

Выделение из многоуровневого представления системы булевых функций подсистем для совместной логической минимизации

П.Н. Бибило
Н.А. Кириенко
В.И. Романов

Ссылка для цитирования

Бибило П.Н., Кириенко Н.А., Романов В.И. Выделение из многоуровневого представления системы булевых функций подсистем для совместной логической минимизации // Программные продукты и системы. 2023. Т. 36. № 4. С. 509–522. doi: 10.15827/0236-235X.142.509-522

Информация о статье

Поступила в редакцию: 31.05.2023

После доработки: 04.07.2023

Принята к публикации: 10.07.2023

Аннотация. В статье приводятся результаты экспериментальных исследований эффективности программ минимизации многоуровневых алгебраических представлений систем булевых функций, выполняемых при синтезе комбинационных схем. Результирующие минимизированные логические описания представлены в виде формул разложений Шеннона или формул, задающих булевы сети. Исследуются три подхода: совместная минимизация многоуровневых представлений систем булевых функций, отдельная минимизация и выделение из исходной системы связанных подсистем. При этом каждая из этих подсистем минимизируется отдельно, а функции, составляющие их, совместно. После получения минимизированных описаний схем, заданных в виде совокупности взаимосвязанных формул разложения Шеннона либо двухоперандных логических уравнений, соответствующих булевым сетям, осуществляется синтез логических схем в одной и той же библиотеке проектирования заказных цифровых сверхбольших интегральных схем, выполненных по КМОП СБИС (комплементарной метал-оксид-полупроводник технологии). Полученные логические схемы сравниваются по площади кристалла и по быстродействию (временной задержке). Были проведены эксперименты на 39 промышленных примерах схем. Результаты показали конкурентоспособность и целесообразность использования на практике всех трех рассмотренных подходов. Улучшение параметров схем (площадь, временная задержка) при выделении из исходной системы связанных подсистем достигается за счет того, что каждая выделенная подсистема минимизируется на основе разложений Шеннона по своей (для каждой подсистемы) перестановке переменных разложения. При этом для одной половины схем более эффективным является минимизация многоуровневых представлений на основе разложений Шеннона для исходных матричных описаний систем функций, а для другой – на основе разложений Шеннона систем функций, представленных в виде логических уравнений. Практическая значимость проведенного исследования заключается в том, что использование разработанной программы, реализующей предложенный алгоритм выделения подсистем булевых функций, позволяет во многих случаях сокращать площадь и увеличивать быстродействие функциональных блоков заказных КМОП СБИС.

Ключевые слова: система булевых функций, многоуровневое представление, дизъюнктивная нормальная форма, BDD, булева сеть, синтез логической схемы, VHDL, СБИС, КМОП-технология

Введение. Минимизация двухуровневых либо многоуровневых представлений (описаний) булевых функций и систем является первым и важнейшим этапом синтеза комбинационных схем, от которого зависят важнейшие параметры логической схемы – площадь, быстродействие и энергопотребление. Двухуровневыми (И-ИЛИ) называют представления функций в виде *дизъюнктивных нормальных форм* (ДНФ), многоуровневыми – различные формы функциональных разложений. Компактность минимизированных описаний схем достигается за счет нахождения общих частей в функциональных описаниях различного вида. Нахождение общих частей в функциональных двухуровневых описаниях систем функций легко проиллюстрировать при решении задачи минимизации булевых функций в классе ДНФ. При схемной реализации минимизированных систем ДНФ на программируемых логических

матрицах преимущество получили методы и алгоритмы совместной минимизации, при которой одна и та же элементарная конъюнкция может входить в минимизированные ДНФ нескольких функций системы. Однако при синтезе схем из библиотечных логических элементов отдельная минимизация булевых функций в классе ДНФ не потеряла своего значения, так как промышленные синтезаторы логических схем часто извлекают из функциональных исходных описаний систем описания отдельных функций и осуществляют их отдельную схемную реализацию.

При минимизации многоуровневых скобочных представлений на основе методов факторизации выделяются общие части как элементарных конъюнкций и дизъюнкций, так и алгебраических подвыражений, что является одним из видов совместной минимизации многоуровневых описаний схем и оказалось полез-

ным для уменьшения площади схем заказных СБИС. В качестве структур данных при решении задач факторизации использованы булевы сети, представленные ориентированными бесконтурными графами DAG (Directed-Acyclic-Graph). При декомпозиции систем функций были разработаны различные методы совместной декомпозиции, при которой одна и та же промежуточная функция может использоваться совместно – участвовать в разложениях (суперпозициях) нескольких функций исходной системы.

Несколько последних десятилетий в качестве основных методов технологически независимой оптимизации многоуровневых представлений систем булевых функций применяются методы минимизации графовых BDD (Binary Decision Diagram)-представлений [1] и их модификаций, при этом внимание чаще уделяется совместной минимизации [1, 2] и декомпозиции [3, 4] BDD-представлений систем функций. Следует заметить, что структуры данных в виде BDD-представлений широко используются также при верификации логических схем, построении тестов и решении сложных задач в различных областях информатики. Размерность задач проектирования СБИС возрастает в связи с совершенствованием технологических процессов производства СБИС и уменьшением технологических норм [1]. Решение задач логической оптимизации начинается совместно с задачами топологического и физического проектирования СБИС с широким использованием программ проверки выполнимости *конъюнктивной нормальной формы* (КНФ) булевых функций, называемых SAT-solvers (Boolean Satisfiability problem – задача выполнимости булевой формулы, представленной в виде КНФ) [5–7].

После BDD для задания систем булевых функций появились новые структуры данных, которые используются при решении задач технологически независимой логической оптимизации [5–7]. Это AIG (AND-Inverter-Graph), XMG (XOR-Majority-Graph) [8], MIG (Majority-Inverter-Graph) [9–11], XAIG (XOR-AND-Inverter-Graph) [12]. Структуры AIG положены в основу системы ABC (<http://www.eecs.berkeley.edu/~alanmi/abc>) – известной академической системы оптимизации представлений систем булевых функций и синтеза логических схем. Характерной особенностью решаемых с помощью этих структур оптимизационных задач является то, что они используются для совместной реализации систем булевых функций.

Раздельная минимизация многоуровневых представлений и выделение подсистем для их совместных реализаций не изучались.

Одинаковость частей в таблицах истинности булевых функций привела к идее использования связанности (общности) областей определений булевых функций при синтезе многовыходных комбинационных схем. Выделение связанных подсистем функций по таблицам истинности и ДНФ функций рассматривалось в [13]. Такой подход ограничен размерностью задач: число аргументов системы функций не превышает трех десятков. Для BDD-представлений задача выделения связанных по областям определения подсистем функций решалась в [13], где были предложены различные меры связанности функций, в том числе и основанная на общности заданного числа формул разложений Шеннона для всех функций выделяемой подсистемы. Такую связанность можно назвать сильной.

В данной работе предложены новая мера связанности подсистем функций и алгоритм последовательного формирования связанных подсистем как для BDD и их модификаций, задаваемых формулами разложений Шеннона, так и для булевых сетей, задаваемых одно- и двухоперандными логическими формулами. Алгоритм может быть легко обобщен на другие структуры данных, используемые для многоуровневых представлений систем булевых функций. Проведены программная реализация алгоритма и его экспериментальное исследование на потоке промышленных примеров, когда число аргументов и функций системы превышает сотню, а число формул (уравнений) достигает нескольких десятков тысяч. Выполнено сравнение эффективности предложенного подхода с процедурами совместной и раздельной минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона. Экспериментально установлена целесообразность использования предложенного алгоритма выделения подсистем функций при синтезе логических схем в библиотеке проектирования заказных цифровых КМОП СБИС как для BDD-представлений, так и для булевых сетей.

Совместная и раздельная минимизация многоуровневых представлений систем булевых функций на основе разложения Шеннона

BDDI-представления системы булевых функций. Под BDDI (*Binary Decision Diagram*

with Inverse cofactors) понимается ориентированный бесконтурный граф, задающий последовательные разложения Шеннона булевой функции $f(x)$ либо системы $F = \{f^1(x), \dots, f^m(x)\}$ булевых функций по всем переменным x_1, \dots, x_n при заданном порядке (перестановке) переменных, по которым проводятся разложения, при условии нахождения пар взаимно инверсных кофакторов. Разложением Шеннона булевой функции $f(x) = f(x_1, \dots, x_n)$ по переменной x_i является представление $f(x)$ в виде

$$f(x) = x_i f_0 \vee x_i f_1.$$

Каждый из кофакторов (cofactors) $f_0 = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f_1 = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ может быть разложен по одной из переменных 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. Если не ищутся взаимно инверсные кофакторы, то в результате последовательных разложений Шеннона строятся ROBDD-представления (Reduced Ordered Binary Decision Diagram – сокращенная упорядоченная бинарная диаграмма решений). В ROBDD функциональная вершина реализует один кофактор, в то время как в BDDI взаимно инверсные кофакторы представляются парой взаимосвязанных вершин. Таким образом, под BDDI будут пониматься совместные ROBDD, в которых на каждом уровне найдены пары взаимно инверсных кофакторов. Алгоритм построения ROBDD описан в [2], он был модифицирован для минимизации BDDI-представлений.

