

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

LOGICAL DESIGN



УДК 519.714
<https://doi.org/10.37661/1816-0301-2021-18-2-7-32>

Оригинальная статья
Original Paper

Минимизация многоуровневых представлений систем полностью определенных булевых функций с использованием разложений Шеннона и алгебраических представлений кофакторов

П. Н. Бибило[✉], 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. В системах проектирования цифровых СБИС (сверхбольших интегральных схем) графовый аппарат BDD (Binary Decision Diagram – бинарная диаграмма решений) применяется при верификации СБИС, а также при технологически независимой оптимизации, выполняемой как первый этап синтеза логических схем в различных технологических базах. BDD представляет собой ациклический граф, задающий булеву функцию либо систему булевых функций. Каждой вершине этого графа соответствует полная или редуцированная формула разложения Шеннона. После получения BDD-представлений систем булевых функций предлагается выполнять дополнительные логические оптимизации на основе описываемого в статье метода поиска алгебраических представлений кофакторов (подфункций разложения Шеннона) одного уровня BDD в виде дизъюнкции, конъюнкции либо суммы по модулю два подфункций того же уровня либо нижних уровней BDD. Ориентированный граф BDD для системы функций строится на основе разложений Шеннона всех компонентных функций системы по одной и той же переменной. Метод позволяет уменьшать число литералов путем замены формул разложений Шеннона более простыми логическими формулами и сокращать число литералов в описании системы булевых функций. Число литералов в алгебраических многоуровневых представлениях систем полностью определенных булевых функций является основным критерием логической оптимизации при синтезе комбинационных схем из библиотечных логических элементов.

Ключевые слова: система булевых функций, дизъюнктивная нормальная форма, Binary Decision Diagram, разложение Шеннона, синтез логической схемы, СБИС

Для цитирования. Бибило, П. Н. Минимизация многоуровневых представлений систем полностью определенных булевых функций с использованием разложений Шеннона и алгебраических представлений кофакторов / П. Н. Бибило, В. И. Романов // Информатика. – 2021. – Т. 18, № 2. – С. 7–32. <https://doi.org/10.37661/1816-0301-2021-18-2-7-32>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 25.01.2021
Подписана в печать | Accepted 21.04.2021
Опубликована | Published 26.06.2021

Minimization of binary decision diagrams for systems of completely defined Boolean functions using Shannon expansions and algebraic representations of cofactors

Petr N. Bibilo[✉], 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. In the systems of digital VLSI design (Very Large Integrated Circuits), the BDD (Binary Decision Diagram) is used for VLSI verification, as well as for technologically independent optimization as the first stage in the synthesis of logic circuits in various technological bases. The BDD is an acyclic graph defining a Boolean function or a system of Boolean functions. Each vertex of this graph corresponds to the complete or reduced Shannon expansion formula. When BDD representation for systems of Boolean functions is constructed, it is possible to perform additional logical optimization based on the proposed method of searching for algebraic representations of cofactors (subfunctions) of the same BDD level in the form of a disjunction, conjunction either exclusive-or of cofactors of the same level or lower levels of BDD. A directed BDD graph for a system of functions is constructed on the basis of Shannon expansion of all component functions of the system by the same permutation of variables. The method allows to reduce the number of literals by replacing the Shannon expansion formulas with simpler formulas that are disjunctions or conjunctions of cofactors, and to reduce the number of literals in specifying a system of Boolean functions. The number of literals in algebraic multilevel representations of systems of fully defined Boolean functions is the main optimization criterion in the synthesis of combinational circuits from librarian logic elements.

Keywords: system of Boolean functions, disjunctive normal form, Binary Decision Diagram, Shannon expansion, digital logic synthesis, VLSI

For citation. Bibilo P. N., Romanov V. I. Minimization of binary decision diagrams for systems of completely defined Boolean functions using Shannon expansions and algebraic representations of cofactors. *Informatika*, 2021, vol. 18, no. 2, pp. 7–32 (In Russ.). <https://doi.org/10.37661/1816-0301-2021-18-2-7-32>

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

Введение. Минимизация различных форм представления систем булевых функций выполняется в качестве начального этапа синтеза многовыходных комбинационных схем. Оптимизация двухуровневых (И-ИЛИ) представлений систем булевых функций сводится к различным способам совместной либо раздельной минимизации функций в классе ДНФ (дизъюнктивных нормальных форм) [1] и эффективно применяется при синтезе схем программируемых логических матриц [2]. Минимизация многоуровневых представлений обычно выполняется как этап технологически независимой оптимизации при синтезе схем из библиотечных логических элементов [3] и структур FPGA (Field-Programmable Gate Array – программируемая пользователем вентильная матрица). При синтезе логических схем в составе FPGA также используются библиотеки параметризованных логических элементов, состоящие из сотен логических ячеек с различным числом входов, триггеров, мультиплексоров и универсальных логических элементов, настраиваемых на реализацию булевых функций от ограниченного числа $k = 4, 6$ переменных [4]. Развитие методов многоуровневой оптимизации началось с алгебраической факторизации – выделения общих частей в алгебраических представлениях функций [5–9]. Такие представления часто задавались в виде булевых сетей. В булевых сетях (графах) используются функциональные описания вершин (узлов), узлы при покрытии (технологическом отображении – втором этапе синтеза логической схемы) объединяются в подсхемы, а каждая подсхема реализуется одним библиотечным логическим элементом. Методы факторизации были реализованы в промышленных программах [5, 7] автоматизации проектирования логических схем в различных библиотеках. Нахождение многоуровневых представлений осуществляется на основе ре-

шения различных задач декомпозиции (разложения) систем функций [10–14]. Среди многочисленных видов функциональных разложений и соответствующих им форм представлений систем функций, пожалуй, наиболее эффективными для технологически независимой оптимизации оказались многоуровневые представления систем функций на основе разложения Шеннона, называемые в литературе BDD-представлениями [15–21]. Широкое применение BDD – ориентированных графов, задающих формулы разложений Шеннона, основывается на компактности графов и эффективности вычислительных процедур над такими графами [20]. Эффективность вычислений обусловлена тем обстоятельством, что BDD-представления булевых функций соответствуют ортогонализированным формам задания и исходных функций, и всех подфункций (кофакторов) разложений Шеннона.

В настоящей статье предлагается использовать алгебраические представления кофакторов BDD в виде дизъюнкции, конъюнкции либо исключающего ИЛИ (суммы по модулю два) других кофакторов той же BDD. При этом применяется инверсирование кофакторов и учитывается сложность самих кофакторов, участвующих в алгебраических представлениях. Предложенный в статье метод основывается на вычислениях и сравнениях функций кофакторов и позволяет улучшать результаты логической минимизации по числу литералов булевых переменных – основному критерию технологически независимой оптимизации [3] при синтезе комбинационных схем из библиотечных логических элементов. Уменьшение сложности многоуровневых представлений систем функций приводит к меньшим по площади кристаллам заказных цифровых СБИС и цифровых логических блоков, размещаемых в составе FPGA.

Основные определения. Булевыми называются двоичные $(0, 1)$ функции $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ двоичных (булевых) переменных x_1, x_2, \dots, x_n , литералом булевой переменной – переменная x_i либо ее инверсия \bar{x}_i . Пусть $V^{\mathbf{x}}$ – булево пространство, построенное над переменными булева вектора $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Элементами этого пространства являются n -компонентные наборы (векторы) \mathbf{x}^* нулей и единиц. Булева функция, значения 0, 1 которой определены на всех элементах $\mathbf{x}^* \in V^{\mathbf{x}}$, называется *полностью определенной булевой функцией*. Полностью определенная булева функция (далее – булева функция) принимает единичное значение на элементах \mathbf{x}^* подмножества M_f^1 булева пространства $V^{\mathbf{x}}$ и нулевое значение на элементах подмножества M_f^0 . Очевидно, что $M_f^1 \cap M_f^0 = \emptyset$, $V^{\mathbf{x}} = M_f^0 \cup M_f^1$. Булевы функции $f_1(\mathbf{x})$ и $f_2(\mathbf{x})$ равны, если и только если $M_{f_1}^1 = M_{f_2}^1$, $M_{f_1}^0 = M_{f_2}^0$. Будем называть булевы функции $f_1(\mathbf{x})$ и $f_2(\mathbf{x})$ *взаимно инверсными* ($f_2 = \bar{f}_1$, $f_1 = \bar{f}_2$), если и только если $M_{f_1}^1 = M_{f_2}^0$, $M_{f_1}^0 = M_{f_2}^1$. Будем говорить, что рассматривается *инверсная форма* булевой функции f , если рассматривается инверсия \bar{f} этой функции, и *прямая форма* функции f , если речь идет о f . Под *векторной булевой функцией* $\mathbf{f}(\mathbf{x})$ будем понимать упорядоченную систему булевых функций $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, значениями векторных функций на элементах \mathbf{x}^* булева пространства являются m -компонентные двоичные векторы $\mathbf{f}(\mathbf{x}^*)$.

Разложением Шеннона полностью определенной булевой функции $f = f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, по переменной x_i называется представление

$$f = f(\mathbf{x}) = \bar{x}_i f_0(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \vee x_i f_1(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n). \quad (1)$$

Функции $f_0 = f_0(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f_1 = f_1(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ называются *кофакторами* (cofactors) разложения по переменной x_i либо *подфункциями*. Они получаются из функции $f = f(x_1, \dots, x_n)$ подстановкой вместо переменной x_i констант 0 и 1. Каждая из под-

функций f_0 и f_1 может быть разложена по одной из переменных из множества $x_1, \dots, \{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\}$. Процесс разложения подфункций заканчивается, когда все n переменных будут использованы для разложения либо когда все подфункции вырождаются до констант 0, 1. На каждом шаге разложения выполняется сравнение на равенство полученных подфункций и оставляется одна из нескольких попарно равных подфункций.

Под *BDD-представлением* полностью определенной векторной булевой функции $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ понимается ациклический ориентированный граф, задающий последовательные разложения Шеннона всех компонентных функций $f_i(\mathbf{x})$, $i = 1, \dots, m$, по всем ее переменным x_1, x_2, \dots, x_n при одном и том же заданном порядке (перестановке) переменных, по которым проводятся разложения [21]. Для упрощения графа листовые вершины 0, 1 обычно дублируются, а ориентация дуг на изображениях графов BDD не показывается, так как всегда принимается, что дуги ориентированы сверху вниз.

Наиболее близкими к графам BDD для векторных функций являются широко известные в литературе сокращенные упорядоченные BDD (англ. Reduced Ordered BDD, ROBDD) для одной полностью определенной булевой функции, в которых каждой функциональной вершине соответствует одна функция (подфункция разложения Шеннона), при этом функциональные вершины лишь подразумеваются (отождествляются с вершинами-переменными). Подробное описание OBDD (упорядоченных BDD) дано в работах [18, 19], ROBDD – в [20]. Далее под BDD будут пониматься ROBDD для систем функций (векторных функций) [21, с. 40]. BDD векторных функций иногда называются в литературе совместными BDD. Будем изображать на рисунках графы BDD, которые содержат функциональные вершины, соответствующие разлагаемым функциям, вершины-переменные и листовые вершины, соответствующие константам 0, 1 [21, с. 16]. BDD-представлению соответствуют формулы разложения Шеннона, каждой функциональной вершине – своя полная либо сокращенная формула разложения Шеннона. Функциональные вершины обычно показываются на рисунках для удобства изложения методов минимизации BDD и иллюстрации зависимостей переменных в формулах разложений Шеннона. Рисунки классических BDD не содержат функциональных вершин. BDD нашли очень широкое применение не только при проектировании логических схем, но и в информатике в целом [20].

