УДК 004.021

Метод бинарно-векторного полиномиального разложения булевых функций

А.А. Акинин, Ю.С. Акинина, С.В. Тюрин

Воронежский государственный технический университет, svturin@mail.ru

***Аннотация* — Рассматривается оригинальный подход к полиномиальному разложению булевых функций на основе композиции векторного метода обратных конечных разностей и бинарного метода фрактальных преобразований.**

***Ключевые слова* — поляризованный полином, базис Жегалкина, логический преобразователь, булева функция, разложение Давио.**

I. Введение

Известно, что представление булевой функции в виде полинома Жегалкина может быть короче её представления в виде минимальной дизъюнктивной нормальной формы [1], а среди поляризованных полиномов Рида-Маллера могут быть найдены формы в 1.5 раза короче, чем полином Жегалкина [2]. Однако уникальным достоинством полиномиальных форм является то, что для структурного тестирования полиномиальных логических преобразователей используются тестовые векторы с унифицированной битовой структурой, не зависящей от реализуемой логической функции. При этом для обнаружения любой одиночной константной неисправности достаточное и необходимое количество тестовых векторов не превосходит величины 3n [3, 4], где n – количество аргументов булевой функции.

Следует отметить [5], что преобразование булевой функции к полиномиальной форме относится к NP-трудным задачам, в связи с чем вычислительная и объемная сложности алгоритмов их решения имеют оценку $O(2^n)$, где *n* – количество аргументов преобразуемой булевой функции. Однако вычислительная и объемная сложности программ, реализующих подобные алгоритмы преобразования, могут существенно отличаться друг от друга, значительно превышая обобщенную (точнее, минимальную) оценку $O(2^n)$ вычислительной и объемной сложности всех возможных алгоритмов преобразования булевой функции.

Поиск лучших программных реализаций алгоритмов полиномиального преобразования булевых функций позволит практически их применять в ПЛИС, как при серийном, так и при единичном производстве цифровой аппаратуры, используя в качестве инструментальных средств персональную вычислительную технику.

II. Разновидности полиномиальных логических структур

Полиномиальные логические преобразователи (ПЛП) реализуются на матричных структурах [3, 4], приведенных на рис. 1, которые ориентированы на представление логических функций в виде полиномиальных нормальных форм (ПНФ) и в которых используется логический базис Жегалкина [6]: логические функции «И» (AND), «Исключающее ИЛИ» (EXOR) и «Константа 1».

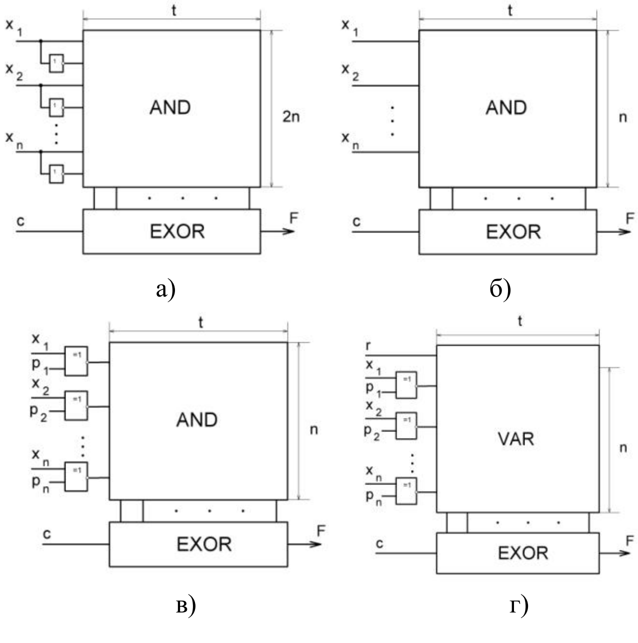

Рис. 1. Матричные структуры для реализации полиномиальных логических преобразователей

На рис. 1а представлена матричная структура ПЛП, которая базируется на следующем логическом уравнении (1):

$$F(x_1, x_2 ... x_n) = C \oplus K_1 \oplus K_2 \oplus ... \oplus K_t, \qquad (1)$$

где K_i - ортогональные элементарные конъюнкции, в каждую из которых переменные $x_1, x_2 ... x_n$ могут входить как с инверсией, так и без инверсии;

$\oplus$ - знак логической операции «исключающее ИЛИ» (exclusive-or – EXOR), которую часто называют «сумма по модулю 2»;

$C = \{0,1\}$ – признак неинвертирования (C=0) или инвертирования (C=1) функции $F(x_1, x_2 ... x_n)$.

