ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ
LOGICAL DESIGN

УДК 519.714.5
https://doi.org/10.37661/1816-0301-2024-21-1-28-47

Оригинальная статья
Original Paper

Технологически независимая оптимизация при реализации в заказных СБИС разреженных систем дизъюнктивных нормальных форм булевых функций

П. Н. Бибило✉, С. Н. Кардаш

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

Аннотация

Цели. Рассматривается проблема выбора лучших методов и программ для схемной реализации в заказных цифровых СБИС разреженных систем дизъюнктивных нормальных форм (ДНФ) полностью определенных булевых функций. Для матричных форм разреженных систем ДНФ троичная матрица, задающая элементарные конъюнкции, содержит большую долю неопределенных значений, соответствующих в алгебраической записи отсутствующим литералам булевых входных переменных, а булева матрица, задающая вхождения конъюнкций в ДНФ функций, содержит большую долю нулевых значений.

Методы. Предлагается исследовать различные методы технологически независимой логической оптимизации, выполняемой на первом этапе логического синтеза: совместную минимизацию систем функций в классе ДНФ, раздельную и совместную минимизацию в классах многоуровневых представлений в виде булевых сетей и BDD-представлений с использованием взаимно инверсных кофакторов, разбиение системы функций на подсистемы с ограниченным числом входных переменных, а также метод блочного покрытия систем ДНФ, ориентированный на минимизацию суммарной площади блоков, образующих покрытие.

Результаты. При реализации в заказных СБИС разреженных систем ДНФ булевых функций наряду с традиционными методами совместной минимизации систем функций в классе ДНФ для технологически независимой оптимизации могут применяться методы оптимизации многоуровневых представлений систем булевых функций на основе разложений Шеннона, при этом раздельная минимизация и совместная минимизация всей системы в целом оказываются менее эффективными по сравнению с блочными разбиениями и покрытиями системы ДНФ и последующей минимизацией многоуровневых представлений. Схемы, полученные в результате синтеза по минимизированным представлениям булевых сетей, чаще имеют меньшую площадь, чем схемы, полученные по минимизированным BDD-представлениям.

Заключение. Для проектирования схем заказных цифровых СБИС показана эффективность комбинированного подхода, использующего сначала программы блочного покрытия системы ДНФ с последующим применением программ минимизации многоуровневых представлений блоков в виде булевых сетей, минимизированных на основе разложений Шеннона.

Ключевые слова: система булевых функций, ДНФ, минимизация ДНФ, бинарная диаграмма решений, булева сеть, разложение Шеннона, блочное покрытие системы ДНФ, синтез логической схемы, заказная СБИС, VHDL

Для цитирования. Бибило, П. Н. Технологически независимая оптимизация при реализации в заказных СБИС разреженных систем дизъюнктивных нормальных форм булевых функций / П. Н. Бибило, С. Н. Кардаш // Информатика. – 2024. − Т. 21, № 1. – С. 28–47.
https://doi.org/10.37661/1816-0301-2024-21-1-28-47

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 26.12.2023
Подписана в печать | Accepted 19.01.2024
Опубликована | Published 29.03.2024

Technology independent optimization when implementing sparse systems of disjunctive normal forms of Boolean functions in ASIC

Petr N. Bibilo✉, Sergey N. Kardash

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 problem of choosing the best methods and programs for circuit implementation as part of digital ASIC (Application-Specific Integrated Circuit) sparse systems of disjunctive normal forms (DNF) of completely defined Boolean functions is considered. For matrix forms of sparse DNF systems, the ternary matrix specifying elementary conjunctions contains a large proportion of undefined values corresponding to missing literals of Boolean input variables, and the Boolean matrix specifying the occurrences of conjunctions in DNF functions contains a large proportion of zero values.

Methods. It is proposed to investigate various methods of technologically independent logical optimization performed at the first stage of logical synthesis: joint minimization of systems of functions in the DNF class, separate and joint minimization in classes of multilevel representations in the form of Boolean networks and BDD representations using mutually inverse cofactors, as well as the division of a system of functions into subsystems with a limited number of input variables and the method of block cover of DNF systems, focused on minimizing the total area of the blocks forming the cover.

Results. When implementing sparse DNF systems of Boolean functions in ASIC, along with traditional methods of joint minimization of systems of functions in the DNF class, methods for optimizing multilevel representations of Boolean function systems based on Shannon expansions can be used for technologically independent optimization, while separate minimization and joint minimization of the entire system as a whole turn out to be less effective compared with block partitions and coatings of the DNF system and subsequent minimization of multilevel representations. Schemes obtained as a result of synthesis using minimized representations of Boolean networks often have a smaller area than schemes obtained using minimized BDD representations.

Conclusion. For the design of digital ASIC, the effectiveness of combined approach is shown, when initially the block coverage programs of the DNF system is used, followed by the use of programs to minimize multilevel block representations in the form of Boolean networks minimized based on Shannon expansion.

Keywords: Boolean function system, DNF, DNF minimization, Binary Decision Diagram, Boolean network, Shannon expansion, block cover of the DNF system, logic synthesis, ASIC, VHDL

For citation. Bibilo P. N., Kardash S. N. *Technology independent optimization when implementing sparse systems of disjunctive normal forms of Boolean functions in ASIC*. Informatika [*Informatics*], 2024, vol. 21, no. 1, pp. 28−47 (In Russ.). https://doi.org/10.37661/1816-0301-2024-21-1-28-47

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

Введение. Проблема эффективной схемной реализации цифровых комбинационных блоков в заказных КМОП СБИС (сверхбольших интегральных схемах, выполненных по комплементарной металл-оксид-полупроводник технологии) по-прежнему актуальна при создании средств автоматизированного проектирования цифровых систем. Важным аспектом этой проблемы является то, что современные синтезаторы логических схем чувствительны к форме задания проектной информации, в качестве которой выступают VHDL- либо Verilog-описания [1] моделей функционирования комбинационных схем – те или иные формы задания систем полностью определенных булевых функций. Синтез логических схем выполняется в два этапа: технологически независимая оптимизация представлений систем булевых функций (первый этап) и технологическое отображение в заданный базис (библиотеку) логических элементов заказной СБИС (второй этап). Важнейшим является первый этап, на котором выбирается форма представления системы булевых функций и осуществляется минимизация этой формы. Результат выполнения первого этапа определяет и важнейшие параметры синтезированной на втором этапе логической схемы – площадь, временную задержку и энергопотребление.

Методы и программы технологически независимой оптимизации традиционно развивались для исходных заданий реализуемых систем булевых функций в виде систем ДНФ. Широко известны методы совместной и раздельной минимизации систем булевых функций в классе ДНФ [2, 3], методы факторизации – выделения общих (одинаковых) частей конъюнкций, дизъюнкций и одинаковых подвыражений в скобочных алгебраических представлениях систем булевых функций [4–6], а также многочисленные методы раздельной и совместной функциональной декомпозиции систем булевых функций [7–10 и др.]. В последнее время в качестве методов технологически независимой оптимизации выступают методы минимизации многоуровневых представлений систем функций на основе разложения Шеннона – это методы минимизации BDD (Binary Decision Diagram, бинарная диаграмма решений) [11–17], модификаций BDD [18] и булевых сетей [19]. Предложены также и другие структуры данных [20–24] для представления систем булевых функций и соответствующие методы минимизации.

В настоящей работе рассматриваются разреженные системы ДНФ полностью определенных булевых функций. Для матричных форм таких систем ДНФ троичная матрица, задающая элементарные конъюнкции, содержит большу́ю долю неопределенных значений, а булева матрица, задающая вхождения конъюнкций в ДНФ функций, содержит большу́ю долю нулевых значений и, следовательно, небольшую долю единичных значений. Для разреженных систем ДНФ булевых функций предлагаются алгоритмы их блочного покрытия. Проводится экспериментальное исследование эффективности применения блочных многоуровневых представлений при синтезе комбинационных блоков заказных СБИС в библиотеке КМОП-элементов. Синтезированные схемы сравниваются по площади и временной задержке. Многоблочные многоуровневые представления сравниваются по результатам синтеза с раздельными и совместными многоуровневыми представлениями исходной системы ДНФ и минимизированными двухуровневыми представлениями, под которыми понимаются совместно минимизированные ДНФ. В качестве базовых многоуровневых представлений использованы бинарные диаграммы решений с инверсными кофакторами (BDDI-представления) и булевы сети (Bool-представления). Минимизация BDDI-представлений выполняется по матричным заданиям систем ДНФ булевых функций, Bool-представлений – по логическим уравнениям, задающим те же системы ДНФ. В результате проведенных экспериментов установлено, что для разреженных систем ДНФ наряду с методами минимизации систем функций в классе ДНФ эффективным методом технологически независимой оптимизации является комбинированный метод, включающий блочное покрытие системы ДНФ и последующую минимизацию многоуровневых представлений бло-

ков, при этом функции блоков предпочтительнее минимизировать в классе булевых сетей с использованием разложений Шеннона.