Булевы сети – Bool-представления системы булевых функций. В статье рассматриваются булевы сети [2], функциями вершин которых могут быть бинарные логические операции И (конъюнкция), ИЛИ (дизъюнкция), а также унарная операция НЕ (инверсия). Логическая минимизация булевых сетей на основе разложения Шеннона описана в [2] и заключается в поиске такой перестановки участвующих в разложении переменных, при которой число литералов в булевой сети является наименьшим. Литерал – это булева переменная либо ее инверсия. Таким образом, после разложения Шеннона по очередной переменной минимизация булевой сети сводится к следующему: ищутся вершины булевой сети, опирающиеся на одинаковые подсети, после чего проводится редукция (сокращение) сети и находятся уравнения, соответствующие редукцированной сети.

Раздельная BDDI и Bool-минимизация заключаются в нахождении своей перестановки

$\langle x_{i_1}, \dots, x_{i_n} \rangle$ переменных разложения для каждой из функций $f^1(x), \dots, f^m(x)$, в то время как при совместной BDDI- и Bool-минимизации используется одна и та же перестановка переменных разложения для всех функций системы.

Связанность функций в многоуровневом представлении системы булевых функций

Пусть система $F = \{f^1(x), \dots, f^m(x)\}$ булевых функций задана на общем множестве Z взаимосвязанных уравнений (формул) в булевом базисе И, ИЛИ, НЕ – это наборы формул разложений Шеннона либо наборы формул, соответствующих булевым сетям. Каждая формула из Z задает одну промежуточную (внутреннюю) булеву переменную либо функцию, которая является выходной. Обозначим через $R(f^j)$ множество всех формул, задающих промежуточные переменные, связанные с формульным заданием только одной функции $f^j(x)$, $j = 1, \dots, m$, $R(f^j) \subseteq Z$. Заметим, что собственно формула для выходной функции f^j в $R(f^j)$ не включается. Для многоуровневого формульного представления системы $F = \{f^1(x), \dots, f^m(x)\}$ булевых функций через $R(F) = R(f^1, \dots, f^m) = \bigcup_{i=1}^m R(f^i)$ обозначим множество внутренних формул для всех функций системы.

Рассмотрим далее понятие связанности представлений функций, которую будем понимать как общность формул в заданиях функций в виде множеств взаимосвязанных алгебраических логических уравнений. Прежде всего следует отметить, что характеристика связанности трактуется как отношение между парой объектов – компонент системы, в роли которых выступает одна или несколько функций, задаваемых уравнениями рассматриваемой системы. Для определения количественных оценок связанности введем ряд обозначений. Пусть F^g, F^h – подсистемы системы F : $F^g \subset F, F^h \subset F$. Назовем *весом* связанности подсистем F^g, F^h число

$$S(F^g, F^h) = |R(F^g) \cap R(F^h)| = \left| \left(\bigcup_{f^i \in F^g} R(f^i) \right) \cap \left(\bigcup_{f^j \in F^h} R(f^j) \right) \right|, \quad (1)$$

где через $|A|$ обозначена мощность множества A .

В целях нормирования количественных оценок связанности наряду с весом (1) введем понятие меры $M(F^g, F^h)$ связанности подсистем F^g, F^h , вычисляемой по формуле

$$M(F^g, F^h) = \frac{S(F^g, F^h)}{\max(|R(F^g)|, |R(F^h)|)}. \quad (2)$$

Когда каждая из подсистем представляет единственную функцию системы F , формулу (2) можно записать в виде

$$M(f^i, f^j) = \frac{S(f^i, f^j)}{\max(|R(f^i)|, |R(f^j)|)} = \frac{R(f^i) \cap R(f^j)}{\max(|R(f^i)|, |R(f^j)|)}. \quad (3)$$

Под $M(f^i, f^j)$ понимается мера связанности многоуровневых представлений пары функций f^i, f^j .

Предлагаемый далее алгоритм выделения связанных подсистем основан на последовательном пополнении уже выделенной подсистемы $F^m = \{f^1, \dots, f^m\}$, $m \geq 2$, новой функцией f^{m+1} . Под мерой связанности системы функций $F^m \cup f^{m+1}$ будем понимать величину

$$M(F^m \cup \{f^{m+1}\}) = \frac{|R(f^1, \dots, f^m) \cap R(f^{m+1})|}{\max(|R(f^1, \dots, f^m)|, |R(f^{m+1})|)}. \quad (4)$$

В дальнейшем, оценивая меру связанности с использованием формулы (3), будем называть ее мерой связанности пары функций системы F в отличие от меры компонентной связанности, вычисляемой по формуле (4). Предлагаемое в данной работе понятие компонентной связанности подсистемы функций базируется на том, что для различных пар функций из этой подсистемы имеются свои подмножества общих уравнений.

Величина меры связанности всегда находится в пределах отрезка $[0,1]$: от полного различия множеств формул исходного представления функций (либо подсистем функций) до их полного совпадения. Это свойство меры связанности дает возможность определять границы выделяемых для совместной минимизации подсистем, причем при проведении экспериментальных исследований эту меру удобно выражать в процентах.

Примем, что рассматриваемая мера связанности обладает уровнем q , если справедливо $q \leq M(f^i, f^j)$ для связанности пары функций f^i, f^j или $q \leq M(F^m \cup \{f^{m+1}\})$ для компонентной связанности при пополнении системы новой функцией. Далее, когда речь пойдет о связанности систем или подсистем функций, будет подразумеваться связанность их многоуровневых представлений с оценкой по мере связанности последнего проведенного пополнения.

Рассмотрим пример системы булевых функций, заданной в таблице 1.

Таблица 1

Таблица истинности системы полностью определенных булевых функций

Table 1

Truth table of a system of fully defined Boolean functions

| $x_1 x_2 x_3 x_4$ | $f^1 f^2 f^3 f^4$ |
|-------------------|-------------------|
| 0 0 0 0 | 0 0 1 1 |
| 0 0 0 1 | 1 1 0 0 |
| 0 0 1 0 | 1 0 0 0 |
| 0 0 1 1 | 1 1 0 0 |
| 0 1 0 0 | 0 0 1 1 |
| 0 1 0 1 | 1 1 0 0 |
| 0 1 1 0 | 0 1 1 1 |
| 0 1 1 1 | 1 1 1 0 |
| 1 0 0 0 | 0 0 1 1 |
| 1 0 0 1 | 1 0 0 0 |
| 1 0 1 0 | 1 1 0 0 |
| 1 0 1 1 | 1 1 0 0 |
| 1 1 0 0 | 0 0 1 1 |
| 1 1 0 1 | 1 0 1 1 |
| 1 1 1 0 | 0 0 0 1 |
| 1 1 1 1 | 1 0 1 0 |

Для данной системы функций BDDI-представление состоит из 12 формул:

$$\begin{aligned} f^1 &= \overline{x_3}x_4 \vee x_3s_1; & f^2 &= \overline{x_3}s_2 \vee x_3s_3; \\ f^3 &= \overline{x_3}s_4 \vee x_3s_5; & f^4 &= \overline{x_3}s_4 \vee x_3\overline{s_1}; \\ s_1 &= \overline{x_4}x_2 \vee x_4; & s_2 &= \overline{x_4}x_1; & s_3 &= \overline{x_4}s_8 \vee x_4\overline{s_6}; \\ s_4 &= \overline{x_4} \vee x_4s_6; & s_5 &= \overline{x_4}s_7 \vee x_4x_2; \\ s_6 &= x_1x_2; & s_7 &= \overline{x_1}x_2; & s_8 &= \overline{x_1}x_2 \vee x_1x_2. \end{aligned} \quad (5)$$

Множество $R(f^1, f^2, f^3, f^4)$ формул, задающих функции системы $F = \{f^1, f^2, f^3, f^4\}$, состоит из восьми формул, так как первые четыре формулы из (5) не включаются в множество $R(f^1, f^2, f^3, f^4)$. Отдельные функции системы заданы следующими множествами: $R(f^1) = \{s_1\}$, $R(f^2) = \{s_2, s_3, s_6, s_8\}$, $R(f^3) = \{s_4, s_5, s_6, s_7\}$, $R(f^4) = \{s_1, s_4, s_6\}$. Здесь и далее будем задавать множества формул, перечисляя только их левые части. Подсистема $\{f^3, f^4\}$ функций имеет вес связанности $S(f^3, f^4) = 2$ (в задании функций имеются две общие формулы – s_4, s_6 . По формуле (3) мера связанности $M(f^3, f^4) = 2/4 = 0,5$, так как $|R(f^3)| = 4$, $|R(f^4)| = 3$ и поэтому $\max(|R(f^3)|, |R(f^4)|) = \max(4,3) = 4$. Легко заметить, что пара f^2, f^3 имеет меру связанности, равную $1/4$. Мера компонентной связанности (4) подсистемы $\{f^2, f^3\} \cup f^4$ равна $2/7$, так как $R(f^2, f^3) = \{s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$, $R(f^4) = \{s_1, s_4, s_6\}$, $R(f^2, f^3) \cap R(f^4) = \{s_4, s_6\}$, $|R(f^4)| = 2$, $\max(|R(f^2, f^3)|, |R(f^4)|) = 7$. Если же к подсистеме $\{f^2, f^3, f^4\}$ добавить функцию f^1 , то мера ком-

понентной связанности (4) системы $\{f^2, f^3, f^4\} \cup f^1$ будет равна $1/7$, так как подсистема $\{f^2, f^3, f^4\}$ имеет одно общее уравнение s_1 с функцией f^1 .

