 x_4 + x_1;$ $s_{45} = \bar{x}_1 x_4 + x_1 \bar{x}_4;$ $s_{50} = \bar{x}_1 + x_1 \bar{x}_4;$ $s_{40} = x_1 \bar{x}_4;$ $s_{36} = x_1 x_4;$	$y_{10} = T_{15} + T_{16};$ $T_{15} = x_1 T_{51};$ $T_{16} = \bar{x}_1 T_{102};$ $y_1 = x_1 T_{106};$ $y_2 = T_{18} + T_{19};$ $T_{18} = x_1 T_{43};$ $T_{19} = \bar{x}_1 T_{73};$ $y_3 = T_{21} + T_{22};$ $T_{21} = x_1 T_{47};$ $T_{22} = \bar{x}_1 T_{74};$ $y_4 = T_{15} + T_{25};$ $T_{25} = \bar{x}_1 T_{78};$ $y_5 = T_{27} + T_{28};$ $T_{27} = x_1 T_{55};$ $T_{28} = \bar{x}_1 T_{82};$ $y_6 = T_{30} + T_{31};$ $T_{30} = x_1 T_{124};$ $T_{31} = \bar{x}_1 T_{86};$ $y_7 = T_{15} + T_{34};$ $T_{34} = \bar{x}_1 T_{90};$ $y_8 = T_{27} + T_{37};$ $T_{37} = \bar{x}_1 T_{94};$ $y_9 = T_{30} + T_{31};$ $T_{43} = T_{44} + T_{45};$ $T_{44} = x_4 T_{107};$ $T_{45} = \bar{x}_4 T_{153};$ $T_{47} = T_{48} + T_{49};$ $T_{48} = x_4 T_{110};$ $T_{49} = \bar{x}_4 T_{157};$ $T_{51} = T_{52} + T_{53};$ $T_{52} = x_4 T_{190};$ $T_{53} = \bar{x}_4 T_{112};$ $T_{55} = T_{56} + T_{57};$ $T_{56} = x_4 T_{118};$ $T_{57} = \bar{x}_4 T_{162};$ $T_{73} = x_4 T_{106};$ $T_{74} = T_{75} + T_{76};$ $T_{75} = x_4 T_{112};$ $T_{76} = \bar{x}_4 T_{168};$ $T_{78} = T_{79} + T_{80};$ $T_{79} = x_4 T_{114};$ $T_{80} = \bar{x}_4 T_{116};$ $T_{82} = T_{83} + T_{53};$ $T_{83} = x_4 T_{122};$ $T_{86} = T_{87} + T_{88};$ $T_{87} = x_4 T_{124};$ $T_{88} = \bar{x}_4 T_{172};$ $T_{90} = T_{75} + T_{92};$ $T_{92} = \bar{x}_4 T_{176};$ $T_{94} = T_{83} + T_{96};$ $T_{96} = \bar{x}_4 T_{180};$ $T_{102} = T_{79} + T_{92};$ $T_{106} = x_2 x_3;$ $T_{107} = \bar{x}_2 x_5;$ $T_{110} = \bar{x}_2 T_{199};$ $T_{112} = x_2 T_{190};$ $T_{114} = T_{112} + T_{116};$ $T_{116} = \bar{x}_2 \bar{x}_2 T_{213};$ $T_{118} = T_{119} + T_{110};$ $T_{119} = x_2 T_{198};$ $T_{122} = \bar{x}_2 \bar{x}_2 T_{199};$ $T_{124} = T_{125} + T_{126};$ $T_{125} = x_2 T_{199};$ $T_{126} = \bar{x}_2 T_{190};$ $T_{153} = T_{112} + T_{155};$ $T_{155} = \bar{x}_2 T_{213};$ $T_{157} = T_{112} + T_{110};$ $T_{162} = \bar{x}_2 \bar{x}_3;$ $T_{168} = x_2 T_{213};$ $T_{172} = T_{173} + T_{126};$ $T_{173} = x_2 \bar{x}_2 T_{198};$ $T_{176} = T_{125} + T_{116};$ $T_{180} = x_2 x_5;$ $T_{190} = \bar{x}_3 x_5;$ $T_{198} = x_3 x_5;$ $T_{199} = x_3 x_5;$ $T_{213} = x_5 \bar{x}_3;$

Связанность формульных представлений булевых функций. Для многоуровневого формульного представления системы $F = \{f^1(x), \dots, f^m(x)\}$ булевых функций обозначим через $R(F)$ множество внутренних формул системы F , т. е. формул, задающих промежуточные (не выходные) булевы переменные для всех функций системы: $R(F) = R(f^1, \dots, f^m) = \bigcup_{i=1}^m R(f^i)$.

Под мерой $q_{ur} = M_{ur}(F^m \cup \{f^{m+1}\})$ связанности системы функций $F^m \cup \{f^{m+1}\}$, где $F^m = \{f^1, \dots, f^m\}$, по уравнениям [16] будем понимать величину

$$q_{ur} = M_{ur}(F^m \cup \{f^{m+1}\}) = \frac{|R(f^1, \dots, f^m) \cap R(f^{m+1})|}{\max(|R(f^1, \dots, f^m)|, |R(f^{m+1})|)}, \quad (2)$$

где $|R(F)|$ – мощность множества $R(F)$.

Через $S(F)$ обозначим число литералов в записи всех формул системы F , т. е. формул, задающих как промежуточные, так и выходные булевы переменные для всех функций системы F .

Под мерой $q_{lit} = M_{lit}(F^m \cup \{f^{m+1}\})$ связанности системы функций $F^m \cup \{f^{m+1}\}$ по литералам будем понимать величину

$$q_{lit} = M_{lit}(F^m \cup \{f^{m+1}\}) = \frac{S(f^1, \dots, f^m, f^{m+1})}{S(f^1, \dots, f^m) + S(f^{m+1})}. \quad (3)$$

При проведении экспериментальных исследований меры q_{ur} , q_{lit} связанности удобно задавать в процентах. В алгоритме (программе) *Splitter* [16] при добавлении функции f^{m+1} в систему F^m проверялось, чтобы мера связанности q_{ur} расширенной подсистемы не была меньше заданного значения q . В предлагаемом далее алгоритме *Splitter2* подсистема связанных функций пополняется тогда, когда мера связанности q_{lit} не больше заданного значения q .