В отечественной литературе форму (1) часто называют «сумма по модулю два элементарных конъюнкций», а в зарубежной – ESOP (exclusive-or sum-of-products).

На рис. 1б представлена матричная структура ПЛП, базирующаяся на следующем логическом уравнении (2):

$$F(x_1, x_2 ... x_n) = C \oplus K_1^m \oplus K_2^m \oplus ... \oplus K_t^m, \qquad (2)$$

где K_i^m - монотонная элементарная конъюнкция, в каждую из которых входят неинвертированными только те переменные $x_1, x_2 ... x_n$, которые на соответствующих входных наборах равны 1.

В отечественной литературе форму (2) называют полиномом Жегалкина или положительно поляризованным полиномом Рида-Маллера, а в зарубежной – PPRM (positive-polarity Reed-Muller expressions).

На рис. 1в представлена матричная структура ПЛП, базирующаяся на следующем логическом уравнении (3):

$$F(x_1^{p_1}, x_2^{p_2} ... x_n^{p_n}) = C \oplus K_1^{m\boldsymbol{P}} \oplus K_2^{m\boldsymbol{P}} \oplus ... \oplus K_t^{m\boldsymbol{P}}, \qquad (3)$$

где $K_i^{m\boldsymbol{P}}$ - поляризованная монотонная элементарная конъюнкция, в которую входят только те переменные $x_1, x_2 ... x_n$, которые на соответствующих входных наборах равны 1, при этом инверсными входят те переменные, для которых соответствующий бит поляризации p_j=1;

$\boldsymbol{P}(p_1, p_2, ... p_n)$ - двоичный вектор поляризации, в котором каждый компонент характеризует полярность соответствующей переменной.

В отечественной литературе форму (3) называют поляризованным полиномом Рида-Маллера с фиксированной полярностью, а в зарубежной – FPRM (fixed-polarity Reed-Muller expressions).

На рис. 1г представлена матричная структура ПЛП, базирующаяся на следующем логическом уравнении (4):

$$F(x_1^{p_1}, x_2^{p_2} ... x_n^{p_n}) = C \oplus V_1^{m\boldsymbol{P}} \oplus V_2^{m\boldsymbol{P}} \oplus ... \oplus V_t^{m\boldsymbol{P}}, \qquad (4)$$

где $V_i^{m\boldsymbol{P}}$ - электронно-перестраиваемые логические функции, реализуемые элементами VAR (variable).

Авторы работы [4] предлагают матричную структуру, представленную на рис. 1г, называть VAR-EXOR.

Сигнал управления r обеспечивает электронную перестройку логических элементов VAR. При $r = 1$ реализуется рабочий режим функционирования ПЛП и уравнение (4) эквивалентно уравнению (3). При $r = 0$ реализуется режим тестирования ПЛП, при котором входные переменные заменяются на псевдослучайные последовательности максимальной длины (М-последовательности) [4]. При этом последовательные двухоперандные операции логического умножения в перестраиваемых логических элементах VAR заменяются на последовательные двухоперандные операции равнозначности (NEXOR), которые инверсны операциям EXOR. На рис. 2 представлена функциональная схема двухоперандного элемента VAR, которая соответствует логическому уравнению (5):

$$y_i = (x_{i-1} \,\&\, x_i) \vee \overline{(x_{i-1} \vee x_i \vee r)}, \qquad (5)$$

где & – знак логического умножения;

$\vee$ – знак логического сложения.