Рассмотрим другое множество формул, задающих те же функции:

$$\begin{aligned} f^1 &= \overline{t_{12}}; \quad f^2 = t_1 \vee t_2; \quad f^3 = t_3 \vee t_4; \\ f^4 &= t_5 \vee t_6; \quad t_1 = x_1 t_{17}; \quad t_2 = \overline{x_1 t_{13}}; \\ t_3 &= x_1 t_7; \quad t_4 = \overline{x_1 t_{18}}; \quad t_5 = x_1 t_{10}; \\ t_6 &= \overline{x_1 t_{12}}; \quad t_7 = t_8 \vee t_9; \quad t_8 = x_4 x_2; \\ t_9 &= \overline{x_4 x_3}; \quad t_{10} = t_{11} \vee t_{12}; \quad t_{11} = x_4 t_{16}; \\ t_{12} &= \overline{x_4 t_{17}}; \quad t_{13} = x_4 \vee t_{15}; \quad t_{14} = x_4 t_{15}; \\ t_{15} &= x_2 x_3; \quad t_{16} = \overline{x_2 x_3}; \quad t_{17} = \overline{x_2 x_3}; \\ t_{18} &= t_{14} \vee t_{12}. \end{aligned} \tag{6}$$

Как видим, каждая формула в (6) состоит только из одного булева оператора и является менее сложной (содержащей меньшее число логических операторов) по сравнению с формулами (5), задающими разложения Шеннона. Формулы (6) описывают булеву сеть, реализующую систему функций из таблицы 1.

Множества уравнений, задающие отдельные функции системы: $R(f^1) = \{t_{12}, t_{17}\}$; $R(f^2) = \{t_1, t_2, t_{13}, t_{15}, t_{17}\}$; $R(f^3) = \{t_3, t_4, t_7, t_8, t_9, t_{12}, t_{14}, t_{15}, t_{17}, t_{18}\}$; $R(f^4) = \{t_5, t_6, t_{10}, t_{11}, t_{12}, t_{16}, t_{17}\}$.

Подсистема $\{f^3, f^4\}$ функций имеет вес связанности $S(f^3, f^4) = 2$ (в задании функций имеются две общие формулы t_{12} и t_{17}) и меру связанности $M(f^3, f^4) = 2/10 = 0,2$, так как $|R(f^3)| = 10$, $|R(f^4)| = 7$ и $\max(|R(f^3)|, |R(f^4)|) = \max(10, 7) = 10$. Примеры показывают, что различные представления (множества формул) одних и тех же функций могут иметь различные меры связанности.

Выделение подсистем с заданной мерой связанности

Предлагаемый эвристический алгоритм является модификацией предложенного в [13] алгоритма и состоит в последовательном формировании (на каждой итерации i) по текущей (остаточной) системе функций очередной подсистемы P^i функций (подсистема P^i характеризуется мерой компонентной связанности, не меньшей q). На первой итерации ($i = 1$) текущую систему функций образуют функции исходной системы.

На каждой итерации требуется выполнить шаги 1–3.

Шаг 1. Рассмотреть $C_m^2 = m(m-1)/2$ неупорядоченных пар функций $\{f^i, f^j\}$, $i, j = 1, \dots, m$,

$i \neq j$, текущей системы и найти такую пару функций L , которая имеет максимальное значение меры связанности, вычисляемое по формуле (3) и не меньше параметра q . Если таких пар функций несколько, то выбирается первая из них. Если указанной пары функций нет, то переход на шаг 4.

Шаг 2. Составить из функций найденной на первом шаге пары L формируемую подсистему P^i , исключив выбранную пару функций L из текущей системы, и добавлять в формируемую подсистему поочередно те функции f^r , которые находятся с помощью следующей эвристики: из множества функций текущей системы выбирается та функция f^r , которая обеспечивает наибольшее возможное значение меры компонентной связанности (4) для подсистемы $P^i \cup \{f^r\}$. Если таких функций несколько, то выбирается и добавляется в формируемую подсистему P^i первая из них.

Шаг 3. Если нет ни одной такой функции f^r , что подсистема $P^i \cup \{f^r\}$ имеет меру компонентной связанности, не меньшую q , то закончить формирование подсистемы P^i и объявить не входящие в нее функции текущей подсистемой. Переход на шаг 1 для формирования подсистемы на итерации $i + 1$.

Шаг 4. Закончить формирование подсистем, когда все функции текущей системы будут включены в формируемые подсистемы, либо в текущей подсистеме нельзя будет найти ни одной пары функций, характеризуемых мерой (3) связанности, не меньшей q , либо в текущей системе имеется только одна функция. Конец алгоритма.

Достоинствами данного алгоритма являются его высокое быстродействие и возможность обобщения на формулы AIG, MIG, XMG, XAIG, а недостатком – зависимость от последовательности формирования результирующих выделенных подсистем. Чтобы подсчитать меру связанности результирующей выделенной подсистемы, надо знать последнюю добавляемую в подсистему функцию и, очевидно, ту подсистему, в которую она добавляется.

Приведем пример работы алгоритма на формулах (6) при значении допустимого уровня связанности $q = 0,2$ (20 %).

Шаг 1. Рассмотрим все неупорядоченные пары функций из формул (6) исходной системы и найдем для каждой из пар меру связанности (3): $M(f^1, f^2) = 1/5$; $M(f^1, f^3) = 2/10$; $M(f^1, f^4) = 2/7$; $M(f^2, f^3) = 2/10$; $M(f^2, f^4) = 1/7$; $M(f^3, f^4) = 2/10$.

Исходя из полученных оценок мер связанности пар функций при уровне связанности $q = 0,2$, результатом выполнения шага 1 является пара $\{f^1, f^4\}$, имеющая наибольшую меру связанности $M(f^1, f^4) = 2/7$.

Шаг 2. К подмножеству $\{f^1, f^4\}$ функций будем добавлять поочередно функции f^2, f^3 и находить меры компонентной связанности: $M(\{f^1, f^4\} \cup \{f^2\}) = 1/7, M(\{f^1, f^4\} \cup \{f^3\}) = 2/10$. Для подсистемы $(\{f^1, f^4\}, f^2): R(f^1, f^4) = \{t_5, t_6, t_{10}, t_{11}, t_{12}, t_{16}, t_{17}\}, R(f^2) = \{t_1, t_2, t_{13}, t_{15}, t_{17}\}, R(f^1, f^4) \cap R(f^2) = \{t_{17}\}, |\{t_{17}\}| = 1, \max(|R(f^1, f^4)|, |R(f^2)|) = \max(7, 5) = 7$, откуда $M(\{f^1, f^4\} \cup \{f^2\}) = 1/7$, что меньше $0,2$.

Шаг 3. Для подсистемы $(\{f^1, f^4\}, f^3)$ значение $M(\{f^1, f^4\} \cup \{f^3\}) = 0,2$, поэтому на шаге 3 выделенной подсистемой объявляется $\{f^1, f^4, f^3\}$. Добавить к этой подсистеме функцию f^2 не удастся, так как полученное значение меры компонентной связанности будет меньше $0,2$. Текущей подсистемой объявляется $\{f^2\}$, и выполняется переход на шаг 1.

Шаг 4. В текущей подсистеме остается одна функция, поэтому алгоритм прекращает работу. Результатом являются одна выделенная подсистема $\{f^1, f^4, f^3\}$ и остаток – подсистема $\{f^2\}$.

Если провести синтез схемы непосредственно по формулам (6) (Bool-представлению), то будет получена схема с площадью $4\,336$ условных единиц, состоящая из 12 элементов, задержка схемы составит $2,78$ нс. Если же провести совместную минимизацию в классе булевых сетей подсистемы $\{f^1, f^4, f^3\}$ и совместную минимизацию подсистемы $\{f^2\}$

(для этой подсистемы совместная минимизация вырождается в раздельную одной функции), то в результате синтеза получим схему (см. рисунок) с площадью $4\,262$ условные единицы, состоящую из 12 элементов, задержка схемы составит $3,21$ нс. Функции логических элементов схемы и их площади приведены в таблице 2. Таким образом, параметры схемы изменились – площадь схемы уменьшилась, задержка увеличилась.