Алгоритм выделения подсистем функций для заданного значения q меры связанности q_{lit} . Описываемый ниже алгоритм *Splitter2* выделения подсистем для заданного значения q меры связанности q_{lit} подобен алгоритму *Splitter* и состоит в последовательном формировании (на каждой итерации i) по текущей системе функций очередной подсистемы P^i функций (подсистема P^i характеризуется мерой связанности, не большей q). На первой итерации $i=1$ текущую систему функций образуют функции исходной системы, а остаточная подсистема P^{ost} является пустой. Перед выполнением итераций алгоритма *Splitter2* требуется упорядочить функции системы (например, в лексикографическом порядке имен функций), составить соответствующий список W и найти для каждой функции из списка W число литералов в уравнениях, задающих функцию.

На каждой итерации требуется выполнить шаги 1–3:

Шаг 1. Рассмотреть по очереди функции f^i из списка W и найти такую пару $L = \{f^i, f^j\}$, $i, j = 1, \dots, m$, $i \neq j$, которая имеет наименьшее (но не большее q) значение меры связанности. В этом случае при подсчете меры связанности (3) полагается $F^m = \{f^i(x)\}$. Если таких пар функций несколько, то выбирается первая из них. Если указанной пары функций нет, то переход на шаг 4. Если для какой-то функции f^i все остальные функции образуют с функцией f^i такие пары L , которые имеют меру связанности, бóльшую q , то функция f^i исключается из текущей системы и помещается в остаточную подсистему P^{ost} . Если система функций содержит m функций и для всех $C_m^2 = m(m-1)/2$ пар выяснено, что их связанность q_{lit} превышает q , то из системы нельзя выделить ни одной подсистемы с требуемой связанностью q .

Шаг 2. Составить из функций найденной на шаге 1 пары L формируемую подсистему P^i , исключив функции выбранной пары L из текущей системы, и добавлять в формируемую подсистему поочередно те функции f^r , которые находятся с помощью следующей эвристики: из множества функций текущей системы выбирается та функция f^r , которая обеспечивает наименьшее (но не большее q) возможное значение меры связанности (3) для подсистемы $P^i \cup \{f^r\}$. Если таких функций несколько, то выбирается и добавляется в формируемую подсистему P^i первая из них.

Шаг 3. Если нет ни одной функции f^r , такой, что подсистема $P^i \cup \{f^r\}$ имеет меру компонентной связанности, не меньшую q , то закончить формирование подсистемы P^i и объявить не вхо-

дящие в нее функции текущей системой. Переход на шаг 1 для формирования подсистемы на итерации $i+1$.

Шаг 4. Закончить формирование подсистем, когда все функции текущей системы будут включены в формируемые подсистемы. Формирование подсистем заканчивается (алгоритм прекращает работу) также и тогда, когда в текущей подсистеме нельзя найти ни одной пары функций, характеризующих мерой связанности, не меньшей q , либо когда в текущей системе имеется только одна функция, которая добавляется в остаточную подсистему P^{ost} . Конец алгоритма.

Пример. Рассмотрим пример, иллюстрирующий алгоритм *Splitter2*, для BDDI-представления, заданного в левой части табл. 2, и значения $q=0,7$ (70 %). Перед выполнением итераций алгоритма составим в лексикографическом порядке (по именам функций) список пар: <имя функции, число литералов в ней>: < y_1 , 4>, < y_{10} , 25>, < y_2 , 28>, < y_3 , 23>, < y_4 , 19>, < y_5 , 34>, < y_6 , 20>, < y_7 , 25>, < y_8 , 38>, < y_9 , 20>. Функция y_1 задана уравнениями $y_1=x^3*s_{21}$; $s_{21}=x^2*x_1$, в правых частях которых содержатся четыре литерала. В уравнениях для функции y_5 имеется наибольшее число литералов – 34.

Итерация 1.

Шаг 1. Легко видеть, что для функции y_1 нельзя подобрать ни одной другой функции, для которой связанность пары функций будет не меньше 0,7. Функция y_1 помещается в остаточную подсистему. Следующей рассматривается функция y_{10} . Перебираются пары $\{y_{10}, y_i\}$, $i=2, \dots, 9$, и для них вычисляется связанность по формуле (3). Все пары имеют связанность, большую 0,7, за исключением пары $\{y_{10}, y_7\}$, которая имеет связанность по литералам, меньшую 0,7: $q_{107}=(33)/(25+25)=0,66$, поэтому на шаге 1 выбирается пара L (подсистема) $\{y_{10}, y_7\}$. Это легко проверить, рассмотрев уравнения

```
y10=^x5*sf4+x5*sf11;
sf4=x3*sf20;
sf11=^x3*sf28+x3*sf18;
sf20=^x2*^x1;
sf28=^x2*sf36+x2*sf57;
sf18=x2*sf39;
sf36=x1*x4;
sf57=^x1*x4+x1;
sf39=^x1*^x4;
```

для функции y_{10} , содержащие 25 литералов, и уравнения

```
y7=^x5*sf10+x5*sf11;
sf10=x3*sf19;
sf11=^x3*sf28+x3*sf18;
sf19=^x2*sf39;
sf28=^x2*sf36+x2*sf57;
sf18=x2*sf39;
sf39=^x1*^x4;
sf36=x1*x4;
sf57=^x1*x4+x1;
```

для функции y_7 , содержащие также 25 литералов. Уравнения

```
y10=^x5*sf4+x5*sf11;
sf4=x3*sf20;
sf11=^x3*sf28+x3*sf18;
sf20=^x2*^x1;
sf28=^x2*sf36+x2*sf57;
sf18=x2*sf39;
sf36=x1*x4;
```



```
sf57=^x1*x4+x1;  
sf39=^x1*^x4;  
y7=^x5*sf10+x5*sf11;  
sf10=x3*sf19;  
sf19=^x2*sf39;
```

для подсистемы $\{y_{10}, y_7\}$ содержат 33 литерала, так как функции y_{10}, y_7 имеют шесть общих (выделенных жирным шрифтом) уравнений. Поэтому $q_{lit} = (33)/(25+25) = 0,66$.