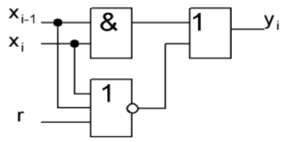

Рис. 2. Функциональная схема двухоперандного элемента VAR

Уникальной особенностью VAR-EXOR полиномиальных логических преобразователей является то, что в режиме тестирования одним и тем же генератором на максимальной рабочей частоте логического преобразователя одновременно формируются как тестовые М-последовательности, так и эталонная М-последовательность. При этом и тестовые, и эталонная последовательности принадлежат одному и тому же замкнутому классу, а их различие состоит лишь в фазовом сдвиге относительно друг друга. Фазовый сдвиг эталонной М-последовательности зависит от реализуемой логической функции и должен быть предварительно определен.

III. Обоснование подхода к бинарно-векторному разложению булевых функций

Анализ многочисленных зарубежных и отечественных литературных источников, часть из которых представлены в библиографиях работ [5, 7], позволяет разделить методы полиномиальных разложений булевых функций на две разновидности: базирующиеся на дискретных моделях преобразования булевых функций с побитовыми вычислениями; базирующиеся на дискретных моделях преобразования булевых функций с векторными вычислениями.

Среди методов побитовых преобразований наименьшей вычислительной сложностью обладает алгоритм, дискретная модель которого подробно рассмотрена в [8] и который назван «алгоритмом фрактального разложения». Данный алгоритм основан на математическом методе полиномиального разложения булевой функции, который представлен в [9], и назван в [9] как «метод, базирующийся на преобразовании вектора значений функции». Однако такое название метода, по нашему мнению, не раскрывает его отличительной сути. По этой причине в дальнейшем будем называть этот метод «методом фрактальных разложений». Метод фрактальных разложений реализуется алгоритмом, требующим для своей реализации объема основной оперативной памяти в 2^n машинных слов, или, в лучшем случае, 2^n бит, и характеризуется вычислительной сложностью порядка $n2^{n-1}$ проходов.

Среди методов векторных преобразований весьма эффективен метод, названный i-поляризацией булевой функции и представленный в [5]. Суть данного метода такова.

Булева функция n переменных представляется 2^n-вектором с единицами, показывающими минтермы, а сами минтермы задаются наборами аргументов булевой функции.

Пусть вектор **f** представляет булеву функцию $f(x_1, x_2, x_3, x_4)$ с девятью минтермами: $x_1^/ x_2^/ x_3^/ x_4^/$, $x_1 x_2 x_3^/ x_4^/$, $x_1^/ x_2^/ x_3 x_4^/$ и т.д. (см. табл. 1). Под $x_i^/$ следует понимать инверсное значение i-го аргумента.

Таблица 1

Представление булевой функции $f(x_1, x_2, x_3, x_4)$

f	1 0 0 1	1 1 0 0	1 0 1 1	1 0 0 1
x_4	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1
x_3	0 0 0 0	1 1 1 1	0 0 0 0	1 1 1 1
x_2	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1
x_1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1
j	0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15

Тогда операция i-поляризации заключается в следующем [5]: часть вектора **f**, составленная только из компонент, где переменная x_i имеет значение 0, сдвигается на 2^{i-1} позиций вправо, и результат складывается по модулю 2 с исходным вектором **f**.

Пример i-поляризации по переменной $x_3^/$:

1001	1100	1011	1001	**f**
1111	0000	1111	0000	$x_3^/$
1001	0000	1011	0000	$g := f \,\&\, x_3^/$
0000	1001	0000	1011	$g := g{:}2^{3-1}$
1001	0101	1011	0010	$\mathbf{f}(x_3^/) := f \oplus g$

Как следует из представленного примера, для получения вектора значений полинома Жегалкина потребуется n последовательных i-поляризаций исходной функции f (по каждой отдельной переменной). Порядок выбора переменных может быть произвольным, а для каждой последующей поляризации должен использоваться результат предшествующей поляризации.

Для всех i-поляризаций потребуется (n+2) вектора, индивидуальной размерностью 2^n бит: n векторов для хранения инверсных значений для каждой переменной, один вектор для хранения значений исходной булевой функции и её поляризованных значений, а также один вспомогательный вектор для векторных вычислений и сдвига промежуточных результатов.

