

П. Н. Бибилло, д-р техн. наук, проф., зав. лаб., bibilo@newman.bas-net.by,
В. И. Романов, вед. науч. сотр., доц., rom@newman.bas-net.by,
Объединенный институт проблем информатики Национальной академии наук Беларуси,
Минск

Экспериментальное исследование алгоритмов минимизации BDDI-представлений систем булевых функций с использованием алгебраических разложений кофакторов

Рассматриваются многоуровневые разложения Шеннона систем полностью определенных булевых функций, графической формой задания которых являются модификации бинарных диаграмм решений (BDD - Binary Decision Diagram), называемые BDDI (Binary Decision Diagram with Inverse cofactors). В отличие от BDD, в BDDI могут быть пары взаимно инверсных подфункций (кофакторов). Описываются программно реализованные алгоритмы дополнительной логической оптимизации BDDI-представлений систем булевых функций на основе поиска алгебраических разложений кофакторов одного уровня BDDI в виде дизъюнкции либо конъюнкции других инверсных либо безынверсных кофакторов данного уровня BDDI. Приведены результаты применения соответствующих программ при синтезе логических схем в библиотеке проектирования заказных сверхбольших интегральных схем.

Ключевые слова: система булевых функций, дизъюнктивная нормальная форма (ДНФ), Binary Decision Diagram (BDD), разложение Шеннона, синтез логической схемы, VHDL, СБИС

Введение

Автоматизация логического проектирования функциональных блоков комбинационной логики, входящих в состав заказных цифровых СБИС (сверхбольших интегральных схем), по-прежнему остается актуальной научной проблемой, так как размерности решаемых задач возрастают, появляются более жесткие требования на энергопотребление логических схем, изменяются библиотеки логических элементов. Логическое проектирование комбинационных схем в библиотеках проектирования заказных СБИС традиционно разбивается на два этапа — логическую оптимизацию и технологическое отображение, на котором оптимизированные логические функциональные описания проектируемых логических схем покрываются функциональными описаниями логических элементов. Логическая оптимизация, часто называемая технологически независимой оптимизацией, обычно не ориентируется на используемый базис (библиотеку) логических элементов, а стремится уменьшить число литералов и число двухоперандных операций дизъюнкции и конъюнкции в минимизированных алгебраических представлениях систем полностью определенных булевых функций. В практике проектирования давно было замечено [1, с. 44],

что сокращение числа литералов и логических операций в функциональных описаниях положительно влияет на сокращение в схеме числа элементов технологической библиотеки (базиса синтеза), т. е. по сути, на сокращение суммарного числа транзисторов, входящих во все элементы схемы. Уменьшение числа транзисторов позволяет уменьшать энергопотребление схем.

Для логической минимизации в настоящее время широко используются методы оптимизации многоуровневых представлений булевых функций на основе разложения Шеннона, графические формы таких представлений, часто используемые как структуры данных, называются бинарными диаграммами решений [2–7]. В работе [8] был предложен метод дополнительной минимизации BDD-представлений систем булевых функций, основанный на поиске дизъюнктивных, конъюнктивных и модулярных разложений подфункций, входящих в оптимизированные BDD-представления. Метод позволяет заменять формулы разложения Шеннона двухоперандными формулами дизъюнкции $g_3 = g_1 \vee g_2$, конъюнкции $g_3 = g_1 \& g_2$ либо модулярными формулами вида $g_3 = g_1 \oplus g_2$ (здесь g_1, g_2, g_3 — кофакторы одного и того же уровня BDDI). Такая замена уменьшает число литералов и логических операций в резуль-

тирующих логических формулах, по которым выполняется технологическое отображение, т. е. заключительный этап синтеза схемы. В настоящей работе метод [8] доведен до алгоритмов и программ, эффективность которых исследована на потоках примеров, входящих в известную библиотеку примеров, используемых для проверки и сравнения программ логической оптимизации и синтеза логических схем. Для разложения кофакторов используются только дизъюнктивные и конъюнктивные разложения, причем кофакторы в таких разложениях могут быть как в инверсной, так и безынверсной форме.

Бинарные диаграммы решений и алгебраические разложения кофакторов

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

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

Функции $f_0 = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f_1 = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ в правой части (1) называются *кофакторами* (*cofactors*, англ.) разложения по переменной x_i . Они получаются из функции f подстановкой вместо переменной x_i константы 0 и 1 соответственно. Каждый из кофакторов f_0 и f_1 может быть разложен по одной из переменных из множества $\{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\}$. Процесс разложения кофакторов заканчивается, когда все n переменных будут использованы для разложения, либо когда все кофакторы вырождаются до констант 0, 1. На каждом шаге разложения выполняется сравнение на равенство полученных кофакторов.

Пусть $\mathbf{f}(\mathbf{x})$ — упорядоченная система полностью определенных компонентных булевых функций $f_i(\mathbf{x})$, $i = 1, \dots, m$ (векторная булева функция $\mathbf{f} = (f_1, \dots, f_m)$): $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Под BDD-представлением векторной булевой функции $\mathbf{f}(\mathbf{x})$ понимается ориентированный ациклический граф, задающий последовательные разложения Шеннона всех компонентных функций $f_i(\mathbf{x})$, $i = 1, \dots, m$, по всем ее переменным x_1, \dots, x_n при одном и том же заданном порядке (перестановке) переменных, по которым проводятся разложения. Рассматриваемые в статье BDD соответствуют совместным (*shared*) сокращенным упорядоченным BDD для систем функций (*Reduced Ordered BDD*, *ROBDD*) [5, с. 18]. В совместных ROBDD функциональным вершинам BDD соответствуют кофакторы, общие для функций системы, при этом функциональные вершины лишь подразумеваются (отождествляются с вершинами-переменными). Описание OBDD (упорядоченных BDD) дано в работах [2, с. 90, 5, с. 16], ROBDD — в работе [3, с. 243]. Далее под BDD будем понимать совместные ROBDD для систем функций (векторных функций).

BDD-представлению соответствуют формулы разложения Шеннона, каждой функциональной вершине BDD [6, с. 18] соответствует своя формула,

в которой оба кофактора выступают в безынверсной форме. По BDD-представлению можно найти задание каждой из компонентных функций $f_i(\mathbf{x})$ в виде двух ортогонализированных ДНФ (дизъюнктивных нормальных форм): одна из таких ДНФ задает область $M_{f_i}^1$ единичных значений функции $f_i(\mathbf{x})$, другая ДНФ — область $M_{f_i}^0$ нулевых значений. Аналогично можно найти представления в виде ДНФ каждого из кофакторов. Данные переходы подробно описаны в работе [6, с. 37]. На рис. 1, а изображена BDD, которой соответствуют следующие взаимосвязанные уравнения разложения Шеннона:

$$\begin{aligned} f_1 &= \bar{x}_3 g_1 \vee x_3 x_1; & f_2 &= \bar{x}_3 g_2 \vee x_3 g_4; \\ f_3 &= \bar{x}_3 g_5 \vee x_3 g_3; & f_4 &= \bar{x}_3 \vee x_3 g_6; \\ g_1 &= \bar{x}_1 s_1 \vee x_1 s_2; & g_2 &= \bar{x}_1 s_1; & g_3 &= x_1 s_2; \\ g_4 &= \bar{x}_1 s_2; & g_5 &= x_1 s_1; & g_6 &= \bar{x}_1 s_2 \vee x_1 s_1; \\ s_1 &= x_2; & s_2 &= \bar{x}_2. \end{aligned}$$

Кофакторы g_1, \dots, g_6 находятся на втором уровне BDD и зависят от двух переменных x_1, x_2 , кофакторы на первом уровне зависят от переменной x_2 : $s_1 = x_2$, $s_2 = \bar{x}_2$ (см. рис. 1, а). На нижнем уровне BDD всегда находятся константы 0, 1, которые могут дублироваться для упрощения изображения графа BDD на рисунке. На рис. 1, б показан граф той же BDD в символической, принятой в зарубежной литературе: штриховые линии, имеющие на рис. 1, соответствующие пометкам 0, сплошные — пометкам 1. По графу BDD легко найти кофакторы, находящиеся на одном уровне BDD, а от задания кофактора (либо функции исходной системы) в виде ДНФ можно перейти к его заданию в виде СДНФ (совершенной ДНФ) либо таблицы истинности. Рассмотрим кофакторы $g_1, g_2, g_3, g_4, g_5, g_6$ второго уровня BDD (рис. 1, а), элиминируем промежуточные переменные, получим совершенные ДНФ:

$$\begin{aligned} g_1 &= \bar{x}_1 s_1 \vee x_1 s_2 = \bar{x}_1 x_2 \vee x_1 \bar{x}_2; \\ g_2 &= \bar{x}_1 s_1 = \bar{x}_1 x_2; & g_3 &= x_1 s_2 = x_1 \bar{x}_2; \\ g_4 &= \bar{x}_1 s_2 = \bar{x}_1 \bar{x}_2; & g_5 &= x_1 s_1 = x_1 x_2; \\ g_6 &= \bar{x}_1 s_2 \vee x_1 s_1 = \bar{x}_1 \bar{x}_2 \vee x_1 x_2. \end{aligned}$$

Зададим СДНФ кофакторов и их инверсий таблицей истинности (табл. 1).