Таблица 2

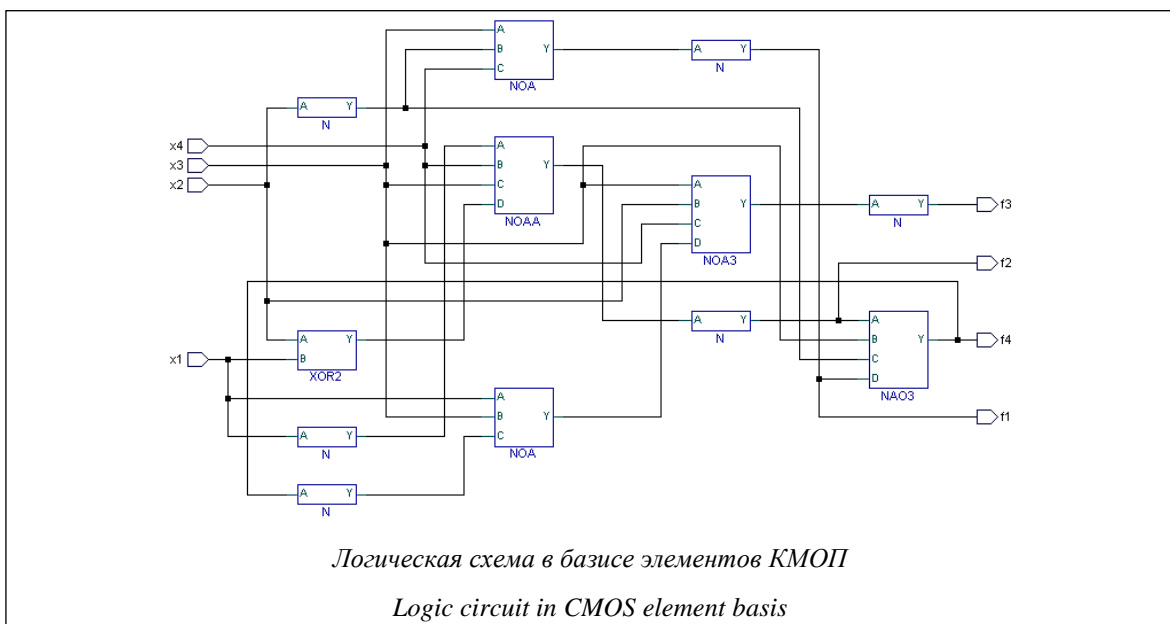
Функции и площади элементов КМОП логической схемы

Table 2

Functions and areas of CMOS logic circuit elements

| Элемент | Функция | Площадь |
|---------|--|---------|
| N | $y = \bar{A}$ | 223 |
| NOA | $y = \overline{(AB) \vee C}$ | 363 |
| NOA3 | $y = \overline{(A \vee B \vee C) \bar{D}}$ | 491 |
| XOR2 | $y = A \oplus B$ | 692 |
| NOAA | $y = \overline{AB \vee CD}$ | 485 |
| NOA3 | $y = \overline{(ABC) \vee D}$ | 530 |

Если же реализовать схему по уравнениям (5) (BDDI-представлению), то получим схему с площадью $4\,659$ условных единиц, состоящую из 14 элементов, задержка схемы составит $1,72$ нс. Данный пример показывает, что различные способы минимизации многоуровневых представлений позволяют получать схемы различной площади и задержки. Проведенные про-



граммные эксперименты проверяют конкурентоспособность трех подходов к минимизации представлений на основе двух моделей – BDDI и Bool: это совместная минимизация, раздельная минимизация и выделение отдельных подсистем и их (подсистем) совместная минимизация.

Исходные данные для экспериментов

Исходными описаниями блоков комбинационной логики являлись 39 примеров систем булевых функций на языке SF [14] в виде матричных описаний систем ДНФ – SDF описаний языка SF.

В матричной форме система ДНФ

$$\begin{aligned}
 f^1 &= \overline{x_2}x_3 \vee x_4; \\
 f^2 &= \overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3 \vee x_1x_4; \\
 f^3 &= \overline{x_1}x_2x_3 \vee x_1x_2\overline{x_3} \vee x_2x_3x_4 \vee \overline{x_3}x_4; \\
 f^4 &= \overline{x_1}x_2x_3 \vee x_2x_4 \vee \overline{x_3}x_4
 \end{aligned}
 \tag{7}$$

задается парой матриц: строки троичной матрицы T^x представляют элементарные конъюнкции, а единичные значения элементов в булевой матрице B^f отмечают вхождения соответствующих конъюнкций в ДНФ функций (табл. 3).

Таблица 3

Система ДНФ булевых функций

Table 3

DNF system of Boolean functions

| T^x | B^f |
|-------------------|-------------------|
| $x_1 x_2 x_3 x_4$ | $f^1 f^2 f^3 f^4$ |
| – 0 1 – | 1 0 0 0 |
| 0 1 1 – | 0 1 1 0 |
| – – – 1 | 1 0 0 0 |
| 1 0 1 – | 0 1 0 0 |
| 1 1 0 – | 0 0 1 1 |
| – 1 – 0 | 0 0 0 1 |
| – 1 1 1 | 0 0 1 0 |
| 0 – – 1 | 0 1 0 0 |
| – – 0 0 | 0 0 1 1 |

Заметим, что данная система ДНФ задается логическими уравнениями (формат LOG языка SF) и представляет ту же систему булевых функций, которая приведена в таблице 1. Таблице истинности соответствует система совершенных ДНФ (СДНФ), если перейти к формульной записи функций.

Из 39 примеров четыре примера (Ttt2, X1, X3, X4) многоуровневых представлений систем функций были преобразованы в матричный формат SDF (добавлялось окончание имени _matr). Примеры Mult_7, Mult_8 – системы ДНФ, задающие функции семи- и вось-

миразрядных умножителей. Описания функций умножителей были получены командой unpar в синтезаторе LeonardoSpectrum из схемных реализаций умножителей, после чего был осуществлен переход к ДНФ. Заметим, что так может быть осуществлено перепроектирование устройств комбинационной логики из одного базиса логических элементов в другой базис. Примеры Verg_1, Verg_2, Sist_4 взяты из практики проектирования цифровых устройств. Пример Psevdo_1 – сгенерированный псевдослучайный пример. Остальные примеры взяты из библиотеки примеров схем (<http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex>), представленных в формате PLA, при этом описания из формата PLA переводились в формат SDF системы FLC-2 [14]. Было установлено, что примеры GARY и IN0 задают одну и ту же систему булевых функций в виде различных систем ДНФ, то есть ДНФ с различными множествами элементарных конъюнкций.

Параметры примеров систем функций заданы в таблице 4: n – число аргументов; m – число функций системы ДНФ, заданной на k общих элементарных конъюнкциях; Area – суммарная площадь элементов схемы в условных единицах; Delay – временная задержка схемы (нс). Если $k = 2^n$, то исходной является матричная форма таблицы истинности системы функций. В таблицах 5–7 символом «*» выделены лучшие решения (в рамках отдельной таблицы) для испытываемого примера – меньшие значения параметров площади и временной задержки.

Организация экспериментов

В экспериментах использовались система FLC-2 [14] и следующие программы.

BDD_Builder – программа минимизации раздельных и совместных BDDI-представлений системы булевых функций.

BoolNetOpt2 – программа минимизации раздельных и совместных Bool-представлений системы булевых функций.

Splitter – программа выделения подсистем функций (при заданном значении q меры связанности) из совместных BDDI и из совместных Bool-представлений систем функций.

AutoSplit – программа нахождения лучшего значения q меры связанности по критерию минимальности общего числа литералов в BDDI-либо в Bool-представлениях систем функций.

Использовались также перечисленные далее стратегии обработки иерархических пред-

Таблица 4
**Параметры систем функций,
 участвующих в экспериментах**
 Таблица 4
**Parameters of function systems
 participating in experiments**

| № | Пример | <i>n</i> | <i>m</i> | <i>k</i> |
|----|-----------|----------|----------|----------|
| 1 | Add6 | 12 | 7 | 1 092 |
| 2 | B12 | 15 | 9 | 431 |
| 3 | B9 | 16 | 5 | 123 |
| 4 | Dist | 8 | 5 | 256 |
| 5 | Gary | 15 | 11 | 442 |
| 6 | Ibm | 48 | 17 | 173 |
| 7 | In0 | 15 | 11 | 138 |
| 8 | In1 | 16 | 17 | 110 |
| 9 | In2 | 19 | 10 | 137 |
| 10 | Intb | 15 | 7 | 664 |
| 11 | Jbp | 36 | 57 | 166 |
| 12 | M181 | 15 | 9 | 430 |
| 13 | M3 | 8 | 16 | 128 |
| 14 | M4 | 8 | 16 | 256 |
| 15 | Max1024 | 10 | 6 | 1 024 |
| 16 | Max512 | 9 | 6 | 512 |
| 17 | Mpl4 | 8 | 8 | 256 |
| 18 | Mp2d | 14 | 14 | 123 |
| 19 | Mult_7 | 14 | 14 | 13 060 |
| 20 | Mult_8 | 16 | 16 | 52 810 |
| 21 | P82 | 5 | 14 | 24 |
| 22 | Psevdo1 | 10 | 20 | 1 000 |
| 23 | Radd | 8 | 5 | 120 |
| 24 | Root | 8 | 5 | 256 |
| 25 | Sist_4 | 17 | 12 | 370 |
| 26 | Soar | 83 | 94 | 529 |
| 27 | Sqn | 7 | 3 | 96 |
| 28 | Tial | 14 | 8 | 640 |
| 29 | Tms | 8 | 16 | 30 |
| 30 | Ttt2_matr | 24 | 21 | 222 |
| 31 | Verg_1 | 17 | 61 | 2 004 |
| 32 | Verg_2 | 18 | 63 | 2 129 |
| 33 | Vtx1 | 27 | 6 | 110 |
| 34 | X1_matr | 51 | 35 | 324 |
| 35 | X3_matr | 135 | 99 | 915 |
| 36 | X4_matr | 94 | 71 | 371 |
| 37 | X9dn | 27 | 7 | 120 |
| 38 | Z4 | 7 | 4 | 128 |
| 39 | Z5xpl | 7 | 10 | 128 |

ставлений систем перечисленных далее функций.

«Представить листья сети в виде LOG» – листья иерархического описания логической сети представлялись в формате LOG (логических уравнений), пример – уравнения (7).