Для каждой i-поляризации потребуется всего три векторных операции, среди которых присутствует одна операция, являющаяся операцией целочисленного деления на 2 или, что эквивалентно, операцией арифметического сдвига вправо. Тогда количество основных алгоритмических операций, необходимых для полной поляризации булевой функции и получения вектора значений полинома Жегалкина можно оценить как

$$K = \sum_{i=1}^{n} 2^{i-1} = (2^{n-1} - 1) \quad , \qquad (6)$$

Оценку (6) можно считать достаточно точной только в том случае, если метод векторной i-поляризации будет реализован аппаратурно. Для случая программной реализации метода векторной i-поляризации уточненную оценку вычислительной сложности провести достаточно сложно, однако можно предположить, что вычислительная сложность будет сопоставима с величиной $n2^{n-1}$. Для данного предположения имеются следующие аргументы.

Во-первых, в качестве операндов для векторных преобразований может быть выбран битовый массив dynamic_bitset из библиотеки BOOST C++. Эта библиотека представляет собой собрание множества кроссплатформенных библиотек, созданных независимыми разработчиками и тщательно проверенными на различных платформах (www.boost.org). Отличительными особенностями массива dynamic_bitset являются: возможность динамического изменения размера массива в ходе выполнения программы, поддержка быстрого доступа по индексу к произвольному элементу массива, поддержка элементарных логических операций (регистрового сдвига, сложения, умножения, инверсии и т.д.). Размер массива dynamic_bitset ограничен величиной 2^{31}, что ограничивает сверху максимальное число аргументов исходной булевой функции до 31. Ясно, что при ограниченной разрядности вычислительной техники не могут быть выполнены «чисто» векторные операции – они будут выполняться параллельно-последовательно по m бит, где m - разрядность вычислительной техники. Данное обстоятельство обязательно приведёт к ухудшению оценки (6).

Во-вторых, и алгоритм фрактального разложения, и метод i-поляризации имеют одинаковую геометрическую интерпретацию, которая представлена на рис. 3 на примере булевой функции, зависящей от трех аргументов.

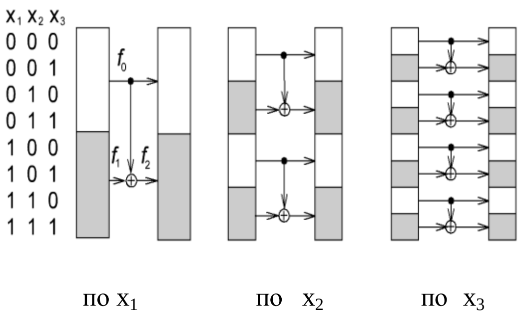

Рис. 3. Геометрическая интерпретация полиномиального разложения булевой функции

Следует заметить, что представленная на рис. 3 геометрическая интерпретация характерна также для разложения булевой функции методом, который в зарубежной литературе называют «позитивное разложение Давио» (pD) [10]. Суть такого разложения (преобразования) поясняется следующими соотношениями:

$$f(x_1,,,x_i,...,x_n) = f_0 \oplus x_i f_2 \ , \qquad (7)$$

где

$$f_0 = f(x_1,...,x_{i-1},0,x_{i+1},...x_n) \ , \qquad (8)$$

$$f_1 = f(x_1,...,x_{i-1},1,x_{i+1},...x_n) \ , \qquad (9)$$

$$f_2 = f_0 \oplus f_1 \ . \qquad (10)$$

Рассмотрим возможность бинарно-векторного полиномиального разложения булевых функций, ориентированного на программную реализацию персональной вычислительной техникой.