Пример BDD-представления векторной булевой функции. Рассмотрим векторную булеву функцию, задаваемую системой ДНФ (табл. 1). Эта функция может быть представлена BDD (рис. 1), построенной по перестановке переменных $\langle x_1, x_2, x_5, x_3, x_4 \rangle$. Функциональным вершинам такой BDD соответствуют следующие формулы разложения Шеннона:

$$\begin{aligned} f_1 &= \bar{x}_1 h_1 \vee x_1 h_2; \quad f_2 = \bar{x}_1 h_2 \vee x_1 h_3; \quad f_3 = \bar{x}_1 h_4 \vee x_1 h_5; \quad f_4 = \bar{x}_1 h_4 \vee x_1 h_6; \quad f_5 = \bar{x}_1 h_6 \vee x_1 h_7; \\ h_1 &= \bar{x}_2 t_1 \vee x_2 t_3; \quad h_2 = \bar{x}_2 t_2 \vee x_2 t_8; \quad h_3 = \bar{x}_2 t_5 \vee x_2 t_6; \quad h_4 = \bar{x}_2 t_6 \vee x_2 t_7; \quad h_5 = \bar{x}_2 t_6 \vee x_2 t_{10}; \\ h_6 &= \bar{x}_2 t_4 \vee x_2 t_9; \quad h_7 = \bar{x}_2 t_{11} \vee x_2 t_{10}; \quad t_1 = \bar{x}_5 t_{15} \vee x_5 t_{16}; \quad t_2 = \bar{x}_5 \bar{x}_4 \vee x_5 t_{12}; \quad t_3 = \bar{x}_5 t_{13} \vee x_5 t_{14}; \\ t_4 &= \bar{x}_5 t_{12} \vee x_5 \bar{x}_4; \quad t_5 = \bar{x}_5 t_{13} \vee x_5 \bar{x}_4; \quad t_6 = \bar{x}_5 \bar{x}_3 \vee x_5 t_{14}; \quad t_7 = \bar{x}_5 x_4 \vee x_5 \bar{x}_4; \quad t_8 = \bar{x}_5 \bar{x}_4 \vee x_5 \bar{x}_3; \\ t_9 &= \bar{x}_5 t_{13}; \quad t_{10} = \bar{x}_5 t_{17} \vee x_5 t_{15}; \quad t_{11} = \bar{x}_5 x_4 \vee x_5 x_3; \quad t_{12} = \bar{x}_3 x_4; \quad t_{13} = \bar{x}_3 \vee x_3 x_4; \\ t_{14} &= \bar{x}_3 \bar{x}_4 \vee x_3; \quad t_{15} = \bar{x}_3 x_4 \vee x_3; \quad t_{16} = \bar{x}_3 \vee x_3 \bar{x}_4; \quad t_{17} = x_3 \bar{x}_4. \end{aligned} \quad (2)$$

Корневыми вершинами BDD (рис. 1) являются f_1, \dots, f_5 , листовыми – 0, 1. На первом уровне BDD располагаются два неконстантных кофактора x_4, \bar{x}_4 и два кофактора, равные константам 0, 1; на втором уровне – кофакторы $t_{12}, \dots, t_{17}, x_3, \bar{x}_3$ и константа ноль; на третьем уровне – кофакторы t_1, \dots, t_{11} ; на четвертом – кофакторы h_1, \dots, h_7 . Многоуровневое описа-

ние (2) содержит 106 литералов булевых переменных. Уравнения кофакторов могут содержать различное число литералов. В частности, кофактор t_1 содержит четыре литерала, t_{15} – три, t_{17} – два литерала. Если перейти к алгебраическому символьному заданию системы ДНФ, то можно получить число (162) литералов в системе ДНФ, заданной в табл. 1. При этом каждый троичный вектор в левой части таблицы задает элементарную конъюнкцию переменных x_1, \dots, x_5 , а единицы в правой части табл. 1 отмечают вхождения элементарных конъюнкций в ДНФ компонентных функций f_1, \dots, f_5 . Например, последнему троичному вектору (1110–) соответствует элементарная конъюнкция $x_1x_2x_3\bar{x}_4$, входящая в ДНФ функций f_3 и f_5 . Эта конъюнкция внесет восемь литералов в их общее число при оценке сложности системы ДНФ по числу литералов.

Таблица 1
Матричное представление системы ДНФ векторной функции

$x_1 x_2 x_3 x_4 x_5$	$f_1 f_2 f_3 f_4 f_5$
1--00	10000
110-1	10000
-0011	10000
010-0	10001
0111-	10000
011-1	10000
0001-	10000
000-1	10000
0010-	10000
001-0	10000
-100-	10000
0--00	01000
010-1	01000
0-011	01000
1--01	01000
110-0	01010
111-1	01000
1000-	01100
1-1-1	00101
-1-10	00010
-0-01	00010
-0010	00111
0000-	00110
001-1	00110
0--01	00110
01-10	00101
00-01	00001
10-10	01000
1011-	00001
11-11	00101
1110-	00101

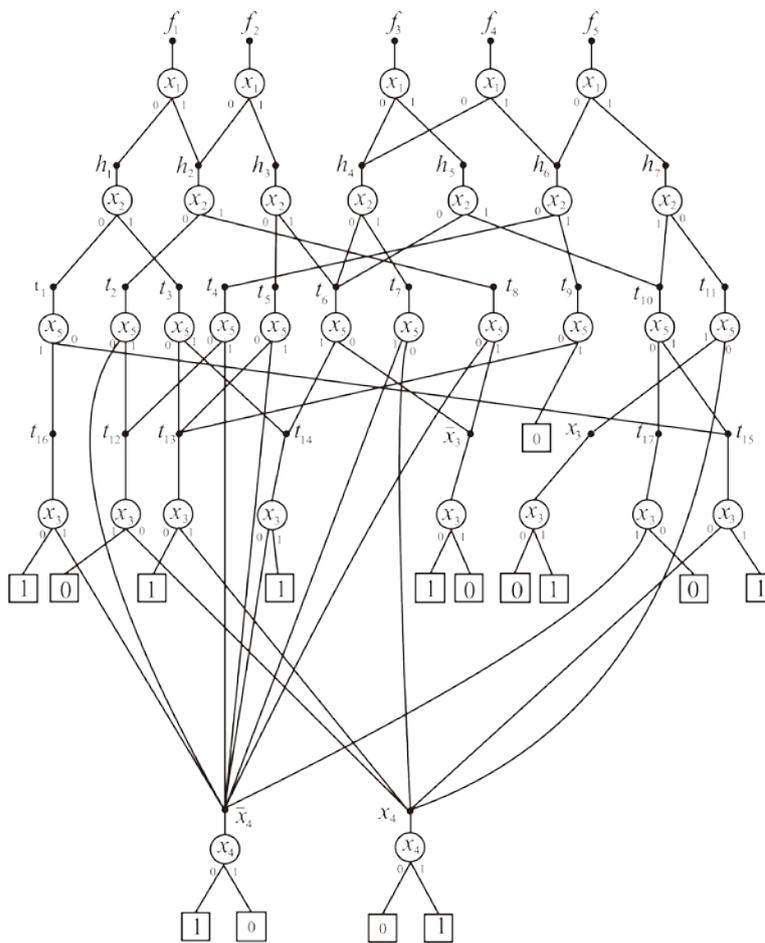


Рис. 1. BDD для векторной булевой функции (табл. 1)

Метод минимизации многоуровневого представления векторной булевой функции. Исходными данными для предлагаемого метода являются граф BDD, представляющий исходную векторную полностью определенную булеву функцию, и логические уравнения (формулы разложения Шеннона), соответствующие функциональным вершинам BDD и задающие многоуровневое описание компонентных функций. Данные уравнения легко могут быть записаны по графу BDD.

Предлагаемый метод включает этапы, выполняемые для кофакторов каждого уровня BDD, за исключением корневого, листового и первого. На первом уровне BDD располагаются кофакторы, получаемые в результате разложения по последней переменной в заданной перестав-

новке переменных, по которым ведется разложение Шеннона. Кофакторы первого уровня BDD зависят от одной переменной, кофакторы второго уровня – не более чем от двух переменных и т. д.

В методе уровни BDD рассматриваются сверху вниз, т. е. начиная от кофакторов, получаемых при разложении по первой переменной, и заканчивая кофакторами второго уровня BDD. Метод ориентирован на замену наибольшего числа формул разложения Шеннона формулами дизъюнкции, конъюнкции либо суммы по модулю два, что позволяет сокращать общее число литералов в многоуровневом задании векторной булевой функции. Для рассматриваемого уровня BDD решается задача нахождения наибольшего числа кофакторов, которые представимы в алгебраическом виде с использованием других кофакторов данного уровня. Для этого осуществляется переход на функциональный уровень представления кофакторов. С целью сокращения общего числа литералов в искомом многоуровневом представлении исходной системы булевых функций выбирается форма (прямая либо инверсная) реализации кофакторов и новая перестановка переменных, по которой строится граф BDD для кофакторов, не представленных в виде двухоперандных логических выражений (уравнений). Для рассматриваемого уровня BDD выполняются следующие восемь этапов.

Этап 1. Получение функций кофакторов, зависящих от входных переменных. Нахождение взаимно инверсных кофакторов. Выбирается форма представления кофакторов рассматриваемого уровня BDD (таблица истинности; пара ДНФ, задающая области определения кофактора) и находятся кофакторы как в прямой, так и инверсной форме. Находятся пары взаимно инверсных кофакторов.

Этап 2. Выбор формы реализации кофакторов. Для каждого кофактора осуществляется выбор, в какой форме (прямой либо инверсной) его реализовать в BDD (*задача 1*). Строится новый подграф нижних уровней BDD для системы кофакторов по новой перестановке переменных, если найдены пары взаимно инверсных кофакторов.

Этап 3. Нахождение всех вариантов алгебраических представлений кофакторов с использованием их инверсий. Находится множество уравнений, задающих всевозможные алгебраические представления кофакторов, т. е. кофакторы данного уровня, содержащие более двух литералов, выражаются в виде дизъюнкции, конъюнкции либо суммы по модулю два других кофакторов этого же уровня BDD. Кофакторы, которые могут быть выражены в виде уравнения, называются *представимыми*. При составлении уравнений для представимых кофакторов используются прямые и инверсные формы кофакторов, участвующих в уравнениях.

Этап 4. Нахождение множества реализуемых кофакторов. Из множества представимых кофакторов выделяется максимальное по мощности множество *реализуемых кофакторов*. Для этого из множества всех уравнений алгебраического представления одного и того же кофактора выбирается одно уравнение либо оно вообще не выбирается. Чтобы решить проблему выбора, строится ориентированный граф G зависимости кофакторов и выполняется редукция циклического графа G к ациклическому подграфу, если граф G не является ациклическим (*задача 2*). Решение задачи 2 позволяет выбрать из множества всех уравнений реализуемые уравнения, которые будут записаны позже в результирующее оптимизированное многоуровневое представление исходной векторной функции. По оптимизированному представлению в дальнейшем проектировании осуществляется синтез логической схемы в той или иной технологической (целевой) библиотеке.

Этап 5. Построение BDD для кофакторов, которые не являются реализуемыми. Для нереализуемых кофакторов строится подграф BDD по новой перестановке переменных рассматриваемого уровня BDD. Новая перестановка переменных ориентирована на уменьшение числа вершин BDD и, следовательно, числа формул разложения Шеннона, соответствующих нереализуемым кофакторам.

Этап 6. Составление уравнений для нереализуемых кофакторов с использованием кофакторов нижнего уровня BDD. Данный этап выполняется для BDD, построенной на этапе 5, с целью увеличения числа реализуемых кофакторов, т. е. числа алгебраических уравнений, полученных на этапе 4. Составляются уравнения для нереализованных кофакторов рассматриваемого

мого уровня BDD с использованием кофакторов нижнего уровня. Для этого множество переменных каждого кофактора нижнего уровня расширяется – добавляется несущественная переменная, по которой проводилось разложение Шеннона на рассматриваемом уровне BDD.

Этап 7. Расширение ациклического подграфа по новым уравнениям. Если на этапе 6 получено большое число (более 100) уравнений, то выполняется последовательная проверка на отсутствие циклов поочередно добавляемых в граф вершины и пары дуг, соответствующих одному уравнению. Если цикл не образуется, то это означает, что кофактор, находящийся в левой части добавляемого уравнения, является реализуемым. Если на этапе 6 получено небольшое число (менее 100) уравнений, то по ним в ациклический подграф, полученный на этапе 4, добавляются все соответствующие вершины и дуги и решается задача 2, т. е. осуществляется редукция расширенного подграфа к результирующему ациклическому подграфу, по которому определяется результирующее множество реализуемых кофакторов.

Этап 8. Корректировка многоуровневого BDD-представления векторной функции. Исходное множество формул разложений Шеннона, задающих BDD-представление (т. е. векторную булеву функцию), корректируется: формулы разложения Шеннона реализуемых кофакторов заменяются новыми формулами дизъюнкции, конъюнкции либо суммы по модулю два. Используются операции инверсии для записи формул одного из кофакторов, составляющих пару взаимно инверсных кофакторов. Здесь следует отметить, что алгоритмы, реализующие предлагаемый метод, могут начинать работу с произвольно выбранного уровня BDD.

Проиллюстрируем на примере BDD (см. рис. 1) предложенный метод, начиная с третьего уровня BDD, кофакторы t_1, \dots, t_{11} которого зависят от трех переменных x_3, x_4, x_5 .

Нахождение алгебраических представлений кофакторов на третьем уровне BDD осуществляется в восемь этапов.

Этап 1. Получение прямой и инверсной формы представления кофакторов. Нахождение взаимно инверсных кофакторов. Из BDD можно получить области определения как исходных функций, так и любого из кофакторов в виде ортогонализированных ДНФ (см. подробное описание в работе [21]). В ортогонализированной ДНФ каждая пара элементарных конъюнкций является ортогональной (их произведение равно нулю). Каждому пути из корневой вершины BDD, помеченной функцией f_j , к листовой вершине 1 соответствует элементарная конъюнкция, включающая дуги (литералы) x_i, \bar{x}_i на данном пути. При этом дуге, помеченной символом 0, соответствует отрицательный литерал \bar{x}_i ; дуге, помеченной символом 1, – положительный литерал x_i , а всем путям между указанными вершинами – дизъюнкция получаемых элементарных конъюнкций, образующая ортогонализированную ДНФ $D_{f_j}^1$. Пути из корневой вершины f_j к листовой вершине 0 задают ортогонализированную ДНФ $D_{f_j}^0$. При построении ДНФ, описывающих области определения кофактора, вместо корневой вершины BDD рассматривается функциональная вершина данного кофактора. Например, области нулевых и единичных значений кофакторов t_8, t_{11} , полученные из графа BDD на рис. 1, задаются ортогонализированными ДНФ $D_{t_8}^0 = x_4 \bar{x}_5 \vee x_3 x_5, D_{t_8}^1 = \bar{x}_4 \bar{x}_5 \vee \bar{x}_3 x_5, D_{t_8}^0 = D_{t_{11}}^1, D_{t_8}^1 = D_{t_{11}}^0$. Матричные формы данных ДНФ заданы в табл. 2. Видно, что кофакторы являются взаимно инверсными: $t_8 = \bar{t}_{11}, t_{11} = \bar{t}_8$.