По таблицам истинности легко проверить представление кофакторов одного и того же уровня BDD в виде *алгебраических разложений*:

дизъюнктивного

$$g_p = g_i \vee g_j \quad (2)$$

либо *конъюнктивного*

$$g_r = g_i \& g_j. \quad (3)$$

Можно также использовать инверсии \bar{g}_i, \bar{g}_j кофакторов g_i, g_j в разложениях вида (2), (3). Для кофакто-

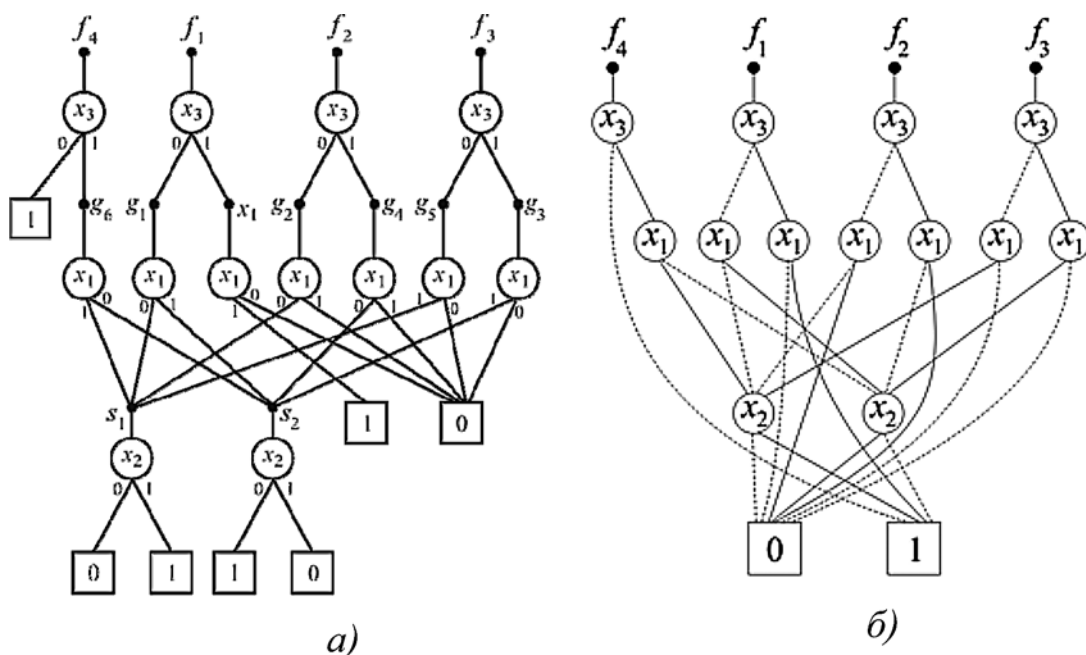


Рис. 1. BDD:

a — с выделенными кофакторами; b — традиционное изображение

Таблица 1

Кофакторы второго уровня BDD (рис. 1)

$x_1 x_2$	g_1	g_2	g_3	g_4	g_5	g_6	\bar{g}_1	\bar{g}_2	\bar{g}_3	\bar{g}_4	\bar{g}_5	\bar{g}_6
0 0	0	0	0	1	0	1	1	1	1	0	1	0
0 1	1	1	0	0	0	0	0	0	0	1	1	1
1 0	1	0	1	0	0	0	0	1	0	1	1	1
1 1	0	0	0	0	1	1	1	1	1	1	0	0

ров, которые имеют в своей записи четыре литерала (это кофакторы g_1, g_6), запишем уравнения их алгебраических разложений через другие кофакторы:

$$g_1 = \bar{g}_4 \& \bar{g}_5; \quad g_1 = g_2 \vee g_3; \quad g_6 = \bar{g}_2 \& \bar{g}_3; \\ g_6 = g_4 \vee g_5$$

и представим полученные уравнения в виде орграфа (рис. 2).

Легко видеть (рис. 3), что кофакторы $g_1 = \bar{x}_1 s_1 \vee x_1 s_2$, $g_6 = \bar{x}_1 s_2 \vee x_1 s_1$ могут быть выражены формулами алгебраических разложений через два кофактора g_2, g_3 : $g_1 = g_2 \vee g_3$; $g_6 = \bar{g}_2 \& \bar{g}_3$. Замена формул $g_1 = \bar{x}_1 s_1 \vee x_1 s_2$, $g_6 = \bar{x}_1 s_2 \vee x_1 s_1$ более простыми позволяет сокращать число литералов и логических операций в функциональном описании BDD (см. рис. 1, a).

Заметим, что кофакторы g_1, g_6 являются взаимно инверсными, поэтому далее в качестве исходных рассматриваются формулы разложений Шеннона, соответствующие BDDI. Под BDDI (*Binary Decision Diagram with Inverse cofactors*) далее понимается ориентированный ациклический граф, задающий последовательные разложения Шеннона булевой функции

$f(x_1, \dots, x_n)$ по всем ее переменным x_1, \dots, x_n при заданном порядке (перестановке) переменных, по которым проводятся разложения, при условии нахождения пар взаимно инверсных кофакторов [9].

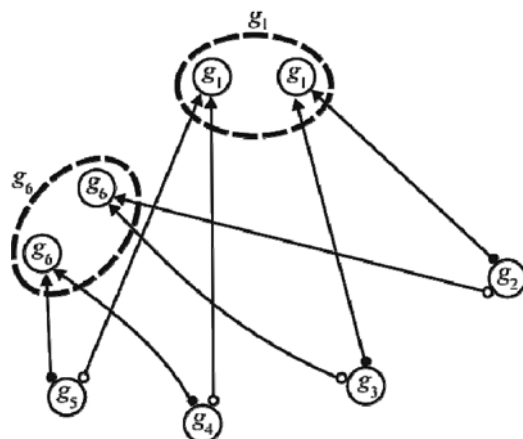


Рис. 2. Задание уравнений в виде орграфа

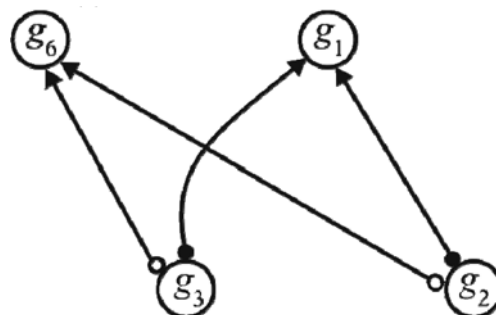
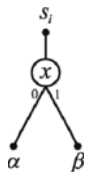
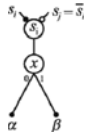


Рис. 3. Сокращенный бесконтурный орграф

Таблица 2

Соответствие графических элементов BDDI и логических формул

Графический элемент	Формула
	$s_i = \bar{x}\alpha \vee x\beta$
	$s_i = \bar{x}\alpha \vee x\beta$ $s_j = \bar{s}_i$

Графы BDDI содержат четыре вида вершин: корневые вершины, соответствующие функциям; функциональные вершины, соответствующие парам разлагаемых взаимно инверсных кофакторов (один из элементов пары может отсутствовать, если такового нет среди кофакторов на данном уровне BDDI); вершины-переменные; листовые вершины, соответствующие константам 0, 1. Функциональная вершина BDDI (табл. 2) реализует один кофактор s_i либо два кофактора — s_i и инверсию \bar{s}_i . Формула $s_i = \bar{x}\alpha \vee x\beta$ в правом столбце табл. 2 является формулой (1) разложения Шеннона кофактора s_i по переменной x : $\alpha = f_0$, $\beta = f_1$.

Заметим, что в BDD функциональная вершина реализуют один кофактор, взаимно инверсные кофакторы представляются парой вершин. BDDI соответствуют совместным ROBDD, в которых пара взаимно инверсных вершин-кофакторов задается одной функциональной вершиной. В уравнениях разложений Шеннона, соответствующих BDDI, кофакторы могут быть как со знаком инверсии, так и в безынверсной форме.

Далее предлагаются алгоритмы нахождения максимальных множеств алгебраических разложений вида (2), (3) кофакторов, находящихся на одном уровне BDDI. Основная проблема заключается в том, что при задании алгебраических уравнений в виде орграфа могут возникать контуры, что недопустимо при схемной реализации комбинационной логики. В рассматриваемом простом примере орграфа (см. рис. 2), иллюстрирующем задание уравнений, контуров нет. Переход к результирующему бесконтурному орграфу (см. рис. 3), в котором каждый кофактор представляется только одним уравнением алгебраического разложения, является сложной комбинаторной задачей для графов, содержащих сотни и тысячи вершин, и в которых может быть огромное число контуров, от которых надо избавляться. Предварительное нахождение пар взаимно инверсных кофакторов (использование BDDI вместо BDD) сокращает комбинаторный перебор при нахождении вариантов алгебраических разложений кофакторов, далее это будет пояснено.

Алгоритм 1 нахождения алгебраических разложений кофакторов в BDDI

Исходными данными для предлагаемого алгоритма 1 нахождения алгебраических представлений кофакторов являются граф BDDI, представляющий исходную векторную полностью определенную булеву функцию, и логические уравнения (формулы разложения Шеннона), соответствующие функциональным вершинам BDDI и задающие многоуровневое описание компонентных функций. Данные уравнения легко могут быть записаны по графу BDDI.

Предлагаемый алгоритм включает этапы, выполняемые для кофакторов каждого уровня BDDI, за исключением корневого, листового и первого. На первом уровне BDDI располагаются кофакторы, получаемые в результате разложения по последней переменной в заданной перестановке переменных, по которым ведется разложение Шеннона. Кофакторы первого уровня BDDI зависят от одной переменной, кофакторы второго уровня зависят от не более чем двух переменных и т. д. В алгоритме уровни BDDI рассматриваются "сверху вниз", т. е. от кофакторов, получаемых при разложении по первой переменной, и заканчивая кофакторами второго уровня BDD. Алгоритм ориентирован на замену наибольшего числа формул разложения Шеннона формулами дизъюнкции либо конъюнкции, что позволяет сокращать общее число литералов в многоуровневом задании системы булевых функций. Для рассматриваемого уровня BDDI решается задача нахождения наибольшего числа кофакторов, которые представимы в виде дизъюнкции либо конъюнкции других кофакторов данного уровня (ранее такие представления были названы алгебраическими разложениями).

Этап 1. Разбить множество уравнений по уровням BDDI.

Этап 2. Для каждого уровня BDDI (за исключением корневого, листового и первого) найти максимальное множество реализуемых уравнений (реализуемых кофакторов).