IV. АЛГОРИТМ БИНАРНО-ВЕКТОРНОГО РАЗЛОЖЕНИЯ БУЛЕВЫХ ФУНКЦИЙ

Основываясь на результатах работы [11] можно предложить следующий векторный алгоритм полиномиального разложения булевой функции F:

Алгоритм AL:

1. Ввод количества (n) переменных x_n функции F;
2. Ввод вектора $\mathbf{F}(x_0, x_1,\dots x_k, \dots x_{n-1}) = (f_0, f_1, \dots f_i, \dots f_{2^n-1})$;
3. Подготовка вспомогательных 2^n-разрядных двоичных векторов **S** =(11…1), **G** =(00…0);
4. **G** := **F** & **S**;
5. **G** := **G**/2;
6. **G** := **F** $\oplus$ **G**;
7. **S**:= **S**/2;
8. **F** := **G**;
9. Если **S** > 0, то перейти на пункт 4;
10. Конец алгоритма: вектор $\mathbf{F}(x_0, x_1\dots x_{n-1})$ преобразован в вектор коэффициентов полинома Жегалкина $\mathbf{P}(p_0, p_1, \dots p_i, \dots p_{2^n-1})$.

Из представленного алгоритма следует, что его вычислительная сложность может быть оценена как

$$L = 2^n - 1. \qquad (11)$$

Оценка (11) справедлива только в том случае, если алгоритм AL будет реализован аппаратно, или если он реализуется программно на ЭВМ с разрядностью $m=2^n$.

Ясно, что исходный вектор $\mathbf{F}(x_0, x_1,\dots x_k, \dots x_{n-1}) = (f_0, f_1, \dots f_i, \dots f_{2^n-1})$ можно разбить на подвекторы $\mathbf{F}(x_0, x_1,\dots x_k, \dots x_{n-1}) = (\mathbf{A}_0, \mathbf{A}_1, \dots \mathbf{A}_s, \dots \mathbf{A}_r)$, где $r = (2^n/m) - 1$, а $\mathbf{A}_0 = (f_0, f_1, \dots f_{m-1})$; $\mathbf{A}_1 = (f_m, f_{m+1}, \dots f_{2m-1})$ и т.д. Тогда, применяя к каждому подвектору $\mathbf{A}_s$ алгоритм AL, а затем, применяя к преобразованным подвекторам $\mathbf{A}_s^*$ алгоритм фрактального разложения, приходим к алгоритму бинарно-векторного преобразования булевых функций к полиному Жегалкина.

Суть предлагаемого алгоритма бинарно-векторного полиномиального разложения булевых функций 5-ти аргументов (n=5) рассмотрим на примере булевой функции $Q(x_0, x_1, x_2, x_3, x_4) = \mathbf{Q}(f_0, f_1, \dots f_{31})$.

Предположим, что разрядность инструментальной ЭВМ m = 4. Тогда исходный вектор $\mathbf{Q}(f_0, f_1, \dots f_{31})$ следует разбить на $2^n/m$ подвекторов:

$\mathbf{A}_0 = (f_0, f_1, f_2, f_3)$; $\mathbf{A}_1 = (f_4, f_5, f_6, f_7)$;

$\mathbf{A}_2 = (f_8, f_9, f_{10}, f_{11})$; $\mathbf{A}_3 = (f_{12}, f_{13}, f_{14}, f_{15})$;

$\mathbf{A}_4 = (f_{16}, f_{17}, f_{18}, f_{19})$; $\mathbf{A}_5 = (f_{20}, f_{21}, f_{22}, f_{23})$;

$\mathbf{A}_6 = (f_{24}, f_{25}, f_{26}, f_{27})$; $\mathbf{A}_7 = (f_{28}, f_{29}, f_{30}, f_{31})$;

Указанный порядок разбиения исходного вектора $\mathbf{Q}(f_0, f_1, \dots f_{31})$ на подвекторы нарушать не допускается.