«Представить листья сети в виде SDF» – листья иерархического описания логической сети

представлялись в виде SDF (матричных описаний), пример – таблица 3.

«Представить листья сети в виде BDDI» – каждый лист SDF иерархического описания логической сети представлялся в виде уравнений, соответствующих BDDI-описанию, пример таких описаний – уравнения (5).

«Представить листья сети в виде Bool» – каждый лист SDF иерархического описания логической сети представлялся в виде уравнений, соответствующих Bool-описанию, пример таких описаний – уравнения (6).

Исходными для программы BoolNetOpt2 [2] являются логические уравнения, поэтому исходные матричные представления (формат SDF системы FLC-2) переводились в формат LOG логических уравнений. Для отдельной Bool-минимизации для такого перевода использовалась стратегия «Представить листья сети в виде LOG». После при отдельной BDDI-минимизации и после выделения подсистем программой Splitter проводилась их BDDI-минимизация с помощью стратегий «Представить листья сети в виде SDF» и «Представить листья сети в виде BDDI».

При Bool-минимизации отдельные функции системы (либо выделенные программой Splitter подсистемы) обрабатывались с помощью стратегий «Представить листья сети в виде SDF», «Представить листья сети в виде LOG», «Представить листья сети в виде Bool». При выполнении стратегии «Представить листья сети в виде Bool» для каждого листа в иерархическом описании схемы (листу соответствует подсистема функций) применялась программа BoolNetOpt2 со следующим ограничением – 10 минут для совместной минимизации подсистемы. При отдельной Bool-минимизации задавалась одна минута для минимизации булевой сети, реализующей одну функцию исходной системы.

Аппарат стратегий обеспечивал для каждой функции сети (либо подсистемы) нахождение своей перестановки переменных при BDDI-либо Bool-минимизации.

Эксперименты были организованы следующим образом. Сначала выполнялась логическая минимизация, затем минимизированные описания многоуровневых представлений систем функций конвертировались в VHDL-описания [15] и подавались на вход того же синтезатора LeonardoSpectrum, который использовался в экспериментах [2]. Для каждого из примеров синтез осуществлялся с одними и теми же опциями управления синтезом и для одной и той же, что и в работе [2], целевой биб-

лиотеки синтеза – библиотеки проектирования заказных цифровых КМОП СБИС.

Эксперименты

Эксперимент 1. Совместная и раздельная минимизация в классе BDDI-представлений. При раздельной минимизации исходные матричные представления систем функций представлялись в виде логических сетей, элементами которых являются отдельные функции системы. Затем данные сети обрабатывались при помощи стратегии «Представить листья сети в виде BDDI». Для минимизации BDDI-

представлений всей системы и отдельных функций использовалась программа BDD_Builder.

Эксперимент 2. Исходными для эксперимента 2 являлись совместно минимизированные BDDI-представления, для которых выполнялась программа Splitter выделения связанных подсистем функций для двух фиксированных значений меры связанности $q = 10\%$, $q = 20\%$. Полученные сети подсистем обрабатывались с помощью стратегий «Представить листья сети в виде SDF», «Представить листья сети в виде BDDI». Результаты экспериментов 1 и 2 отражены в таблице 5.

Таблица 5

Результаты экспериментов 1 и 2 для BDDI-представлений

Table 5

Results of experiments 1 and 2 for BDDI-views

| № | Пример | Эксперимент 1 | | | | Эксперимент 2. Выделение подсистем | | | |
|----|----------------|-----------------|---------------|-----------------|---------------|------------------------------------|---------------|-----------------|---------------|
| | | Совместные BDDI | | Раздельные BDDI | | $q = 10\%$ | | $q = 20\%$ | |
| | | Area | Delay | Area | Delay | Area | Delay | Area | Delay |
| 1 | Add6 | *16 534 | *6.82 | 20 518 | 7.80 | 17 839 | 7.56 | 17 839 | 7.56 |
| 2 | B12 | 17 064 | 3.04 | 15 496 | 2.51 | 15 819 | 2.81 | *15 239 | *2.50 |
| 3 | B9 | 32 247 | *5.32 | *31 376 | 5.72 | 31 722 | 6.27 | 31 879 | 6.03 |
| 4 | Dist | *62 808 | 6.41 | 73 991 | 6.46 | *62 808 | *6.05 | 63 362 | 6.31 |
| 5 | Gary | *91 395 | 7.32 | 93 984 | *6.61 | 92 829 | 8.57 | 93 007 | 7.60 |
| 6 | Ibm | 66 776 | 5.62 | *53 406 | 5.37 | 59 667 | *5.32 | 66 776 | 5.62 |
| 7 | In0 | 93 058 | 7.45 | *90 156 | *6.23 | 92 349 | 6.93 | 93 007 | 7.60 |
| 8 | In1 | *166 758 | *7.54 | 283 525 | 7.58 | 263 449 | 8.87 | 263 393 | 9.68 |
| 9 | In2 | *70 291 | 7.32 | 87 439 | *6.20 | 82 138 | 6.81 | 79 309 | 6.47 |
| 10 | Intb | 235 883 | 9.51 | 285 930 | 9.52 | *233 707 | *8.75 | 234 600 | 9.17 |
| 11 | Jbp | 98 308 | *5.22 | *91 205 | 7.17 | 96 339 | 5.58 | 95 044 | 5.69 |
| 12 | M181 | 16 595 | 3.60 | *15 557 | *2.60 | 15 875 | 3.22 | 16 667 | 2.69 |
| 13 | M3 | *51 632 | 5.13 | 60 437 | 4.59 | 57 519 | 5.33 | 57 892 | *4.29 |
| 14 | M4 | *78 181 | 5.87 | 99 006 | *4.68 | 88 990 | 5.44 | 90 563 | 5.31 |
| 15 | Max1024 | *148 846 | 7.66 | 202 660 | 8.07 | 191 974 | *6.75 | 181 272 | 7.44 |
| 16 | Max512 | 84 682 | 5.91 | *82 138 | *5.68 | 85 904 | 5.74 | 88 337 | 5.89 |
| 17 | Mlp4 | 71 095 | 6.09 | *68 070 | *5.41 | 72 389 | 7.17 | 72 546 | 5.51 |
| 18 | Mp2d | 18 805 | 4.87 | 19 173 | *3.74 | 18 124 | 3.94 | *17 376 | 5.17 |
| 19 | Mult_7 | 2 492 681 | 14.28 | 2 734 390 | 15.20 | *2 372 437 | *13.44 | 2 793 560 | 15.26 |
| 20 | Mult_8 | 7 069 051 | *15.99 | 7 291 481 | 16.31 | *6 735 735 | 16.01 | 7 069 051 | *15.99 |
| 21 | P82 | 21 355 | 3.49 | 20 216 | *2.92 | 20 869 | 3.13 | *20 205 | 2.95 |
| 22 | Psevdo1 | *766 686 | 9.01 | 813 464 | *8.04 | 858 622 | 9.28 | 851 302 | 8.79 |
| 23 | Radd | 11 802 | *3.69 | *8 911 | 4.77 | 9 480 | 3.85 | 9 480 | 3.85 |
| 24 | Root | *26 075 | 4.78 | 30 584 | *4.13 | *26 075 | 4.78 | *26 075 | 4.78 |
| 25 | Sist_4 | 112 722 | 7.21 | 120 037 | *6.82 | 118 569 | 6.94 | *112 565 | 7.56 |
| 26 | Soar | 163 957 | 6.14 | 158 388 | 5.58 | *147 976 | 6.05 | 156 368 | *5.20 |
| 27 | Sqn | 22 878 | 3.99 | 22 683 | *3.33 | 22 526 | 3.74 | *20 975 | 3.55 |
| 28 | Tial | *279 458 | 9.08 | 384 172 | 10.96 | 292 721 | *8.99 | 330 197 | 9.08 |
| 29 | Tms | *39 004 | 4.65 | 44 952 | 5.38 | 43 708 | 5.13 | 46 560 | *4.57 |
| 30 | Ttt2_matr | 76 429 | 5.01 | *45 683 | *4.41 | 64 321 | 5.52 | 77 439 | 6.00 |
| 31 | Verg_1 | *465 032 | 14.41 | 939 081 | *12.21 | 801 790 | 12.77 | 845 839 | 12.28 |
| 32 | Verg_2 | *536 037 | 14.15 | 1 082 989 | 13.28 | 917 430 | 15.31 | 988 687 | 13.12 |
| 33 | Vtx1 | 31 600 | *4.23 | 40 103 | 4.89 | 28 285 | 4.66 | *27 370 | 5.05 |
| 34 | X1_matr | 107 912 | 6.45 | *71 720 | *4.46 | 82 528 | 5.52 | 108 542 | 6.91 |
| 35 | X3_matr | 211 633 | *5.73 | 223 222 | 6.51 | *199 318 | 5.82 | 207 308 | 6.01 |
| 36 | X4_matr | *120 031 | 6.17 | 133 959 | *5.99 | 157 417 | 6.51 | 127 503 | 6.08 |
| 37 | X9dn | 34 719 | 6.58 | 41 939 | 5.09 | *30 327 | *4.31 | 34 278 | 5.34 |
| 38 | Z4 | *6 640 | 4.25 | 7 700 | *3.57 | 6 640 | 4.25 | 6 640 | 4.25 |
| 39 | Z5xp1 | *22 582 | 5.24 | 30 417 | 3.51 | 24 926 | *2.97 | 26 120 | 3.30 |
| | Лучших решений | 17 | 8 | 10 | 18 | 8 | 8 | 7 | 5 |

Эксперимент 3. Совместная и раздельная минимизация в классе булевых сетей (Bool-представлений). При раздельной минимизации в классе Bool-представлений элементы логической сети (отдельные функции системы) обрабатывались с помощью стратегий «Представить листья сети в виде SDF», «Представить листья сети в виде LOG», «Представить листья сети в виде Bool». Базовой программой являлась программа минимизации булевых сетей BoolNetOpt2 [2].