Шаги 2, 3. Перебирая оставшиеся функции системы (за исключением y_1) по одной и добавляя их поочередно в подсистему $\{y_{10}, y_7\}$, убеждаемся, что ни одна из проверяемых подсистем не удовлетворяет ограничению $q < 0,7$. Завершаем построение первой выделенной подсистемы: $P^1 = \{y_{10}, y_7\}$.

Итерация 2.

Шаг 1. Перебираем пары $\{y_2, y_3\}, \{y_2, y_4\}, \{y_2, y_5\}, \{y_2, y_6\}, \{y_2, y_8\}, \{y_2, y_9\}$ и убеждаемся, что для каждой из них мера связанности превышает 0,7, поэтому функция y_2 помещается в остаточную подсистему. Аналогичная ситуация наблюдается и для функций y_3, y_4 , которые также помещаются в остаточную подсистему $P^{ost} = \{y_1, y_2, y_3, y_4\}$. Для очередной функции y_5 рассматриваются пары $\{y_5, y_6\}, \{y_5, y_8\}, \{y_5, y_9\}$. Только для пары $\{y_5, y_8\}$ мера связанности 0,64 меньше 0,7, поэтому следующей подсистемой становится $P^2 = \{y_5, y_8\}$.

Шаги 2, 3. Расширить подсистему $P^2 = \{y_5, y_8\}$ нельзя, поэтому второй выделенной подсистемой является $P^2 = \{y_5, y_8\}$.

Итерация 3.

Шаги 1–3. Легко проверить, что для оставшихся равных функций y_6, y_9 мера связанности равна 0,525, поэтому третьей выделенной подсистемой является $P^3 = \{y_6, y_9\}$.

Результат работы алгоритма *Splitter2*: $P^1 = \{y_{10}, y_7\}, P^2 = \{y_5, y_8\}, P^3 = \{y_6, y_9\}, P^{ost} = \{y_1, y_2, y_3, y_4\}$.

Алгоритмы нахождения лучших значений меры связанности. Алгоритмы основаны на переборе значений q меры q_{ur} (либо меры q_{lit}) связанности, выполнении для заданного значения q алгоритма *Splitter2* (*Splitter*) и выборе лучшего варианта выделения подсистем по различным критериям.

Алгоритм AutoSplit – выбор по минимальной сумме числа литералов в подсистемах. Это базовый алгоритм, который используется в двух других алгоритмах. В алгоритме (программе) *AutoSplit* ищутся лучшие значения параметра связанности q для BDDI- либо Bool-представлений путем итеративного выполнения программы *Splitter2* для меры связанности 95, 90, 85, 80, ..., 5 %, оценивается суммарное число литералов в каждой полученной логической сети и лучший вариант решения выбирается по минимуму оценки, в качестве которой выступает минимальная сумма числа литералов в подсистемах. Естественно, что если для значения q меры связанности q_{lit} подсистемы не выделялись, то программа *AutoSplit* прекращала испытывать меньшие, чем q , значения меры связанности и переходила к выбору решения по критерию меньшего числа литералов.

Алгоритм *AutoSplit* может вызывать при своей работе как *Splitter2*, так и *Splitter* [16], в котором используются значения меры связанности q_{ur} . Если *AutoSplit* вызывает *Splitter*, то значения q меры связанности q_{ur} перебираются в порядке 5, 10, 15, ..., 95 %, и если ни одна подсистема не выделилась, то *AutoSplit* прекращала испытывать большие, чем q , значения меры связанности q_{ur} [16]. Аналогично и для описываемых далее алгоритмов *AutoSplit_Func* и *AutoSplit_Iter*, которые используют как *Splitter*, так и *Splitter2*.

Алгоритм AutoSplit_Func – выбор по минимальному числу функций в остаточной подсистеме. В алгоритме *AutoSplit_Func* ищутся лучшие значения параметра связанности q для BDDI- либо Bool-представлений путем итеративного выполнения программы *Splitter* для q_{ur} – меры связанности, равной 5, 10, 15, 20, ..., 95 %. Лучшее решение выбирается по минимуму числа функций в остаточной подсистеме P^{ost} . Если есть несколько таких вариантов, то выбирается тот из них, где меньше подсистем. Если есть несколько и таких вариантов, то выбирается тот, где меньше суммарное число литералов в подсистемах.

Алгоритм *AutoSplit_Iter* – выбор, основанный на итерационном применении алгоритма *AutoSplit* для остаточной подсистемы. Если после применения алгоритма *AutoSplit* остаточной подсистемы нет, то результат работы алгоритма *AutoSplit_Iter* совпадает с результатом работы алгоритма *AutoSplit*. Если остаточная подсистема содержит одну либо две функции, то алгоритм *AutoSplit_Iter* дальше не работает. Если остаточная подсистема содержит три и более функции, то алгоритм *AutoSplit* выполняется для этой остаточной подсистемы, т. е. выполняется очередная итерация работы алгоритма *AutoSplit*. Для этого извлекается остаточная подсистема P^{ost} и рассматривается как исходная для нового выполнения *AutoSplit*: перебираются меры связанности 5, 10, 15, 20, ..., 95 % и выделяются очередные подсистемы. Проверяется условие останова итераций, и если оно выполняется, то конец итерациям. Если хотя бы одна подсистема выделилась, то итерации продолжаются для новой остаточной подсистемы. Если ни одна подсистема не выделилась в результате работы алгоритма *AutoSplit*, то алгоритм *AutoSplit_Iter* прекращает свою работу – конец итерациям применения алгоритма *AutoSplit*.

Эксперименты. Входными данными были логические уравнения, задающие те же ДНФ-представления 39 исходных систем булевых функций, которые испытывались в работе [16]. Это были примеры из библиотеки примеров¹. Состав проведенных 12 экспериментов и используемых в них алгоритмов (и реализующих эти алгоритмы программ) дан в табл. 3.

Таблица 3
Состав экспериментов

Table 3
Contents of the experiments

Уравнения для всей системы <i>Equations for the whole system</i>	Алгоритм <i>Algorithm</i>	Минимизация выделенных подсистем <i>Minimization of extractable subsystems</i>	Выбор решения при переборе значений параметра связанности q <i>Choosing a solution when trying out the values of the parameter q</i>		
			<i>AutoSplit</i>	<i>AutoSplit_Func</i>	<i>AutoSplit_Iter</i>
BDDI	<i>Splitter</i>	BDDI+Bool	Эксп. 1	Эксп. 2	Эксп. 3
	<i>Splitter2</i>	BDDI+Bool	Эксп. 4	Эксп. 5	Эксп. 6
Bool	<i>Splitter</i>	Bool	Эксп. 7	Эксп. 8	Эксп. 9
	<i>Splitter2</i>	Bool	Эксп. 10	Эксп. 11	Эксп. 12

Каждый из экспериментов состоял из четырех этапов:

1. *Подготовка исходных данных.* Исходные данные были получены в двух видах логических уравнений в результате совместной BDDI- и Bool-минимизации каждой системы булевых функций (табл. 3).