Выполнение этапа 2 сводится к следующим шагам.

Шаг 2.1. Элиминировать промежуточные переменные и построить таблицу истинности кофакторов заданного (рассматриваемого) уровня BDDI.

Шаг 2.2. Добавить в таблицу истинности инверсии кофакторов.

Шаг 2.3. Найти все варианты алгебраических представлений безынверсных кофакторов в виде двухоперандных дизъюнкций либо конъюнкций других кофакторов (либо их инверсий) рассматриваемого уровня BDDI. В результате один и тот же кофактор, назовем его *представимым*, может быть представлен различными уравнениями.

Шаг 2.4. Найти множество реализуемых кофакторов.

Из множества представимых кофакторов выделяется максимальное по мощности подмножество *реализуемых* кофакторов, для этого из множества уравнений алгебраического представления одного и того же кофактора выбирается одно уравнение,

либо уравнение не выбирается. Этот выбор определяется решением задачи 1, которая сформулирована в терминах теории графов.

Эман 3. Скорректировать исходное многоуровневое BDDI-представление системы булевых функций.

Исходное множество формул разложений Шеннона изменяется — формулы разложения Шеннона реализуемых кофакторов заменяются найденными формулами дизъюнкции либо конъюнкции.

Оценим число перебираемых вариантов на шаге 2.3. Пусть k — число безынверсных кофакторов g_1, \dots, g_k . Тогда требуется рассмотреть $12 \times C_k^3$ (C_k^3 — число сочетаний из k кофакторов по 3, $k \geq 3$) вариантов проверки дизъюнктивных разложений: число всех различных неупорядоченных троек кофакторов g_p, g_i, g_j равно C_k^3 , а для каждой тройки $\{g_p, g_i, g_j\}$ надо проверять 12 дизъюнктивных разложений:

$$g_p = \bar{g}_i \vee g_j; \quad g_p = g_i \vee \bar{g}_j; \quad g_p = \bar{g}_i \vee \bar{g}_j; \quad g_p = g_i \vee g_j;$$

$$g_i = \bar{g}_p \vee g_j; \quad g_i = g_p \vee \bar{g}_j; \quad g_i = \bar{g}_p \vee \bar{g}_j; \quad g_i = g_p \vee g_j;$$

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

Аналогично, для каждой из C_k^3 троек кофакторов надо перебирать 12 вариантов для построения конъюнктивных разложений.

Теперь легко объяснить, что использование BDDI вместо BDD сокращает комбинаторный перебор вариантов при нахождении формул дизъюнктивных и конъюнктивных разложений. Например, если предварительно не установлено, что кофакторы g_p, g_s являются взаимно инверсными, то при поиске разложений и нахождении формулы $g_p = g_i \vee g_j$ для инверсии $\bar{g}_p = g_s$ всегда будет найдена двойственная формула $g_s = \bar{g}_i \& \bar{g}_j$. Таким образом, для каждой формулы алгебраического разложения безынверсного кофактора будет найдена двойственная формула для инверсного кофактора и перебор троек кофакторов станет более трудоемким.

Нахождение множества реализуемых кофакторов

Построим ориентированный граф G , задающий формулы алгебраических представлений кофакторов (будем называть их логическими уравнениями либо просто уравнениями). Вершинам орграфа G соответствуют кофакторы в прямой форме, которые упоминаются в уравнениях. Обозначим это множество кофакторов K_Y , а множество кофакторов, для которых не построены уравнения, — 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 пометкой — номером уравнения). Исходящие из вершины представимого кофактора

дуги (см. рис. 3) могут соответствовать прямой форме кофактора (черный кружок) либо инверсной форме кофактора (светлый кружок).

Введем понятие *кластера* (подмножества вершин графа G): в кластер входят вершины, соответствующие уравнениям с одним и тем же кофактором в левой части. Именем одинаковых кофакторов и обозначается кластер.

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

Задача 1. В орграфе G требуется оставить в каждом кластере только одну вершину (условие А) и удалить из графа G такое подмножество пар одинаково помеченных дуг, чтобы орграф G стал бесконтурным (условие В) и содержал наибольшее число неизолированных вершин, в каждую из которых заходит только одна пара одинаково помеченных дуг (условие С).

Смысл оставления в кластере одной вершины (условие А) очевиден — кофактор при схемной реализации достаточно представить только одним уравнением. Требование отсутствия контуров (условие В) продемонстрируем на примере двух уравнений $g_6 = g_1 \vee g_4, \quad g_1 = g_6 \& g_7$ из рассматриваемого далее примера. Данные уравнения в орграфе G создают контур, т. е. логическое противоречие в требовании алгебраического (и схемного) представления кофакторов, заключающееся в следующем. Чтобы выразить g_6 в виде конъюнкции $g_6 = g_1 \vee g_4$, требуется схема, реализующая кофактор g_1 . Однако чтобы получить g_1 в виде $g_1 = g_6 \& g_7$, требуется схема, реализующая g_6 . Условие С требует, чтобы как можно больше кофакторов было представлено в виде конъюнкции либо дизъюнкции.

После решения задачи 1 множество K_Y вершин орграфа G разбивается на три попарно непересекающихся подмножества. Вершины, не имеющие заходящих дуг, но имеющие исходящую дугу, соответствуют нереализуемым кофакторам. Вершины, имеющие одну пару заходящих дуг (с одной и той же пометкой), соответствуют реализуемым кофакторам. Вершины, которые оказались изолированными, например, при удалении дуг для удовлетворения условия В, 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.

Алгоритм решения задачи 1

Эман А1. Выбор одного уравнения в каждом кластере.

Шаг А1.1. Упорядочить кластеры по возрастанию (неубыванию) числа содержащихся в них вершин и рассматривать кластеры в этом порядке.

Шаг А1.2. Оставить только одну вершину в каждом 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 .

Уравнения для BDDI (рис. 4)

Номер уровня BDDI	Уравнения
5	$f_1 = \bar{x}_1 h_1 \vee x_1 h_2; \quad f_2 = \bar{x}_1 h_2 \vee x_1 h_3; \quad f_3 = \bar{x}_1 h_4 \vee x_1 h_5;$ $f_4 = \bar{x}_1 h_4 \vee x_1 h_6; \quad f_5 = \bar{x}_1 h_6 \vee x_1 h_7;$
4	$h_1 = \bar{x}_2 \bar{g}_1 \vee x_2 \bar{g}_3; \quad h_2 = \bar{x}_2 \bar{g}_2 \vee x_2 \bar{g}_8; \quad h_3 = \bar{x}_2 \bar{g}_5 \vee x_2 \bar{g}_6;$ $h_4 = \bar{x}_2 \bar{g}_6 \vee x_2 \bar{g}_7; \quad h_5 = \bar{x}_2 \bar{g}_6 \vee x_2 \bar{g}_{10}; \quad h_6 = \bar{x}_2 \bar{g}_4 \vee x_2 \bar{g}_9;$ $h_7 = \bar{x}_2 \bar{g}_8 \vee x_2 \bar{g}_{10};$
3	$g_1 = \bar{x}_4 s_2 \vee x_4 s_3; \quad g_2 = \bar{x}_4 \bar{x}_5 \vee x_4 s_1; \quad g_3 = \bar{x}_4 s_0 \vee x_4 s_1;$ $g_4 = \bar{x}_4 s_5 \vee x_4 s_2; \quad g_5 = \bar{x}_4 s_0 \vee x_4 s_5; \quad g_6 = \bar{x}_4 \bar{s}_0 \vee x_4 s_4;$ $g_7 = \bar{x}_4 \bar{x}_5 \vee x_4 s_5; \quad g_8 = \bar{x}_4 s_3 \vee x_4 \bar{s}_1; \quad g_9 = \bar{x}_4 s_2 \vee x_4 \bar{x}_5;$ $g_{10} = \bar{x}_4 x_3 \vee x_4 s_5;$
2	$s_0 = \bar{x}_5 x_3; \quad s_1 = x_5 \bar{x}_3; \quad s_2 = \bar{x}_5 \bar{x}_3; \quad s_3 = x_5 x_3;$ $s_4 = \bar{x}_5 \bar{x}_3 \vee x_5 x_3.$
1	Литералы x_3, \bar{x}_3
0	Константы 0, 1

Для выбора оставляемой в первом кластере вершины применяется эвристика 1.

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

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

Для следующих рассматриваемых кластеров используется эвристика 2.

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 эвристика 2 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 A1.2 и этап A1 считаются выполненными, когда в каждом кластере останется по одной вершине.

Этап A2. Приведение орграфа G к бесконтурному графу.

Итеративно выполняются шаги A2.1—A2.3, пока орграф не станет бесконтурным.

Шаг A2.1. Проверка рассматриваемого орграфа на отсутствие контуров.

На данном шаге осуществляется построение множества простых контуров графа. Для этого реализуется последовательный перебор всех вершин графа. Для очередной выбранной вершины выполняется поиск в глубину для построения всевозможных путей в графе, начинающихся с этой вершины. Такая процедура является модификацией программной реализации [10] в языке C# известного алгоритма Джонсона [11].

Если в результате проверки выявлены контуры, то выполняется шаг A2.2. Если орграф не имеет контуров, то осуществляется переход на шаг A2.4.

Шаг A2.2. Нахождение уравнения, удаляемого из графа, т. е. пары дуг с одинаковыми пометками, соответствующих уравнению и заходящих в одну и ту же вершину графа.

Каждая пара дуг с одинаковой пометкой (номером уравнения) p оценивается числом C_p контуров, которые могут быть разорваны при удалении из графа данной пары дуг.

Шаг A2.3. Удаляется пара дуг (уравнение) с пометкой p , которой соответствует максимальное значение числа C_p . Переход на шаг A2.1.

Шаг A2.4. Конец.

Пример выполнения алгоритма 1

Проиллюстрируем предложенный алгоритм 1 на примере BDDI, изображенной на рис. 4. Будем рассматривать третий уровень BDDI, кофакторы которого зависят от переменных x_3, x_4, x_5 .