Под Bool-представлениями в системе FLC-2 понимаются булевы сети, описываемые логическими уравнениями с одним логи-

ческим оператором дизъюнкции либо конъюнкции над двумя литералами булевых переменных.

Эксперимент 4. Исходными для эксперимента 4 являлись совместно минимизированные Bool-представления, для которых выполнялась программа Splitter выделения q -связанных подсистем функций для двух фиксированных значений меры связанности $q = 5\%$, $q = 20\%$. Полученная сеть подсистем обрабатывалась с помощью тех же трех стратегий, что и в эксперименте 3.

Результаты экспериментов 3 и 4 сведены в таблицу 6.

Таблица 6

Результаты экспериментов 3 и 4 для Bool-представлений

Table 6

Results of experiments 3 and 4 for Bool-views

| № | Пример | Эксперимент 3 | | | | Эксперимент 4. Выделение подсистем | | | |
|----------------|-----------|-----------------|--------------|-------------------|---------------|------------------------------------|---------------|-----------------|--------------|
| | | Совместные Bool | | Раздельные Bool | | $q = 5\%$ | | $q = 20\%$ | |
| | | Area | Delay | Area | Delay | Area | Delay | Area | Delay |
| 1 | Add6 | *18 314 | 6.18 | 29 574 | 4.24 | 20 389 | *4.18 | 20 389 | *4.18 |
| 2 | B12 | 15 936 | 3.55 | 17 075 | 3.38 | 15 295 | *2.80 | *15 105 | 2.99 |
| 3 | B9 | *22 454 | 4.69 | 25 607 | 4.06 | 28 793 | *3.42 | 30 316 | 4.20 |
| 4 | Dist | 66 134 | *5.33 | 80 603 | 5.98 | *63 249 | 5.52 | 69 834 | 5.79 |
| 5 | Gary | 97 795 | 7.94 | 91 802 | 7.96 | 92 996 | *6.57 | *89 821 | 7.31 |
| 6 | Ibm | 66 776 | 5.62 | 63 891 | 6.45 | 90 647 | 7.33 | *61 720 | *4.83 |
| 7 | In0 | *95.792 | *6.20 | 96 506 | 6.87 | *95 792 | 6.95 | 98 727 | 6.75 |
| 8 | In1 | *162 478 | 8.52 | 254 325 | *7.86 | 187 650 | 8.78 | 191 210 | 10.64 |
| 9 | In2 | 93 716 | 6.25 | *85 592 | *5.84 | 87 807 | 6.46 | 95 044 | 6.04 |
| 10 | Intb | *253 299 | 9.30 | 338 594 | 9.58 | 254 816 | *9.05 | 255 458 | 9.08 |
| 11 | Jbp | 98 308 | *5.22 | 97 114 | 5.69 | 103 358 | 5.84 | *94 827 | 5.29 |
| 12 | M181 | 16 204 | 3.02 | *15 546 | *2.93 | 16 070 | 3.39 | 16 316 | 3.02 |
| 13 | M3 | 65 414 | 5.14 | *59 823 | *4.66 | 74 292 | 6.67 | 61 603 | 5.23 |
| 14 | M4 | *93 153 | 5.24 | 98 308 | *4.79 | 94 732 | 4.93 | 106 617 | 4.89 |
| 15 | Max1024 | 129 478 | 7.38 | 200 026 | 7.67 | 129 272 | *7.09 | *123 112 | 7.81 |
| 16 | Max512 | *72 300 | 6.26 | 98 945 | 6.74 | 79 721 | 6.99 | 73 840 | *5.49 |
| 17 | Mlp4 | 68 773 | 4.95 | 74 560 | 4.96 | *67 987 | 5.83 | 68 277 | *4.82 |
| 18 | Mp2d | *17 203 | 3.81 | 19 329 | *3.67 | 17 432 | 5.05 | 18 833 | 4.01 |
| 19 | Mult_7 | 2 298 056 | 13.34 | 2 068 489 | 13.00 | *2 062 212 | *12.61 | 2 367 148 | 13.26 |
| 20 | Mult_8 | 5 938 638 | 15.43 | *5 536 956 | 15.82 | 6 503 066 | *14.96 | 5 960 868 | 15.80 |
| 21 | P82 | *19 056 | 3.23 | 19 792 | *2.79 | 19 748 | 3.23 | 19 167 | 3.26 |
| 22 | Psevdo1 | 735 796 | 8.91 | 796 361 | *8.29 | *729 507 | 8.42 | 731 376 | 9.55 |
| 23 | Radd | *7 399 | 3.99 | 12 282 | *3.26 | *7 399 | 3.99 | *7 399 | 3.99 |
| 24 | Root | 31 773 | *4.10 | 38 139 | 4.43 | 36 331 | 5.49 | *28 341 | 4.41 |
| 25 | Sist_4 | 128 351 | 7.38 | *121 845 | *6.27 | 153 941 | 8.06 | 139 271 | 8.59 |
| 26 | Soar | 151 787 | 6.52 | 151 793 | *5.41 | *140 019 | 6.10 | 152 485 | 5.59 |
| 27 | Sqn | 26 695 | 4.23 | 23 715 | 3.96 | *22 008 | *3.50 | 22 532 | 3.55 |
| 28 | Tial | 351 607 | 9.79 | *337 099 | *9.85 | 389 679 | 11.13 | 375 595 | 10.59 |
| 29 | Tms | 50 471 | 6.88 | *46 202 | *4.88 | 55 080 | 6.23 | 52 435 | 5.26 |
| 30 | Ttt2_matr | 52 653 | *4.05 | *46 721 | 4.93 | 51 777 | 5.94 | 48 752 | 4.52 |
| 31 | Verg_1 | *634 954 | 19.25 | 1 145 446 | *11.23 | 999 679 | 15.64 | 1 051 674 | 13.74 |
| 32 | Verg_2 | *694 425 | 18.90 | 1 212 986 | *12.21 | 987 481 | 15.84 | 1 059 960 | 13.80 |
| 33 | Vtx1 | *28 112 | *4.31 | 32 353 | 5.55 | 31 376 | 4.51 | 29 892 | 5.31 |
| 34 | X1_matr | 107 912 | 6.45 | *71 441 | *4.41 | 93 727 | 7.15 | 90 848 | 7.02 |
| 35 | X3_matr | *211 633 | *5.73 | 212 910 | 5.82 | 213 162 | 7.41 | 219 785 | 7.15 |
| 36 | X4_matr | 120 031 | 6.17 | 119 194 | 4.84 | *109 680 | 6.31 | 116 432 | *4.51 |
| 37 | X9dn | *20 741 | 5.48 | 21 310 | *4.89 | 49 924 | 5.22 | 38 725 | 5.62 |
| 38 | Z4 | 6 685 | 3.83 | 10 284 | *3.11 | 6 685 | 3.83 | *6 411 | 3.74 |
| 39 | Z5xp1 | 26 014 | 3.76 | 26 594 | *3.44 | *20 267 | 4.64 | 23 536 | 5.75 |
| Лучших решений | | 15 | 7 | 9 | 19 | 10 | 9 | 8 | 5 |

Эксперимент 5. Оценка выделения связанных подсистем. Для выделения совместно минимизируемых связанных подсистем с помощью программы AutoSplit ищутся лучшие значения параметра связанности q для BDDI-представлений путем итеративного выполнения программы Splitter для меры связанности 5 %, 10 %, 15 %, 20 %, ..., 95 %. Оценивается суммарное число литералов в каждой полученной логической сети, и по минимуму найденной оценки выбирается лучший вариант решения. Естественно, если для значения q меры

связанности подсистемы не выделялись, то программа AutoSplit прекращала испытывать большие, чем q , значения меры связанности и переходила к выбору решения по критерию меньшего числа литералов. Полученная логическая сеть обрабатывалась с помощью тех стратегий, что и в эксперименте 2. Понятно, что выбор лучшего значения связанности целесообразно находить после выполнения синтеза схемы, но тогда трудоемкая процедура синтеза должна выполняться многократно, поэтому вычислялась погрешность δ эвристического