В общем случае проверка равенства $t_i = t_j$ сводится к проверке одновременного выполнения двух условий: $D_{t_i}^0 \bar{D}_{t_j}^0 = 0, D_{t_i}^1 \bar{D}_{t_j}^1 = 0$. Алгоритмы перемножения и инверсирования матричных форм ДНФ описаны в работе [22 с. 34], структуры данных – в [23]. В работе [24] приведены алгоритмы и примеры перехода от ДНФ к полиномам Жегалкина, для которых легко выполняются логические операции инверсирования и перемножения булевых функций. Проверка условий $D_{t_i}^0 \bar{D}_{t_j}^0 = 0, D_{t_i}^1 \bar{D}_{t_j}^1 = 0$ может быть также сведена к задаче «выполнимость» [25, 26], если

перевести ДНФ в конъюнктивную нормальную форму. Переход от задания булевой функции t_i парой ДНФ $D_{t_i}^0, D_{t_i}^1$ к заданию функции в виде таблицы истинности не вызывает затруднений [21].

Таблица 2

Задание областей значений кофакторов t_8, t_{11} в виде ДНФ

Кофактор	$x_3 x_4 x_5$	Значение	ДНФ	Область значений
t_8	-10 1-1	0 0	$D_{t_8}^0$	$M_{t_8}^0$
	-00 0-1	1 1	$D_{t_8}^1$	$M_{t_8}^1$
t_{11}	-00 0-1	0 0	$D_{t_{11}}^0$	$M_{t_{11}}^0$
	-10 1-1	1 1	$D_{t_{11}}^1$	$M_{t_{11}}^1$

Этап 2. Выбор формы реализации кофакторов. Введем в рассмотрение k -компонентный булев вектор $\alpha = (\alpha_1, \dots, \alpha_k)$, называемый далее вектором поляризации для кофакторов (функций) t_1, \dots, t_k . Обозначим $\alpha_i = 1$, если рассматривается кофактор t_i , и $\alpha_i = 0$, если берется инверсия \bar{t}_i кофактора t_i .

Задача 1. Для кофакторов t_1, \dots, t_k требуется найти вектор поляризации $\alpha = (\alpha_1, \dots, \alpha_k)$, для которого значение оценки сложности BDD, построенной по общей для всех кофакторов перестановке переменных, будет минимально.

В качестве оценки сложности BDD в большинстве научных работ рассматривается число вершин графа BDD. Однако уравнения разложения Шеннона, соответствующие вершинам BDD, могут содержать различное число литералов. Поэтому в настоящем исследовании целесообразно в качестве оценок сложности BDD (и других алгебраических представлений) использовать критерий «число литералов» в уравнениях, соответствующих BDD. Решение задачи 1 представляет особый интерес и в данной работе не рассматривается. Далее для демонстрации предлагаемого метода некоторые кофакторы при реализации выбраны в инверсной форме, что привело к получению новой перестановки $\langle x_1, x_2, x_4, x_5, x_3 \rangle$ переменных (рис. 2) при реализации кофакторов третьего уровня BDD и некоторому сокращению числа литералов. Многоуровневое представление (3) содержит 98 литералов булевых переменных:

$$\begin{aligned}
 f_1 &= \bar{x}_1 h_1 \vee x_1 h_2; f_2 = \bar{x}_1 h_2 \vee x_1 h_3; f_3 = \bar{x}_1 h_4 \vee x_1 h_5; f_4 = \bar{x}_1 h_4 \vee x_1 h_6; f_5 = \bar{x}_1 h_6 \vee x_1 h_7; h_1 = \bar{x}_2 \bar{g}_1 \vee x_2 \bar{g}_3; \\
 h_2 &= \bar{x}_2 g_2 \vee x_2 \bar{g}_8; h_3 = \bar{x}_2 \bar{g}_5 \vee x_2 g_6; h_4 = \bar{x}_2 g_6 \vee x_2 \bar{g}_7; h_5 = \bar{x}_2 g_6 \vee x_2 g_{10}; h_6 = \bar{x}_2 g_4 \vee x_2 g_9; \\
 h_7 &= \bar{x}_2 g_8 \vee x_2 g_{10}; g_1 = \bar{x}_4 s_2 \vee x_4 s_3; g_2 = \bar{x}_4 \bar{x}_5 \vee x_4 s_1; g_3 = \bar{x}_4 s_0 \vee x_4 s_1; g_4 = \bar{x}_4 x_5 \vee x_4 s_2; \\
 g_5 &= \bar{x}_4 s_0 \vee x_4 x_5; g_6 = \bar{x}_4 s_5 \vee x_4 s_6; g_7 = \bar{x}_4 \bar{x}_5 \vee x_4 x_5; g_8 = \bar{x}_4 s_3 \vee x_4 s_4; g_9 = \bar{x}_4 s_2 \vee x_4 \bar{x}_5; \\
 g_{10} &= \bar{x}_4 x_3 \vee x_4 x_5; s_0 = \bar{x}_5 x_3; s_1 = x_5 \bar{x}_3; s_2 = \bar{x}_5 \bar{x}_3; s_3 = x_5 x_3; s_4 = \bar{x}_5 \vee x_5 x_3; s_5 = \bar{x}_5 \bar{x}_3 \vee x_5; \\
 s_6 &= \bar{x}_5 \bar{x}_3 \vee x_5 x_3.
 \end{aligned} \tag{3}$$

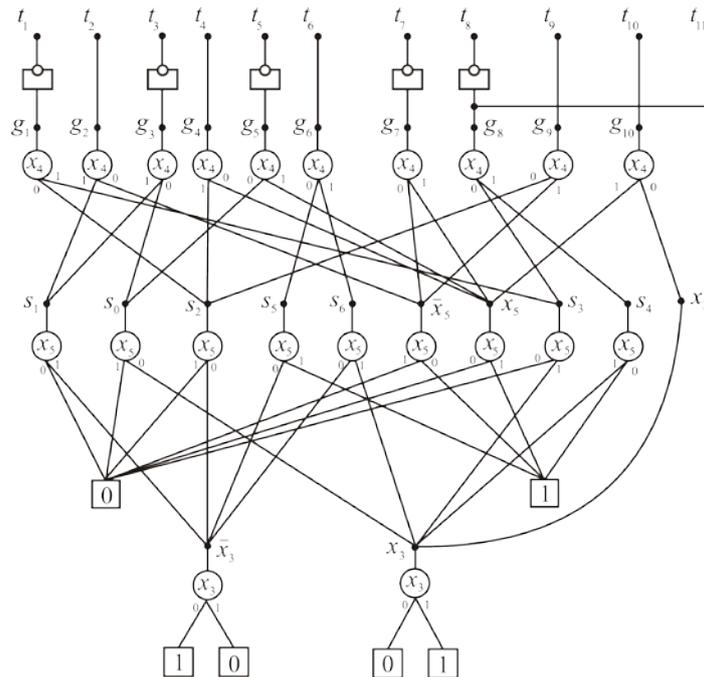


Рис. 2. Результат выполнения этапа 2 – реализация инверсных форм кофакторов t_1, t_3, t_5, t_7, t_8

Этап 3. Нахождение всех вариантов алгебраических представлений кофакторов с использованием их инверсий. Для получения уравнений, задающих алгебраические представления кофакторов, будем использовать обобщенно ортогонализированные формы векторных булевых функций. Нахождение обобщенно ортогонализированной формы для векторной булевой функции $f(x) = (f_1(x), \dots, f_m(x))$ сводится к задаче нахождения ее минимального дизъюнктивного базиса. Зададим каждую компонентную булеву функцию $f_i(x)$ двумя полностью определенными булевыми функциями $\lambda_i^0(x), \lambda_i^1(x)$, такими, что функция $\lambda_i^0(x)$ имеет характеристическое множество $M_{f_i}^0$, а функция $\lambda_i^1(x)$ – характеристическое множество $M_{f_i}^1$. Минимальным дизъюнктивным базисом для $f(x) = (f_1(x), \dots, f_m(x))$ называется минимальная по мощности система попарно ортогональных полностью определенных булевых функций $\Pi = \{\Pi_1(x), \dots, \Pi_k(x)\}$, такая, что каждая полностью определенная функция $\lambda_i^1(x), \lambda_i^0(x)$ равна дизъюнкции некоторого подмножества функций системы Π . Полностью определенная булева функция $f_i(x)$ задает двухблочное разбиение $R_i = \{M_{f_i}^0, M_{f_i}^1\}$ булева пространства V^x на непересекающиеся подмножества. Нахождение для $f(x) = (f_1(x), \dots, f_m(x))$ минимального дизъюнктивного базиса (в теоретико-множественной интерпретации) может быть сведено к нахождению произведения разбиений $R_i, i = 1, \dots, m$, и составлению из непустых блоков полученного произведения разбиений каждого из исходных блоков $M_{f_i}^0, M_{f_i}^1$ [27, с. 12]. Пусть каждая компонентная функция f_i задана парой попарно ортогональных ДНФ $D_{f_i}^0, D_{f_i}^1$, данные ДНФ представляют функции $\lambda_i^0(x), \lambda_i^1(x)$ соответственно. В этом случае нахождение минимального дизъюнктивного базиса Π может быть сведено к выполнению операций перемножения, инверсирования и сравнения на равенство ДНФ (алгоритм приведен в работе [28, с. 124]).

Опишем алгоритм получения уравнений (алгебраических представлений) для кофакторов одного уровня BDD. Рассмотрим кофакторы g_1, \dots, g_{10} третьего уровня BDD (рис. 2). Для каждой неупорядоченной пары g_i, g_j , где $i, j = 1, \dots, 10, i \neq j$, построим обобщенно

ортогонализированную форму, затем для каждой из функций g_p , $p = 1, \dots, 10$, $p \neq i$, $p \neq j$, будет проверяться выполнение равенств

$$g_p = g_i \vee g_j, \quad (4)$$

$$g_p = g_i \& g_j, \quad (5)$$

$$g_p = g_i \oplus g_j. \quad (6)$$

При этом каждый из кофакторов рассматривается как в прямой, так и инверсной форме. Например, для дизъюнкции проверяется выполнение равенств

$$g_p = g_i \vee g_j, \quad g_p = \bar{g}_i \vee g_j, \quad g_p = g_i \vee \bar{g}_j, \quad g_p = \bar{g}_i \vee \bar{g}_j, \\ \bar{g}_p = g_i \vee g_j, \quad \bar{g}_p = \bar{g}_i \vee g_j, \quad \bar{g}_p = g_i \vee \bar{g}_j, \quad \bar{g}_p = \bar{g}_i \vee \bar{g}_j.$$

Подобные равенства проверяются для конъюнкции и суммы по модулю два. Если равенство выполняется, то оно записывается в виде уравнения. Легко понять, что согласно законам булевой алгебры будут появляться двойственные уравнения. Например, если выполняется равенство $g_p = \bar{g}_i \vee \bar{g}_j$, то обязательно будет записано двойственное равенство $\bar{g}_p = g_i \& g_j$ и т. д. Если выполняется равенство $g_p = g_i \oplus g_j$, то при таких проверках будут получены еще три двойственных равенства: $g_p = \bar{g}_i \oplus \bar{g}_j$, $\bar{g}_p = \bar{g}_i \oplus g_j$, $\bar{g}_p = g_i \oplus \bar{g}_j$. Для проведения проверок выполнения всех перечисленных равенств по обобщенно ортогонализированной форме векторной функции $g_{i,j} = (g_i, g_j)$ строится обобщенно ортогонализированная форма векторной функции $g = (g_p, g_i, g_j)$, по которой легко проверяется выполнение данных равенств. Приведем пример: по BDD на рис. 2 выпишем элементарные конъюнкции, которые задают в виде ДНФ области нулевых и единичных значений кофакторов g_1 , g_7 , g_6 , g_8 . Соответствующие ДНФ приведены в табл. 3.

Таблица 3

Задание областей значений кофакторов g_1 , g_7 , g_6 , g_8 в виде ДНФ

Компонентная функция	$x_3 x_4 x_5$	Значение	ДНФ	Область значений	Функция λ
g_1	-01 -10 011 100	0 0 0 0	$D_{g_1}^0$	$M_{g_1}^0$	λ_1^0
	000 111	1 1	$D_{g_1}^1$	$M_{g_1}^1$	λ_1^1
g_7	-01 -10	0 0	$D_{g_7}^0$	$M_{g_7}^0$	λ_7^0
	-00 -11	1 1	$D_{g_7}^1$	$M_{g_7}^1$	λ_7^1
g_6	100 011 110	0 0 0	$D_{g_6}^0$	$M_{g_6}^0$	λ_6^0
	000 -01 111 010	1 1 1 1	$D_{g_6}^1$	$M_{g_6}^1$	λ_6^1
	-00 010 011	0 0 0	$D_{g_8}^0$	$M_{g_8}^0$	λ_8^0
	101 -10 111	1 1 1	$D_{g_8}^1$	$M_{g_8}^1$	λ_8^1

ДНФ Π_i минимального дизъюнктивного базиса для векторной функции $\mathbf{g}_{7,6} = (g_7, g_6)$ приведены в табл. 4. Представим ДНФ, задающие области нулевых и единичных значений кофакторов g_7, g_6 , в виде дизъюнкций ДНФ минимального дизъюнктивного базиса:

$$D_{g_7}^0 = \Pi_2 \vee \Pi_3, D_{g_7}^1 = \Pi_1 \vee \Pi_4, D_{g_6}^0 = \Pi_3 \vee \Pi_4, D_{g_6}^1 = \Pi_1 \vee \Pi_2.$$

Обобщенно ортогонализированные формы векторных функций $\mathbf{g}_{7,6} = (g_7, g_6)$ и $\mathbf{g} = (g_1, g_7, g_6)$ даны в табл. 5 и 6 соответственно. Кофактор g_1 может быть выражен на тех же четырех ДНФ Π_i : $D_{g_1}^0 = \Pi_2 \vee \Pi_3 \vee \Pi_4$; $D_{g_1}^1 = \Pi_1$. Функция g_1 может быть выражена как конъюнкция функций g_7, g_6 , если рассматривать логическую операцию $\&$ над соответствующими значениями функций g_7, g_6 из табл. 5, т. е. $g_1 = g_7 \& g_6$. По табл. 6 легко проверить, что $g_8 \neq g_7 \vee g_6, g_8 \neq g_7 \oplus g_6$.