Этап 1. Разбить множество уравнений по уровням BDDI.

Уравнения для каждого из уровней BDDI заданы в табл. 3.

Этап 2. Нахождение множества реализуемых кофакторов.

Шаги 2.1—2.3. С помощью программы [12] элиминации промежуточных переменных построим таблицу истинности кофакторов третьего уровня BDDI и найдем инверсии кофакторов (табл. 4). После этого проведем полный комбинаторный перебор вариантов нахождения дизъюнктивных и конъюнктивных разложений неинверсных кофакторов из табл. 4. Полученные алгебраические формулы запишем в табл. 5.

Шаг 2.4.

Этап A1 решения задачи 1. Выбор уравнений в кластерах (сокращение кластеров).

Шаг A1.1. Порядок рассмотрения кластеров согласно возрастанию (неубыванию) весов кластеров: $\langle g_6, g_2, g_4, g_1, g_5, g_7, g_3 \rangle$.

Шаг A1.2.

1. Рассматривается кластер g_6 , оставляется уравнение $g_6 = g_1 \vee g_4$, множество $T = \{g_1, g_4, g_6\}$.

2. Рассматривается кластер g_2 , оставляется уравнение $g_2 = \bar{g}_4 \& g_8$, реализуется эвристика 1, множество T пополняется вершинами из оставляемого уравнения: $T = \{g_1, g_2, g_4, g_6, g_8\}$.

3. Рассматривается кластер g_4 , максимальный вес $W_{q_4} = 18$ имеет вершина g_4 для уравнения $g_4 = g_6 \& \bar{g}_1$, оставляется уравнение $g_4 = g_6 \& \bar{g}_1$, реализуется эвристика 1, множество не изменяется: $T = \{g_1, g_2, g_4, g_6, g_8\}$.

Аналогично оставляются уравнения в других кластерах g_1, g_5, g_7, g_3 . Результат выполнения этапа A1:

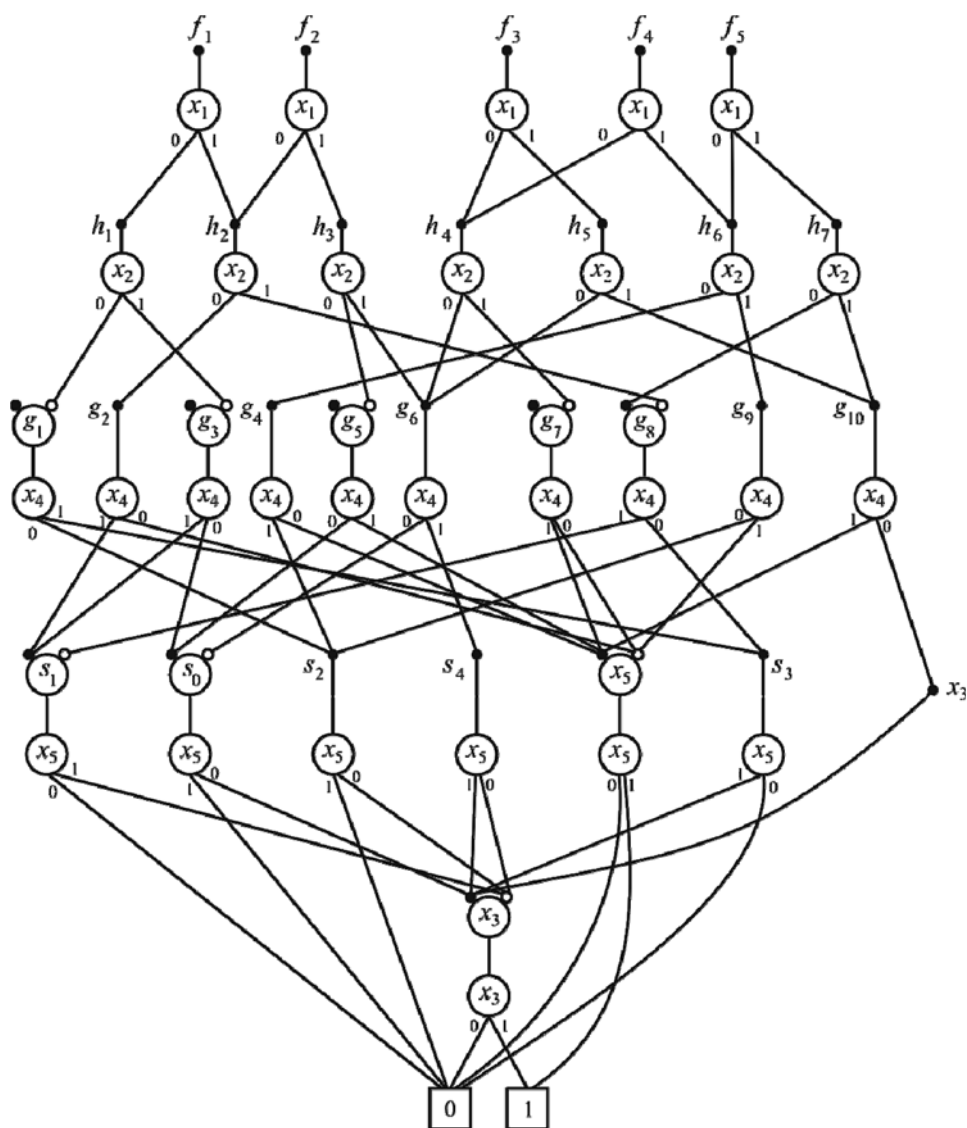


Рис. 4. Граф BDDI

Таблица 4

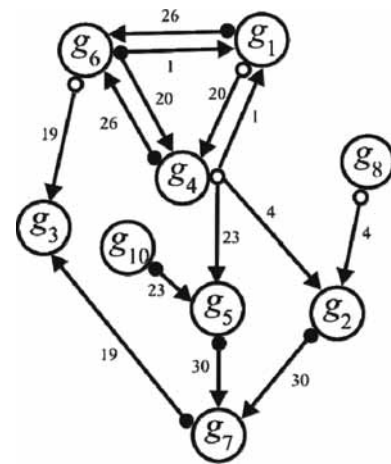
Кофакторы и их инверсии третьего уровня BDDI (рис. 4)

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}	\bar{g}_1	\bar{g}_2	\bar{g}_3	\bar{g}_4	\bar{g}_5	\bar{g}_6	\bar{g}_7	\bar{g}_8	\bar{g}_9	\bar{g}_{10}
0	0	0	1	1	0	0	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0	1
0	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	0	1	0	1	1	1	1
0	1	0	0	0	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	0	1
0	1	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1	0	1	0	1	1	0
1	0	0	0	1	1	0	1	0	1	0	0	1	1	0	0	1	0	1	0	1	1	0
1	0	1	0	0	0	1	0	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	1
1	1	1	1	0	0	0	1	1	1	1	0	1	0	1	1	1	0	0	0	0	1	0

Таблица 5

Уравнения алгебраических разложений кофакторов третьего уровня BDDI

Номер уравнения	Уравнение	Кластер	Вес кластера
1	$g_1 = g_6 \& \bar{g}_4$	g_1	3
2	$g_1 = g_6 \& g_7$		
3	$g_1 = g_7 \& \bar{g}_3$		
4	$g_2 = \bar{g}_4 \& \bar{g}_8$	g_2	2
5	$g_2 = g_7 \& \bar{g}_8$		
6	$g_3 = \bar{g}_6 \& \bar{g}_8$	g_3	14
7	$g_3 = \bar{g}_6 \& \bar{g}_9$		
8	$g_3 = g_{10} \& \bar{g}_6$		
9	$g_3 = g_{10} \& \bar{g}_8$		
10	$g_3 = g_2 \& \bar{g}_1$		
11	$g_3 = g_2 \& \bar{g}_6$		
12	$g_3 = g_2 \& \bar{g}_9$		
13	$g_3 = g_2 \& g_{10}$		
14	$g_3 = g_2 \& g_5$		
15	$g_3 = g_5 \& \bar{g}_1$		
16	$g_3 = g_5 \& \bar{g}_6$		
17	$g_3 = g_5 \& \bar{g}_8$		
18	$g_3 = g_7 \& \bar{g}_1$		
19	$g_3 = g_7 \& \bar{g}_6$		
20	$g_4 = g_6 \& \bar{g}_1$	g_4	2
21	$g_4 = g_6 \& \bar{g}_7$		
22	$g_5 = \bar{g}_4 \& \bar{g}_9$	g_5	4
23	$g_5 = g_{10} \& \bar{g}_4$		
24	$g_5 = g_7 \& \bar{g}_9$		
25	$g_5 = g_7 \& g_{10}$	g_6	1
26	$g_6 = g_1 \vee g_4$		
27	$g_7 = g_1 \vee g_2$		
28	$g_7 = g_1 \vee g_3$		
29	$g_7 = g_1 \vee g_5$		
30	$g_7 = g_2 \vee g_5$	g_7	4

Рис. 5. Орграф G после сокращения кластеров

орграф (рис. 5) и соответствующие ему уравнения (табл. 6).

Таблица 6

Уравнения, реализованные в графе G , после сокращения кофакторов

Номер уравнения	Уравнение	Кластер
1	$g_1 = g_6 \& \bar{g}_4$	g_1
4	$g_2 = \bar{g}_4 \& \bar{g}_8$	g_2
19	$g_3 = g_7 \& \bar{g}_6$	g_3
20	$g_4 = g_6 \& \bar{g}_1$	g_4
23	$g_5 = g_{10} \& \bar{g}_4$	g_5
26	$g_6 = g_1 \vee g_4$	g_6
30	$g_7 = g_2 \vee g_5$	g_7

Этап A2 решения задачи 1. Приведение орграфа G к бесконтурному графу.

В качестве исходного рассматривается орграф на рис. 5.