2. *Выделение подсистем для совместной логической минимизации.* Для выделения совместно минимизируемых подсистем использовались алгоритмы *Splitter* и *Splitter2*, что отражено во втором столбце табл. 3. Для перебора вариантов выделения подсистем использовались три алгоритма: *AutoSplit*, *AutoSplit_Func* и *AutoSplit_Iter*. Результатом выполнения этапа 2 для системы булевых функций является логическая сеть, элементами которой были выделенные (и остаточная) подсистемы.

3. *Совместная минимизация подсистем.* Для каждого примера системы булевых функций этап 3 выполнялся двумя способами.

Способ 1, обозначенный в табл. 3 как BDDI+Bool, применялся в экспериментах 1–6 и состоял в выполнении трех преобразований каждой подсистемы логической сети, полученной на этапе 1. Сначала осуществлялся переход от логических уравнений, задающих разложения Шеннона, к матричной форме системы ДНФ. Затем для полученной системы ДНФ выполнялась совместная BDDI-минимизация (находилась первая лучшая перестановка переменных) и полу-

¹The Tests in the Monograph "Logic Minimization Algorithms for VLSI Synthesis". – URL: <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex> (date of access: 20.01.2020).

ченные уравнения подвергались совместной Bool-минимизации – находилась вторая перестановка переменных при минимизации булевой сети.

Способ 2, обозначенный в табл. 3 как Bool, применялся в экспериментах 7–12 и заключался в выполнении следующих преобразований для каждой подсистемы логической сети: сначала осуществлялся переход от логических уравнений, задающих разложения Шеннона, к матричной форме системы ДНФ, затем от матричной формы системы ДНФ осуществлялся переход к булевой сети, для которой выполнялась совместная Bool-минимизация.

4. *Синтез логической схемы в синтезаторе LeonardoSpectrum.* На этом этапе происходила конвертация данных из языка SF в VHDL-описание [20] минимизированной на этапе 3 логической сети. VHDL-описания подавались на вход синтезатора LeonardoSpectrum – выполнялся синтез логических схем в библиотеке проектирования заказных КМОП СБИС. Синтезатор LeonardoSpectrum перерабатывает входное описание, выполняет свою логическую минимизацию, получает внутреннее описание, по которому и синтезируется схема. Библиотекой синтеза являлась та же библиотека проектирования заказных цифровых КМОП СБИС, которая использовалась в работах [7, 16].

Для каждой системы булевых функций в каждом из экспериментов 1–6 строилось по две логические схемы, которые различались этапами минимизации: сначала выполнялась BDDI-минимизация, затем для результатов такой минимизации выполнялась дополнительная Bool-минимизация. В экспериментах 7–12 для каждого примера системы функций строилась одна логическая схема (одна Bool-минимизация для выделенных подсистем). Таким образом, всего получалось 18 схем для каждого из 39 примеров.

Для рассматриваемого примера двух видов исходных описаний (см. табл. 2) результаты экспериментов 4 и 10 приведены в табл. 4. Видно, что дополнительная Bool-минимизация (эксперимент 4) уменьшает площадь схемы с 18 777 до 18 314, однако задержка схемы при этом увеличивается более чем в два раза.

Таблица 4
Результаты экспериментов 4, 10 для системы функций (табл. 1)

Table 4
Results of experiments 4, 10 for the system of functions (Table 1)

Эксперимент Experiment	Выделение подсистем – разбиение системы функций на подсистемы <i>Extraction of subsystems – partitioning of the system of functions into subsystems</i>				Минимизация подсистем <i>Minimization of subsystems</i>					
					Способ 1 <i>Method 1</i>				Способ 2 <i>Method 2</i>	
	BDDI		BDDI+Bool		Bool					
	P^1	P^2	P^3	P^{ost}	Area	Delay	Area	Delay	Area	Delay
4	y_{10}, y_7	y_5, y_8	y_6, y_9	y_1, y_2, y_3, y_4	18 777	3,04	18 314	6,49	-	-
10	y_{10}, y_7	y_5, y_8	y_6, y_9	y_1, y_2, y_3, y_4	-	-	-	-	19 307	3,58

Хотя алгоритм *Splitter2* выделил одинаковые подсистемы как для BDDI- (эксперимент 4), так и для Bool-описания (эксперимент 10), площади схем оказались разными. Все дело в различных совокупностях логических уравнений в подсистемах: в эксперименте 4 это BDDI-описания (затем Bool-описания), в эксперименте 7 – Bool-описания. Жирным шрифтом выделены лучшие решения.

Результаты экспериментов и их обсуждение. Видно, что каждому эксперименту соответствует свой маршрут технологически независимой оптимизации, состоящей из нескольких оптимизационных процедур. Сравнение результатов экспериментов проведем в три этапа, результаты экспериментов зададим в табл. 5 и 6. Обозначения в таблицах:

n – число аргументов;

m – число функций системы ДНФ, заданной на k общих элементарных конъюнкциях;

Area – суммарная площадь элементов схемы (в условных единицах);

Delay – временная задержка схемы (нс);

q – лучшая мера связанности, для которой были выделены подсистемы и построена логическая схема. По результатам экспериментов 3, 6, 9, 12 (алгоритм *AutoSplit_Iter*) значение q не указано, так как несколько раз могут быть выделены подсистемы с разными значениями q .