Таблица 4

Минимальный дизъюнктивный базис для векторной функции $\mathbf{g} = (g_7, g_6)$

Π_i	$x_3 \ x_4 \ x_5$
Π_1	0 0 0 1 1 1
Π_2	0 0 1 0 1 0 1 0 1
Π_3	1 1 0
Π_4	0 1 1 1 0 0

Таблица 5

Обобщенно ортогонализированная форма векторной функции $\mathbf{g} = (g_7, g_6)$

Π_i	$g_7 \ g_6$
Π_1	1 1
Π_2	0 1
Π_3	0 0
Π_4	1 0

Таблица 6

Обобщенно ортогонализированная форма векторной функции $\mathbf{g} = (g_1, g_7, g_6)$

Π_i	$g_1 \ g_7 \ g_6$
Π_1	1 1 1
Π_2	0 0 1
Π_3	0 0 0
Π_4	0 1 0

Если же рассматривать кофактор g_8 , то для проверки равенств

$$g_8 = g_7 \vee g_6, \tag{7}$$

$$g_8 = g_7 \& g_6, \tag{8}$$

$$g_8 = g_7 \oplus g_6 \tag{9}$$

построим обобщенно ортогонализированную форму векторной функции $\mathbf{g} = (g_8, g_7, g_6)$, перемножая ДНФ Π^i (табл. 4) с ДНФ $D_{g_8}^0, D_{g_8}^1$ (табл. 3). Тогда в результате построения минимального дизъюнктивного базиса для $\mathbf{g} = (g_8, g_7, g_6)$ выяснится, что $\Pi_1 = P_1 \vee P_2, \Pi_2 = P_3 \vee P_4, \Pi_3 = P_5, \Pi_4 = P_6$, где P_i даны в табл. 7. В результате получается обобщенно ортогонализированная форма векторной функции $\mathbf{g} = (g_8, g_7, g_6)$ (табл. 8), которой соответствуют формулы $D_{g_7}^0 = P_3 \vee P_4 \vee P_5, D_{g_7}^1 = P_1 \vee P_2 \vee P_6, D_{g_6}^0 = P_5 \vee P_6, D_{g_6}^1 = P_1 \vee P_2 \vee P_3 \vee P_4, D_{g_8}^0 = P_1 \vee P_3 \vee P_6, D_{g_8}^1 = P_2 \vee P_4 \vee P_5$. По табл. 8 легко проверить, что кофактор g_8 не может быть представлен формулами (7)–(9), т. е. $g_8 \neq g_7 \vee g_6, g_8 \neq g_7 \& g_6, g_8 \neq g_7 \oplus g_6$. Использование инверсий кофакторов g_8, g_7, g_6 также не приводит к получению алгебраических представлений кофактора g_8 . Такой вывод можно сделать, если представить каждую ДНФ $P_i, i = 1, \dots, 6$, минимального дизъюнктивного базиса в виде совершенной ДНФ, а затем перейти к таблицам истинности кофакторов g_8, g_7, g_6 .

Таблица 7

Минимальный дизъюнктивный
базис для векторной функции

$$\mathbf{g} = (g_8, g_7, g_6)$$

P_i	$x_3 \ x_4 \ x_5$
P_1	0 0 0
P_2	1 1 1
P_3	0 0 1
P_4	0 1 0 1 0 1
P_5	1 1 0
P_6	0 1 1 1 0 0

Таблица 8

Обобщенно ортогонализованная
форма векторной функции

$$\mathbf{g} = (g_8, g_7, g_6)$$

P_i	$g_8 \ g_7 \ g_6$
P_1	0 1 1
P_2	1 1 1
P_3	0 0 1
P_4	1 0 1
P_5	1 0 0
P_6	0 1 0

Обобщенно ортогонализованная форма – это компактная форма задания таблицы истинности полностью определенной векторной булевой функции. Компактность может быть обеспечена за счет представления ДНФ минимального дизъюнктивного базиса минимизированными формами, например кратчайшими ДНФ [2].

Утверждение. Проверка условий существования функциональных представлений

$$\varphi(\mathbf{x}) = f_1(\mathbf{x}) \vee f_2(\mathbf{x}), \quad \varphi(\mathbf{x}) = f_1(\mathbf{x}) \& f_2(\mathbf{x}), \quad \varphi(\mathbf{x}) = f_1(\mathbf{x}) \oplus f_2(\mathbf{x})$$

может выполняться как по таблице истинности векторной функции $\mathbf{f} = (\varphi, f_1, f_2)$, так и по таблице задания обобщенно ортогонализованной формы этой векторной функции.

В рассматриваемых примерах функций от трех переменных обобщенно ортогонализированные формы (табл. 5, 6, 8) содержат четыре либо шесть строк, что мало отличается от таблиц истинности, содержащих для данных примеров по восемь строк. В ситуациях, когда решаются практические задачи большой размерности (десятки переменных), обобщенно ортогонализированные формы могут быть гораздо более компактнее таблиц истинности.

Зададим все кофакторы таблицей истинности (табл. 9), что позволит проверять правильность алгебраического представления кофакторов третьего уровня BDD. В результате выполнения этапа 3 будет получено 48 уравнений, не считая 48 двойственных. Приведем их в двух таблицах: уравнения без использования инверсий – в табл. 10, с использованием инверсий – в табл. 11.

Таблица 9

Кофакторы третьего уровня BDD (см. рис. 2)

$x_3 \ x_4 \ x_5$	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
0 0 0	1	1	0	0	0	1	1	0	1	0
0 0 1	0	0	0	1	0	1	0	0	0	0
0 1 0	0	0	0	1	0	1	0	1	1	0
0 1 1	0	1	1	0	1	0	1	0	0	1
1 0 0	0	1	1	0	1	0	1	0	0	1
1 0 1	0	0	0	1	0	1	0	1	0	1
1 1 0	0	0	0	0	0	0	0	1	1	0
1 1 1	1	0	0	0	1	1	1	1	0	1

Для каждого из уравнений с операторами \vee , $\&$ в табл. 10 и 11 имеется двойственное уравнение, а для уравнений с операторами \oplus из табл. 11 – два двойственных.

Таблица 10

Варианты представления кофакторов третьего уровня BDD без использования инверсий операндов

Номер уравнения	Уравнение
1	$g_1 = g_2 \oplus g_5$
2	$g_1 = g_6 \oplus g_4$
3	$g_1 = g_7 \oplus g_3$
4	$g_1 = g_7 \& g_6$
5	$g_2 = g_5 \oplus g_1$
6	$g_3 = g_2 \& g_{10}$
7	$g_3 = g_2 \& g_5$
8	$g_3 = g_7 \oplus g_1$
9	$g_4 = g_6 \oplus g_1$
10	$g_5 = g_2 \oplus g_1$
11	$g_5 = g_7 \& g_{10}$
12	$g_6 = g_1 \oplus g_4$
13	$g_6 = g_1 \vee g_4$
14	$g_7 = g_2 \vee g_1$
15	$g_7 = g_2 \vee g_5$
16	$g_7 = g_3 \oplus g_1$
17	$g_7 = g_3 \vee g_1$
18	$g_7 = g_5 \vee g_1$

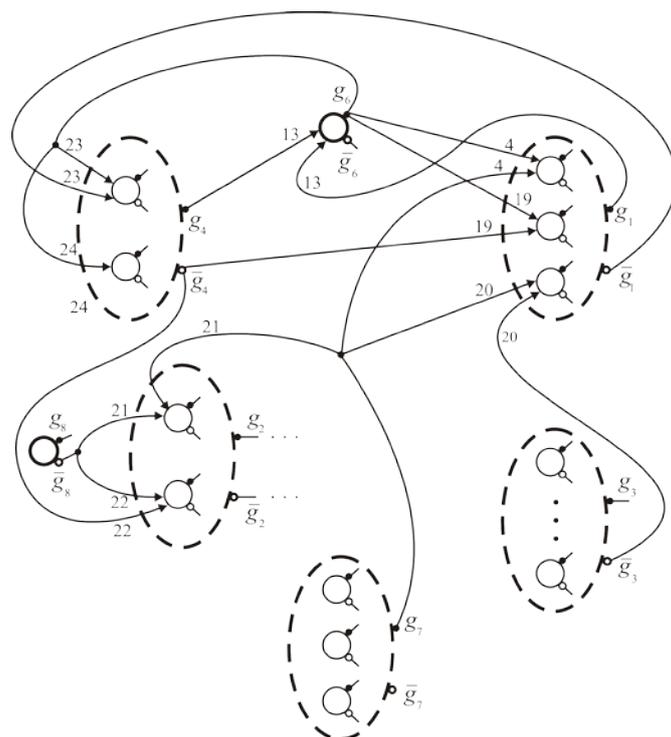
Таблица 11

Варианты представления кофакторов третьего уровня BDD с использованием инверсий операндов

Номер уравнения	Уравнение	Номер уравнения	Уравнение
19	$g_1 = g_6 \& \bar{g}_4$	34	$g_3 = \bar{g}_8 \& g_5$
20	$g_1 = g_7 \& \bar{g}_3$	35	$g_3 = \bar{g}_8 \& \bar{g}_6$
21	$g_2 = \bar{g}_8 \& g_7$	36	$g_3 = \bar{g}_9 \& \bar{g}_6$
22	$g_2 = \bar{g}_8 \& \bar{g}_4$	37	$g_5 = g_{10} \& \bar{g}_4$
23	$g_4 = g_6 \& \bar{g}_1$	38	$g_5 = g_7 \& \bar{g}_9$
24	$g_4 = g_6 \vee \bar{g}_7$	39	$g_5 = \bar{g}_9 \& \bar{g}_4$
25	$g_3 = g_{10} \& \bar{g}_6$	40	$g_1 = \bar{g}_2 \oplus \bar{g}_5$
26	$g_3 = g_2 \& \bar{g}_1$	41	$g_1 = \bar{g}_6 \oplus \bar{g}_4$
27	$g_3 = g_2 \& \bar{g}_6$	42	$g_1 = \bar{g}_7 \oplus \bar{g}_3$
28	$g_3 = g_2 \& \bar{g}_9$	43	$g_2 = \bar{g}_5 \oplus \bar{g}_1$
29	$g_3 = g_5 \& \bar{g}_1$	44	$g_3 = \bar{g}_7 \oplus \bar{g}_1$
30	$g_3 = g_5 \& \bar{g}_6$	45	$g_4 = \bar{g}_6 \oplus \bar{g}_1$
31	$g_3 = g_7 \& \bar{g}_1$	46	$g_5 = \bar{g}_2 \oplus \bar{g}_1$
32	$g_3 = g_7 \& \bar{g}_6$	47	$g_6 = \bar{g}_1 \oplus \bar{g}_4$
33	$g_3 = \bar{g}_8 \& g_{10}$	48	$g_7 = \bar{g}_3 \oplus \bar{g}_1$

Этап 4. Нахождение множества реализуемых кофакторов. Построим ориентированный граф G , задающий формулы алгебраических представлений кофакторов (будем называть их логическими уравнениями либо просто уравнениями). Вершинам графа G соответствуют кофакторы в прямой форме, которые упоминаются в уравнениях. Обозначим это множество кофакторов K_Y , а множество кофакторов, не участвующих в уравнениях, K_Z . Дуги (ориентированные ребра), соответствующие операндам алгебраически представленного кофактора (результату логической операции), помечены одним и тем же числом – номером уравнения. Каждое уравнение задается подграфом с тремя вершинами и двумя помеченными дугами: вершина, соответствующая представимому кофактору, имеет две заходящие дуги (с одной и той же меткой – номером уравнения). Исходящие из вершины представимого кофактора дуги (рис. 3) могут соответствовать прямой форме кофактора (черный кружок) либо инверсной форме кофактора (светлый кружок).

Вершины, соответствующие операндам уравнения, инцидентны исходящим дугам. Например, уравнению 13 $g_6 = g_1 \vee g_4$ соответствует подграф с тремя вершинами g_1, g_4, g_6 : в вершину g_6 заходят из вершин g_1, g_4 дуги, помеченные числом 13 (номером уравнения). Если кофактор в уравнении разложения Шеннона содержит один либо два литерала, то для него на этапе 2 не строятся алгебраические представления, поэтому соответствующая данному кофактору вершина в графе G не имеет заходящих дуг. Введем понятие кластера (подмножества вершин графа G): в кластер входят вершины, соответствующие уравнениям с одним и тем же кофактором (возможно, в инверсном виде) в левой части. Именем одинаковых кофакторов и обозначается кластер. В рассматриваемом примере граф G будет содержать 10 вершин, которые соединяются между собой 48 парами дуг. Фрагмент графа показан на рис. 3. Заметим, что каждой из вершин графа G соответствует кофактор, уравнение которого содержит четыре литерала (см. выражение (3) для кофакторов третьего уровня BDD).