Многоуровневые BDDI-представления систем булевых функций. Под *BDDI-представлением* (BDDI – Binary Decision Diagram with Inverse cofactors) понимается ориентированный бесконтурный граф, задающий последовательные разложения Шеннона булевой функции $f(\boldsymbol{x})=f(x_1,\ldots,x_n)$, $\boldsymbol{x}=(x_1,\ldots,x_n)$, либо системы $\boldsymbol{f}(\boldsymbol{x})=(f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x}))$ булевых функций по всем переменным $x_1, x_2, \ldots, x_n$ при заданном порядке (перестановке) переменных, по которым проводятся разложения, при условии нахождения пар взаимно инверсных кофакторов.

Разложением Шеннона булевой функции $f(\boldsymbol{x})$ по переменной x_i называется представление

$$f(\boldsymbol{x})= \overline{x}_i\, f_0 \vee x_i\, f_1 . \qquad (1)$$

Функции $f_0=f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$, $f_1=f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n)$ в правой части представления (1) называются кофакторами (англ. cofactors) разложения по переменной x_i. Каждый из кофакторов $f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$, $f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n)$ может быть разложен по одной из переменных из множества $\{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n\}$. Процесс разложения кофакторов заканчивается, когда все n переменных будут использованы для разложения. 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 кофакторов является наименьшим [18]. Если для каждой функции системы соответствующая ей BDDI строится независимо, то такая минимизация называется *раздельной*. При построении раздельных BDDI могут появляться одинаковые кофакторы в BDDI различных функций системы, однако данный факт при построении BDDI для каждой отдельной функции во внимание не принимается.

Многоуровневые Bool-представления систем булевых функций. *Bool-представление* системы булевых функций соответствует булевой сети (ориентированному бесконтурному графу), функциями вершин которой могут быть логические операции «конъюнкции» либо «дизъюнкции» (возможно с инверсией) над литералами булевых переменных. *Литерал* – это булева переменная либо ее инверсия. Таким образом, вершина булевой сети имеет две заходящие дуги и может иметь одну либо две исходящие дуги, соответствующие прямому и инверсному выходу. При этом предполагается, что доступны как прямые, так и инверсные значения входных переменных [18]. Логическая минимизация булевых сетей на основе разложения Шеннона заключается в поиске такой перестановки переменных разложения, при которой число литералов в булевой сети является наименьшим. В булевой сети разложение Шеннона записывается виде трех формул

$$f(\boldsymbol{x})= w_0 \vee w_1 ;\ w_0 = \overline{x}_i\, f_0 ;\ w_1 = x_i\, f_1 . \qquad (2)$$

Формулы (2) содержат шесть литералов, в то время как формула (1) содержит четыре литерала булевых переменных. Это обстоятельство следует иметь в виду при сравнении сложностей BDDI- и Bool-представлений по числу литералов. Однако как формула (1), так и формулы (2) содержат по три логических оператора – два оператора конъюнкции и один оператор дизъюнкции. 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 [19].

Раздельные BDDI- и Bool-минимизации для выделенных подсистем системы булевых функций (либо для отдельных функций системы) заключаются в нахождении своей перестанов-

ки $< x_1, x_2, ..., x_n >$ переменных разложения для каждой из подсистем функций, в то время как при совместной BDDI- и Bool-минимизации используется одна и та же перестановка переменных разложения для всех функций $f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x})$ системы $\boldsymbol{f}(\boldsymbol{x})$. Для некоторых систем функций преимущество при синтезе имеет совместная минимизация, для других – раздельная BDDI- либо Bool-минимизация. Обычно раздельная минимизация позволяет получать схемы, характеризуемые бóльшим быстродействием.

Минимизация в классе ДНФ. Кратчайшей системой ДНФ D_f для системы булевых функций $\boldsymbol{f}(\boldsymbol{x})=(f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x}))$ называется система ДНФ, содержащая минимальное число общих элементарных конъюнкций, на которых заданы ДНФ D_{f^i}, i=1,…, m, всех функций $f^i(\boldsymbol{x})$ системы $\boldsymbol{f}(\boldsymbol{x})=(f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x}))$. Задача совместной минимизации системы булевых функций заключается в нахождении кратчайшей системы ДНФ D_f для заданной системы $\boldsymbol{f}(\boldsymbol{x})=(f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x}))$ булевых функций. Совместная минимизация систем булевых функций в экспериментах выполнялась программой Espresso IIC [3].

Блочное покрытие системы ДНФ булевых функций. Пусть (T^x, B^f) – пара матриц, задающая матричную форму системы ДНФ булевых функций $\boldsymbol{f}(\boldsymbol{x})=(f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x}))$, $\boldsymbol{x}=(x_1,\ldots,x_n)$, где T^x – троичная матрица, задающая общие элементарные конъюнкции, B^f – булева матрица, единичные элементы которой отмечают вхождения элементарных конъюнкций в ДНФ функций [2]. Система ДНФ

$$
\begin{aligned}
f^1 &= x_1\bar{x}_2\bar{x}_3x_4 \vee x_2x_3\bar{x}_5 \vee \bar{x}_1x_4x_5;\\
f^2 &= x_5x_6\bar{x}_7\bar{x}_8\bar{x}_9 \vee \bar{x}_4x_5\bar{x}_6x_8x_9x_{10} \vee x_4x_5x_8;\\
f^3 &= x_5x_6\bar{x}_7\bar{x}_8\bar{x}_9 \vee \bar{x}_4x_5\bar{x}_6x_8x_9x_{10} \vee x_1\bar{x}_2\bar{x}_4x_5 \vee x_2x_3\bar{x}_5 \vee \bar{x}_1x_4x_5;\\
f^4 &= x_5x_6\bar{x}_7\bar{x}_8\bar{x}_9 \vee \bar{x}_4x_5\bar{x}_6x_8x_9x_{10} \vee \bar{x}_1\bar{x}_2x_8x_{10} \vee x_1\bar{x}_8x_{10} \vee x_4x_5x_8;\\
f^5 &= x_2x_8\bar{x}_9 \vee x_1\bar{x}_8x_{10};\\
f^6 &= x_1x_2\bar{x}_8x_9x_{10} \vee \bar{x}_1\bar{x}_2x_8x_{10} \vee x_2x_8\bar{x}_9
\end{aligned}
\qquad (3)
$$

задается парой матриц (T^x, B^f) в табл. 1.

Таблица 1
Система ДНФ булевых функций

Table 1
The DNF system of Boolean functions

Номер строки *Line number*	Троичная матрица T^x *Ternary matrix* T^x										Булева матрица B^f *Boolean matrix* B^f					
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	f^1	f^2	f^3	f^4	f^5	f^6
1	1	1	–	–	–	–	–	0	1	1	0	0	0	0	0	1
2	–	–	–	–	1	1	0	0	0	–	0	1	1	1	0	0
3	–	–	–	0	1	0	–	1	1	1	0	1	1	1	0	0
4	0	0	–	–	–	–	–	1	–	1	0	0	0	1	0	1
5	1	0	–	0	1	–	–	–	–	–	1	0	1	0	0	0
6	–	1	1	–	0	–	–	–	–	–	1	0	1	0	0	0
7	–	1	–	–	–	–	–	1	0	–	0	0	0	0	1	1
8	1	–	–	–	–	–	–	0	–	1	0	0	0	1	1	0
9	0	–	–	1	1	–	–	–	–	–	1	0	1	0	0	0
10	–	–	–	1	1	–	–	1	–	–	0	1	0	1	0	0

Число n переменных равно 10 (n=10), число функций m равно 6 (m=6), ДНФ заданы на k=10 общих элементарных конъюнкциях, число d операторов дизъюнкции в системе (3) равно 15 (d=15). Площадь $Q(T^x, B^f)$ матриц будем вычислять по формуле (4) и выражать в числе бит:

$$Q(T^x, B^f)=(n+m)k. \quad (4)$$

Рассмотрим систему ДНФ, каждая элементарная конъюнкция которой включает не более t литералов. Это значит, что в каждой строке троичной матрицы T^x находится не более t определенных 0, 1 элементов (остальные элементы равны «–»). Рассмотрим пару (T_{H_i}, B_{H_i}) подматриц, где T_{H_i} – строчная (образованная некоторыми строками матрицы T^x) подматрица матрицы T^x, B_{H_i} – подматрица матрицы B^f, заданная на том же подмножестве строк, что и T_{H_i}. Назовем пару (T_{H_i}, B_{H_i}) *блоком* H_i. Пусть $p \geq t$.

Блок H_i назовем (p,s,q)-*ограниченным* (рис. 1), если одновременно выполняются следующие условия:

– число столбцов T_{H_i}, содержащих определенные элементы 0, 1, не превышает p;

– число ненулевых столбцов матрицы B_{H_i} не превышает q;

– число строк подматриц T_{H_i}, B_{H_i} не превышает s.