Итерация 1 (далее будут выполняться итерации 2 и 3).

На шаге A2.1 находятся все простые контуры: $\{g_1, g_6, g_1\}$, $\{g_1, g_4, g_1\}$, $\{g_1, g_4, g_6, g_1\}$, $\{g_1, g_6, g_4, g_1\}$, $\{g_4, g_6, g_4\}$.

На шаге A2.2 для каждого из уравнений (пар одинаково помеченных дуг) 1, 20, 26 подсчитывается число разрываемых контуров, если удалить данную пару дуг из графа. Легко видеть, что удаление каждой из пар дуг приводит к разрыву четырех контуров.

На шаге A2.3 выбирается пара дуг с пометкой 1 (рис. 6), которые удаляются из графа.

Итерация 2. На шаге A2.1 находится один простой контур $\{g_4, g_6, g_4\}$, который образуют две дуги с пометками 20, 26.

На шаге A2.2 выясняется, что удаление любой из пар дуг с пометками 20, 26 разрывает единственный простой контур $\{g_4, g_6, g_4\}$.

На шаге A2.3 удаляется пара дуг с пометкой 20, получается граф, представленный на рис. 7.

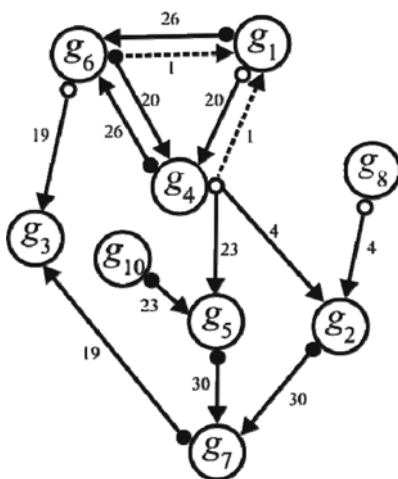


Рис. 6. Шаг 1 приведения графа к бесконтурному (удаляются штриховые дуги, соответствующие уравнению 1)

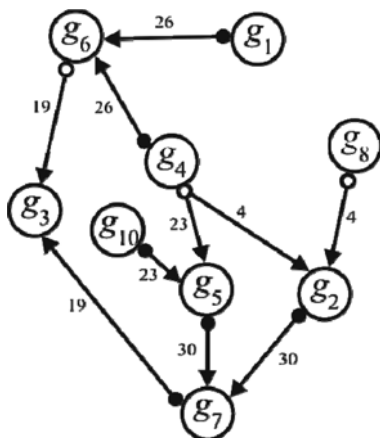


Рис. 7. Результирующий бесконтурный орграф, задающий реализуемые кофакторы

Итерация 3 до конца не выполняется, так как на шаге А2.1 выясняется, что граф (рис. 7) является бесконтурным.

Результат выполнения этапа А2: уравнения для найденных реализуемых кофакторов (табл. 7). На рис. 8 показана логическая схема, соответствующая найденному множеству реализуемых кофакторов, она не имеет обратных связей, так как соответствует ориентированному бесконтурному графу.

Таблица 7

Уравнения, соответствующие реализуемым кофакторам

Номер уравнения	Уравнение	Кластер
4	$g_2 = \bar{g}_4 \& \bar{g}_8$	g_2
19	$g_3 = g_7 \& \bar{g}_6$	g_3
23	$g_5 = g_{10} \& \bar{g}_4$	g_5
26	$g_6 = g_1 \vee g_4$	g_6
30	$g_7 = g_2 \vee g_5$	g_7

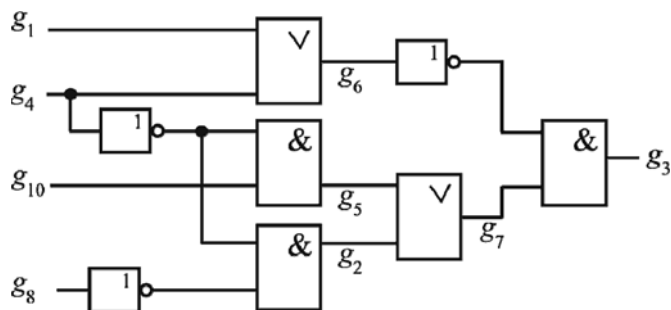


Рис. 8. Логическая схема, соответствующая результирующему бесконтурному орграфу

Этап 3. Корректировка исходного многоуровневого BDDI-представления векторной булевой функции.

На этапе 3 алгоритма 1 в функциональном описании (см. табл. 3) уравнения разложения Шеннона для кофакторов g_2, g_3, g_5, g_6, g_7 заменяются формулами алгебраических разложений из табл. 7 и получается функциональное описание:

$$\begin{aligned}
 f_1 &= \bar{x}_1 h_1 \vee x_1 h_2; & f_2 &= \bar{x}_1 h_2 \vee x_1 h_3; & f_3 &= \bar{x}_1 h_4 \vee x_1 h_5; \\
 f_4 &= \bar{x}_1 h_4 \vee x_1 h_6; & f_5 &= \bar{x}_1 h_6 \vee x_1 h_7; & h_1 &= \bar{x}_2 \bar{g}_1 \vee x_2 \bar{g}_3; \\
 h_2 &= \bar{x}_2 g_2 \vee x_2 \bar{g}_8; & h_3 &= \bar{x}_2 \bar{g}_5 \vee x_2 g_6; & h_4 &= \bar{x}_2 g_6 \vee x_2 \bar{g}_7; \\
 h_5 &= \bar{x}_2 g_6 \vee x_2 g_{10}; & h_6 &= \bar{x}_2 g_4 \vee x_2 g_9; \\
 h_7 &= \bar{x}_2 g_8 \vee x_2 g_{10}; & g_1 &= \bar{x}_4 s_2 \vee x_4 s_3; & g_2 &= \bar{g}_4 \& \bar{g}_8; \\
 g_3 &= g_7 \& \bar{g}_6; & g_4 &= \bar{x}_4 x_5 \vee x_4 s_2; & g_5 &= g_{10} \& \bar{g}_4; \\
 g_6 &= g_1 \vee g_4; & g_7 &= g_2 \vee g_5; & g_8 &= \bar{x}_4 s_3 \vee x_4 s_1; \\
 g_9 &= \bar{x}_4 s_2 \vee x_4 \bar{x}_5; & g_{10} &= \bar{x}_4 x_3 \vee x_4 x_5; \\
 s_1 &= x_5 \bar{x}_3; & s_2 &= \bar{x}_5 \bar{x}_3; & s_3 &= x_5 x_3.
 \end{aligned}$$

После замены формул выяснилось, что уравнения

$$s_0 = \bar{x}_5 x_3; \quad s_4 = \bar{x}_5 \bar{x}_3 \vee x_5 x_3$$

оказались неиспользуемыми (лишними), они отсутствуют в составе взаимосвязанных формул (4). В результате применения предложенного метода логической оптимизации в функциональном BDDI-описании сократилось число литералов и были получены лучшие по площади результаты логического синтеза (табл. 8). Информация об используемой библиотеке логических элементов будет дана далее при описании вычислительного эксперимента.

Таблица 8

Результаты логического синтеза

Функциональное описание	Число литералов	Логическая схема		
		Число логических элементов	Площадь S_{ASIC}	Задержка τ , нс
Исходное (формулы, табл. 2)	100	45	16 031	2,89
Результирующее (формулы (4))	84	37	13 855	3,36

Алгоритмы 2—4 нахождения алгебраических разложений кофакторов в BDDI

Чтобы описать отличия алгоритмов 2—4 от описанного алгоритма 1, заметим, что в **алгоритме 1** в виде алгебраических разложений представляются все кофакторы, т. е. те, которые имеют в своей записи два, три либо четыре литерала.

Формулы разложения Шеннона, включающие только два литерала, назовем *короткими*, а формулы, состоящие из трех либо четырех литералов — *длинными*. Уравнения вида (2), (3), которыми заменяются формулы разложения Шеннона, будем называть также подставляемыми уравнениями либо *подстановками*.

Алгоритм 2 отличается от алгоритма 1 тем, что на этапе 3 корректировки исходного многоуровневого BDDI-представления исходные короткие уравнения не заменяются найденными дизъюнктивными либо конъюнктивными разложениями.

Алгоритм 3 отличается от алгоритма 1 тем, что алгебраические дизъюнктивные и конъюнктивные разложения ищутся только для кофакторов, представленных только длинными уравнениями, при этом для искомым разложений используются кофакторы, представляемые в исходной записи как длинными, так и короткими уравнениями.

Алгоритм 4 отличается от алгоритма 3 дополнительным шагом: если в найденных результирующих формулах алгебраических разложений

$$g_p = \bar{g}_i \vee \bar{g}_j; \quad g_r = \bar{g}_i \& \bar{g}_j;$$

используются два инверсных кофактора, то, согласно аксиомам булевой алгебры (формулам де Моргана), в решение записываются эквивалентные подстановки, т. е. формулы

$$g_p = \neg(g_i \& g_j); \quad g_r = \neg(g_i \vee g_j),$$

содержащие один оператор инверсии.

Программная реализация

Программная реализация представленных выше алгоритмов была выполнена на языке C++ в рамках кроссплатформенной среды Qt [13], при этом активно применялись ранее разработанные библиотеки работы с булевыми объектами [14] и для обработки описаний, выполненных на языке SF [15, с. 51]. В целях практического использования общая организация программы выполнена по правилам разработки проектных процедур в системе FLC-2 [16].