Рис. 3. Фрагмент графа G

С использованием графа G задача нахождения множества реализуемых кофакторов сводится к следующей задаче.

Задача 2. В графе G требуется оставить в каждом кластере только одну вершину (условие A) и удалить из графа G такое подмножество пар одинаково помеченных дуг, чтобы граф G стал ациклическим (условие B) и содержал наибольшее число неизолированных вершин, в каждую из которых заходит только одна пара одинаково помеченных дуг (условие C).

Смысл оставления в кластере одной вершины (условие A) очевиден – кофактор при схемной реализации достаточно представить только одним уравнением. Требование ациклическости (условие B) продемонстрируем на примере двух уравнений ($g_6 = g_1 \vee g_4$, $g_1 = g_6 \& g_7$) из рассматриваемого примера. Данные уравнения в графе G создают цикл, т. е. логическое противоречие в требовании алгебраического (и схемного) представления кофакторов, заключающееся в следующем. Чтобы выразить g_6 в виде конъюнкции $g_6 = g_1 \vee g_4$ (уравнение 13), требуется схема, реализующая кофактор g_1 . Однако, чтобы получить g_1 в виде $g_1 = g_7 \& g_6$ (см. уравнение (4)), требуется схема, реализующая g_6 . Условие C требует, чтобы как можно больше кофакторов было представлено в виде конъюнкции, дизъюнкции либо суммы по модулю два.

После решения задачи 2 множество K_Y вершин графа G разбивается на три попарно непересекающихся подмножества. Вершины, не имеющие входящих дуг, но имеющие исходящую дугу, соответствуют нереализуемым кофакторам. Вершины, имеющие одну пару входящих дуг (с одной и той же пометкой), соответствуют реализуемым кофакторам. Вершины, которые оказались изолированными, например, при удалении дуг для удовлетворения условия B , не будут участвовать в уравнениях и зачисляются в множество K_Z также нереализуемых кофакторов. Реализуемый кофактор будет записан в виде уравнения в результирующее многоуровневое представление системы булевых функций.

Для решения задачи 2 предлагается использовать следующий *эвристический алгоритм*, основанный на последовательном рассмотрении уравнений, по которым строятся соответствующие подграфы графа G :

Шаг 1. Построить подграф G_{NOTI} графа G , образованный вершинами и дугами уравнений без использования инверсий.

Шаг 2. Удалить из каждого кластера подграфа G_{NOTI} вершины (и заходящие в них дуги), соответствующие уравнениям, которые содержат операцию \oplus в тех случаях, если имеются вершины, соответствующие уравнениям, которые содержат операцию \vee либо $\&$.

Шаг 3. Сократить число вершин в кластерах подграфа G_{SNOTI} и оставить в каждом кластере только одну вершину.

Шаг 3.1. Упорядочить кластеры по возрастанию (неубыванию) числа содержащихся в них вершин и рассматривать кластеры в этом порядке.

Шаг 3.2. Удалить вершины в кластере.

Положить пустым текущее множество T вершин: $T = \emptyset$. Каждая оставленная вершина g_r в кластере, в которую заходят одноименно помеченные дуги, инцидентные вершинам g_i, g_j , пополняет множество T элементами g_i, g_j, g_r .

Для выбора оставляемой в первом кластере вершины применяется

Эвристика 1. Каждая вершина g_r кластера, имеющая заходящие дуги из вершин g_i, g_j , оценивается суммарным числом W_{qr} исходящих дуг из вершин g_i, g_j . Оставляется в кластере та вершина g_s , которой соответствует максимальное значение W_{qr} .

Вершины, которые не остались в кластере, удаляются из графа вместе с заходящими в них дугами. В множество T заносятся три вершины: вершина g_s и две вершины, инцидентные заходящим в вершину g_s дугам.

Для следующих рассматриваемых кластеров используется

Эвристика 2. В кластере оставляется та вершина, которая имеет две заходящие дуги из вершин множества T . Если такой вершины нет, то оставляется та вершина, которая имеет одну заходящую дугу из вершин множества T . Если нет вершин, для которых данная эвристика выполняется, то для оставления вершины в кластере используется эвристика 1. Вершины, которые не остались в кластере, удаляются из графа вместе с заходящими в них дугами.

Шаг 3 считается выполненным, когда в каждом кластере останется по одной вершине.

Шаг 4. Удалить пары одинаково помеченных дуг для получения ациклического подграфа G_{SNOTI} . (Выполняется, если граф G_{SNOTI} содержит циклы, т. е. не является ациклическим.)

Для графа G_{SNOTI} , содержащего циклы, выполняются шаги 4.1–4.3 до тех пор, пока граф не станет ациклическим.

Шаг 4.1. Найти все циклы графа G_{SNOTI} .

Шаг 4.2. Каждую пару дуг с одинаковой пометкой (номером) p оценить числом C_p циклов, которые могут быть разорваны при удалении из графа данной пары дуг.

Шаг 4.3. Удалить пару дуг с пометкой p , которой соответствует максимальное значение числа C_p . Переход на шаг 4.1.

Шаг 5. Из множества уравнений с инверсиями кофакторов удалить уравнения для реализованных кофакторов. Поочередно добавлять в полученный на шаге 4 ациклический граф дуги и вершины, соответствующие нереализованным кофакторам, и проверять получаемый граф на ациклическость. Дуги и вершины из множества уравнений для кофакторов с инверсиями не добавляются в граф, если они вносят цикл. Если рассмотрены все уравнения из множества оставшихся уравнений с инверсиями, то выполнить переход на шаг 6.

Шаг 6. Конец.

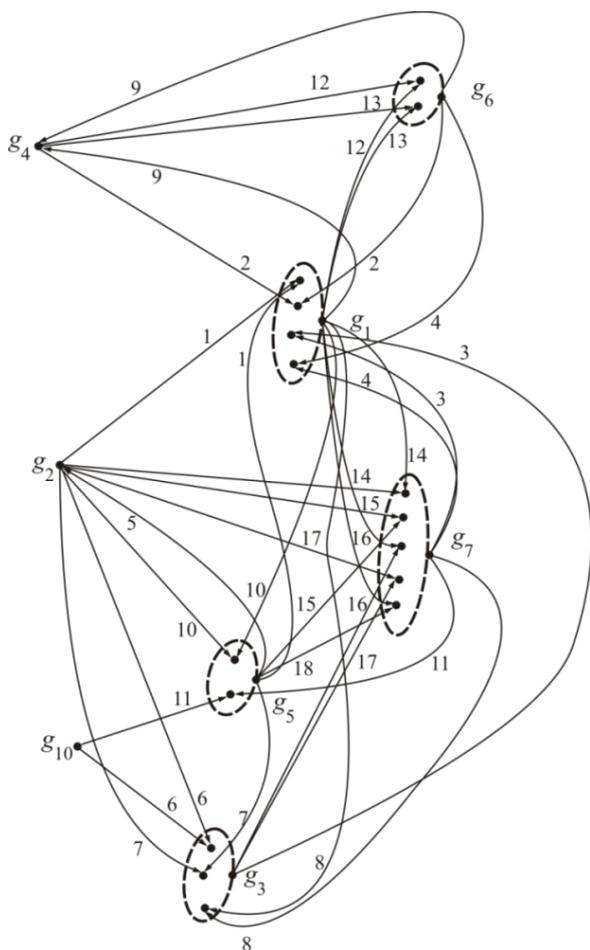


Рис. 4. Подграф G_{NOTI} для 18 уравнений из табл. 10

Рассмотрим алгоритм на примере графа G_{SNOTI} (рис. 4).

Шаг 1. Подграф G_{SNOTI} изображен на рис. 4, содержит восемь вершин, из них вершины g_2, g_4, g_{10} не являются вершинами-кластерами, вершин g_8, g_9 нет в подграфе G_{SNOTI} , так как кофакторы g_8, g_9 не входят в уравнения из табл. 10.

Замечание: если граф G построен по уравнениям, в которых нет операций инверсий операндов, то вершины графа представляются на рис. 4 точками (черными кружками, а светлые кружки отсутствуют).

Шаг 2. Уравнения, по которым строится подграф G_{SNOTI} графа G_{NOTI} , заданы в табл. 12, полученный подграф G_{SNOTI} показан на рис. 5.

Таблица 12

Варианты алгебраического представления кофакторов после шага 2 алгоритма решения задачи 2

Номер уравнения	Уравнение	Номер уравнения	Уравнение
4	$g_1 = g_7 \& g_6$	13	$g_6 = g_1 \vee g_4$
7	$g_3 = g_2 \& g_5$	17	$g_7 = g_3 \vee g_1$
6	$g_3 = g_2 \& g_{10}$	18	$g_7 = g_5 \vee g_1$
9	$g_4 = g_6 \oplus g_1$	14	$g_7 = g_2 \vee g_1$
11	$g_5 = g_7 \& g_{10}$	15	$g_7 = g_2 \vee g_5$

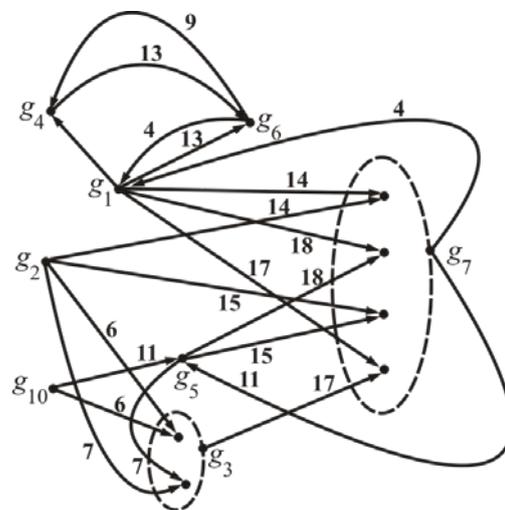


Рис. 5. Подграф G_{SNOTI}

Шаг 3. Сокращение числа вершин в кластерах подграфа G_{SNOTI} .

Шаг 3.1. Порядок рассмотрения кластеров $\langle g_3, g_7 \rangle$.

Шаг 3.2. По эвристике 1 на этом шаге в кластере g_3 оставляем вершину с заходящими дугами g_2, g_5 , так как суммарное число W_{q_3} исходящих дуг для нее равно семи. Полагаем $T = \{g_2, g_5, g_3\}$. Для кластера g_7 выбираем вершину с заходящими дугами g_2, g_5 (эвристика 2), так как в этом случае в множестве $T = \{g_2, g_5, g_3\}$ имеются две общие вершины g_2, g_5 . После этого граф G принимает вид, показанный на рис. 6, а.

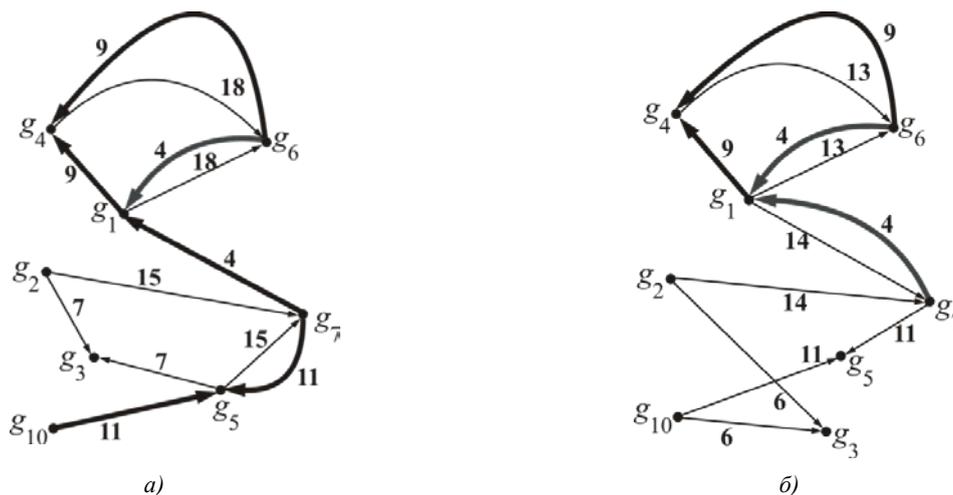


Рис. 6. Подграф G_{SNOTI} после выполнения шага 3 алгоритма решения задачи 2:
а) по эвристике 1; б) другой вариант

Шаг 4. Чтобы разорвать циклы (первый цикл – $\{g_1, g_6\}$, второй – $\{g_5, g_7\}$, третий – $\{g_4, g_6\}$, четвертый – $\{g_1, g_4, g_6\}$), удаляем пары дуг с пометками 4, 11, 9; результирующий граф, в вершины g_3, g_7, g_6 которого заходят попарно дуги с одинаковыми пометками, соответствует трем выбранным (реализуемым) уравнениям: $g_3 = g_2 \& g_5, g_7 = g_2 \vee g_5, g_6 = g_1 \vee g_4$. Полученное решение означает, что три представимых кофактора (g_3, g_7, g_6) могут быть представлены в виде дизъюнкции либо конъюнкции четырех кофакторов (g_2, g_5, g_1, g_4).

Применяемые эвристики позволяют получать приближенное решение задачи 2. Если на шаге 3 в первом рассматриваемом кластере g_3 будет оставлена вершина, имеющая заходящие дуги из вершин g_2, g_{10} , то можно получить четыре алгебраически представимых кофактора. Покажем это, еще раз выполнив шаги 3 и 4.