Блок (T_{H_i}, B_{H_i}) назовем (p,q)-*ограниченным*, если значение параметра s не ограничивается, т. е. всегда предполагается $s=k$. Блок назовем p-*ограниченным*, если значения параметров s и q не ограничиваются, т. е. всегда предполагается s=k и q=m.

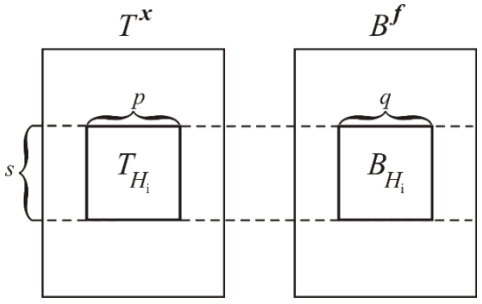


Рис. 1. (p,s,q)-ограниченный блок H_i матричной формы системы ДНФ

Fig. 1. (p,s,q) is a limited H_i block of the matrix form of the DNF system

Видно, что каждой паре (T_{H_i}, B_{H_i}) соответствует своя система ДНФ булевых функций. Рассмотрим три пары подматриц, заданных в табл. 2–4.

Таблица 2
Система ДНФ булевых функций блока H_1

Table 2
DNF system of Boolean block functions H_1

Номер строки *Line number*	Троичная матрица T_{H_1} *Ternary matrix* T_{H_1}	Булева матрица B_{H_1} *Boolean matrix* B_{H_1}
	x_1 x_2 x_8 x_9 x_{10}	f_1^4 f^5 f^6
1	1 1 0 1 1	0 0 1
4	0 0 1 – 1	1 0 1
7	– 1 1 0 –	0 1 1
8	1 – 0 – 1	1 1 0

Таблица 3
Система ДНФ булевых функций блока H_2

Table 3
DNF system of Boolean block functions H_2

Номер строки *Line number*	Троичная матрица T_{H_2} *Ternary matrix* T_{H_2}	Булева матрица B_{H_2} *Boolean matrix* B_{H_2}
	$x_4\ x_5\ x_6\ x_7\ x_8\ x_9$	$f^2\ f_1^3\ f_2^4$
2	– 1 1 0 0 0	1 1 1
3	0 1 0 – 1 1	1 1 1
10	1 1 – – 1 –	1 0 1

Таблица 4
Система ДНФ булевых функций блока H_3

Table 4
DNF system of Boolean block functions H_3

Номер строки *Line number*	Троичная матрица T_{H_3} *Ternary matrix* T_{H_3}	Булева матрица B_{H_3} *Boolean matrix* B_{H_3}
	$x_1\ x_2\ x_3\ x_4\ x_5$	$f^1\ f_2^3$
5	1 0 – 0 1	1 1
6	– 1 1 – 0	1 1
9	0 – – 1 1	1 1

Пара (T_{H_1}, B_{H_1}) является (5,4,3)-ограниченной и имеет площадь 32 бит, пара (T_{H_2}, B_{H_2}) является (6,3,3)-ограниченной и имеет площадь 27 бит, пара (T_{H_3}, B_{H_3}) является (5,3,2)-ограниченной и имеет площадь 21 бит.

Множество $\{H_1,\ldots, H_v\}=\{(T_{H_1}, B_{H_1}),\ldots, (T_{H_v}, B_{H_v})\}$ блоков назовем блочным дизъюнктивным покрытием (далее *блочным покрытием*) пары матриц (T^x, B^f), если каждый единичный элемент матрицы B^f входит только в одну из подматриц B_{H_i}, а каждая строка матрицы T^x входит хотя бы в одну из подматриц T_{H_i}, i=1,…,v.

Блочное покрытие (p,q)-ограниченными блоками будет являться разбиением исходной системы ДНФ на непересекающиеся подсистемы функций, если все функции каждого блока H_i (рис. 2) будут зависеть от одного и того же подмножества переменных, мощность которого не более p. Если же каждая строка троичной матрицы T^x будет входить только в один блок, то блочное покрытие будет задавать разбиение множества строк матрицы T^x на непересекающиеся подмножества T_{H_i}.

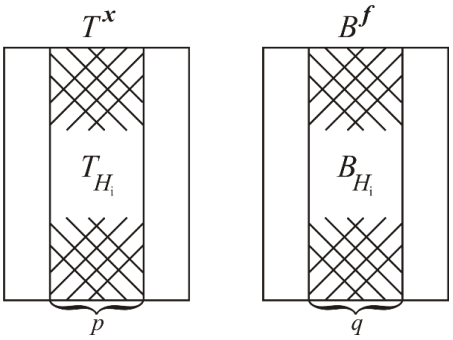


Рис. 2. Блок H_i содержит q функций, зависящих от p переменных

Fig. 2. The H_i block contains q functions depending on p variables

Если же для подсистемы, состоящей из q функций, число ее аргументов превышает число p, то для реализации блочного разложения требуются операции дизъюнкции для некоторых (либо всех) функций нескольких (возможно всех) блоков. На рис. 3 показан этот случай. Для m-входового дизъюнктора площадь будем подсчитывать по формуле $6(m–1)$ бит.

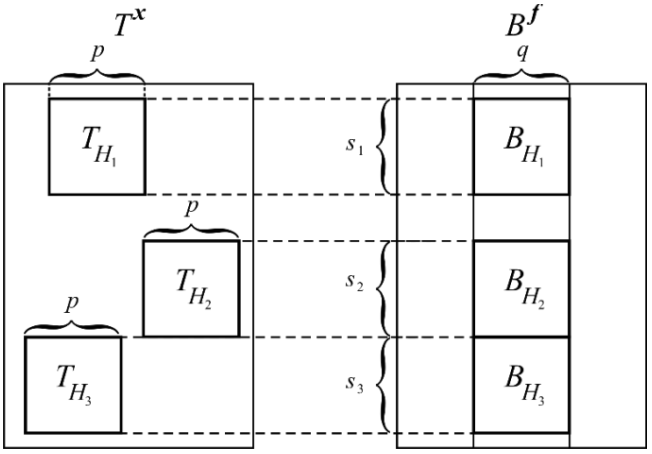


Рис. 3. Подсистема, содержащая q функций, которые зависят более чем от p переменных

Fig. 3. A subsystem containing q functions that depend on more than p variables

Задача 1 блочного покрытия системы ДНФ по критерию минимальности числа блоков: найти покрытие пары (T^x, B^f) возможно меньшим числом (p,q)-ограниченных блоков (T_{H_i}, B_{H_i}), $i=1,\ldots, v$.

Задача 2 блочного покрытия системы ДНФ по критерию минимальности площади блоков: найти покрытие пары (T^x, B^f) возможно меньшим числом p-ограниченных блоков (T_{H_i}, B_{H_i}), $i=1,\ldots, w$, имеющих возможно меньшую суммарную площадь с учетом площадей логических элементов, реализующих дизъюнкции функций.

Алгоритмы и программы решения задач блочного покрытия системы ДНФ описаны в работе [25]. Они являются эвристическими и итерационными – на каждой итерации формируется один блок на основе эвристик выбора столбцов и строк соответствующих подматриц очередного блока. После этого обнуляются те единичные элементы булевой матрицы B^f, которые принадлежат сформированному блоку. Процесс формирования блоков заканчивается, когда все элементы матрицы B^f становятся нулевыми.

Применение программы решения задачи 2 (для $p=6$) блочного покрытия с минимизацией площади блоков для системы ДНФ из табл. 1 позволяет получить три блока H_1, H_2, H_3 (рис. 4), заданных в табл. 2–4 соответственно.

В табл. 3 ДНФ функций f^2, f_2^4 одинаковы, т. е. $f^2 = f_2^4$, в табл. 4 также задаются одинаковые ДНФ $f_1^4 = f_2^3$, поэтому можно сказать, что блочное покрытие является приемом логической оптимизации и позволяет выделять общие подфункции в дизъюнктивных разложениях системы ДНФ. Заметим, что факты равенства ДНФ функций, принадлежащих одному и тому же блоку, устанавливаются при последующей многоуровневой минимизации функций этого блока.

Если не стремиться уменьшить число блоков в блочном покрытии системы ДНФ, то решение задачи 2 по критерию минимальной суммарной площади можно свести к нахождению k-блочного покрытия, где каждый блок формируется по одной из k строк матриц T^x, B^f. В примере таким покрытием будет 10-блочное покрытие с общей суммарной площадью, равной 59 бит, при этом число дополнительных дизъюнкций для представления логической сети будет равно 15, как и в формулах (3).