Рассмотрим последовательное преобразование алгоритмом AL вектора $\mathbf{A}_0 = (f_0, f_1, f_2, f_3)$

$$\mathbf{A}_0^* = \begin{vmatrix} f_0^1 = f_0 \oplus 0; f_1^1 = f_0 \oplus f_1; f_2^1 = f_1 \oplus f_2; f_3^1 = f_2 \oplus f_3 \\ f_0^2 = f_0^1 \oplus 0; f_1^2 = f_1^1 \oplus 0; f_2^2 = f_1^1 \oplus f_2^1; f_3^2 = f_2^1 \oplus f_3^1 \\ f_0^3 = f_0^2 \oplus 0; f_1^3 = f_1^2 \oplus 0; f_2^3 = f_2^2 \oplus 0; f_3^3 = f_2^2 \oplus f_3^2 \end{vmatrix} \qquad (12)$$

В (12) под f_i^j следует понимать i-ый компонент вектора $\mathbf{A}_0 = (f_0, f_1, f_2, f_3)$, значение которого преобразуется на j-ом проходе алгоритма AL, а под $\oplus$ - поразрядную сумму векторов по модулю 2.

Аналогичным образом алгоритмом AL последовательно преобразуется каждый подвекор A_s.

В последующем к подвекторам A_s^*, как к индивидуальным объектам, применяется бинарный алгоритм фрактального разложения. Суть этого алгоритма для рассматриваемого примера сводится к следующим преобразованиям (13):

$$\begin{aligned}
&A_1^{1*}= A_0^{1*}\oplus A_1^{1*};\ A_3^{1*}= A_2^{1*}\oplus A_3^{1*};\\
&A_5^{1*}= A_4^{1*}\oplus A_5^{1*};\ A_7^{1*}= A_6^{1*}\oplus A_7^{1*};\\
&\\
&A_2^{2*}= A_0^{1*}\oplus A_2^{1*};\ A_3^{2*}= A_1^{1*}\oplus A_3^{1*};\\
&A_6^{2*}= A_4^{1*}\oplus A_6^{1*};\ A_7^{2*}= A_5^{1*}\oplus A_7^{1*}; \qquad (13)\\
&\\
&A_4^{3*}= A_0^{2*}\oplus A_4^{2*};\ A_5^{3*}= A_1^{2*}\oplus A_5^{2*};\\
&A_6^{3*}= A_2^{2*}\oplus A_6^{2*};\ A_7^{3*}= A_3^{2*}\oplus A_7^{2*}.
\end{aligned}$$

В (13) под A_s^{v*} следует понимать модификацию s-го подвектора на v-ом проходе алгоритма фрактального разложения, а под $\oplus$ - поразрядную сумму векторов по модулю 2.

В результате проведенных преобразований приходим к соотношению (14):

$$\begin{aligned}
&\mathbf{F}(x_0, x_1,\dots x_k, \dots x_4) = (f_0, f_1, \dots f_i, \dots f_{31}) \equiv\\
&\equiv (\mathbf{A}_0, \mathbf{A}_1^{1*}, \mathbf{A}_2^{2*}, \mathbf{A}_3^{2*}; \mathbf{A}_4^{3*}, \mathbf{A}_5^{3*}, \mathbf{A}_6^{3*}, \mathbf{A}_7^{3*}) \equiv \qquad (14)\\
&\equiv \mathbf{P}(p_0, p_1, \dots p_i, \dots p_{2^n-1}).
\end{aligned}$$

В (14) $\mathbf{P}(p_0, p_1, \dots, p_i, \dots, p_{2^n-1})$ есть вектор коэффициентов полинома Жегалкина, при этом, если $p_i=1$, то только эта, i-ая монотонная конъюнкция входит в полиномиальную нормальную форму. Равенство $p_0 =1$, учитывает тот факт, в исходной булевой функции присутствует конъюнкция, состоящая из инверсных значений всех её аргументов, которая в полиномиальной форме всегда приравнивается 1.

Анализ соотношения (13) показывает, что предлагаемая обработка подвекоров A_s^* имеет ту же геометрическую интерпретацию, которая представлена на рис. 3. Отличие заключается лишь в том, что изменена последовательность преобразований, что правомочно, так как логическая операция суммы по модулю 2 обладает свойством ассоциативности.

Вычислительная сложность предлагаемого алгоритма бинарно-векторного полиномиального разложения булевых функций может быть оценена следующей величиной (15):

$$W=(m-1)\frac{2^n}{m}+(n-\log_2 m)2^{(n-1-\log_2 m)} \quad , \qquad (15)$$

где *n*- количество аргументов булевой функции;
m – разрядность инструментальной ЭВМ.