Программа является параметрически настраиваемой. В качестве параметров выступают:

-i <src> — путь доступа к исходному файлу описания схемы, представленного в виде BDD-файла — системы логических уравнений на языке SF;

-o <tar> — путь доступа к результирующему, оптимизированному по числу литералов файлу;

-r <config> — путь доступа к файлу конфигурации — файлу формата INI системы Windows, содер-

жащему в своем составе секции с набором ключевых параметров, устанавливающих режимы работы программы:

```
[BDD _ OPT]
...
inversion=2
donothandleshort=2
...
[DATA]
RPT _ Name=e:/FLC2/workDir/protokol.txt
```

Здесь параметр `inversion` может принимать значения из множества {0,1,2}: "0" означает, что для подбора оптимизирующих подстановок не будут использованы инверсные функции; "1" — равнозначное использование как прямых, так и инверсных функций; "2" — кроме использования инверсных функций будет осуществлен переход к инверсному представлению для подстановок:

$$g_p = \bar{g}_i \vee \bar{g}_j \rightarrow g_p = \neg(g_i \& g_j);$$

$$g_r = \bar{g}_i \& \bar{g}_j \rightarrow g_r = \neg(g_i \vee g_j).$$

Параметр `donothandleshort` устанавливает правила рассмотрения уравнений на предмет поиска подстановки кофакторов и также может принимать значения из множества {0,1,2}: "0" означает, что проверке на замену будут подвергаться все уравнения рассматриваемой системы; "1" — отсеиваются возможные подстановки, способных сократить число литералов в уравнении, будет выполнен на этапе формирования результирующего файла; "2" — отсеивается будет проведен на этапе поиска подстановок.

Параметр `RPT _ Name` из секции `DATA` определяет путь доступа к файлу протокола сеанса, содержащего информацию об ошибках в принятых исходных данных.

Рассматриваемая программа имеет ряд ограничений. Среди них фиксированными являются только ограничения по числу n аргументов реализуемой системы булевых функций $n \leq 27$; числу возможных представлений отдельного кофактора (не более 96 вариантов) и числу дуг графа G (не более 30 000). Однако достижения таких значений на практике не наблюдалось в силу требуемого объема проводимых вычислений. Известны случаи нормального срабатывания программы при размерах списков смежности описания графа более 15 000 элементов (дуг графа) и порядка 90 шагов срабатывания алгоритма приведения орграфа к бесконтурному виду.

Вычислительный эксперимент

Был проведен эксперимент по проверке эффективности использования программ, реализующих предложенные алгоритмы 1—4, для сокращения площади и увеличения быстродействия блоков комбинационной логики, реализуемых в составе заказных цифровых СБИС (ASIC).

Исходными описаниями *первого набора примеров* комбинационной логики являлись системы ДНФ

булевых функций на языке SF [15, с. 53], взятые из библиотеки [17] примеров схем, представленных в формате PLA, при этом описания из формата PLA переводились в формат SF системы FLC-2 логической оптимизации [16].

Второй набор примеров составляли системы булевых функций, задающих SF-описания таблиц истинности модулярных сумматоров [18].

Для каждого из примеров систем полностью определенных булевых функций выполнялась BDDI-минимизация с помощью программы BDD_Builder [9], также включенной в FLC-2. Полученные схемные реализации были названы *базовыми* для эксперимента. BDDI-описания и дополнительно минимизированные BDDI-описания (дополнительная оптимизация выполнялась с помощью программ, реализующих предложенные в данной работе алго-

ритмы 1—4 алгебраического разложения кофакторов) конвертировались в VHDL-описания и подавались на вход синтезатора LeonardoSpectrum [19, с. 241]. Для каждого из примеров синтез логической схемы осуществлялся с одними и теми же опциями управления синтезом и для одной и той же целевой библиотеки синтеза. Целевой являлась библиотека проектирования заказных цифровых КМОП СБИС, состав библиотеки приведен в работе [20].

Параметры систем функций и BDDI-описаний для первого набора примеров заданы в табл. 9. Было установлено, что примеры GARY и IN0 задают одну и ту же систему функций в виде различных систем ДНФ, т. е. ДНФ с различными множествами элементарных конъюнкций. Для второго набора примеров имя примера Mod_i соответствует сумматору по модулю i .

Таблица 9

Параметры исходных систем функций и BDDI

Схема	Система ДНФ			BDDI		
	n	m	k	Число S_{BDDI} уравнений	Число пар взаимно инверсных кофакторов	Суммарное число литералов P
ADD6	12	7	1 092	27	10	91
ADDM4	9	8	512	66	41	590
ADR4	8	5	256	17	6	57
ALU1	12	8	19	16	0	50
B12	15	9	431	66	3	209
B9	16	5	123	73	14	253
BR1	12	8	34	119	4	319
BR2	12	8	35	85	2	216
DC2	8	7	58	58	9	179
DIST	8	5	256	115	32	424
EX7	16	5	123	73	14	253
F51M	8	8	256	37	13	124
GARY	15	11	214	317	17	964
IN0	15	11	138	317	17	964
IN1	16	17	110	756	20	2 552
IN2	19	10	137	261	25	861
INTB	15	7	664	681	105	2 465
LIFE	9	1	512	36	5	133
LOG8MOD	8	5	47	62	10	215
M1	6	12	32	48	7	134
M181	15	9	430	67	3	212
M2	8	16	96	116	28	362
M3	8	16	128	130	30	413

Схема	Система ДНФ			BDDI		
	n	m	k	Число S_{BDDI} уравнений	Число пар взаимно инверсных кофакторов	Суммарное число литералов P
M4	8	16	256	175	46	604
MAX1024	10	6	1 024	333	118	1 242
MAX46	9	1	46	72	0	248
MAX512	9	6	512	186	56	686
MLP4	8	8	256	147	37	528
MP2D	14	14	123	78	2	219
NEWAPLA	12	10	17	44	4	112
NEWAPLA1	12	7	10	24	0	55
NEWCPLA1	9	16	38	80	8	212
NEWILL	8	1	8	15	0	47
NEWTAG	8	1	8	8	0	23
NEWTPLA	15	5	23	56	4	147
P82	5	14	24	53	5	162
RADD	8	5	120	17	6	57
RD53	5	3	32	15	5	53
RD73	7	3	147	29	9	107
ROOT	8	5	256	56	15	175
SEX	9	14	23	51	0	142
SQN	7	3	96	46	5	161
SQR6	6	12	64	62	11	199
T3	12	8	152	87	5	248
TIAL	14	8	640	582	126	2 099
VTX1	27	6	110	100	2	298
X9DN	27	7	120	102	2	305
Z4	7	4	128	15	5	51
Z5XP1	7	10	128	40	11	131

Далее в таблицах, задающих результаты экспериментов, используются следующие обозначения для параметров систем функций $\mathbf{f}(\mathbf{x}) = (f_1(x), \dots, f_m(\mathbf{x}))$, $\mathbf{x} = (x_1, \dots, x_n)$: n — число переменных x_1, \dots, x_n ; m — число функций; k — число общих элементарных конъюнкций, входящих в ДНФ всех компонентных функций $f_j(\mathbf{x})$, $j = 1, \dots, m$.

Результирующие логические схемы оценивались двумя параметрами: S_{ASIC} — суммарная площадь всех элементов логической схемы; τ — задержка схемы (нс). Результаты эксперимента приведены в табл. 10–12. Лучшие решения (схемы меньшей площади либо с меньшей задержкой) отмечены жирным шрифтом, символом * помечены решения, улучшающие исходные базовые

решения, которыми, как уже говорилось, являются логические схемы, построенные по исходным BDDI. В табл. 11 для каждого из алгоритмов 1–4 приведены числа исключаемых (лишних) уравнений после применения соответствующего алгоритма. Такую информацию выдает синтезатор LeonardoSpectrum, если при синтезе выявляются неиспользуемые сигналы в функциональных описаниях, по которым ведется синтез схем. Если для какого-либо примера для каждого из алгоритмов 1–4 число исключаемых уравнений равно нулю, то информация по такому примеру отсутствует, например, для первого примера ADD6 информация в табл. 11 не приводится, так как лишних уравнений не оказывается ни для одного из алгоритмов.

Результаты эксперимента для первого набора примеров

Схема	Исходные BDDI (базовые решения)		Алгоритм 1		Алгоритм 2		Алгоритм 3		Алгоритм 4	
	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс
ADD6	12 806	8,03	*11 952	*7,72	12 806	8,03	12 583	9,05	12 583	9,05
ADDM4	80 782	7,84	79 643	8,01	*73 974	8,17	78 672	11,35	81 602	*7,03
ADR4	8 074	4,90	*7 589	*4,82	8 074	4,90	*7 851	5,60	*7 851	5,06
ALU1	7 109	1,12	7 109	1,12	7 109	1,12	7 109	1,12	7 109	1,12
B12	18 358	3,59	21 327	3,77	21 293	*3,46	21 293	*3,46	18 492	*3,40
B9	26 081	4,91	26 204	*3,89	26 081	4,91	*25 701	4,65	25 701	4,65
BR1	23 843	6,36	27 867	*5,65	24 530	*5,06	26 193	*5,46	28 547	*5,26
BR2	21 371	6,34	21 790	*5,57	*20 032	*4,43	*20 032	*4,43	21 371	6,34
DC2	23 302	4,28	22 845	5,34	20 780	4,88	*19 976	4,30	*19 067	4,73
DIST	60 085	6,08	55 627	8,34	*54 159	6,74	56 062	7,14	54 807	*6,01
EX7	26 081	4,91	26 204	*3,89	26 081	4,91	*25 701	*4,65	*25 701	*4,65
F51M	18 353	7,67	17 622	*5,39	*16 824	*5,45	17 298	*3,21	*16 037	*3,54
GARY	94 648	6,74	107 264	7,14	96 032	6,81	101 299	*6,68	*93 978	6,43
IN0	94 648	6,74	107 264	7,14	96 032	6,81	98 409	*6,50	*93 978	6,43
IN1	192 655	8,26	235 638	14,12	216 493	10,05	227 262	13,16	221 805	11,62
IN2	75 414	6,73	71 619	11,23	*69 471	9,13	70 102	10,01	*73 851	10,75
INTB	272 555	8,67	*241 168	13,90	*237 758	13,26	*248 561	13,50	*248 316	15,73
LIFE	18 146	4,60	*15 256	7,44	*16 450	7,26	*15 150	7,36	*17 359	7,37
LOG8MOD	24 022	4,26	25 160	*4,05	24 440	4,89	24 161	4,48	24 457	4,76
M1	15 312	3,46	17 605	*2,63	16 121	*2,63	16 857	*2,47	16 037	*2,77
M181	18 849	3,48	20 356	3,73	20 914	*3,42	20 914	*3,42	*18 576	3,75
M2	45 086	5,20	*45 058	*4,87	*43 362	5,39	*42 776	*5,19	*42 787	*5,07
M3	52 580	4,49	54 656	6,62	*52 435	4,92	*51 754	5,81	*49 947	5,33
M4	78 181	5,87	78 405	*5,27	78 879	6,19	*77 473	6,18	*76 502	*4,84
MAX1024	146 888	7,18	153 606	11,02	150 777	10,30	148 567	10,50	*144 076	9,79
MAX46	36 125	4,89	*34 892	5,61	*32 035	4,93	*35 344	4,89	*35 344	4,89
MAX512	84 643	5,98	*84 537	7,29	*79 827	7,72	*81 228	6,29	*79 582	6,46
MLP4	68 439	5,60	70 224	6,28	*66 586	6,70	68 628	9,17	*68 210	6,98
MP2D	17 471	3,56	20 239	6,15	19 351	3,48	18 771	4,37	17 767	3,61
NEWAPLA	11 087	3,84	*10 184	3,90	11 087	3,84	*10 184	3,90	11 087	3,84
NEWAPLA1	6 869	3,35	*6 702	3,63	6 869	3,35	6 869	3,35	6 869	3,35
NEWCPLA1	20 585	3,87	22 158	4,53	20 596	4,58	*19 206	4,02	22 951	5,14
NEWILL	5 122	3,22	5 736	4,76	5 736	4,76	5 736	4,76	5 122	3,22
NEWTAG	2 126	1,90	2 126	1,90	2 126	1,90	2 126	1,90	2 126	1,90