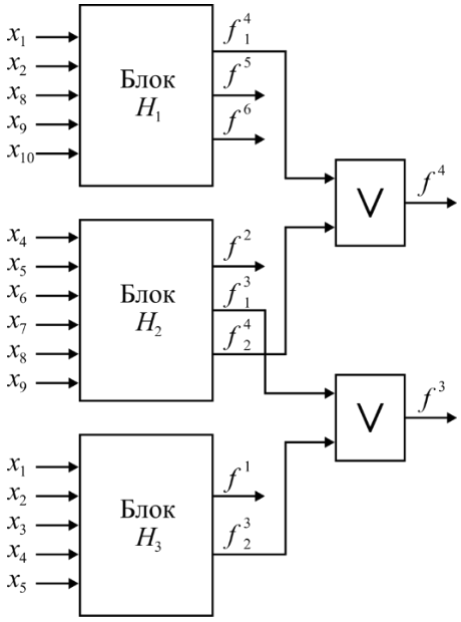


Рис. 4. Логическая схема, соответствующая блочному покрытию (табл. 2–4) системы ДНФ (табл. 1)

Fig. 4. The circuit corresponding to the block cover (Tables 2–4) of the DNF system (Table 1)

Площадь матриц T^x, B^f из табл. 1 составляет $Q(T^x,B^f)=(n+m)k=160$ (бит), суммарная площадь трех блоков, заданных в табл. 2–4, составляет 32+27+21=80 (бит), что в два раза меньше площади матриц из табл. 1, при этом понадобятся только две двухвходовые дизъюнкции (рис. 4). Матричная форма оператора двухвходовой дизъюнкции имеет площадь 6 бит. Поэтому общая площадь составляет 80+12=92 (бит). Если для той же системы ДНФ (см. табл. 1) решить задачу 1, уменьшив значение параметра p, положив p=5, q=3 и выполнив программу блочного покрытия по критерию минимальности числа блоков, то получим четыре блока в покрытии ДНФ с суммарной площадью блоков, равной 77 и 30 бит. В этой логической сети имеется один оператор двухвходовой дизъюнкции и два оператора трехвходовой дизъюнкции с общей площадью 30 бит, так как матричная форма трехвходового оператора дизъюнкции составляет 12 бит.

Таким образом, изменяя параметры p, q, можно получать различные блочные покрытия системы ДНФ, характеризуемые разной суммарной площадью, разным числом блоков и разным числом дизъюнкций, требуемых для формирования выходных функций.

Разреженные системы ДНФ. Под *разреженностью* α *троичной матрицы* T^x будем понимать отношение числа неопределенных элементов «–» к числу всех элементов этой матрицы и выражать это отношение в процентах. Например, троичная матрица T^x (табл. 1) содержит 62 неопределенных значения «–», общее число элементов матрицы T^x равно 100 (матрица состоит из 10 столбцов и 10 строк). Следовательно, α =62 %.

Под *разреженностью* β *булевой матрицы* B^f будем понимать долю числа ее нулевых элементов, выраженную в процентах. В булевой матрице B^f (табл. 1) число нулевых элементов равно 39. Следовательно, β=39/60=0,65, что составляет 65 %. Чем большее значение имеют параметры α и β, тем более разреженной является матричная форма системы ДНФ.

Исходные данные для экспериментов. Системы булевых функций для экспериментов (табл. 5) были заданы в двух формах – матричной и форме логических уравнений. В табл. 5 используются следующие обозначения: n – число аргументов системы ДНФ булевых функций, m – число функций, k – число общих элементарных конъюнкций, d – число дизъюнкций в системе ДНФ булевых функций, α – разреженность (в процентах) троичной матрицы T^x, β –

разреженность (в процентах) булевой матрицы B^f. Примеры Pozd_1, Pozd_2 – это «блочные» системы ДНФ. Параметры трехблочной системы ДНФ Pozd_1 (рис. 5): n_1=n_2=n_3=12, m_1=m_2=m_3=10, k_1=263, k_2=358, k_3=133; параметры трехблочной системы ДНФ Pozd_2: n_1=n_2=n_3=12, m_1=m_2=m_3=10, k_1=263, k_2=137, k_3=205. Соседние блоки в примерах Pozd_1 и Pozd_2 имеют две и четыре общие входные переменные соответственно.

Таблица 5
Исходные данные – разреженные системы ДНФ булевых функций

Table 5
Initial data – sparse DNF systems of Boolean functions

Пример *Example*	*n*	*m*	*k*	*d*	α	β
C8	28	18	70	103	89,5	78,0
DALU	75	16	194	1145	94,0	58,7
LAL	26	19	117	67	83,8	71,8
PM1	16	13	42	27	83,6	70,0
SCT	19	15	64	76	98,9	63,3
TTT2	24	21	222	203	80,7	71,9
Alu4	14	8	1 028	1 020	45,2	40,3
Apex5	117	88	1 227	1 142	95,0	94,3
I2c	147	142	1 357	1 251	98,6	97,8
X1	51	35	324	289	87,0	78,0
X3	135	99	915	523	92,4	93,9
X4	94	71	371	277	94,5	90,8
Blocki1	15	16	355	506	58,1	71,5
Blocki2	15	16	90	101	64,2	75,4
Pozd_1	30	30	754	1 516	79,2	82,5
Pozd_2	30	30	605	1 805	75,2	77,8

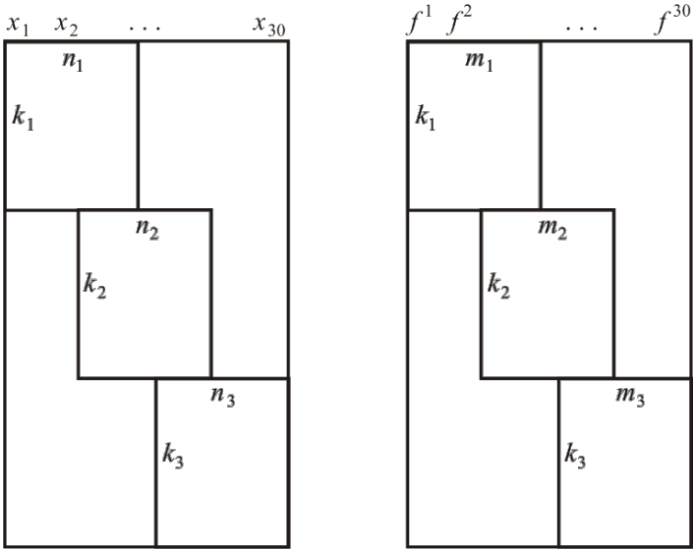


Рис. 5. Структура систем ДНФ примеров Pozd_1 и Pozd_2

Fig. 5. Structure of DNF systems of examples Pozd_1 and Pozd_2

Для примера Blocki1 (двухблочной системы ДНФ) n_1=11, n_2=7, m_1=m_2=8, k_1=255, k_2=100. Для примера Blocki2 (двухблочной системы ДНФ) n_1=10, n_2=7, m_1=m_2=8, k_1=42, k_2=48. Блоки имеют две общие входные переменные для обоих примеров Blocki1 и Blocki2. Пример I2c взят

из библиотеки (http://lsi.epfl.ch/benchmarks) примеров описаний (логических уравнений) графов AIG (And-Inverter Graph). Остальные 11 примеров (табл. 5) – это разреженные системы ДНФ, взятые из библиотеки примеров LGSynth91. Исходные описания примеров Alu4, Apex5 даны в библиотеке в формате PLA, остальные девять примеров – в формате Blif. Все примеры были переведены в матричный формат (SDF) языка SF в системе FLC-2 [26]. Исходные функциональные описания примеров C8, DALU, LAL, PM1, SCT, TTT2, X1, X3, X4 не содержат инверсий литералов входных переменных, т. е. троичные матрицы T^x для этих примеров содержат только 1 и «–». Основными критериями выбора примеров являлись практическая размерность (десятки аргументов и функций) и возможно большая разреженность матричных заданий систем ДНФ. Рассмотрим матричную форму примера Apex5. Троичная матрица T^x включает 117 столбцов (переменных) и 1227 строк (элементарных конъюнкций), площадь этой матрицы 117 × 1227=143 559, она содержит 136 453 неопределенных элемента «–», разреженность этой матрицы 136 453/143 559=0,95, т. е. 95 %. Булева матрица B^f включает 88 столбцов (функций) и 1227 строк, площадь этой матрицы 88 × 1227=107 976, данная матрица содержит 101 759 нулевых элементов, ее разреженность 101 759/107 976=0,943, т. е. 94,3 %. Пример Alu4 является наименее разреженным – доля неопределенных элементов в троичной матрице T^x меньше половины, доля единичных элементов в булевой матрице B^f также меньше половины.

Эксперименты. Всего было проведено 10 экспериментов по выяснению эффективности алгоритмов и программ технологически независимой оптимизации, используемых при синтезе функциональных блоков заказных КМОП СБИС. Этапы экспериментов показаны на рис. 6.