Как следует из (15), вычислительная сложность метода бинарно-векторного полиномиального разложения булевой функции постепенно уменьшается с увеличением разрядности инструментальной ЭВМ, при этом следует иметь в виду, что величина *m* должна быть кратна 2^d, где d изменяется от 0 до n. Все векторные и бинарные операции производятся только над машинными словами.

V. Преобразование булевых функций к поляризованным полиномам

При обосновании подхода к бинарно-векторному разложению булевых функций (БФ) предполагалось, что исходная булева функция должна быть представлена в виде упорядоченного вектора своих значений на возрастающих в лексиграфическом порядке наборах значений аргументов. Иначе говоря, исходная булева функция представляется в виде таблицы истинности, соответствующей совершенной дизъюнктивной нормальной форме. При таком исходном представлении БФ и применении алгоритма бинарно-векторного разложения может быть получен вектор значений коэффициентов полинома Жегалкина, который идентичен положительно поляризованному полиному Рида-Маллера. Общее количество всех возможных форм поляризованных полиномов Рида-Маллера определяется величиной 2^n, где n-количество аргументов БФ. Для нахождения всего множества поляризованных полиномов возможны следующие подходы: последовательная поляризация в определенном порядке полинома Жегалкина; многократная предварительная поляризация исходной булевой функции с последующим её преобразованием в полином Жегалкина, однако в действительности будут получены поляризованные полиномы Рида-Маллера.

Эффективный векторный метод и соответствующий алгоритм нахождения поляризованных полиномов из полинома Жегалкина рассмотрен в [5]. Данный метод базируется на векторной операции, которую в [5] именуют как «смена i-полярности». Данная операция заключается в следующем: часть вектора $\mathbf{P}(p_0, p_1, \dots, p_i, \dots, p_{2^n-1})$, составленная из компонент, где переменная x_i имеет значение 1, сдвигается на 2^{i-1} позиций влево, и результат складывается по модулю 2 с исходным вектором $\mathbf{P}(p_0, p_1, \dots p_i, \dots p_{2^n-1})$. Используя эту операцию, получают один за другим все 2^n вектора коэффициентов поляризованных полиномов Рида-Маллера и находят наилучший из них – с минимальным числом коэффициентов, равных 1.

Перебор различных вариантов полярности переменных рекомендуют выполнять в таком порядке, чтобы соседние варианты поляризации отличались только по одной переменной.

Вычислительная и объемная сложности одной операции смены i-полярности такие же, как и у операции i-поляризации. При этом следует учитывать, что операция смены i-полярности также требует выполнения поразрядного суммирования по модулю 2 и операций сдвига, которые при ограниченной разрядности средств вычислительной техники могут быть выполнены только параллельно-последовательно.

Известно, что применение алгоритма преобразования поляризованной по каким–либо переменным булевой функции к полиному Жегалкина эквивалентно нахождению поляризованного по тем же переменным полинома Рида-Маллера.

Известно также, что поляризации булевой функции по каким-либо переменным приводит, в конечном счете, к определенной перестановке компонент неполяризованного вектора значений булевой функции $\mathbf{F}(x_0, x_1,\dots x_k, \dots x_{n-1}) = (f_0, f_1, \dots f_i, \dots f_{2^n-1})$. В [7] со ссылкой на [12] представлена систематическая процедура определения перестановок компонент в векторах поляризованных значений булевых функций и приведен пример таких перестановок для булевой функции трех аргументов. В табл. 2 приведены все перестановки компонент неполяризованной функции $\mathbf{F}(x_0, x_1,\dots x_k, \dots x_{n-1}) = (f_0, f_1, \dots f_i, \dots f_{2^n-1})$ в зависимости от соответствующего вектора поляризации.