Таблица 5
Результаты экспериментов 1–6

Table 5
Results of experiments 1–6

Номер примера <i>Example number</i>	Пример (схема) <i>Example (circuit)</i>	n	m	k	Группа I (эксперименты 1–3) <i>Group I (experiments 1–3)</i>				Группа II (эксперименты 4–6) <i>Group II (experiments 4–6)</i>			
					$q, \%$	Area	Delay	Эксперимент	$q, \%$	Area	Delay	Эксперимент
1	Add6	12	7	1 092	5	17 839	7,56	1,3	-	14 826	5,76	6
2	B12	15	9	431	-	15 507	3,40	3	95	15 518	3,47	4
3	B9	16	5	123	5	*23 492 D	4,24	2	95	24 413 D	4,49	4
4	Dist	8	5	256	-	*62 195 D	6,34	3	-	*62 195 D	6,34	6
5	Gary	15	11	442	20	*91 601 D	6,62	1	85	92 673	*6,23	4
6	Ibm	48	17	173	10	*59 667	*5,32	1	-	-	-	-
7	In0	15	11	138	20	*88 901 D	6,76	1	85	92 673	*6,23	4
8	In1	16	17	110	80	183 242 D	*8,76	1	95	162 445 D	9,43	4
9	In2	19	10	137	-	77 255	6,67	3	-	*71 033 D	*5,58	6
10	Intb	15	7	664	-	*225 159 D	9,52	3	-	234 600	9,17	6
11	Jbp	36	57	166	5	92 784	*4,80	2	-	*91 272	5,74	6
12	M181	15	9	430	5	*15 585	3,19	1,2	-	*15 585	3,19	6
13	M3	8	16	128	40	*51 632	5,13	1	95	52 703 D	5,70	4
14	M4	8	16	256	60	75 648 D	5,28	1	55	*74 811 D	5,17	4
15	Max1024	10	6	1 024	5	122 191 D	7,90	2	-	131 080 D	7,60	6
16	Max512	9	6	512	5	*67 038 D	*5,37	1	85	72 071 D	6,21	4
17	Mlp4	8	8	256	5	68 121 D	*5,69	2	95	70 191 D	6,53	4
18	Mp2d	14	14	123	5	16 461 D	*3,64	2	95	*16 076 D	5,28	5
19	Mult_7	14	14	13 060	10	2 226 230 D	13,68	1	-	2 384 909 D	14,73	4
20	Mult_8	16	16	52 810	10	6 735 735	*16,01	1	-	7 821 190 D	17,00	4
21	P82	5	14	24	10	19 759 D	2,96	1	-	*18 810	*2,94	6
22	Psevdo1	10	20	1 000	25	781 981	11,03	1	95	762 719 D	*7,51	5
23	Radd	8	5	120	5	9 480	3,85	1	-	9 480	*3,85	6
24	Root	8	5	256	50	24 585	5,21	1	75	*22 683 D	8,60	4
25	Sist_4	17	12	370	5	112 973 D	7,65	2	95	*101 154	7,44	5
26	Soar	83	94	529	-	137 854 D	*5,43	3	-	*136 063 D	6,19	6
27	Sqn	7	3	96	15	*20 094 D	5,31	2	85	*20 094 D	5,31	4,5
28	Tial	14	8	640	5	*282 571 D	9,52	1	95	285 049 D	*8,63	4
29	Tms	8	16	30	5	41 320	*4,96	2	-	*40 522	5,56	6
30	Tt2_matr	24	21	222	5	*43 429	*3,77	2	95	48 261 D	4,86	5
31	Verg_1	17	61	2 004	90	454 536 D	16,10	1	55	*380 584	*12,97	4
32	Verg_2	18	63	2 129	85	608 555 D	15,76	1	95	*489 098	*13,26	4
33	Vtx1	27	6	110	-	23 866	5,14	3	-	28 285	*4,66	6
34	X1_matr	51	35	324	5	*79 398 D	6,32	2	95	82 830 D	7,28	5
35	X3_matr	135	99	915	-	*205 696 D	6,81	3	-	207 358	*6,12	6
36	X4_matr	94	71	371	5	117 705 D	*5,93	1	-	114 033 D	6,80	6
37	X9dn	27	7	120	5	*20 741 D	5,48	2	95	20 919 D	5,44	5
38	Z4	7	4	128	50	6 640	4,25	1,3	-	6 640	4,25	6
39	Z5xp1	7	10	128	5	22 186 D	4,01	2	-	21 226 D	4,95	5
Лучшие решения					15	11				14	11	