Сначала осуществлялся перевод всех функциональных описаний в стандартные форматы, используемые в системе FLC-2. Затем (кроме эксперимента 1) выполнялась технологически независимая оптимизация, включающая раздельную, совместную либо блочную (комбинированную) логическую минимизацию.

После логической минимизации минимизированные описания представлений систем функций в виде логических уравнений конвертировались в VHDL-описания [1, 27] и подавались на вход синтезатора LeonardoSpectrum. Для всех примеров синтез осуществлялся с одними и теми же опциями управления синтезом и для одной и той же целевой библиотеки синтеза. Синтезатор LeonardoSpectrum [27] имеет свои средства логической минимизации, он перерабатывает входное описание, получает свое (внутреннее) описание, по которому и синтезируется схема. Библиотекой синтеза являлась библиотека проектирования заказных цифровых КМОП СБИС, ее состав приведен в работе [28].

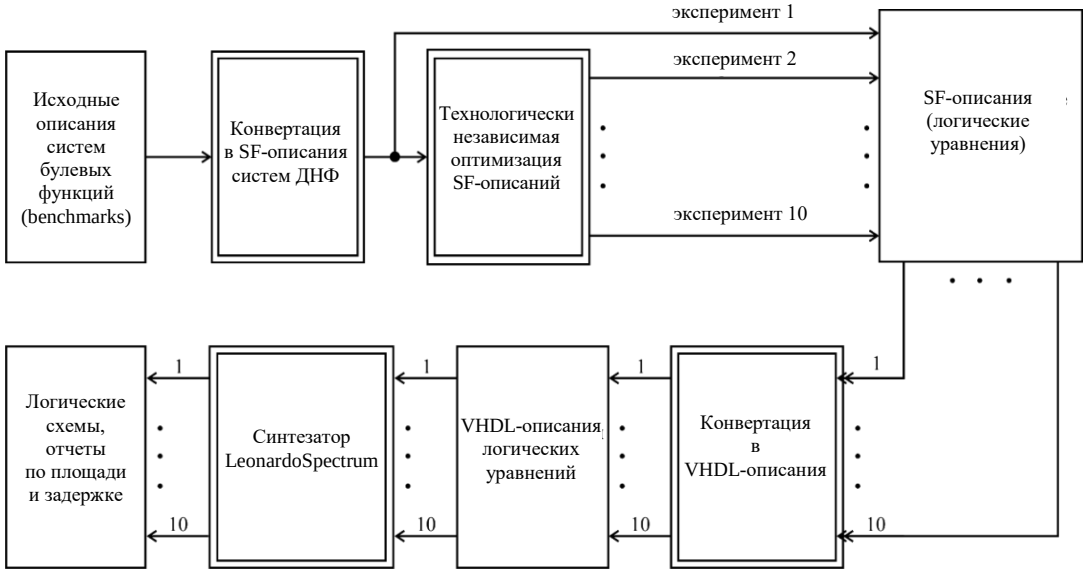


Рис. 6. Этапы экспериментов

Fig. 6. Stages of experiments

BDDI-минимизация выполнялась для матричных форм, Bool-минимизация – для логических уравнений, задающих те же системы функций. Программы блочного покрытия выполнялись только для матричных представлений, обработка многоблочных логических сетей осуществлялась с помощью стратегий системы логической оптимизации FLC2 [26]. Перечислим программы системы FLC2, участвующие в экспериментах. BDDI-минимизация отдельных ДНФ и совместная BDDI-минимизация систем ДНФ выполнялись с помощью программы BDD_Builder [18]; раздельная и совместная Bool-минимизация – с помощью модификации программы BoolNet_Opt [18]. Для решения задачи 1 блочного покрытия систем ДНФ использовалась программа RAZ [25], для решения задачи 2 – программа RAZ_Area [25]. В реализованном варианте программ RAZ, RAZ_Area блочного покрытия предполагается, что каждый единичный элемент матрицы B^f покрывается только одной из подматриц B_H. Улучшение результатов многоуровневой минимизации функций блока (T_{H_i}, B_{H_i}) возможно при реализации многократного покрытия единичного значения в матрице B^f несколькими подматрицами B_{H_i} с целью уменьшения числа различных ДНФ в блоке (T_{H_i}, B_{H_i}). Как уже говорилось, совместная минимизация систем булевых функций в классе ДНФ выполнялась программой Espresso IIC [3].

Эксперимент 1. Логическая оптимизация не выполнялась, исходные матричные описания систем ДНФ переводились в логические уравнения и сразу конвертировались в VHDL-описания. Полученные в эксперименте 1 схемные реализации систем ДНФ названы *базовыми*.

Эксперимент 2. Совместная минимизация систем функций в классе ДНФ с помощью программы Espresso IIC [3].

Эксперимент 3. Разбиение системы на подсистемы, состоящие из отдельных функций, затем для каждой из функций исходной системы, заданной в матричной форме, выполнение раздельной BDDI-минимизации.

Эксперимент 4. Разбиение системы на подсистемы, состоящие из отдельных функций, затем перевод в логические уравнения матричного описания каждой из функций, после этого выполнение для каждой из функций раздельной Bool-минимизации.

Эксперимент 5. Совместная BDDI-минимизация исходной системы ДНФ булевых функций.

Эксперимент 6. Перевод матричного описания системы ДНФ булевых функций в логические уравнения, затем для системы функций, заданных логическими уравнениями, выполнение совместной Bool-минимизации.

Эксперимент 7. Блочное покрытие матричной формы системы ДНФ по критерию минимальности числа (p, q)-ограниченных блоков, затем для функций каждого блока выполнение совместной BDDI-минимизации.

Эксперимент 8. Блочное покрытие матричной формы системы ДНФ по критерию минимальности числа (p, q)-ограниченных блоков, затем для каждого блока выполнение совместной Bool-минимизации.

Эксперимент 9. Блочное покрытие матричной формы системы ДНФ по критерию минимальности суммарной площади блоков, затем для функций каждого блока выполнение совместной BDDI-минимизации.

Эксперимент 10. Блочное покрытие матричной формы системы ДНФ по критерию минимальности суммарной площади блоков, затем для функций каждого блока выполнение совместной Bool-минимизации.

Результаты экспериментов и их обсуждение. Для системы ДНФ функций (см. табл. 1) результаты экспериментов для заказных СБИС даны в табл. 6. Обозначения, используемые в табл. 6, будут пояснены далее. Можно отметить, что результаты экспериментов 1 и 2 для данного примера полностью совпадают. Это связано с тем, что программа Espresso минимизации системы ДНФ (см. табл. 1) не позволяет изменить исходную систему ДНФ – функции являются неминимизируемыми в классе ДНФ.

Таблица 6
Результаты экспериментов для системы ДНФ (табл. 1)

Table 6
Experimental results for the DNF system (Table 1)

Номер эксперимента *Experiment number*	*Z*	*p, q*	*Area*	*Delay*	*r*
1	81	–	8 643	2,74	1
2	81	–	8 643	2,74	1
3	148	–	14 910	***2,53**	6
4	136	–	9 614	3,48	6
5	162	–	9 631	3,06	1
6	80	–	8 643	2,74	1
7	89	5, 3	8 828	2,75	4
8	87	5, 3	9 201	2,78	4
9	80	6, 0	***8 493**	2,64	3
10	81	6, 0	8 643	2,63	3

Результаты экспериментов для примеров из табл. 5 представлены в табл. 7–11, где символом * отмечены лучшие решения для испытываемого примера – меньшие значения параметров площади и временной задержки. При этом сравнение проводилось по всем 10 экспериментам.

Символом # помечены решения, улучшающие базовые решения, т. е. отмечены меньшие значения параметра площади либо задержки, чем соответствующие значения параметров из эксперимента 1.