Шаг 3 (повторно). Порядок рассмотрения кластеров $\langle g_3, g_7 \rangle$. Оставляем вершину кластера g_3 , имеющую заходящие дуги из вершин g_2, g_{10} , полагаем $T = \{g_2, g_{10}, g_3\}$. Для второго кластера g_7 выбираем вершину с дугами, имеющими пометку 14 и инцидентными вершинам g_1, g_2 , так как $g_2 \in T$, где $T = \{g_2, g_{10}, g_3\}$. По эвристике 1 имеется одна вершина, входящая в множество T . В результате выполнения первого шага граф G принимает вид, представленный на рис. 6, б.

Шаг 4 (повторно). В графе (рис. 6, б) имеются четыре цикла: $\{g_1, g_6\}, \{g_1, g_7\}, \{g_4, g_6\}, \{g_1, g_4, g_6\}$. Удаление пары дуг с пометками 4, 9 разрывает оба цикла. Результирующий ациклический граф, полученный в результате шага 4 эвристического алгоритма, показан на рис. 7. Данному графу соответствуют четыре уравнения: $g_3 = g_2 \& g_{10}, g_5 = g_7 \& g_{10}, g_6 = g_1 \vee g_4,$

$g_7 = g_1 \vee g_2$. Четыре кофактора (g_2, g_{10}, g_1, g_4) позволяют получить алгебраические представления других четырех кофакторов (g_3, g_5, g_6, g_7). Эти представления кофакторов будем использовать далее.

Шаг 5. Из табл. 11 уравнений с инверсиями удалим уравнения для четырех реализуемых кофакторов (g_3, g_5, g_6, g_7), оставшиеся уравнения заданы в табл. 13. В ациклический граф (рис. 7, а) будем добавлять поочередно вершины и дуги, соответствующие уравнениям из табл. 13 (всего 11 уравнений). Каждое из уравнений вносит цикл в граф, кроме уравнения 22 ($g_2 = \bar{g}_8 \& \bar{g}_4$) (рис. 7, б). Таким образом, использование инверсий позволяет найти выражение $g_2 = \bar{g}_8 \& \bar{g}_4$, свидетельствующее о том, что кофактор g_2 стал реализуемым. Реализуемые кофакторы третьего уровня с использованием инверсий: $g_3 = g_2 \& g_{10}$; $g_5 = g_7 \& g_{10}$; $g_6 = g_1 \vee g_4$; $g_7 = g_1 \vee g_2$; $g_2 = \bar{g}_8 \& \bar{g}_4$.

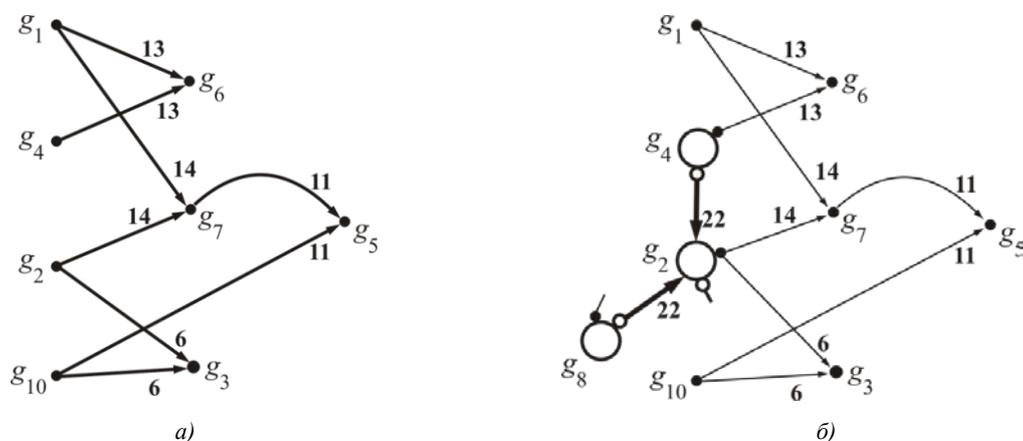


Рис. 7. Ациклический граф G после выполнения: а) шага 4 эвристического алгоритма решения задачи 2; б) эвристического алгоритма решения задачи 2

Таблица 13

Варианты алгебраического представления
нереализованных кофакторов

Номер уравнения	Уравнение	Номер уравнения	Уравнение
19	$g_1 = g_6 \& \bar{g}_4$	40	$g_1 = \bar{g}_2 \oplus \bar{g}_5$
20	$g_1 = g_7 \& \bar{g}_3$	41	$g_1 = \bar{g}_6 \oplus \bar{g}_4$
21	$g_2 = \bar{g}_8 \& g_7$	42	$g_1 = \bar{g}_7 \oplus \bar{g}_3$
22	$g_2 = \bar{g}_8 \& \bar{g}_4$	43	$g_2 = \bar{g}_5 \oplus \bar{g}_1$
23	$g_4 = g_6 \& \bar{g}_1$	45	$g_4 = \bar{g}_6 \oplus \bar{g}_1$
24	$g_4 = g_6 \vee \bar{g}_7$		

Этап 5. Построение BDD для кофакторов, которые не являются реализуемыми. Нереализуемыми кофакторами являются $g_1, g_4, g_8, g_9, g_{10}$. Построение BDD для них приводит к перестановке $\langle x_4, x_5, x_3 \rangle$ переменных и следующим формулам разложения Шеннона:

$$g_1 = \bar{x}_4 w_0 \vee x_4 w_2; g_4 = \bar{x}_4 x_5 \vee x_4 w_0; g_8 = \bar{x}_4 w_2 \vee x_4 w_3; g_9 = \bar{x}_4 w_0 \vee x_4 w_5; \quad (10)$$

$$g_{10} = \bar{x}_4 x_3 \vee x_4 x_5; w_0 = \bar{x}_5 \bar{x}_3; w_1 = x_5; w_2 = x_5 x_3; w_3 = \bar{x}_3 \bar{x}_5 \vee x_3; w_4 = x_3; w_5 = \bar{x}_5$$

(сначала по переменной x_4 , затем по переменной x_5). На первом уровне – разложение по переменной x_3 . Кофакторы второго уровня BDD, зависящие от переменных x_5, x_3 , заданы в табл. 14.

Таблица 14

Кофакторы второго уровня BDD

$x_3 x_5$	w_0	$w_1 = x_5$	w_2	w_3	$w_4 = x_3$	$w_5 = \bar{x}_5$
0 0	1	0	0	1	0	1
0 1	0	1	0	0	0	0
1 0	0	0	0	1	1	1
1 1	0	1	1	1	1	0

Этап 6. Составление уравнений для нереализуемых кофакторов с использованием кофакторов нижнего уровня BDD. Составляются уравнения для нереализуемых кофакторов $g_1, g_4, g_8, g_9, g_{10}$ с использованием реализуемых кофакторов и кофакторов второго уровня, заданных в табл. 14. Для этого табл. 14 преобразуется в табл. 15; переменная x_4 является несущественной для кофакторов, приведенных в табл. 15. Полученные уравнения заданы в табл. 16.

Таблица 15

Кофакторы второго уровня BDD с несущественной переменной x_4

$x_3 x_4 x_5$	w_0	$w_1 = x_5$	w_2	w_3	$w_4 = x_3$	$w_5 = \bar{x}_5$
0 0 0	1	0	0	1	0	1
0 0 1	0	1	0	0	0	0
0 1 0	1	0	0	1	0	1
0 1 1	0	1	0	0	0	0
1 0 0	0	0	0	1	1	1
1 0 1	0	1	1	1	1	0
1 1 0	0	0	0	1	1	1
1 1 1	0	1	1	1	1	0

Этап 7. Расширение ациклического подграфа по новым уравнениям. В полученный на этапе 4 граф (рис. 7, б) последовательно (по одному уравнению) добавляются вершины и дуги, соответствующие уравнениям из табл. 16. Каждое из уравнений, кроме уравнения 54 ($g_9 = \bar{g}_{10} \& w_5$), преобразует граф в циклический. Добавление уравнения 54 оставляет граф ациклическим (рис. 8) с шестью реализуемыми кофакторами.

Таблица 16

Варианты алгебраического представления кофакторов второго уровня BDD

Номер уравнения	Уравнение
49	$g_{10} = g_3 \oplus w_2$
50	$g_{10} = g_3 \vee w_2$
51	$g_{10} = \bar{g}_3 \oplus \bar{w}_2$
52	$g_{10} = g_5 \vee w_2$
53	$g_8 = \bar{g}_2 \& w_3$
54	$g_9 = \bar{g}_{10} \& w_5$
55	$g_9 = \bar{g}_{10} \& w_3$
56	$g_9 = g_3 \oplus w_5$
57	$g_9 = \bar{g}_5 \oplus w_5$

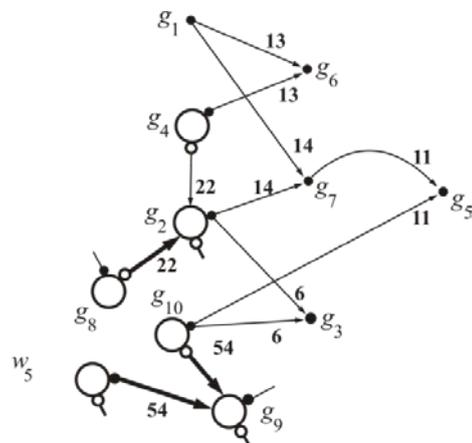


Рис. 8. Результирующий ациклический граф G

Реализуемыми являются шесть кофакторов: $g_3 = g_2 \& g_{10}$, $g_5 = g_7 \& g_{10}$, $g_6 = g_1 \vee g_4$, $g_7 = g_1 \& g_2$, $g_2 = \bar{g}_8 \& \bar{g}_4$, $g_9 = \bar{g}_{10} \& w_5$, где $w_5 = \bar{x}_5$. Уравнение $g_9 = \bar{x}_4 w_0 \vee x_4 w_5$ разложения Шеннона из списка (10) заменено алгебраическим выражением $g_9 = \bar{g}_{10} \& w_5$.

Этап 8. Корректировка многоуровневого BDD-представления векторной функции. После выполнения этапа 7 многоуровневое представление кофакторов третьего уровня BDD записывается в виде следующих уравнений, которым соответствует рис. 9:

$$\begin{aligned}
 g_3 &= g_2 \& g_{10}; & g_5 &= g_7 \& g_{10}; & g_6 &= g_1 \vee g_4; & g_7 &= g_1 \& g_2; & c_2 &= g_4 \vee g_8; & g_2 &= \bar{c}_2; \\
 c_{10} &= \bar{g}_{10}; & g_9 &= w_5 \& c_{10}; & g_1 &= \bar{x}_4 w_0 \vee x_4 w_2; & g_8 &= \bar{x}_4 w_2 \vee x_4 w_3; & g_{10} &= \bar{x}_4 x_3 \vee x_4 x_5; \\
 g_4 &= \bar{x}_4 x_5 \vee x_4 w_0; & w_0 &= \bar{x}_5 \bar{x}_3; & w_2 &= x_5 x_3; & w_3 &= \bar{x}_3 \bar{x}_5 \vee x_3; & w_5 &= \bar{x}_5.
 \end{aligned}
 \tag{11}$$

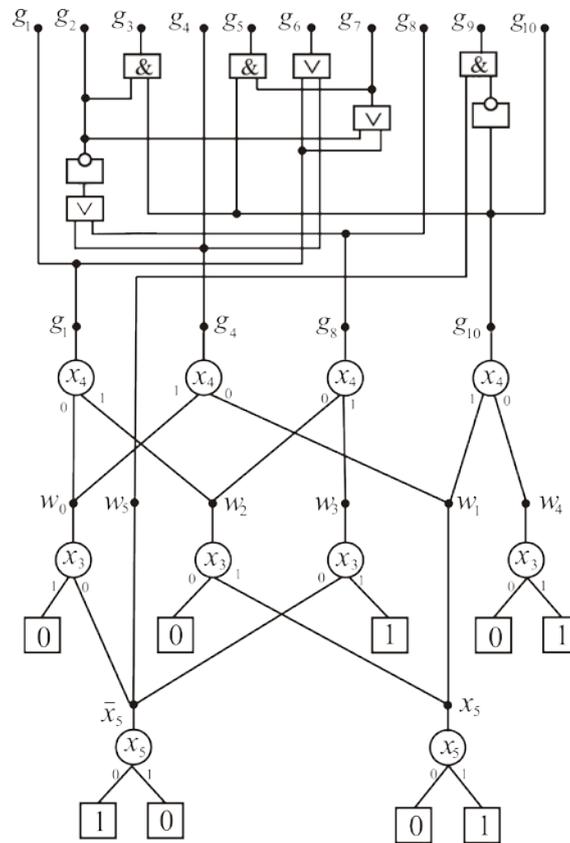


Рис. 9. Реализация подфункций g_1, \dots, g_{10} в виде BDD и алгебраических разложений

Нахождение алгебраических представлений кофакторов на втором уровне BDD. Для тривиальных кофакторов второго уровня, равных переменным либо их инверсиям, алгебраические представления не ищутся. Тривиальными кофакторами второго уровня BDD (см. табл. 14) являются $w_1 = x_5$, $w_4 = x_3$, $w_5 = \bar{x}_5$, нетривиальными – w_0, w_2 . Уравнения для w_0, w_2 , использующие кофакторы второго уровня BDD, заданы в табл. 17. Заметим, что кофактор w_3 не выражается через другие кофакторы, а w_0 представляет собой элементарную конъюнкцию: $w_0 = \bar{x}_3 \bar{x}_5$. Реализация в виде BDD кофакторов второго уровня не может быть упрощена за счет их алгебраических представлений.