Таблица 7

Результаты экспериментов 5 и 6

Table 7

Results of experiments 5 and 6

| № | Пример | Эксперимент 5 | | | | Эксперимент 6 | | | |
|---------------------|-----------|---------------|-------------------|-------|--------------|---------------|-------------------|-------|--------------|
| | | $q, \%$ | Area | Delay | $\delta, \%$ | $q, \%$ | Area | Delay | $\delta, \%$ |
| 1 | Add6 | 5 | 17 839 | 7,56 | 0 | 5 | 20 389 | 4,18 | 0 |
| 2 | B12 | 5 | *15 819 | 2,81 | 3,8 | 10 | *15 674 | 3,13 | 3,7 |
| 3 | B9 | 5 | 31 879 | 6,03 | 0,5 | 5 | 28 793 | 3,42 | 6,1 |
| 4 | Dist | 5 | 62 808 | 6,41 | 0 | 5 | *63 249 | 5,52 | 0 |
| 5 | Gary | 20 | 93 007 | 7,6 | 0,2 | 45 | 93 113 | 7,47 | 3,6 |
| 6 | Ibm | 10 | 59 667 | 5,32 | 0 | 35 | 93 247 | 8,42 | 51 |
| 7 | In0 | 20 | 93 007 | 7,6 | 0,7 | 20 | 98 727 | 6,75 | 5,2 |
| 8 | In1 | 80 | 186 947 | 7,9 | 0 | 75 | 164 616 | 9,12 | 0 |
| 9 | In2 | 20 | 79 309 | 6,47 | 0 | 5 | 87 807 | 6,46 | 3,4 |
| 10 | Intb | 10 | *233 707 | 8,75 | 0 | 5 | 254 816 | 9,05 | 0 |
| 11 | Jbp | 55 | 100 596 | 5,89 | 5,8 | 60 | 109 591 | 6 | 16 |
| 12 | M181 | 5 | 15 875 | 3,22 | 0 | 10 | 15 808 | 2,72 | 0 |
| 13 | M3 | 40 | 51 632 | 5,13 | 0 | 30 | 70 570 | 5,2 | 14,5 |
| 14 | M4 | 60 | 78 237 | 5,91 | 0 | 45 | 96 043 | 5,59 | 1,3 |
| 15 | Max1024 | 5 | 191 974 | 6,75 | 5,9 | 5 | *129 272 | 7,09 | 5 |
| 16 | Max512 | 5 | 85 904 | 5,74 | 0 | 5 | 79 721 | 6,99 | 7,9 |
| 17 | Mlp4 | 5 | 72 389 | 7,17 | 0 | 5 | *67 987 | 5,83 | 0 |
| 18 | Mp2d | 95 | *18 369 | 4,87 | 5,7 | 55 | 17 410 | 4,37 | 0 |
| 19 | Mult_7 | 10 | *2 372 437 | 13,44 | 0 | 5 | *2 062 212 | 12,61 | 0 |
| 20 | Mult_8 | 10 | *6 735 735 | 16,01 | 0 | 45 | 6 145 572 | 16,66 | 4 |
| 21 | P82 | 10 | 20 869 | 3,13 | 3,2 | 50 | 20 886 | 3,3 | 8,9 |
| 22 | Pseudol | 25 | 781 981 | 11,03 | 0 | 5 | *729 507 | 8,42 | 0 |
| 23 | Radd | 5 | 9 480 | 3,85 | 0 | 5 | 7 399 | 3,99 | 0 |
| 24 | Root | 50 | 24 585 | 5,21 | 0 | 25 | 33 195 | 4,33 | 17,1 |
| 25 | Sist_4 | 25 | 130 432 | 8,38 | 15,8 | 35 | 183 286 | 9,58 | 31,6 |
| 26 | Soar | 90 | 171 836 | 5,79 | 16,1 | 75 | *148 417 | 5,81 | 5,9 |
| 27 | Sqn | 30 | *22 526 | 3,74 | 2,3 | 5 | *22 008 | 3,5 | 0 |
| 28 | Tial | 5 | 294 724 | 8,73 | 0,6 | 5 | 389 679 | 11,13 | 3,7 |
| 29 | Tms | 55 | 43 814 | 5,2 | 0,2 | 80 | 48 161 | 5,62 | 0 |
| 30 | Ttt2_matr | 20 | 77 439 | 6 | 22,5 | 30 | 49 517 | 4,89 | 1,5 |
| 31 | Verg_1 | 90 | 484 924 | 15,05 | 0 | 95 | 672 239 | 15,73 | 0 |
| 32 | Verg_2 | 85 | 634 385 | 14,16 | 0 | 90 | 759 990 | 16,69 | 0 |
| 33 | Vtx1 | 5 | *28 285 | 4,66 | 3,3 | 5 | 31 376 | 4,51 | 16,3 |
| 34 | X1_matr | 5 | 120 232 | 8,34 | 45,6 | 10 | 108 503 | 8,02 | 19,4 |
| 35 | X3_matr | 70 | 211 912 | 5,76 | 6,3 | 15 | 217 001 | 8,47 | 1,8 |
| 36 | X4_matr | 5 | 126 962 | 6,9 | 0 | 5 | *109 680 | 6,31 | 0 |
| 37 | X9dn | 55 | *33 430 | 5,18 | 10,2 | 70 | 43 368 | 5,17 | 17,6 |
| 38 | Z4 | 50 | 6 640 | 4,25 | 3,5 | 40 | 6 685 | 3,83 | 4,2 |
| 39 | Z5xp1 | 5 | 26 416 | 3,39 | 1,1 | 5 | *20 267 | 4,64 | 0 |
| Средняя погрешность | | | | | 3,9 | - | - | - | 6,4 |

критерия «Общее число литералов» при выборе решения – лучшего результата выделения связанных подсистем. Например, в таблице 7 для примера B12 (BDDI-представление) указана погрешность $\delta = 3,8\%$. Погрешность δ вычислялась следующим образом: программа AutoSplit нашла лучшее значение $q = 5\%$ меры связанности, программа Splitter выделила подсистемы, которые были минимизированы, синтез схемы по полученному описанию подсистем позволил получить схему с площадью Area = 15 819, однако ручной перебор значений q (табл. 5) и синтез двух схем позволили получить схему площадью Area = 15 239 для значения $q = 20\%$. Погрешность составила $(15\,819 - 15\,239) / 15\,239 = 0,038$, что эквивалентно $3,8\%$.

Эксперимент 6. Совпадает с экспериментом 5, при этом исходными были совместно минимизированные Bool-представления, а результирующие логические сети обрабатывались с помощью тех же стратегий, что и в эксперименте 3.

Итоги экспериментов 5 и 6 отражены в таблице 7.

Обсуждение результатов экспериментов

Результаты экспериментов 1 и 2 (табл. 5) показывают, что предложенный алгоритм выделения подсистем в 13 примерах схем (из 39 примеров) позволил получить схемы меньшей площади для исходных совместных BDDI-представлений, из которых выделялись подсистемы. Для 17 примеров схем лучшие результаты показала совместная минимизация, для 10 – раздельная минимизация BDDI-представлений. Получение схем с меньшей задержкой чаще было обеспечено применением раздельной BDDI-минимизации и алгоритмом выделения подсистем.

При выделении подсистем из совместных Bool-представлений (табл. 6) примерно в половине случаев лучшие решения были получены совместной и раздельной минимизацией, а в другой половине – алгоритмом выделения подсистем. Предложенный алгоритм выделения связанных подсистем на основе критерия компонентной связанности оказался эффективнее предложенного в [13] алгоритма, основанного

на другой мере связанности функций выделяемых подсистем, предполагающей общие уравнения для всех функций подсистемы.

Критерий минимальности общего числа литералов в функциональных BDDI-представлениях оказался эффективным при автоматическом выборе лучшего решения для программы, реализующей предложенный алгоритм выделения подсистем. Сравнение с ручным выбором лучшего значения меры связанности показало погрешность в $3,9\%$ для BDDI-представлений и $6,4\%$ для Bool-представлений (табл. 7). Программа AutoSplit обеспечила восемь лучших решений для BDDI-представлений и десять – для Bool-представлений.

Проведенные эксперименты показали, что для одной половины примеров систем функций более эффективным является аппарат минимизации на основе матричных представлений, для другой – аппарат минимизации булевых сетей.

Заключение

Экспериментально установлено, что для некоторых систем функций при синтезе логических схем из библиотечных элементов более эффективна совместная минимизация, для других – раздельная минимизация, среднюю позицию (по виду синтезированных схем) занимает оптимизационная процедура выделения подсистем для совместной минимизации. Эффект достигается за счет того, что каждая выделенная подсистема минимизируется на основе разложений Шеннона по своей (для каждой подсистемы) перестановке переменных разложения. При этом для промышленных примеров описаний проектов многовыходных комбинационных схем, математическими моделями поведения которых являются системы булевых функций, целесообразно испытывать две модели функциональных описаний и, соответственно, два подхода: первый – минимизация на основе разложений Шеннона и аппарата матричных описаний, второй – минимизация на основе разложений Шеннона для логических уравнений и последующая структурная минимизация полученных булевых сетей.

Список литературы

1. Amaru L.G. New Data Structures and algorithms for logic synthesis and verification. Springer Publ., 2017, 156 p. doi: 10.1007/978-3-319-43174-1.
2. Бибило П.Н., Ланкевич Ю.Ю. Экспериментальное сравнение эффективности алгоритмов оптимизации BDD-представлений систем булевых функций // Программные продукты и системы. 2020. Т. 33. № 3. С. 449–463. doi: 10.15827/0236-235X.131.449-463.
3. Kubica M., Kania D. SMTBDD: New form of BDD for logic synthesis. IJET, 2016, vol. 62, no. 1, pp. 33–41. doi: 10.1515/eletel-2016-0004.