Таблица 7
Результаты экспериментов 1 и 2

Table 7
Results of experiments 1 and 2

Пример *Example*	Эксперимент 1 (базовые решения) *Experiment 1 (basic solutions)*			Эксперимент 2 *Experiment 2*				
	Z	*Area*	*Delay*	*k*	k_{Espr}	*Z*	*Area*	*Delay*
C8	204	21 500	***2,20**	70	70	204	***#21 494**	2,36
DALU	1 404	106 249	10,79	194	194	1 404	**#47 865**	**#4,60**
LAL	529	26 343	3,80	117	117	529	27 889	**#3,54**
PM1	124	***10 764**	2,58	42	42	124	11 099	**#2,44**
SCT	253	20 406	3,20	64	64	253	**#19 976**	**#3,15**
TTT2	1 263	43 652	***3,85**	222	222	1 263	45 019	4,72
Alu4	7 875	487 848	9,94	1 028	***575***	5 493	**#327 736**	**#8,59**
Apex5	7 106	188 559	8,77	1 227	***1 088***	7 227	***#188 180**	8,80
I2c	7 112	280 601	9,12	1 357	***805***	5 816	***#254 906**	11,70
X1	2 148	67 546	3,99	324	***274***	1 974	***#67 256**	3,99
X3	5 045	203 659	6,75	915	915	5 045	208 151	**#6,47**
X4	2 649	***95 781**	5,38	371	371	2 649	96 400	5,39
Blocki1	3 174	275 607	7,43	355	***240***	3 177	**#243 260**	**#6,61**
Blocki2	648	71 402	4,62	90	***80***	645	**#70 989**	5,07
Pozd_1	9 572	714 586	8,66	754	***555***	9 381	***#615 926**	9,56
Pozd_2	13 543	768 522	9,51	605	***442***	13 497	***#653 089**	10,07
Улучшено базовых решений							11	7

Таблица 8
Результаты экспериментов 3 и 4

Table 8
Results of experiments 3 and 4

Пример *Example*	Эксперимент 3 *Experiment 3*			Эксперимент 4 *Experiment 4*		
	Z	*Area*	*Delay*	*Z*	*Area*	*Delay*
C8	213	22 314	***2,20**	235	21 943	***2,20**
DALU	1 243	**#67 033**	**#6,22**	992	**#52 285**	**#4,89**
LAL	575	26 829	**#3,73**	527	26 377	3,80
PM1	173	11 841	**#2,38**	135	11 082	**#2,38**
SCT	390	**#19 926**	**#3,18**	340	**#20 149**	**#3,18**
TTT2	724	45 683	4,41	764	**#42 631**	4,21
Alu4	3 469	373 492	11,83	3 680	**#378 815**	**#9,16**
Apex5	5 059	196 081	**#7,42**	6 257	277 678	8,86
I2c	4 982	277 320	9,65	7 112	292 398	***#7,24**
X1	1 739	71 720	4,46	1 300	73 835	5,08
X3	3 833	223 222	**#6,51**	3 408	213 625	**#6,66**
X4	3 082	133 959	5,99	2 895	124 373	5,72
Blocki1	6 001	603 358	7,89	4 570	267 243	**#6,80**
Blocki2	1 070	73 338	**#4,20**	1 142	72 730	4,83
Pozd_1	29 913	3 281 191	10,16	15 516	747 480	**#8,39**
Pozd_2	36 180	4 268 153	10,99	19 704	815 160	**#9,49**
Улучшено базовых решений		2	8	–	4	10

Таблица 9
Результаты экспериментов 5 и 6

Table 9
Results of experiments 5 and 6

Пример *Example*	Эксперимент 5 *Experiment 5*			Эксперимент 6 *Experiment 6*		
	Z	*Area*	*Delay*	*Z*	*Area*	*Delay*
C8	225	22 839	***2,20**	269	22 314	***2,20**
DALU	955	**#82 316**	**#7,17**	1 148	**#57 770**	**#6,57**
LAL	405	43 139	5,15	307	27 331	4,12
PM1	161	12 923	**#2,33**	126	***10 764**	2,58
SCT	253	27 744	4,88	289	20 507	***#2,80**
TTT2	506	49 154	5,72	860	44 657	4,58
Alu4	2 129	**#300 835**	**#9,80**	2792	***#252 060**	**#8,21**
Apex5	6 791	221 587	9,11	6999	**#196 583**	**#8,12**
I2c	5245	301 415	**#9,10**	5890	283 765	**#8,24**
X1	2 664	107 912	**#6,45**	2 405	83 527	**#4,70**
X3	4 591	211 633	5,73	4 096	225 895	7,46
X4	4 235	120 031	7,23	3 028	165 759	6,94
Blocki1	5 395	659 534	9,35	2 768	**#274 787**	7,53
Blocki2	1 171	92 472	4,71	742	***#70 542**	4,82
Pozd_1	27 544	3 508 196	11,41	1 417	726 031	**#8,65**
Pozd_2	29 502	3 925 803	11,39	7 104	798 944	***#9,11**
Улучшено базовых решений		2	6	–	6	9

Таблица 10
Результаты экспериментов 7 и 8

Table 10
Results of experiments 7 and 8

Пример *Example*	Эксперимент 7 *Experiment 7*					Эксперимент 8 *Experiment 8*				
	Z	*p,q*	*Area*	*Delay*	*r*	*Z*	*p,q*	*Area*	*Delay*	*r*
C8	262	12,7	21 935	***2,20**	5	279	12,7	22 114	2,98	5
DALU	1391	20,4	**#68 199**	**#4,79**	12	1 228	20,4	**#50 187**	***#4,37**	12
LAL	531	14,8	26 829	**#3,73**	4	515	14,8	***#25 925**	***#3,29**	4
PM1	189	8,4	12 131	**#2,38**	6	173	8,4	11 099	**#2,44**	6
SCT	428	10,5	20 674	**#3,18**	6	335	10,5	20 674	**#3,18**	6
TTT2	958	15,5	64 482	5,11	7	845	15,5	48 457	5,98	7
Alu4	3 327	14,4	**#327 825**	***#7,68**	2	2 784	14,4	**#261 038**	**#8,10**	2
Apex5	6 711	25,20	**#246 005**	**#7,87**	12	6234	25,20	292 900	9,52	12
I2c	4861	24,20	298 497	**#9,09**	20	7 283	24,20	295 400	**#8,71**	20
X1	1 709	25,10	81 652	4,66	7	1 468	25,10	69 867	***#3,93**	7
X3	4 460	25,10	**#198 760**	***#4,98**	14	4 265	25,10	**#201 940**	**#6,43**	14
X4	3 123	15,10	139 238	7,23	21	2 412	15,10	105 473	***#4,31**	21
Blocki1	5 428	10,8	641 739	9,22	10	2 798	10,8	**#274 212**	***#6,58**	10
Blocki2	1 162	10,8	98 264	5,12	2	748	10,8	71 870	***#4,05**	2
Pozd_1	25 322	12,10	3 204 304	11,18	3	7 074	12,10	727 537	8,72	3
Pozd_2	27 065	12,10	3 664 347	10,42	3	7 200	12,10	803 180	**#9,26**	3
Улучшено базовых решений			4	9	–	–	–	5	12	–

Таблица 11
Результаты экспериментов 9 и 10

Table 11
Results of experiments 9 and 10

Пример *Example*	Эксперимент 9 *Experiment 9*					Эксперимент 10 *Experiment 10*				
	Z	*p*	*Area*	*Delay*	*r*	*Z*	*p*	*Area*	*Delay*	*r*
C8	266	12	22 169	***2,20**	5	255	12	22 314	***2,20**	21
DALU	1 305	20	**#68 305**	**#5,33**	12	1 030	20	***#48 044**	**#4,63**	34
LAL	588	14	27 710	**#3,25**	4	523	14	26 829	**#3,73**	27
PM1	198	8	11 127	***#2,20**	6	168	8	11 099	**#2,44**	20
SCT	397	10	***#19 301**	**#2,93**	6	356	10	**#20 116**	**#2,95**	25
TTT2	1 106	15	57 189	4,64	7	1 056	15	***41 415**	5,11	40
Alu4	3 469	14	**#373 492**	11,83	7	3 512	14	**#315 415**	**#7,70**	7
Apex5	6 126	40	195 903	***#6,94**	117	5 949	50	242 981	**#6,98**	107
I2c	6 807	40	**#278 258**	**#6,56**	183	4 859	40	**#259 565**	**#7,63**	88
X1	1 709	25	81 652	4,66	7	1 492	25	73 087	4,71	7
X3	4 238	25	229 427	**#6,74**	135	3 766	25	***#198 252**	**#6,67**	135
X4	3 402	15	123 720	5,75	189	3 030	15	103 799	**#4,75**	189
Blocki1	5 418	10	341 451	**#7,10**	10	4 558	10	***#265 290**	**#6,92**	16
Blocki2	1 070	10	73 338	**#4,20**	2	1 138	10	**#70 609**	**#4,22**	16
Pozd_1	25 975	12	3 233 861	11,15	3	7 892	12	731 611	***#7,94**	6
Pozd_2	27 065	12	3 665 887	11,43	3	7 178	12	803 180	**#9,26**	3
Улучшено базовых решений			4	10	–	–	–	8	14	–

В табл. 6–11 используются следующие обозначения:

Z – число литералов в задании системы булевых функций;

k_{Espr} – число элементарных конъюнкций в совместно минимизированной системе ДНФ с помощью программы Espresso;

Area – суммарная площадь элементов схемы в условных единицах;