Результирующее многоуровневое представление векторной булевой функции (см. табл. 1) содержит 86 литералов:

$$\begin{aligned}
 f_1 &= \bar{x}_1 h_1 \vee x_1 h_2; & f_2 &= \bar{x}_1 h_2 \vee x_1 h_3; & f_3 &= \bar{x}_1 h_4 \vee x_1 h_5; & f_4 &= \bar{x}_1 h_4 \vee x_1 h_6; & f_5 &= \bar{x}_1 h_6 \vee x_1 h_7; \\
 h_1 &= \bar{x}_2 \bar{g}_1 \vee x_2 \bar{g}_3; & h_2 &= \bar{x}_2 g_2 \vee x_2 \bar{g}_8; & h_3 &= \bar{x}_2 \bar{g}_5 \vee x_2 g_6; & h_4 &= \bar{x}_2 g_6 \vee x_2 \bar{g}_7; & h_5 &= \bar{x}_2 g_6 \vee x_2 g_{10}; \\
 h_6 &= \bar{x}_2 g_4 \vee x_2 g_9; & h_7 &= \bar{x}_2 g_8 \vee x_2 g_{10}; & g_3 &= g_2 \& g_{10}; & g_5 &= g_7 \& g_{10}; & g_6 &= g_1 \vee g_4; \\
 g_7 &= g_1 \vee g_2; & c_2 &= g_4 \vee g_8; & g_2 &= \bar{c}_2; & c_{10} &= \bar{g}_{10}; & g_9 &= w_5 \& c_{10}; & g_1 &= \bar{x}_4 w_0 \vee x_4 w_2; \\
 g_8 &= \bar{x}_4 w_2 \vee x_4 w_3; & g_{10} &= \bar{x}_4 x_3 \vee x_4 x_5; & g_4 &= \bar{x}_4 x_5 \vee x_4 w_0; & w_0 &= \bar{x}_5 \bar{x}_3; & w_2 &= x_5 x_3; & w_3 &= \bar{x}_3 \bar{x}_5 \vee x_3; & w_5 &= \bar{x}_5.
 \end{aligned}
 \tag{12}$$

В исходный граф BDD (см. рис. 1) внесены графические изображения логических уравнений реализуемых кофакторов и операторов инверсии, в результате чего получен рис. 10. При составлении результирующих уравнений следует иметь в виду тот факт, что два последовательно идущих инвертора можно удалить из схемы реализации кофакторов.

Таблица 17
Варианты алгебраического представления нетривиальных кофакторов второго уровня BDD

Номер уравнения	Уравнение
58	$w_0 = w_3 \oplus w_4$
59	$w_0 = w_3 \& \bar{w}_4$
60	$w_0 = \bar{w}_3 \oplus \bar{w}_4$
61	$w_0 = w_5 \& \bar{w}_4$
62	$w_2 = w_1 \oplus \bar{x}_3$
63	$w_2 = w_3 \oplus w_5$
64	$w_2 = w_3 \& w_1$
65	$w_2 = w_4 \& w_1$

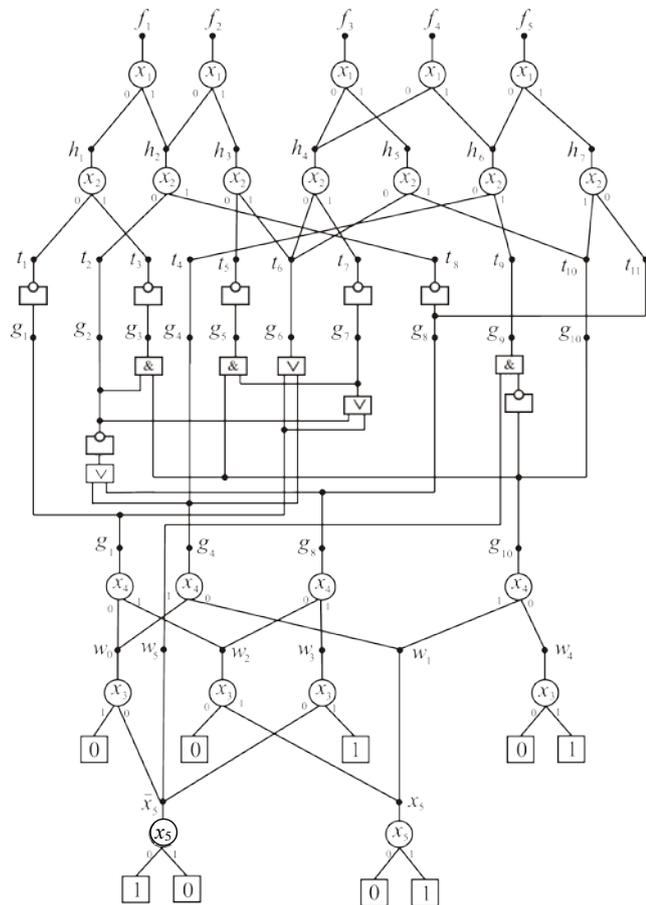


Рис. 10. Результирующий граф BDD

Предложенный метод оптимизации многоуровневых представлений векторных булевых функций может быть обобщен по нескольким направлениям.

Во-первых, может быть расширен список алгебраических выходных функций φ , по которым осуществляется разложение $g_p = \varphi(g_i, g_j)$ кофакторов. К выходным функциям φ , определяющим логические операции $\vee, \&, \oplus$, могут быть добавлены другие булевы функции, существенно зависящие от двух либо большего числа переменных.

Во-вторых, на этапах 6 и 7 метода уравнения могут составляться с применением не одного, а нескольких нижних уровней BDD.

В-третьих, желательно проверять, позволяет ли использование операции \oplus в качестве выходной функции алгебраического представления кофактора g_p сократить сложность BDD нижнего уровня (за счет исключения реализации на BDD кофактора g_p) на пять и более литералов, так как будем считать, что сложность реализации операции $g_p = g_i \oplus g_j = \bar{g}_i g_j \vee g_i \bar{g}_j$ составляет четыре литерала.

Поиск алгебраических представлений кофакторов может осуществляться в процессе построения BDD для векторной булевой функции, которая задана в виде системы ДНФ компонентных функций. (Методы, алгоритмы и программы построения BDD по системе ДНФ булевых функций описаны в работе [21].) В этом случае приходится проверять возможность алгебраического представления кофакторов, заданных произвольными (неортогонализированными) ДНФ, элементарные конъюнкции которых находятся в различных отношениях, таких как поглощение, склеивание, ортогональность, неортогональность [2]. При использовании метода, когда нижние уровни BDD еще не построены, этапы 6 и 7 метода будут отсутствовать. Метод станет более быстродействующим, однако качество решений может ухудшиться.

Оценки сложности функциональных описаний и логических схем для различных способов оптимизации. В табл. 18 для различных способов логической оптимизации одной и той же векторной булевой функции (см. табл. 1) даны оценки сложности функциональных и схемных реализаций (логических схем). Для получения минимизированной системы ДНФ (см. строку 1 табл. 18) использовалась программа [29], минимизация BDD-представлений осуществлялась во всех случаях программой из работы [30].

Таблица 18

Сравнение способов логической оптимизации

Функциональное описание	Сложность функционального описания				Сложность логической схемы из библиотечных элементов		
	α	d	k	P	Площадь	Число логических элементов	Задержка схемы, нс
1. Минимизированная система ДНФ (табл. 1), программа [29]	0	38	119	162	17 443	47	3,96
2. Булева сеть, полученная программой BoolNetOpt [31]	60	25	40	130	16 556	46	3,29
3. BDD по перестановке $\langle x_1, x_2, x_4, x_3, x_5 \rangle$, полученная программой [30]	26	28	55	114	14 793	41	3,62
4. BDD с нахождением инверсий кофакторов [24] по перестановке $\langle x_1, x_2, x_5, x_3, x_4 \rangle$, формулы (13)	21	23	47	96	15 797	44	2,88
5. Исходная BDD по перестановке $\langle x_1, x_2, x_5, x_3, x_4 \rangle$, формулы (2)	24	26	51	106	13 827	38	3,36
6. Алгебраическое представление кофакторов третьего уровня BDD без инверсий кофакторов и без операции \oplus , формулы (14)	21	21	42	89	14 486	42	3,66
7. Предложенный метод, формулы (12)	23	20	38	86	13 492	35	4,49

В строке 2 табл. 18 применялась предварительная логическая минимизация на основе аппарата булевых сетей [31]; в строке 3 – оптимизация BDD по перестановке $\langle x_1, x_2, x_4, x_3, x_5 \rangle$ с использованием программы [24], в строке 4 – оптимизация с использованием программы BDD-Builder, которая при построении BDD находит взаимно инверсные кофакторы на каждом уровне BDD и получает следующие формулы:

$$\begin{aligned}
 f_1 &= \bar{x}_1 s_6 \vee x_1 s_2; f_2 = \bar{x}_1 s_2 \vee x_1 s_3; f_3 = \bar{x}_1 s_0 \vee x_1 s_5; f_4 = \bar{x}_1 s_0 \vee x_1 s_1; f_5 = \bar{x}_1 s_1 \vee x_1 s_9; \\
 s_0 &= \bar{x}_2 s_{11} \vee x_2 s_{23}; s_1 = \bar{x}_2 s_{18} \vee x_2 s_{19}; s_2 = \bar{x}_2 s_{12} \vee x_2 s_{13}; s_3 = \bar{x}_2 s_{10} \vee x_2 s_{11}; \\
 s_5 &= \bar{x}_2 s_{11} \vee x_2 s_{15}; s_6 = \bar{x}_2 s_{20} \vee x_2 s_{21}; s_9 = \bar{x}_2 s_{13} \vee x_2 s_{15}; s_{10} = \bar{x}_5 s_{30} \vee x_5 \bar{x}_4; \\
 s_{11} &= \bar{x}_5 \bar{x}_3 \vee x_5 \bar{s}_{27}; s_{12} = \bar{x}_5 \bar{x}_4 \vee x_5 s_{27}; s_{13} = \bar{x}_5 \bar{x}_4 \vee x_5 \bar{x}_3; s_{15} = \bar{x}_5 s_{30} \vee x_5 s_{31}; \\
 s_{18} &= \bar{x}_5 s_{27} \vee x_5 \bar{x}_4; s_{19} = \bar{x}_5 s_{30}; s_{20} = \bar{x}_5 s_{31} \vee x_5 s_{42}; s_{21} = \bar{x}_5 s_{30} \vee x_5 s_{27}; \\
 s_{23} &= \bar{x}_5 x_4 \vee x_5 \bar{x}_4; s_{27} = \bar{x}_3 x_4; s_{30} = x_3 \bar{x}_4; s_{31} = \bar{x}_3 x_4 \vee x_3; s_{42} = \bar{x}_3 \vee x_3 \bar{x}_4.
 \end{aligned} \tag{13}$$

Строка 5 в табл. 18 соответствует оценке сложности функционального описания согласно формулам (2), строка 6 – формулам (14), полученным в результате применения предложенного в статье метода без поиска инверсий кофакторов и без использования операции \oplus :

$$\begin{aligned}
 f_1 &= \bar{x}_1 h_1 \vee x_1 h_2; f_2 = \bar{x}_1 h_2 \vee x_1 h_3; f_3 = \bar{x}_1 h_4 \vee x_1 h_5; f_4 = \bar{x}_1 h_4 \vee x_1 h_6; f_5 = \bar{x}_1 h_6 \vee x_1 h_7; \\
 h_1 &= \bar{x}_2 \bar{g}_1 \vee x_2 \bar{g}_3; h_2 = \bar{x}_2 g_2 \vee x_2 \bar{g}_8; h_3 = \bar{x}_2 \bar{g}_5 \vee x_2 g_6; h_4 = \bar{x}_2 g_6 \vee x_2 \bar{g}_7; h_5 = \bar{x}_2 g_6 \vee x_2 g_{10}; \\
 h_6 &= \bar{x}_2 g_4 \vee x_2 g_9; h_7 = \bar{x}_2 g_8 \vee x_2 g_{10}; g_1 = \bar{x}_4 r_0 \vee x_4 r_2; g_2 = \bar{x}_4 \bar{x}_5 \vee x_4 r_1; g_3 = g_2 \& g_{10}; \\
 g_7 &= g_1 \vee g_2; g_6 = g_1 \vee g_4; g_5 = g_7 \& g_{10}; g_9 = \bar{x}_4 r_0 \vee x_4 \bar{x}_5; g_8 = \bar{x}_4 r_2 \vee x_4 r_3; \\
 g_{10} &= \bar{x}_4 x_3 \vee x_4 x_5; g_4 = \bar{x}_4 x_5 \vee x_4 r_0; r_0 = \bar{x}_3 \bar{x}_5; r_1 = \bar{x}_3 x_5; r_2 = x_3 x_5; r_3 = \bar{x}_3 \bar{x}_5 \vee x_3.
 \end{aligned} \tag{14}$$

Все полученные представления векторной функции были записаны на языке VHDL [32], затем по данным пяти VHDL-описаний были синтезированы логические схемы в одной и той же библиотеке логических элементов и одинаковых установках опций, управляющих процессом синтеза в синтезаторе LeonardoSpectrum [33].

В табл. 18 используются следующие обозначения: α – число внутренних переменных в функциональном описании, d – число двухоперандных операций дизъюнкции, k – число двухоперандных операций конъюнкции, P – число литералов в функциональном описании. Площадь схемы (суммарная площадь всех элементов) подсчитывалась в условных единицах. Лучшие решения отмечены жирным шрифтом.