Схема	Исходные BDDI (базовые решения)		Алгоритм 1		Алгоритм 2		Алгоритм 3		Алгоритм 4	
	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс
NEWTPLA	11 316	3,30	13 515	3,85	11 316	3,30	13 264	4,04	11 316	3,30
P82	19 988	2,93	22 292	3,23	*19 965	*2,75	19 982	*2,76	*19 982	*2,76
RADD	8 074	4,90	*7 589	*4,82	8 074	4,90	7 851	5,60	*7 851	5,60
RD53	7 321	3,38	7 779	*3,22	7 321	3,38	7 321	3,38	7 321	3,38
RD73	18 090	4,55	*14 212	*4,49	15 222	5,30	15 669	4,61	*15 669	4,61
ROOT	26 109	4,78	26 455	*4,68	*24 580	*4,23	*22 532	*4,08	*23 581	5,30
SEX	12 566	3,69	15 524	*3,56	12 605	*2,08	12 605	*2,08	12 605	*2,08
SQN	22 303	3,33	*20 077	4,87	*19 642	5,04	*19 245	*3,10	*18 922	5,57
SQR6	28 737	5,72	*28 157	*4,55	*28 023	*4,33	*27 827	*3,77	*27 035	*3,72
T3	17 276	3,59	19 268	5,85	17 454	6,03	17 454	6,03	17 917	4,16
TIAL	255 531	8,75	*221 509	15,25	*212 286	12,65	*209 217	12,15	*211 633	14,72
VTX1	26 996	5,96	*25 930	6,40	*24 223	*5,94	*24 223	*5,94	*24 552	6,16
X9DN	26 996	6,00	25 701	*5,81	*25 155	*5,84	*25 155	*5,84	*26 812	6,65
Z4	6 640	4,25	*6 339	4,32	6 640	4,25	6 417	4,71	*6 417	4,71
Z5XP1	18 442	4,64	*18 252	4,30	18 442	4,53	*16 400	*3,61	*16 400	*3,61
Число лучших решений (полу- жирный шрифт)	16	22	9	12	14	11	15	14	18	16
Число улучшений базовых решений (число *)			17	18	19	12	21	18	28	13

Таблица 11

Результаты эксперимента для первого набора примеров

Схема	Число исключаемых уравнений			Схема	Число исключаемых уравнений		
	Алгоритм 1	Алгоритм 2	Алгоритм 3, 4		Алгоритм 1	Алгоритм 2	Алгоритм 3, 4
ADDM4	1	1	1	MAX1024	3	3	5
B12	1	0	0	MAX512	1	1	2
BR1	4	3	2	MLP4	1	1	1
BR2	6	2	2	MP2D	2	2	2
DC2	4	2	1	NEWAPLA	2	0	0
DIST	5	3	3	NEWCPLA1	2	3	0
GARY	14	6	6	P82	2	0	0
IN0	14	6	6	RD73	1	0	0
IN1	73	60	57	SEX	3	1	3
IN2	22	20	20	SQN	1	1	1
INTB	90	93	96	T3	11	7	7
LIFE	4	4	4	TIAL	59	75	75
M1	5	2	2	VTX1	2	2	2
M181	1	0	0	X9DN	2	2	2
M3	2	0	0	Z5XP1	1	0	0
M4	2	2	2				

Результаты эксперимента для второго набора примеров

Схема	Исходные BDDI (базовые решения)		Алгоритм 1		Алгоритм 2		Алгоритм 3		Алгоритм 4	
	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс	S_{ASIC}	τ , нс
Mod_5	13 520	2,68	*12 923	2,86	13 520	2,68	13 593	3,88	13 520	2,68
Mod_7	16 946	3,51	20 696	4,50	16 946	3,51	19 469	*3,39	18 777	4,00
Mod_9	32 358	6,21	*30 640	*5,64	32 358	6,21	*30 082	6,44	*30 679	*5,01
Mod_15	41 080	5,83	44 768	7,00	41 080	5,83	44 205	6,93	44 428	7,34
Mod_17	54 137	6,02	57 697	7,62	54 137	6,02	*52 703	*5,98	*48 462	8,15
Mod_19	49 483	6,93	58 311	9,00	49 483	9,93	51 593	7,80	*48 752	*5,95
Mod_23	58 746	6,75	*58 171	*6,17	58 746	6,75	60 822	7,81	62 624	8,75
Mod_25	68 043	6,60	75 492	9,00	68 043	6,60	75 570	8,53	73 913	8,35
Mod_27	78 600	6,77	92 422	10,28	78 600	6,77	85 240	8,96	87 059	9,24
Mod_29	77 902	6,51	*77 149	10,02	77 902	6,51	*72 919	10,39	*71 469	11,68
Mod_31	73 003	7,68	*72 652	8,96	73 003	7,68	*72 864	8,08	*69 795	8,77
Mod_37	113 352	6,60	*94 296	9,04	113 352	6,60	*87 483	9,89	*91 395	11,18
Mod_59	149 628	9,35	159 030	16,62	149 628	9,35	150 264	15,73	151 542	13,57
Mod_61	129 863	8,54	*102 466	13,36	129 863	8,54	*100 261	13,83	*93 549	18,04
Число лучших решений (полу- жирный шрифт)	5	10	2	1	4	8	2	2	5	3
Число улучшений базовых решений (число *)			7	2	0	0	6	2	7	2

Проанализируем результаты эксперимента. Первый набор примеров состоял из 59 функциональных описаний схем. Для 10 примеров схем небольшой размерности — это схемы CLPL, CO14, NEWAPLA2, NEWBYTE, NEWCOND, NEWTPLA1, NEWTPLA2, RYY6, SYM10, Z9SYM — применение алгоритмов 1—4 не изменило базовое решение, поэтому исходные данные и результаты вычислительного эксперимента для данных примеров в табл. 9—11 не приводятся.

Для трех примеров B12, LOG8MOD, MP2D применение любого из алгоритмов 1—4 изменения функциональных базовых BDDI описаний приводило к незначительному увеличению площади соответствующей логической схемы. Это связано с тем, что синтезатор LeonardoSpectrum имеет собственные встроенные средства логической оптимизации и технологического отображения. Технологическое отображение заключается в покрытии оптимизированных логических уравнений функциональными описаниями логических элементов, входящих в библиотеку проектирования. Изменение функционального описания приводило к другим результатам технологического отображения. Для 33 примеров схем применение предложенных алгоритмов позволило

улучшить их площадь. Сокращение площади для схем большой размерности может быть практически значимым: для схемы INTB сокращение площади составляет 13 %, для схемы Tial — 18 %. В целом, алгоритмы 3 и 4 являются более эффективными по сравнению с алгоритмами 1 и 2. Алгоритм 4 позволяет улучшить базовые решения (по площади) для 27 схем, алгоритм 3 — позволяет увеличить быстродействие для 18 схем. Алгоритмы 3 и 4 можно рекомендовать для практического применения в первую очередь. Не исключается также вариант того, что применение алгоритмов 1 и 2 также может улучшить базовое решение.

Для второго набора примеров (модулярных сумматоров) применение алгоритмов позволило уменьшить площадь для девяти схем и уменьшить задержки для пяти схем из 14. Площадь схемы Mod_61 уменьшилась на 28 %, площадь схемы Mod_37 уменьшилась на 23 % (табл. 12).

Программный эксперимент также показал, что для сокращения площади схем всегда более эффективны дизъюнктивные и конъюнктивные разложения кофакторов с использованием их инверсий по сравнению с аналогичными разложениями без инверсий кофакторов.

Заключение

Эксперименты [9] показали, что нахождение пар взаимно инверсных кофакторов при оптимизации BDDI значительно улучшает результаты последующего синтеза по сравнению с оптимизацией BDD, выполняемой без нахождения инверсных кофакторов.