Таблица 6
Результаты экспериментов 7–12

Table 6
Results of experiments 7–12

Номер примера Example number	Пример (схема) Example (circuit)	Группа III (эксперименты 7–9) Group III (experiments 7–9)				Группа IV (эксперименты 10–12) Group IV (experiments 10–12)				Группа V (статья [16]) Group V (paper [16])	
		q, %	Area	Delay	Эксперимент	q, %	Area	Delay	Эксперимент	Area	Delay
1	Add6	5	20 389	*4,18	8	-	*14 564	5,33	12	16 534	6,82
2	B12	5	*15 094	3,85	8	90	15 546	*3,15	10	15 496	#2,51
3	B9	5	23 721	*3,82	8	-	26 494	3,94	12	#22 454	4,69
4	Dist	5	63 249	5,52	7	-	66 854	*4,89	12	62 808	6,41
5	Gary	5	92 355	8,27	8	-	93 035	7,90	12	#91 395	7,32
6	Ibm	5	71 095	5,84	8	-	66 162	6,14	12	#53 406	5,37
7	In0	-	91 250	6,38	9	-	92 857	8,05	12	90 156	#6,23
8	In1	75	164 616	9,12	7	95	*154 415	10,00	11	162 478	#8,52
9	In2	-	79 571	5,59	9	-	83 270	6,16	12	#70 291	7,32
10	Intb	5	254 816	*9,05	7	-	244 248	10,97	12	235 883	9,51
11	Jbp	-	98 046	6,83	9	95	97 231	5,80	11	#91 205	7,17
12	M181	10	15 808	*2,72	7	90	15 713	3,02	10	#15 546	2,93
13	M3	-	57 390	*4,70	9	-	64 583	4,99	12	51 632	5,13
14	M4	5	94 732	*4,93	8	-	83 326	5,14	12	78 181	5,87
15	Max1024	5	129 272	*7,09	7	90	*121 449	7,47	10	129 478	7,38
16	Max512	-	73 143	5,56	9	-	71 413	5,89	12	72 300	6,26
17	Mlp4	5	*67 987	5,83	7	-	69 783	5,87	12	68 070	#5,41
18	Mp2d	-	17 159	3,72	9	95	16 846	3,95	10	17 203	3,81
19	Mult_7	-	*2 062 212	*12,61	7	95	2 467 320	13,23	10	2 068 489	13,00
20	Mult_8	-	*6 145 572	16,66	7	-	-	-	-	#5 536 956	#15,82
21	P82	-	*18 810	*2,94	9	-	19 441	3,46	12	19 056	3,23
22	Psevdo1	5	729 507	8,42	7	90	*722 543	8,69	10	735 796	8,91
23	Radd	5	*7 399	3,99	7,8,9	90	*7 399	3,99	10,11,12	7 399	3,99
24	Root	-	26 405	*4,71	9	-	25 244	5,11	12	26 075	4,78
25	Sist_4	-	127 380	7,30	9	95	139 595	*6,87	11	112 722	7,21
26	Soar	5	141 615	7,32	8	-	146 508	6,11	12	151 787	6,52
27	Sqn	5	22 008	*3,5	7					22 683	#3,33
28	Tial	5	389 679	11,13	7,8	95	343 176	11,46	11	#279 458	9,08
29	Tms	80	48 161	5,62	7	-	44 227	4,99	12	#39 004	#4,65
30	Ttt2_matr	30	49 517	4,89	7	-	48 775	5,35	12	45 683	4,41
31	Verg_1	95	672 239	15,73	7	-	721 176	15,34	12	465 032	14,41
32	Verg_2	90	759 990	16,69	7	-	819 189	17,53	12	536 037	14,15
33	Vtx1	-	27 632	5,07	9	70	*23 034	5,51	11	28 112	#4,31
34	X1_matr	5	93 789	*6,09	8	-	90 921	7,75	12	#71 441	#4,41
35	X3_matr	15	217 001	8,47	7	-	216 705	7,81	12	211 633	#5,73
36	X4_matr	5	*109 680	6,31	7	-	116 125	6,05	12	119 194	#4,84
37	X9dn	-	31 706	*5,43	9	-	28 581	5,56	12	20 741	5,48
38	Z4	10	*6 411	3,74	8,9	80	7 444	*3,57	10,11,12	6 640	4,25
39	Z5xp1		*20 127	5,31	8	95	24 100	*3,94	10	22 582	5,24
Лучшие решения			9	13	-	-	6	5	-	10	10

Этап 1. Сначала проанализируем результаты экспериментов отдельно в каждой из четырех групп экспериментов, найдем лучшие решения в каждой группе и запишем их в табл. 5 и 6.

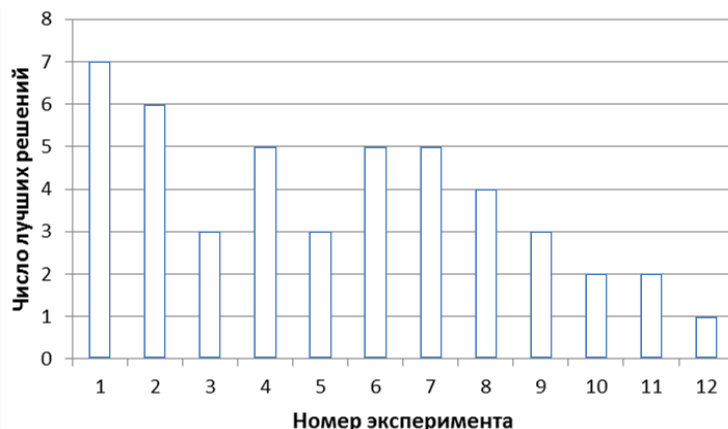
В группу I включены эксперименты 1–3, в группу II – эксперименты 4–6 (табл. 5). Группу III составляют эксперименты 7–9, группу IV – эксперименты 10–12 (табл. 6).

В каждой группе для всех примеров найдем значение меньшей площади и соответствующее («привязанное» к этой площади) значение задержки. Если схема с меньшей площадью получена после дополнительной Bool-минимизации, то после этого значения поставим символ D (дополнительной минимизации). Например, для схемы B9 в группе I экспериментов записано значение *23 492 D площади, символ D свидетельствует о том, что схема с такой площадью была получена в результате BDDI+Bool-минимизации. Символ D не появляется после значений площади в экспериментах 7–12, так как в них выполняется только Bool-минимизация. Символы * появляются после выполнения этапа 2 сравнения результатов экспериментов.

Этап 2. Сравним сначала результаты экспериментов групп I–IV. Для этого в табл. 5 и 6 одним символом * и полужирным шрифтом выделим лучшее решение (по группам I–IV экспериментов) для испытываемого примера – меньшее значение параметра площади либо задержки схемы.

Значение 23 492 D площади для примера B9, помеченное символом *, свидетельствует о том, что это наименьшее значение площади по всем четырем группам и, следовательно, по всем 12 экспериментам. Однако для примера B9 схема с наименьшей задержкой *3,82 получена в группе III. Довольно часто оказывается, что схема с наименьшей площадью имеет и наименьшую задержку. Например, это выполняется (оба значения площади и задержки помечены символом *) для схемы Ibm в группе I экспериментов: Area=*59 667, Delay=*5,32 (см. табл. 5).

На рисунке представлена информация о числе лучших решений (по площади), полученных в каждом из 12 экспериментов (маршрутов оптимизации).



Сравнительная оценка лучших решений (по площади схем) в экспериментах 1–12
Comparative assessment of the best solutions (in terms of circuit area) in experiments 1–12

Этап 3. Сравним лучшие результаты проведенных 12 экспериментов с лучшими результатами выполненных в работе [16] четырех экспериментов по отдельной и совместной BDDI- и Bool-минимизации. Лучшие результаты экспериментов из работы [16] приводятся в двух последних столбцах табл. 6 (группа V).

Для каждого примера системы булевых функций символом # в группе V пометим те значения площади и задержки, которые меньше значений, помеченных символом *. Символ # свидетельствует о том, что алгоритмы выделения подсистем проиграли хотя бы одному из четырех алгоритмов группы V (отдельной либо совместной BDDI- и Bool-минимизации). Например, для системы функций B9 в группе V значение площади (Area) равно #22 454, что меньше лучшего значения площади *23 492 D (см. табл. 5), полученного по всем 12 экспериментам.

В последних строках табл. 5 и 6 приводится число лучших решений по всем группам. В последней строке группы V дается число примеров, для которых выгоднее проводить традиционную совместную либо раздельную минимизацию без выделения подсистем.