Для рассмотренного примера векторной булевой функции применение предложенного в статье метода позволяет получать как более простое функциональное описание, так и менее сложную (по площади) логическую схему, синтезируемую по этому описанию. Рост числа уровней в минимизированном функциональном описании может привести в результате синтеза схемы к увеличению ее задержки, что соответствует общей закономерности, связывающей площади логических схем и их задержки при синтезе многоуровневых логических схем. Сравним (табл. 18) лучшие решения по задержке (строка 4) и площади (строка 7): схема с площадью 13 492 выигрывает 17 % по площади у схемы с площадью 15 797, имеющей задержку 2,88 нс, и проигрывает 36 % по быстродействию, так как задержка схемы с меньшей площадью составляет 4,49 нс.

Заключение. Предложенный в статье метод позволяет дополнительно упрощать многоуровневые BDD-представления систем полностью определенных булевых функций, используя алгебраические разложения подфункций, находящихся на одном уровне BDD, в виде двухоперандных дизъюнкции, конъюнкции и сумм по модулю два. Метод легко обобщается на случай исходного задания функций в виде систем ДНФ и на случай других видов $g_p = \varphi(g_i, g_j)$ разложений подфункций g_p , когда в качестве выходных используются булевы функции φ , существенно зависящие от двух и большего числа переменных. Дополнительная оптимизация по числу литералов в логических уравнениях, соответствующих многоуровневым представлениям систем булевых функций, приводит к более простым функциональным описаниям, по которым осуществляется синтез комбинационных логических схем в различных технологических базисах. Однако уменьшение числа литералов в функциональных описаниях, позволяющее со-

кращать площадь схем, может приводить к увеличению числа логических уровней в минимизированных описаниях и увеличению временных задержек схем.

Вклад авторов. Бибило П. Н. – научное руководство исследованием, написание текста, утверждение окончательного варианта статьи для публикации, согласие нести ответственность за все аспекты работы; Романов В. И. – программная реализация алгоритмов, проведение вычислительных экспериментов, интерпретация полученных результатов.

Список использованных источников

1. Logic Minimization Algorithm for VLSI Synthesis / K. R. Brayton [et al.]. – Boston : Kluwer Academic Publishers, 1984. – 193 p.
2. Закревский, А. Д. Логический синтез каскадных схем / А. Д. Закревский. – М. : Наука, 1981. – 416 с.
3. Брейтон, Р. К. Синтез многоуровневых комбинационных логических схем / Р. К. Брейтон, Г. Д. Хэчтел, А. Л. Санджованни-Винчензелли // ТИИЭР. – 1990. – Т. 78, № 2. – С. 38–83.
4. Тарасов, И. Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования / И. Е. Тарасов. – М. : Горячая линия – Телеком, 2020. – 538 с.
5. Синтез асинхронных автоматов на ЭВМ / под ред. А. Д. Закревского. – Минск : Наука и техника, 1975. – 184 с.
6. Brayton, K. R. Factoring logic functions / K. R. Brayton // IBM J. of Research and Development. – 1987. – Vol. 31, no. 2. – P. 187–198.
7. MIS: a multiple-level logic optimization systems / R. K. Brayton [et al.] // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1987. – Vol. 6, iss. 6. – P. 1062–1081.
8. Boolean decomposition in multilevel logic optimization / S. Devadas [et al.] // IEEE J. of Solid-State Circuits. – 1989. – Vol. 24, no. 2. – P. 399–407.
9. Brayton, R. Boolean factoring and decomposition of logic networks / R. Brayton, A. Mishchenko, S. Chatterjee; eds.: S. P. Khatri, K. Gulati // Advanced Techniques in Logic Synthesis, Optimizations and Applications. – Springer, 2011. – P. 47–66.
10. Curtis, H. A. A New Approach to the Design of Switching Circuit / H. A. Curtis. – Princeton, Van Nostrand, 1962. – 635 p.
11. Scholl, C. Functional Decomposition with Applications to FPGA Synthesis / C. Scholl. – Kluwer Academic Publishers, 2001. – 288 p.
12. Поттосин, Ю. В. Табличные методы декомпозиции систем полностью определенных булевых функций / Ю. В. Поттосин, Е. А. Шестаков. – Минск : Беларус. навука, 2006. – 327 с.
13. Бибило, П. Н. Декомпозиция булевых функций на основе решения логических уравнений / П. Н. Бибило. – Минск : Беларус. навука, 2009. – 211 с.
14. Lin, H.-P. Ashenurst decomposition using SAT and interpolation / H.-P. Lin, J.-H. R. Jiang, R.-R. Lee; eds.: S. P. Khatri, K. Gulati // Advanced Techniques in Logic Synthesis, Optimizations and Applications. – Springer, 2011. – P. 67–86.
15. 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.
16. Drechsler, R. Binary Decision Diagrams: Theory and Implementation / R. Drechsler, B. Becker. – Springer, 1998. – 210 p.
17. Ebdndt, R. Advanced BDD Optimization / R. Ebdndt, G. Fey, R. Drechsler. – Springer, 2005. – 222 p.
18. 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.
19. Meinel, C. Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications / C. Meinel, T. Theobald. – Berlin, Heidelberg : Springer-Verlag, 1998. – 267 p.
20. Кнут, Д. Э. Искусство программирования. Т. 4, А. Комбинаторные алгоритмы. Ч. 1 : пер. с англ. / Д. Э. Кнут. – М. : Вильямс, 2013. – 960 с.
21. Бибило, П. Н. Применение диаграмм двоичного выбора при синтезе логических схем / П. Н. Бибило. – Минск : Беларус. навука, 2014. – 231 с.
22. Уткин, А. А. Анализ логических сетей и техника булевых вычислений / А. А. Уткин. – Минск : Наука и техника, 1979. – 152 с.
23. Романов, В. И. Программные средства для решения логико-комбинаторных задач / В. И. Романов // Информатика. – 2005. – № 4(8). – С. 114–123.

24. 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.
25. Goldberg, E. BerkMin: a fast and robust SAT-solver / E. Goldberg, Y. Novikov // Discrete Applied Mathematics. – 2007. – Vol. 155, no. 12. – P. 1549–1561.
26. Handbook of Satisfiability / eds.: A. Biere, M. Heule, H. van Maaren, T. Walsh. – IOS Press, 2009. – 980 p.
27. Романовский, И. В. Дискретный анализ : учебное пособие / И. В. Романовский. – 4-е изд., испр. и доп. – СПб. : Невский Диалект, БХВ-Петербург, 2008. – 336 с.
28. Бибило, П. Н. Синтез комбинационных схем методами функциональной декомпозиции / П. Н. Бибило, С. В. Енин. – Минск : Наука и техника, 1987. – 189 с.
29. Торопов, Н. Р. Минимизация систем булевых функций в классе ДНФ / Н. Р. Торопов // Логическое проектирование. – Минск : Ин-т техн. кибернетики НАН Беларуси, 1999. – Вып. 4. – С. 4–19.
30. Бибило, П. Н. Минимизация многоуровневых представлений систем булевых функций на основе разложения Шеннона / П. Н. Бибило, Ю. Ю. Ланкевич // Информатика. – 2017. – № 2(54). – С. 45–57.
31. 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.
32. Ashenden, P. J. VHDL–2008. Just the New Stuff / P. J. Ashenden, J. Lewis. – Burlington, MA, USA : Morgan Kaufman Publishers, 2008. – 909 p.
33. Бибило, П. Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum / П. Н. Бибило. – М. : СОЛОН-Пресс, 2005. – 384 с.

References

1. 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.
2. Zakrevskij A. D. Logicheskij sintez kaskadnyh skhem. *Logical Synthesis of Cascading Circuit*. Moscow, Nauka, 1981, 416 p. (In Russ.).
3. Brayton R. K., Hachtel G. D., Sangiovanni-Vincentelli A. L. *Synthesis of multi-level combinational logic circuits*. Trudy Institute inzhenerov po jelektronike i radiotekhnike [Proceedings of the Institute of Electronics and Radio Engineering], 1990, vol. 78, no. 2, pp. 38–83 (In Russ.).
4. 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.
5. Sintez asinhronnyh avtomatov na EHVM. *Synthesis of Asynchronous Automata on a Computer*. In Zakrevskogo A. D. (ed.). Minsk, Nauka i tekhnika, 1975, 184 p.
6. Brayton K. R. Factoring logic functions. *IBM Journal of Research and Development*, 1987, vol. 31, no. 2, pp. 187–198.
7. Brayton R. K., Rudell R., Sangiovanni-Vincentelli A. L., Wang A. R. MIS: a multiple-level logic optimization systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1987, vol. 6, iss. 6, pp. 1062–1081.
8. Devadas S., Wang A. R., Newton A. R., Sangiovanni-Vincentelli A. Boolean decomposition in multilevel logic optimization. *IEEE Journal of Solid-State Circuits*, 1989, vol. 24, no. 2, pp. 399–407.
9. Brayton R., Mishchenko A., Chatterjee S. Boolean factoring and decomposition of logic networks. In Khatri S. P., Gulati K. (eds.). *Advanced Techniques in Logic Synthesis, Optimizations and Applications*. Springer, 2011, pp. 47–66.
10. Curtis H. A. *A New Approach to the Design of Switching Circuit*. Princeton, Van Nostrand, 1962, 635 p.
11. Scholl C. *Functional Decomposition with Applications to FPGA Synthesis*. Kluwer Academic Publishers, 2001, 288 p.
12. Pottosin Yu. V., Shestakov E. A. Tablichnye metody dekompozicii sistem polnost'yu opredelennyh bulevykh funkciy. *Tabular methods for decomposition of systems of completely defined Boolean functions*. Minsk, Belaruskaja navuka, 2006, 327 p. (In Russ.).
13. Bibilo P. N. Dekompoziciya bulevykh funkciy na osnove resheniya logicheskikh uravnenij. *Decomposition of Boolean Functions Based on the Solution of Logical Equations*. Minsk, Belaruskaja navuka, 2009, 211 p. (In Russ.).

14. Lin H.-P., Jiang J.-H. R., Lee R.-R. Ashenurst decomposition using SAT and interpolation. In Khatri S. P., Gulati K. (eds.). *Advanced Techniques in Logic Synthesis, Optimizations and Applications*. Springer, 2011, pp. 67–86.
15. Bryant R. E. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, 1986, vol. 35, no. 8, pp. 677–691.
16. Drechsler R., Becker B. *Binary Decision Diagrams: Theory and Implementation*. Springer, 1998, 210 p.
17. Ebdndt R., Fey G., Drechsler R. *Advanced BDD Optimization*. Springer, 2005, 222 p.
18. 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.
19. Meinel C., Theobald T. *Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications*. Berlin, Heidelberg, Springer-Verlag, 1998, 267 p.
20. Knuth D. E. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Addison-Wesley Professional, 2011, 912 p.
21. Bibilo P. N. Primenenie diagram dvoichnogo vybora pri sinteze logicheskikh shem. *Application of Binary Selection Diagrams in the Synthesis of Logic Circuits*. Minsk, Belaruskaja navuka, 2014, 231 p. (In Russ.).
22. Utkin A. A. Analiz logicheskikh setej i tekhnika bulevykh vychislenij. *The analysis of logical networks and the technique of Boolean calculations*. Minsk, Nauka i tekhnika, 1979, 152 p. (In Russ.).
23. Romanov V. I. *Software tools for the solution of the logical-combinatorial task*. Informatika [Informatics], 2005, no. 4(8), pp. 114–123 (In Russ.).
24. Bibilo P. N., Lankevich Yu. Yu. *The use of Zhegalkin polynomials in minimizing multilevel representations of systems of Boolean functions based on the Shannon expansion*. Programmnyaya inzheneriya [Software Engineering], 2017, no. 8, pp. 369–384 (In Russ.).
25. Goldberg E., Novikov Y. BerkMin: a fast and robust SAT-solver. *Discrete Applied Mathematics*, 2007, vol. 155, no. 12, pp. 1549–1561.
26. *Handbook of Satisfiability*. In Biere A., Heule M., Van Maaren H., Valsh T. (eds.). IOS Press, 2009, 980 p.
27. Romanovskij I. V. Diskretnyj analiz. *Discrete Analysis*. Saint Petersburg, Nevskij Dialekt, BHV-Peterburg, 2008, 336 p. (In Russ.).
28. Bibilo P. N., Enin S. V. Sintez kombinacionnykh skhem metodami funkcional'noj dekompozicii. *Synthesis of Combination Circuits by Methods of Functional Decomposition*. Minsk, Nauka i tekhnika, 1987, 189 p. (In Russ.).
29. Toropov N. R. *Minimization of systems of Boolean functions in the class DNF*. Logicheskoe proektirovanie [Logical Design]. Minsk, Institut tehniceskoy kibernetiki Nacional'noj akademii nauk Belarusi, 1999, iss. 4, pp. 4–19 (In Russ.).
30. Bibilo P. N., Lankevich Yu. Yu. *Minimizing multilevel representations of systems of Boolean functions based on Shannon expansion*. Informatika [Informatics], 2017, no. 2(54), pp. 45–57 (In Russ.).
31. 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.).
32. Ashenden P. J., Lewis J. *VHDL–2008. Just the New Stuff*. Burlington, MA, USA, Morgan Kaufman Publishers, 2008, 909 p.
33. Bibilo P. N. Cistemy proektirovaniya integral'nykh 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.).

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

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

Information about the authors

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

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