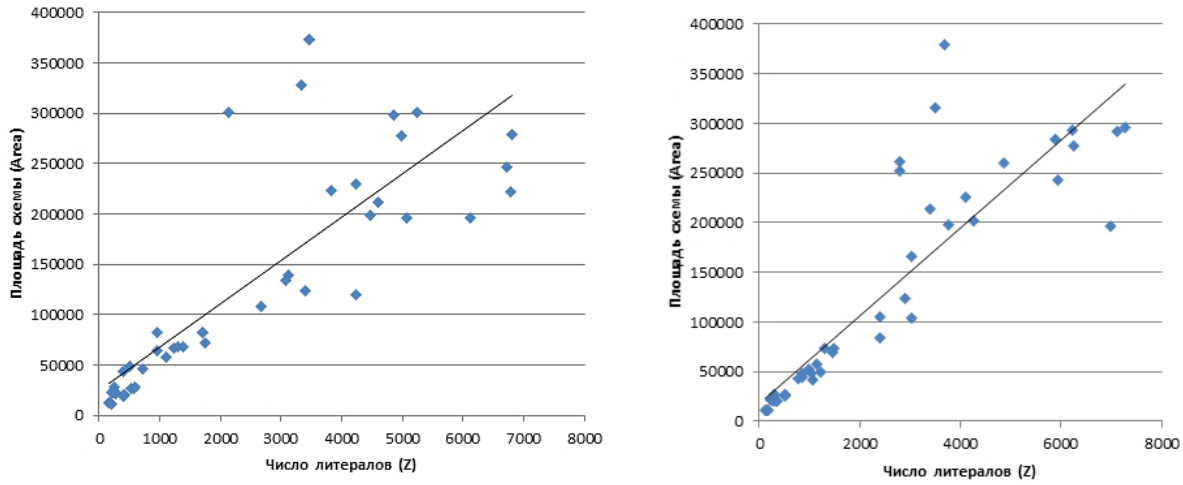
Delay – временная задержка схемы, нс;

p – число входов блока;

q – число выходов блока;

r – число блоков после разбиения системы функций на подсистемы (случай *r*=1 соответствует совместной реализации системы).

Результаты экспериментов позволяют сделать следующие выводы. Для исследованного множества блочных и разреженных систем ДНФ булевых функций значительное преимущество по площади синтезированных многовыходных комбинационных логических схем из библиотечных элементов имеет Bool-минимизация, которая выполняется для систем функций, заданных логическими уравнениями. Синтез схем по матричным представлениям систем функций и последующая BDDI-минимизация являются менее эффективными: для такого вывода достаточно сравнить эксперименты в парах (3, 4), (5, 6), (7, 8), (9, 10). В экспериментах с нечетными номерами используется BDDI-минимизация, в экспериментах с четными номерами – Bool-минимизация. Графики зависимостей площадей схем от числа литералов в BDDI- и Bool-представлениях систем функций показаны на рис. 7. При этом исключены четыре специально сгенерированных примера. График на рис. 7, *a* построен по результатам нечетных (3, 5, 7, 9) экспериментов, график на рис. 7, *b* – по результатам четных (4, 6, 8, 10) экспериментов. Данные графики (тренды) показывают достаточно хорошую линейную зависимость площади схемы от числа литералов в многоуровневом функциональном описании реализуемой системы функций. Поэтому целесообразно в дальнейших исследованиях организовывать переборы блочных покрытий, оценивания их не только по площади, но и по суммарному числу литералов в блоках покрытия. Для разреженных систем ДНФ такой подход может быть более перспективным, чем подход, исследованный в работе [31] и основанный на выделении из формульного задания системы функций таких подсистем, для которых совместная минимизация является более предпочтительной, чем минимизация функций по отдельности либо совместная минимизация всей системы в целом.



a) *b)*

Рис. 7. Зависимость площади схемы (Area) от числа литералов (Z) при BDDI-минимизации (*a*) и Bool-минимизации (*b*)

Fig. 7. Dependence of the area of the scheme (Area) on the number of literals (Z) with BDDI minimization (a) and Bool minimization (b)

Установлено также то, что для разреженных систем ДНФ сравнение кофакторов целесообразно выполнять для их задания матричными формами, так как переход от матричных форм к полиномам Жегалкина [30] становится трудоемким и время BDDI-минимизации возрастает. Сравнение кофакторов в матричном виде позволяет ускорять вычисления.

Алгоритмы и программы [25] блочного покрытия для матричных представлений позволяют выделять блоки в блочных системах ДНФ функций и в разреженных системах ДНФ. Это подтверждается нахождением блочных покрытий для специально сгенерированных примеров Blocki1, Blocki2, Pozd_1, Pozd_2 блочных систем ДНФ. Минимизация блочных покрытий по критерию минимальности числа блоков уступает по результатам синтеза минимизации по критерию (4) минимальности общей площади блоков. Оптимизация блочных покрытий матричных представлений систем ДНФ функций по критерию площади и последующая Bool-минимизация полученных блоков являются достаточно эффективными, так как в большом числе случаев позволяют уменьшить площади схем либо их временные задержки.

Заключение. Система FLC-2 [26] включает разнообразные программы логической минимизации различных форм представлений систем булевых функций. В ней можно провести эффективную технологически независимую оптимизацию разреженных систем ДНФ, используя программы блочного покрытия с последующей минимизацией Bool- либо BDDI-представлений полученных блоков. Это не исключает применения и других программ логической минимизации, особенно в тех случаях, когда система ДНФ не является разреженной. Такая система ДНФ задается матрицей конъюнкций T^x, характеризующейся небольшим процентом неопределенных элементов, а матрица B^f характеризуется небольшим процентом нулевых значений. В этих случаях могут быть эффективными комбинированные методы многоэтапной совместной минимизации многоуровневых представлений [32] либо подходы, основанные на выделении из системы функций таких подсистем, для которых преимущество при синтезе имеет совместная минимизация [31]. После блочного покрытия для логической минимизации могут быть использованы не только представления проектных данных в виде BDDI и Bool, но и другие структуры данных.

Вклад авторов. *С. Н. Кардаш* разработал программные средства для нахождения блочных покрытий систем ДНФ и выбрал лучшую программу блочного покрытия, *П. Н. Бибило* выполнил эксперименты, подготовил текст статьи.

Список использованных источников

1. Тарасов, И. Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования / И. Е. Тарасов. – М. : Горячая линия – Телеком, 2020. – 538 с.

2. Закревский, А. Д. Логический синтез каскадных схем / А. Д. Закревский. – М. : Наука, 1981. – 416 с.

3. Logic Minimization Algorithm for VLSI Synthesis / K. R. Brayton [et al.]. – Boston : Kluwer Academic Publishers, 1984. – 193 p.

4. Синтез асинхронных автоматов на ЭВМ / под ред. А. Д. Закревского. – Минск : Наука и техника, 1975. – 184 с.

5. Brayton, R. K. The decomposition and factorization of Boolean expressions / R. K. Brayton, C. T. McMullen // Proc. of IEEE Intern. Symp. on Circuits and Systems (ISCAS 1982), Rome, Italy, 10–12 May 1982. – Rome, 1982. – P. 49–54.

6. 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. CAD-6, no. 6. – P. 1062–1081.

7. Scholl, C. Functional Decomposition with Applications to FPGA Synthesis / C. Scholl. – Boston : Kluwer Academic Publishers, 2001. – 288 p.

8. Поттосин, Ю. В. Табличные методы декомпозиции систем полностью определенных булевых функций / Ю. В. Поттосин, Е. А. Шестаков. – Минск : Беларус. навука, 2006. – 327 с.

9. Sasao, T. Memory-Based Logic Synthesis / T. Sasao. – N. Y. : Springer, 2011. – 189 p.

10. Бибило, П. Н. Декомпозиция булевых функций на основе решения логических уравнений / П. Н. Бибило. – Минск : Беларус. навука, 2009. – 211 с.

11. Bryant, R. E. Graph-based algorithms for Boolean function manipulation / R. E. Bryant // IEEE Transactions on Computers. – 1986. – Vol. 35, no. 8. – P. 677–691.

12. Drechsler, R. Binary Decision Diagrams: Theory and Implementation / R. Drechsler, B. Becker. – Springer, 1998. – 210 p.

13. Ebendt, R. Advanced BDD Optimization / R. Ebendt, G. Fey, R. Drechsler. – Springer, 2005. – 222 p.

14. Bryant, R. E. Ordered binary decision diagrams / R. E. Bryant, C. Meinel // Logic Synthesis and Verification / eds.: S. Hassoun, T. Sasao, R. K. Brayton. – Kluwer Academic Publishers, 2002. – P. 285–307.

15. Meinel, C. Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications / C. Meinel, T. Theobald. – Berlin, Heidelberg : Springer-Verlag, 1998. – 267 p.

16. Кнут, Д. Э. Искусство программирования. Т. 4, А. Комбинаторные алгоритмы. Ч. 1 : пер. с англ. / Д. Э. Кнут. – М. : Вильямс, 2013. – 960 с.

17. Бибило, П. Н. Применение диаграмм двоичного выбора при синтезе логических схем / П. Н. Бибило. – Минск : Беларус. навука, 2014. – 231 с.

18. 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.

19. Бибило, П. Н. Логическая минимизация булевых сетей с использованием разложения Шеннона / П. Н. Бибило, Ю. Ю. Ланкевич // Информатика. – 2019. – Т. 16, № 2. – С. 73–89.