Результаты экспериментов 1–6 показали, что дополнительная Bool-минимизация булевых сетей часто позволяет улучшать по площади схем результаты BDD-минимизации. Об этом свидетельствует частота появления буквы D (см. табл. 5): в группе I экспериментов 23 лучших результата (они помечены буквой D) получены после дополнительной Bool-минимизации, в группе II – 22 лучших результата. Напомним, что всех примеров было 39. Без буквы D лучшими результатами в группе I (по всем 12 экспериментам) оказались лишь четыре примера: Ibm, M181, M3, Ttt2_matr; в группе II – пять примеров: Jbr, M181, P82, Tms, Verg_1, Verg_2.

Результаты экспериментов подтвердили уже известный факт, что программа BoolNetOpt минимизации булевых сетей гораздо более чувствительна к форме задания исходных данных (логических уравнений) по сравнению с программой Bdd_Builder минимизации BDD-представлений, исходными данными для которой являются матричные представления систем функций.

Анализ результатов, представленных на рисунке, показывает, что преимущество по частоте получения лучших решений имеет алгоритм *AutoSplit* выбора лучших решений по числу литералов в сочетании с алгоритмом *Splitter* (в эксперименте 1 получены семь лучших решений) либо в сочетании с алгоритмом *Splitter2* (в эксперименте 4 получены пять лучших решений). В целом применение алгоритмов, испытанных в группах I и II, дает лучшие результаты по сравнению с алгоритмами, испытанными в группах III и IV. Алгоритм *Splitter2* оказался конкурентоспособным по сравнению с алгоритмом *Splitter*. Алгоритмы *AutoSplit_Func* и *AutoSplit_Iter* также позволили улучшить результаты ранее проведенных экспериментов [16] по сравнению с алгоритмом *AutoSplit*.

По сравнению с результатами работы [16], где алгоритмы *Splitter* и *AutoSplit* выиграли по площади только в 13 случаях, предложенные в данной работе алгоритмы выиграли в 29 случаях (из 39) на том же потоке примеров и с теми же программами многоуровневой минимизации, применяемыми для выделенных подсистем. Тем не менее следует сказать о том, что имеются примеры (в испытанном наборе примеров их 10), для которых совместная либо раздельная многоуровневая оптимизация позволяет получать лучшие результаты синтеза (площадь и поддержку схемы) по сравнению с предложенными алгоритмами выделения подсистем.

Заключение. Программы, реализующие все предложенные в данной работе алгоритмы выделения подсистем функций для совместной многоуровневой минимизации, включены в систему FLC-2 [18] логической оптимизации и часто позволяют улучшать результаты технологически независимой оптимизации блоков комбинационной логики, применяемой при выполнении проектов цифровых систем в заказных цифровых КМОП СБИС.

Вклад авторов. П. Н. Библио выполнил эксперименты, подготовил текст статьи, Н. А. Кириенко разработала программные средства для выделения из многоуровневого представления системы булевых функций подсистем для совместной минимизации, В. И. Романов разработал программные средства для анализа результатов экспериментов и выбора лучших решений.

Список использованных источников

1. Закревский, А. Д. Логический синтез каскадных схем / А. Д. Закревский. – М. : Наука, 1981. – 416 с.
2. Logic Minimization Algorithm for VLSI Synthesis / K. R. Brayton, G. D. Hachtel, C. McMullen, A. L. Sangiovanni-Vincentelli. – Boston : Kluwer Academic Publishers, 1984. – 193 p.
3. Sasao, T. Input variable assignment and output phase optimization of PLA's / T. Sasao // IEEE Transactions on Computers. – 1984. – Vol. C-33, no. 10. – P. 879–894.
4. Wey, C. L. An efficient output assignment for PLA minimization / C. L. Wey, T. Y. Chang // IEEE Transactions on Computer-Aided Design. – 1990. – Vol. 9. – P. 1–7.
5. Ebendt, R. Advanced BDD Optimization / R. Ebendt, G. Fey, R. Drechsler. – Springer, 2005. – 222 p.

6. Кнут, Д. Э. Искусство программирования. Т. 4, А. Комбинаторные алгоритмы. Ч. 1 : пер. с англ. / Д. Э. Кнут. – М. : Вильямс, 2013. – 960 с.
7. Бибило, П. Н. Бинарные диаграммы решений в логическом проектировании / П. Н. Бибило. – М. : Ленанд, 2024. – 560 с.
8. Брейтон, Р. К. Синтез многоуровневых комбинационных логических схем / Р. К. Брейтон, Г. Д. Хэчтел, А. Л. Санджованни-Винченцелли // Труды Института инженеров по электротехнике и радиоэлектронике. – 1990. – Т. 78, № 2. – С. 38–83.
9. Синтез асинхронных автоматов на ЭВМ / под ред. А. Д. Закревского. – Минск : Наука и техника, 1975. – 184 с.
10. MIS: A multiple-level logic optimization systems / R. K. Brayton, R. Rudell, A. L. Sangiovanni-Vincentelli, A. R. Wang // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1987. – Vol. CAD-6, no. 6. – P. 1062–1081.
11. Brayton, K. R. Factoring logic functions / K. R. Brayton // IBM Journal of Research and Development. – 1987. – Vol. 31, no 2. – P. 187–198.
12. Amaru, L. G. New Data Structures and Algorithms for Logic Synthesis and Verification / L. G. Amaru. – Springer, 2017. – 156 p.
13. О сложности инверсных графов, реализующих булевы функции от малого числа переменных / С. А. Ложкин, В. С. Зизов, М. С. Шуплецов [и др.] // Проблемы разработки перспективных микро- и наноэлектронных систем : сб. тр. / под общ. ред. А. Л. Стемповского. – 2020. – Вып. 4. – С. 95–102. – DOI: 10.31114/2078-7707-2020-4-95-102.
14. Optimizing majority-inverter graphs with functional hashing / M. Soeken, L. G. Amaru, P. Gaillardon, G. De Micheli // Proc. of the 2016 Design, Automation & Test in Europe Conf. & Exhibition (DATE), Dresden, Germany, 14–18 March 2016. – Dresden, 2016. – P. 1030–1035.
15. Exact synthesis of majority-inverter graphs and its applications / M. Soeken, L. Amaru, P.-E. Gaillardon, G. De Micheli // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2017. – Vol. 36, no. 11. – P. 1842–1855.
16. Бибило, П. Н. Выделение из многоуровневого представления системы булевых функций подсистем для совместной логической минимизации / П. Н. Бибило, Н. А. Кириенко, В. И. Романов // Программные продукты и системы. – 2023. – Т. 36, № 4. – С. 509–522. – DOI: 10.15827/0236-235X.142.509-522.
17. Бибило, П. Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum / П. Н. Бибило. – М. : СОЛОН-Пресс, 2005. – 384 с.
18. Бибило, П. Н. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов // Проблемы разработки перспективных микро- и наноэлектронных систем. – 2020. – Вып. 4. – С. 9–16. – DOI: 10.31114/2078-7707-2020-4-9-16.
19. Бибило, П. Н. Экспериментальное сравнение эффективности алгоритмов оптимизации BDD-представлений систем булевых функций / П. Н. Бибило, Ю. Ю. Ланкевич // Программные продукты и системы. – 2020. – Т. 33, № 3. – С. 449–463. – DOI: 10.15827/0236-235X.131.449-463.
20. Ashenden, P. J. VHDL–2008. Just the New Stuff / P. J. Ashenden, J. Lewis. – Burlington, MA, USA : Morgan Kaufman Publishers, 2008. – 909 p.