В статье описаны программно реализованные алгоритмы дополнительной минимизации алгебраических многоуровневых BDDI-представлений систем полностью определенных функций на основе дизъюнктивных и конъюнктивных представлений подфункций, находящихся на одном уровне BDDI. Такая дополнительная логическая оптимизация по числу литералов в результирующих логических уравнениях, задающих BDDI, приводит к более простым функциональным описаниям, по которым осуществляется синтез логических схем. Сокращение площади схем, сводящееся к уменьшению числа транзисторов в схемах, позволяет также сокращать энергопотребление схем. Однако сокращение числа литералов при алгебраических разложениях, позволяющее часто сокращать площадь схем из библиотечных КМОП-элементов, может приводить как к уменьшению, так и увеличению временных задержек схем.

Список литературы

1. Брейтон Р. К., Хэтчел Г. Д., Санджованни-Винченцелли А. Л. Синтез многоуровневых комбинационных логических схем // ТИИЭР. 1990. Т. 78, № 2. С. 38–83.
2. Meinel C., Theobald T. Algorithms and Data Structures in VLSI Design: OBDD — Foundations and Applications. — Berlin, Heidelberg. — Springer-Verlag, 1998. 267 p.
3. Кнут Д. Э. Искусство программирования. Том 4. А. Комбинаторные алгоритмы. Часть 1 / Пер. с англ. — М.: Вильямс, 2013. 960 с.
4. Yang S., Ciesielski M. BDS: a BDD-based logic optimization system // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2002. Vol. 21, No. 7. P. 866–876.
5. Ebendt R., Fey G., Drechsler R. Advanced BDD optimization. — Springer, 2005. 222 p.
6. Бибило П. Н. Применение диаграмм двоичного выбора при синтезе логических схем. — Минск: Беларус. навука, 2014. 231 с.
7. Kubica M., Kania D. SMTBDD: New form of BDD for logic synthesis // International Journal of Electronics and Telecommunications. 2016. Vol. 62, No. 1. P. 33–41.
8. Бибило П. Н., Романов В. И. Минимизация многоуровневых представлений систем полностью определенных булевых функций с использованием разложений Шеннона и алгебраических представлений кофакторов // Информатика. 2021. Т. 18, № 2. С. 7–32.
9. Бибило П. Н., Ланкевич Ю. Ю. Использование полиномов Жегалкина при минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона // Программная инженерия. 2017. Т. 8, № 8. С. 369–384.
10. Поиск элементарных циклов в графе. URL: <https://vscode.ru/prog-lessons/poisk-elementarnyih-tsiklov-v-grafe.html>
11. Johnson D. B. Finding all the elementary circuits of a directed graph. // SIAM J. Comput. 1975. Vol. 4, No. 1. P. 77–84.
12. Торопов Н. Р. Преобразование многоярусной комбинационной сети в двухярусную // Логическое проектирование. Вып. 5. — Минск: ИТК НАН Беларуси, 2000. С. 4–14.
13. Шлее М. Qt 5.10. Профессиональное программирование на C++. — СПб.: БХВ-Петербург, 2018. 1072 с.
14. Романов В. И. Программные средства для решения логико-комбинаторных задач // Информатика. 2005. № 4. С. 114–123.
15. Бибило П. Н., Романов В. И. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний. — Минск: Беларус. навука, 2011. — 279 с.
16. Бибило П. Н., Романов В. И. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний // Проблемы разработки перспективных микро- и наноэлектронных систем. — 2020. Сб. трудов / под общ. ред. акад. РАН А. Л. Стемповского. — М.: ИППМ РАН, 2020. № 4. С. 9–16.
17. The Tests in the Monograph "Logic Minimization Algorithms for VLSI Synthesis". URL: <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex> (дата обращения: 20.11.2020).
18. Балака Е. С., Тельпухов Д. В., Осинин И. П., Городецкий Д. А. Сравнительное исследование и анализ методов аппаратной реализации сумматоров по модулю // 7Universum: Технические науки: электрон. научн. журн. 2016. № 1 (23). URL: <http://7universum.com/ru/tech/archive/item/2887>
19. Бибило П. Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum. — М.: СОЛОН-Пресс, 2005. 384 с.
20. Авдеев Н. А., Бибило П. Н. Автоматизированное проектирование цифровых операционных устройств с пониженным энергопотреблением // Программная инженерия. 2021. Т. 12, № 2. С. 63–73.

Experimental Study of Algorithms for Minimization of Binary Decision Diagrams using Algebraic Representations of Cofactors

Bibilo P. N., bibilo@newman.bas-net.by, **V. I. Romanov**, rom@newman.bas-net.by,
The United Institute of Informatics Problems of the National Academy of Sciences of Belarus,
Minsk, 220012, Belarus

Corresponding author:

Bibilo Petr N., Head of Laboratory, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, 220012, Minsk, Belarus
E-mail: bibilo@newman.bas-net.by

*Received on October 25, 2021
Accepted on December 07, 2021*

BDD (Binary Decision Diagram) is used for technology-independent optimization, performed as the first stage in the synthesis of logic circuits in the design of ASIC (application-specific integrated circuit). BDD is an acyclic graph defining a Boolean function or a system of Boolean functions. Each vertex of this graph is associated with the complete or reduced Shannon expansion formula. Binary decision diagrams with mutually inverse subfunctions (cofactors) are considered. We have developed algorithms for finding algebraic representations of cofactors of the same BDD level in the form of a disjunction or conjunction of other inverse or non-inverse cofactors of the same BDD level. The algorithms make it possible to reduce the number of literals by replacing the Shannon expansion formulas with simpler logical formulas and to reduce the number of literals in the description of a system of Boolean functions. We propose to use the developed algorithms for an additional logical optimization of the constructed BDD representations of systems of Boolean functions. Experimental results of the application of the corresponding programs in the synthesis of logic circuits in the design library of custom VLSI CMOS circuits are presented.

Keywords: system of Boolean functions, Disjunctive Normal Form, Binary Decision Diagram, Shannon expansion, digital logic synthesis, VHDL, VLSI

For citation:

Bibilo P. N., Romanov V. I. Experimental Study of Algorithms for Minimization of Binary Decision Diagrams using Algebraic Representations of Cofactors, *Programmnaya Ingeneria*, 2022, vol. 13, no. 2, pp. 51–67.

DOI: 10.17587/prin.13.51-67

References

1. **Brayton R. K., Hachtel G. D., Sangiovanni-Vincentelli A. L.** Synthesis of multi-level combinational logic circuits, *Trudy Instituta inzhenerov po elektronike i radiotekhnike*, 1990, vol. 78, no. 2, pp. 38–83 (in Russian).
2. **Meinel C., Theobald T.** *Algorithms and Data Structures in VLSI Design: OBDD — Foundations and Applications*, Berlin; Heidelberg, Springer-Verlag, 1998, 267 p.
3. **Knuth D. E.** Combinatorial Algorithms, *The Art of Computer Programming*, Pearson Education, 2011, vol. 4A, 883 p. (Russ. ed.: Moscow, 2013, 960 p.).
4. **Yang S., Ciesielski M.** BDS: a BDD-based logic optimization system, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2002, vol. 21, no. 7, pp. 866–876.
5. **Ebendt R., Fey G., Drechsler R.** *Advanced BDD optimization*, Springer, 2005, 222 p.
6. **Bibilo P. N.** *Application of Binary Decision Diagrams in the Synthesis of Logic Circuits*, Minsk, Belaruskaja navuka, 2014, 231 p. (in Russian).
7. **Kubica M., Kania D.** SMTBDD: New form of BDD for logic synthesis, *International Journal of Electronics and Telecommunications*, 2016, vol. 62, no. 1, pp. 33–41.
8. **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, *Informatics*, 2021, vol. 18, no. 2, pp. 18–43 (in Russian).
9. **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, *Programmnaya ingeneria*, 2017, no. 8, pp. 369–384 (in Russian).
10. **Search** for elementary cycles in a graph, available at: <https://vscode.ru/prog-lessons/poisk-elementarnyih-tsiklov-v-grafe.html>
11. **Johnson D. B.** Finding all the elementary circuits of a directed graph, *SIAM J. Comput.*, 1975, vol. 4, no. 1, pp. 77–84.
12. **Toropov N. R.** Transformation of a multi-level combinational network into a two-level one, *Logicheskoe proektirovanie*, Vyp. 5. Minsk, ITK NAN Belarusi, 2000, pp. 4–14 (in Russian).
13. **Shlee M.** *Qt 5.10. Professional programming in C++*, Saint Petersburg, BHV-Peterburg, 2018, 1072 p. (in Russian).
14. **Romanov V. I.** Software tools for solving logic-combinatorial problems, *Informatics*, 2005, no. 4, pp. 114–123 (in Russian).
15. **Bibilo P. N., Romanov V. I.** *Logical Design of Discrete Devices Using a Production-Frame Knowledge Representation Model*, Minsk, Belaruskaja navuka, 2011, 279 p. (in Russian).
16. **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 sistem.* — 2020. Sb. trudov / pod obshch. red. akad. RAN A. L. Stempkovskogo. — Moscow, IPPM RAN, 2020, no. 4, pp. 9–16 (in Russian).
17. **The Tests** in the Monograph "Logic Minimization Algorithms for VLSI Synthesis", available at: <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex>
18. **Balaka E. S., Tel'puhov D. V., Osinin I. P., Gorodeckij D. A.** Comparative study and analysis methods hardware implementation RNS-based adders, *7 Universum: Tekhnicheskie nauki: elektron. nauchn. zhurn.*, 2016, no. 1 (23), available at: <http://7universum.com/ru/tech/archive/item/2887> (in Russian).
19. **Bibilo P. N.** *CAD of integrated circuits based on the VHDL. StateCAD, ModelSim, LeonardoSpectrum*, Moscow, SOLON-Press, 2005, 384 p. (in Russian).
20. **Avdeev N. A., Bibilo P. N.** Design of Digital Operational Units with Low Power Consumption, *Programmnaya Ingeneria*, 2021, no. 2, pp. 63 – 73 (in Russian).