20. A novel basis for logic rewriting / W. Haaswijk [et al.] // Proc. of 22nd Asia and South Pacific Design Automation Conf. (ASP-DAC), Chiba, Japan, 16–19 Jan. 2017. – Chiba, 2017. – P. 151–156. https://doi.org/10.1109/ASPDAC.2017.7858312

21. Optimizing majority-inverter graphs with functional hashing / M. Soeken [et al.] // Proc. of the 2016 Design, Automation & Test in Europe Conf. & Exhibition (DATE), Dresden, Germany, 14–18 March 2016. – Dresden, 2016. – P. 1030–1035.

22. Exact synthesis of majority-inverter graphs and its applications / M. Soeken [et al.] // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2017. – Vol. 36, no. 11. – P. 1842–1855.

23. Size optimization of MIGs with an application to QCA and STMG technologies / H. Riener [et al.] // Proc. of the 14th IEEE/ACM Intern. Symp. on Nanoscale Architectures, Athens, Greece, 17–19 July 2018. – Athens, 2018. – P. 157–162.

24. Harlecek, I. Are XORs in logic synthesis really necessary? / I. Harlecek, P. Fiser, J. Schmidt // IEEE 20th Intern. Symp. on Design and Diagnostics of Electronic Circuits & Systems (DDECS), Dresden, Germany, 19–21 Apr. 2017. – Dresden, 2017. – P. 134–139.

25. Кардаш, С. Н. Построение блочных разбиений систем булевых функций на основе задачи покрытия булевых матриц / С. Н. Кардаш // BIG DATA и анализ высокого уровня = BIG DATA and Advanced Analytics : сб. науч. ст. IX Междунар. науч.-практ. конф., Минск, 17–18 мая 2023 г. : в 2 ч. – Минск : БГУИР, 2023. – Ч. 2. – С. 326–330.

26. 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.

27. Бибило, П. Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum / П. Н. Бибило. – М. : СОЛОН-Пресс, 2005. – 384 с.

28. Авдеев, Н. А. Автоматизированное проектирование цифровых операционных устройств с пониженным энергопотреблением / Н. А. Авдеев, П. Н. Бибило // Программная инженерия. – 2021. – Т. 12, № 2. – С. 63–73.

29. Соловьев, В. В. Архитектуры ПЛИС фирмы Xilinx: FPGA и CPLD 7-й серии / В. В. Соловьев. – М. : Горячая линия – Телеком, 2016. – 392 с.

30. 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.

31. 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. 197–206.

32. Бибило, П. Н. Логическая минимизация многоуровневых представлений систем булевых функций / П. Н. Бибило, Ю. Ю. Ланкевич, В. И. Романов // Информационные технологии. – 2023. – Т. 29, № 2. – С. 59–71.

References

1. 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 Russ.).

2. Zakrevskij A. D. Logicheskij sintez kaskadnyh skhem. *Logical Synthesis of Cascading Circuit.* Moscow, Nauka, 1981, 416 p. (In Russ.).

3. 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.

4. Zakrevskij A. D. (ed.). Sintez asinhronnyh avtomatov na EHVM. *Synthesis of Asynchronous Automata on a Computer*. Minsk, Nauka i tekhnika, 1975, 184 p. (In Russ.).

5. Brayton R. K., McMullen C. T. The decomposition and factorization of Boolean expressions. *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 1982), Rome, Italy, 10–12 May 1982*. Rome, 1982, pp. 49–54.

6. 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.

7. Scholl C. *Functional Decomposition with Application to FPGA Synthesis*. Boston, Kluwer Academic Publishers, 2001, 288 p.

8. Pottosin Yu. V., Shestakov E. A. Tablichnye metody dekompozicii sistem polnost'yu opredelennyh bulevyh funkcij. *Tabular Methods for Decomposition of Systems of Completely Defined Boolean Functions*. Minsk, Belaruskaja navuka, 2006, 327 p. (In Russ.).

9. Sasao T. *Memory-Based Logic Synthesis*. New York, Springer, 2011, 189 p.

10. Bibilo P. N. Dekompoziciya bulevyh funkcij na osnove resheniya logicheskih uravnenij. *Decomposition of Boolean Functions Based on the Solution of Logical Equations*. Minsk, Belaruskaja navuka, 2009, 211 p. (In Russ.).

11. Bryant R. E. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, 1986, vol. 35, no. 8, pp. 677–691.

12. Drechsler R., Becker B. *Binary Decision Diagrams: Theory and Implementation*. Springer, 1998, 210 p.

13. Ebendt R., Fey G., Drechsler R. *Advanced BDD Optimization.* Springer, 2005, 222 p.

14. Bryant R. E., Meinel C. Ordered binary decision diagrams. In S. Hassoun, T. Sasao, R. K. Brayton (eds.). *Logic Synthesis and Verification.* Kluwer Academic Publishers, 2002, pp. 285–307.

15. Meinel C., Theobald T. *Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications.* Berlin, Heidelberg, Springer-Verlag, 1998, 267 p.

16. Knuth D. E. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Addison-Wesley Professional, 2011, 912 p.

17. Bibilo P. N. Primenenie diagram dvoichnogo vybora pri sinteze logicheskih shem. *Application of Binary Selection Diagrams in the Synthesis of Logic Circuits*. Minsk, Belaruskaja navuka, 2014, 231 p. (In Russ.).

18. 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.).

19. Bibilo P. N., Lankevich Yu. Yu. *Logical optimization of Boolean nets using Shannon expansion*. Informatika [*Informatics*], 2019, vol. 16, no. 2, pp. 73–89 (In Russ.).

20. Haaswijk W., Soeken M., Amaru L., Gaillardon P.-E., De Micheli G. A novel basis for logic rewriting. *Proceedings of 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), Chiba, Japan, 16–19 January 2017*. Chiba, 2017, pp. 151–156. https://doi.org/10.1109/ASPDAC.2017.7858312

21. 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.

22. 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.

23. Riener H., Testa E., Amaru L., Soeken M., De Micheli G. Size optimization of MIGs with an application to QCA and STMG technologies. *Proceedings of the 14th IEEE/ACM International Symposium on Nanoscale Architectures, Athens, Greece, 17–19 July 2018*. Athens, 2018, pp. 157–162.

24. Harlecek I, Fiser P., Schmidt J. Are XORs in logic synthesis really necessary? *IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), Dresden, Germany, 19–21 April 2017*. Dresden, 2017, pp. 134–139.

25. Kardash S. N. *Construction of block partitions of Boolean function systems based on the problem of covering Boolean matrices*. BIG DATA i analiz vysokogo urovnja : sbornik nauchnyh statej IX Mezhdunarodnoj nauchno-prakticheskoj konferencii, Minsk, 17–18 maja 2023 g. : v 2 chastjah. Chast' 2 [*BIG DATA and Advanced Analytics : Collection of Scientific Articles of the IX International Scientific and Practical Conference, Minsk, 17–18 May 2023 : in 2 Parts*]. Minsk, Belorusskij gosudarstvennyj universitet informatiki i radiojelektroniki, 2023, part 2, pp. 326–330 (In Russ.).

26. Bibilo P. N., Romanov V. I. *The system of logical optimization of functional structural descriptions of digital circuits based on production-frame knowledge representation model*. Problemy razrabotki perspektivnyh mikro- i nanoelektronnyh system [*Problems of Developing Promising Micro- and Nanoelectronic Systems*], 2020, iss. 4, pp. 9–16 (In Russ.).

27. 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.).

28. Avdeev N. A., Bibilo P. N. *Design of digital operational units with low power consumption*. Programmnaya inzheneriya [*Software Engineering*], 2021, vol. 12, no. 2, pp. 63–73 (In Russ.).

29. 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 Russ.).

30. Bibilo P. N., Lankevich Yu. Yu. *The use of Zhegalkin polynomials for minimization of multilevel represintations of Boolean functions based on Shannon expansion*. Programmnaya inzheneriya [*Software Engineering*], 2017, no. 8, pp. 369–384 (In Russ.).

31. Bibilo P. N., Kirienko N. A., Romanov V. I. *Extraction from a multilevel representation of a system of Boolean functions of subsystems for joint logical minimization*. Programmnye produkty i sistemy [*Software & Systems*], 2023, vol. 36, no. 4, pp. 197–206 (In Russ.).

32. Bibilo P. N., Lankevich Yu. Yu., Romanov V. I. *Logical minimization of multilevel representations of Boolean function systems*. Informacionnye tekhnologii [*Information Technology*], 2023, vol. 29, no. 2, pp. 59–71 (In Russ.).

Информация об авторах

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

Кардаш Сергей Николаевич, кандидат технических наук, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: kardash77@gmail.com

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

Sergey N. Kardash, Ph. D. (Eng.), The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: kardash77@gmail.com