References

1. Zakrevskij A. D. Logicheskij sintez kaskadnyh skhem. *Logical Synthesis of Cascading Circuit*. Moscow, Nauka, 1981, 416 p. (In Russ.).
2. Brayton K. R., Hachtel G. D., McMullen C., Sangiovanni-Vincentelli A. L. *Logic Minimization Algorithm for VLSI Synthesis*. Boston, Kluwer Academic Publishers, 1984, 193 p.
3. Sasao T. Input variable assignment and output phase optimization of PLA's. *IEEE Transactions on Computers*, 1984, vol. C-33, no. 10, pp. 879–894.
4. Wey C. L., Chang T. Y. An efficient output assignment for PLA minimization. *IEEE Transactions on Computer-Aided Design*, 1990, vol. 9, pp. 1–7.
5. Ebendt R., Fey G., Drechsler R. *Advanced BDD Optimization*. Springer, 2005, 222 p.
6. Knuth D. E. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Addison-Wesley Professional, 2011, 912 p.
7. Bibilo P. N. Binarnye diagrammy reshenij v logicheskom proektirovanii. *Binary Decision Diagrams in Logical Design*. Moscow, Lenand, 2024, 560 p. (In Russ.).

8. Brayton R. K., Hachtel G. D., Sangiovanni-Vincentelli A. L. *Synthesis of multilevel combinational logic circuits*. Trudy Instituta inzhenerov po jelektrotehnike i radioelektronike [*Proceedings of the IEEE*], 1990, vol. 78, no. 2, pp. 38–83 (In Russ.).
9. Zakrevskij A. D. (ed.). *Sintez asinhronnyh avtomatov na JeVM. Synthesis of Asynchronous Automata on a Computer*. Minsk, Nauka i tekhnika, 1975, 184 p. (In Russ.).
10. 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. CAD-6, no. 6, pp. 1062–1081.
11. Brayton K. R. Factoring logic functions. *IBM Journal of Research and Development*, 1987, vol. 31, no. 2, pp. 187–198.
12. Amaru L. G. *New Data Structures and Algorithms for Logic Synthesis and Verification*. Springer, 2017, 156 p.
13. Lozhkin S. A., Zizov V. S., Shupletsov M. S., Zhukov V. V., Khzmalian D. E., Belyankov O. O. *On complexity of inverter graphs for Boolean functions of small number of variables*. Problemy razrabotki perspektivnyh mikro- i nanoelektronnyh system : sbornik trudov [*Problems of Developing Promising Micro- and Nanoelectronic Systems : Collection of Works*]. In A. L. Stempkovskogo (ed.). 2020, iss. 4, pp. 95–102 (In Russ.). DOI: 10.31114/2078-7707-2020-4-95-102.
14. Soeken M., Amaru L. G., Gaillardon P., De Micheli G. Optimizing majority-inverter graphs with functional hashing. *Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 14–18 March 2016*. Dresden, 2016, pp. 1030–1035.
15. Soeken M., Amaru L., Gaillardon P.-E., De Micheli G. Exact synthesis of majority-inverter graphs and its applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017, vol. 36, no. 11, pp. 1842–1855.
16. Bibilo P. N., Kirienko N. A., Romanov V. I. *Extracting subsystems from a multilevel representation of a Boolean function system for joint logical minimization*. Programmnye produkty i sistemy [*Software & Systems*], 2023, vol. 36, no. 4, pp. 509–522 (In Russ.). DOI: 10.15827/0236-235X.142.509-522.
17. 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 Russ.).
18. Bibilo P. N., Romanov V. I. *The system of logical optimization of digital circuits functional structural descriptions based on production-frame knowledge representation model*. Problemy razrabotki perspektivnyh mikro- i nanoelektronnyh system [*Problems of Advanced Micro- and Nanoelectronic Systems Development*], 2020, iss. 4, pp. 9–16 (In Russ.). DOI: 10.31114/2078-7707-2020-4-9-16.
19. Bibilo P. N., Lankevich Yu. Yu. *Experimental investigation of effectiveness of algorithms for minimizing BDD representations of Boolean function systems*. Programmnye produkty i sistemy [*Software & Systems*], 2020, vol. 33, no. 3, pp. 449–463 (In Russ.). DOI: 10.15827/0236-235X.131.449-463.
20. Ashenden P. J., Lewis J. *VHDL–2008. Just the New Stuff*. Burlington, MA, USA, Morgan Kaufman Publishers, 2008, 909 p.

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

Кириенко Наталья Алексеевна, кандидат технических наук, доцент, Объединенный институт проблем информатики Национальной академии наук Беларуси. E-mail: kir@newman.bas-net.by

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, D. Sc. (Eng.), Prof., The United Institute of Informatics Problems of the National Academy of Sciences of Belarus. E-mail: bibilo@newman.bas-net.by

Natalia A. Kirienko, Ph. D. (Eng.), Assoc. Prof., The United Institute of Informatics Problems of the National Academy of Sciences of Belarus. E-mail: kir@newman.bas-net.by

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