4. Kubica M., Kania D. Decomposition of multi-output functions oriented to configurability of logic blocks. *Bull. Pol. Acad. Sci. Tech. Sci.*, 2017, vol. 65, no. 3, pp. 317–331. doi: 10.1515/bpasts-2017-0036.
5. Reis A.I., Drechsler R. *Advanced Logic Synthesis*. Springer Publ., 2017, 232 p. doi: 10.1007/978-3-319-67295-3.
6. Lee S.-Y., Riener H., Mishchenko A., Brayton R.K., De Micheli G. A simulation-guided paradigm for logic synthesis and verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022, vol. 41, no. 8, pp. 2573–2586. doi: 10.1109/TCAD.2021.3108704.
7. Haaswijk W., Soeken M., Mishchenko A., De Micheli G. SAT-based exact synthesis: Encodings, topology families, and parallelism. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020, vol. 39, no. 4, pp. 871–884. doi: 10.1109/TCAD.2019.2897703.
8. Haaswijk W., Soeken M., Amaru L., Gaillardon P.-E., De Micheli G. A novel basis for logic rewriting. *Proc. ASP-DAC*, 2017, pp. 151–156. doi: 10.1109/ASPDAC.2017.7858312.
9. Soeken M., Amaru L., Gaillardon P., De Micheli G. Optimizing majority-inverter graphs with functional hashing. *Proc. DATE*, 2016, pp. 1030–1035.
10. Soeken M., Amaru L., Gaillardon P.-E., De Micheli G. Exact synthesis of majority-inverter graphs and its applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017, vol. 36, no. 11, pp. 1842–1855. doi: 10.1109/TCAD.2017.2664059.
11. Riener H., Testa E., Amaru L., Soeken M., De Micheli G. Size optimization of MIGs with an application to QCA and STMG technologies. *Proc. NANOARCH*, 2018, pp. 157–162.
12. Harlecek I., Fiser P., Schmidt J. Are XORs in logic synthesis really necessary? *Proc. IEEE 20th Int. Symposium DDECS*, 2017, pp. 134–139. doi: 10.1109/DDECS.2017.7934583.
13. Бибило П.Н., Позняк А.М. Выделение подсистем связанных функций из многоуровневого представления системы булевых функций // Информатика. 2020. Т. 17. № 1. С. 63–77. doi: 10.37661/1816-0301-2020-17-1-63-77.
14. 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. № 4. С. 9–16. doi: 10.31114/2078-7707-2020-4-9-16.
15. Тарасов И.Е. ПЛИС XILINX. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. М.: Горячая линия; Телеком, 2020. 538 с.

Extracting subsystems from a multilevel representation of a Boolean function system for joint logical minimization

Petr N. Bibilo
Natalia A. Kirienko
Vladimir I. Romanov

For citation

Bibilo, P.N., Kirienko, N.A., Romanov, V.I. (2023) 'Extracting subsystems from a multilevel representation of a Boolean function system for joint logical minimization', *Software & Systems*, 36(4), pp. 509–522 (in Russ.). doi: 10.15827/0236-235X.142.509-522

Article info

Received: 31.05.2023

After revision: 04.07.2023

Accepted: 10.07.2023

Abstract. The paper presents experimental results obtained from studying the effectiveness of programs for minimizing multilevel algebraic representations of Boolean function systems performed in combinational circuit synthesis. The resulting minimized logical descriptions are presented as Shannon expansion formulas or formulas defining Boolean networks. Three approaches are investigated: joint minimization of multilevel representations of Boolean function systems; separate minimization; selection of connected subsystems from an original system, where each connected subsystem is minimized separately, however functions that make up each subsystem are minimized together. After obtaining minimized descriptions of circuits given as a set of interrelated Shannon expansion formulas or as two-operand logical equations corresponding to Boolean networks, the synthesis of logic circuits is performed within the same library for designing custom digital CMOS VLSI (very large integrated circuits made using complementary metal oxide-semiconductor technology). The resulting logic circuits are compared in terms of a crystal area and speed (time delay). Experiments involve 39 industrial examples of circuits. Experimental results show competitiveness and expediency of all three considered approaches in practice. The improvement of circuit parameters (area, time delay) in approach 3 is achieved due to minimizing each selected subsystem

on the basis of Shannon expansions in its own (for each subsystem) permutation of expansion variables. At the same time, minimizing multilevel representations based on Shannon expansions for function system matrix descriptions is more effective for one half of circuits; minimizing based on Shannon expansions of function systems presented as logical equations is effective for the other half. The practical significance of the study is that the developed software, which implements the proposed algorithm for extracting Boolean function subsystems, allows reducing the area and increasing the performance of functional blocks of custom CMOS VLSI in many cases.

Keywords: the system of Boolean functions, disjunctive normal form (DNF), Binary Decision Diagram (BDD), Boolean net, digital logic synthesis, VHDL, VLSI, CMOS

References

1. Amaru, L.G. (2017) *New Data Structures and Algorithms for Logic Synthesis and Verification*. Springer Publ., 156 p. doi: 10.1007/978-3-319-43174-1.
2. Bibilo, P.N., Lankevich, Yu.Yu. (2020) 'Experimental investigation of effectiveness of algorithms for minimizing BDD representations of Boolean function systems', *Software & Systems*, 33(3), pp. 449–463 (in Russ.). doi: 10.15827/0236-235X.131.449-463.
3. Kubica, M., Kania, D. (2016) 'SMTBDD: New form of BDD for logic synthesis', *IJET*, 62(1), pp. 33–41. doi: 10.1515/eletel-2016-0004.
4. Kubica, M., Kania, D. (2017) 'Decomposition of multi-output functions oriented to configurability of logic blocks', *Bull. Pol. Acad. Sci. Tech. Sci.*, 65(3), pp. 317–331. doi: 10.1515/bpasts-2017-0036.
5. Reis, A.I., Drechsler, R. (2017) *Advanced Logic Synthesis*. Springer Publ., 232 p. doi: 10.1007/978-3-319-67295-3.
6. Lee, S.-Y., Riener, H., Mishchenko, A., Brayton, R.K., De Micheli, G. (2022) 'A simulation-guided paradigm for logic synthesis and verification', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(8), pp. 2573–2586. doi: 10.1109/TCAD.2021.3108704.
7. Haaswijk, W., Soeken, M., Mishchenko, A., De Micheli, G. (2020) 'SAT-based exact synthesis: Encodings, topology families, and parallelism', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(4), pp. 871–884. doi: 10.1109/TCAD.2019.2897703.
8. Haaswijk, W., Soeken, M., Amaru, L., Gaillardon, P.-E., De Micheli, G. (2017) 'A novel basis for logic rewriting', *Proc. ASP-DAC*, pp. 151–156. doi: 10.1109/ASPDAC.2017.7858312.
9. Soeken, M., Amaru, L., Gaillardon, P., De Micheli, G. (2016) 'Optimizing majority-inverter graphs with functional hashing'. *Proc. DATE*, pp. 1030–1035.
10. Soeken, M., Amaru, L., Gaillardon, P.-E., De Micheli, G. (2017) 'Exact synthesis of majority-inverter graphs and its applications'. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(11), pp. 1842–1855. doi: 10.1109/TCAD.2017.2664059.
11. Riener, H., Testa, E., Amaru, L., Soeken, M., De Micheli, G. (2018) 'Size optimization of MIGs with an application to QCA and STMG technologies', *Proc. NANOARCH*, pp. 157–162.
12. Harlecek, I., Fiser, P., Schmidt, J. (2017) 'Are XORs in logic synthesis really necessary?' *Proc. IEEE 20th Int. Symposium DDECS*, pp. 134–139. doi: 10.1109/DDECS.2017.7934583.
13. Bibilo, P.N., Pazniak, A.M. (2020) 'The search for subsystems of related functions from multilevel representation of systems of Boolean functions', *Informatics*, 17(1), pp. 63–77 (in Russ.). doi: 10.37661/1816-0301-2020-17-1-63-77.
14. Bibilo, P.N., Romanov, V.I. (2020) 'The system of logical optimization of functional-structural descriptions of digital circuits based on production-frame knowledge representation model', *Problems of Advanced Micro- and Nanoelectronic Systems Development*, (4), pp. 9–16 (in Russ.). doi: 10.31114/2078-7707-2020-4-9-16.
15. Tarasov, I.E. (2020) *XILINX FPGA. Hardware Description Languages VHDL and Verilog, CAD, Design Techniques*. Moscow, 538 p. (in Russ.).

Авторы

Бибилло Петр Николаевич¹, д.т.н., профессор,
заведующий лабораторией,

bibilo@newman.bas-net.by

Кириенко Наталья Алексеевна¹, к.т.н.,

доцент, старший научный сотрудник,

kir@newman.bas-net.by

Романов Владимир Ильич¹, к.т.н., доцент,

ведущий научный сотрудник,

rom@newman.bas-net.by

Authors

Petr N. Bibilo¹, Dr.Sc. (Engineering), Professor,
Head of Laboratory,

bibilo@newman.bas-net.by

Natalia A. Kirienko¹, Cand. of Sci. (Engineering),

Associate Professor, Senior Researcher,

kir@newman.bas-net.by

Vladimir I. Romanov¹, Cand. of Sci. (Engineering),

Associate Professor, Leading Researcher,

rom@newman.bas-net.by

¹ Объединенный институт проблем информатики
Национальной академии наук Беларуси,
г. Минск, 220012, Беларусь

¹ United Institute of Informatics Problems of the
National Academy of Sciences of Belarus (UIIP NASB),
Minsk, 220012, Belarus