Таблица 2

Векторы значений поляризованной БФ

$x_2x_1x_0$	$\mathbf{F}^{\mathbf{P}}(\dots)$ – вектор значений поляризованной БФ							
	$\mathbf{P}(p_2,p_1,p_0)$ – вектор поляризации							
	000	001	010	011	100	101	110	111
0 0 0	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0 0 1	f_1	f_0	f_3	f_2	f_5	f_4	f_7	f_6
0 1 0	f_2	f_3	f_0	f_1	f_6	f_7	f_4	f_5
0 1 1	f_3	f_2	f_1	f_0	f_7	f_6	f_5	f_4
1 0 0	f_4	f_5	f_6	f_7	f_0	f_1	f_2	f_3
1 0 1	f_5	f_4	f_7	f_6	f_1	f_0	f_3	f_2
1 1 0	f_6	f_7	f_4	f_5	f_2	f_3	f_0	f_1
1 1 1	f_7	f_6	f_5	f_4	f_3	f_2	f_1	f_0

В зависимости от значений компонент вектора поляризации $\mathbf{P}(p_2,p_1,p_0)$ в $\mathbf{F}^{\mathbf{P}}(\dots)$ необходимо подставить соответствующие компоненты в том порядке, в котором они перечислены сверху вниз в соответствующем столбце таблицы. На основе данных табл. 2 предлагается следующий метод поляризации булевых функций.

Примем следующие соглашения:

$\mathbf{F}(x_0, \dots x_k, \dots x_{n-1}) = (f_0, f_1, \dots f_i, \dots f_{2^n-1})$ есть исходный вектор значений неполяризованной БФ;

$\boldsymbol{P}(\mathbf{P}_0, \mathbf{P}_1\dots \mathbf{P}_z\dots \mathbf{P}_{2^n-1})$ – множество всех векторов поляризации;

$\mathbf{F}^{\mathbf{P}_z}(x_0,\dots x_k, \dots x_{n-1}) = (f_s, \dots f_j, \dots f_g)$ – вектор значений поляризованной по $\mathbf{P}_k$ БФ. Тогда справедливо:

$$j = i \oplus z \quad (i = \text{от } 0 \text{ до } 2^n-1) \qquad (16)$$

В (16) десятичные значения индексов j, i и z должны представляться двоичными эквивалентами.

VI. Заключение

Предложен метод бинарно-векторного полиномиального разложения булевых функций, предельно ориентированный на реализацию с помощью инструментальных ЭВМ, требующий для реализации объема основной памяти порядка 2^n бит и обладающий наименьшей вычислительной сложностью по сравнению с известными методами.

Литература

[1] Papakonstantinou G. Minimization of modulo-2 sum of products // IEEE Trans. Comput. 1979. № 2. P. 163–167.

[2] Перязев Н.А. Сложность булевых функций в классе полиномиальных поляризованных форм // Алгебра и логика. 1995. Вып. 3. № 34. С. 323-326.

[3] Hirayama T., Nagasawa K., Nishitani Y., Shimizu K. Double Fixed-Polarity Reed-Muller Expressions: A New Class of AND–EXOR Expressions for Compact and Testable Realization // IPSJ Journal. Apr. 2001. Vol. 42. № 4. P. 983-991.

[4] Пат. 2413282 Российская Федерация, МПК[7] – G 06 F 011/22. Способ тестопригодной реализации логических преобразователей [Текст] / Акинина Ю.С., Подвальный С.Л., Тюрин С.В.; заявитель и патентообладатель Воронеж. гос. техн. университет. № 2008151028; заявл. 22.12.2008; опубл. 27.02.2011; Бюл. № 6(III ч.).

[5] Закревский А.Д., Торопов Н.Р. Полиномиальная реализация частичных булевых функций и систем. М.: Едиториал УРСС, 2003. 200 с.

[6] Жегалкин И.И. Арифметизация символической логики // Математический сборник Московского математического общества. 1927. Т. 354. С. 9-28.

[7] Бережная М.А., Рыжикова М.Г., Татаренко Д.А. Синтез комбинационных схем в базисе полиномиальных форм // Радиоэдектроника и информатика.–Харьковский национальный университет радиоэлектроники. 2005. № 3 (32). С. 103-108.

[8] Акинин А.А. Алгоритм фрактального полиномиального разложения булевых функций // Научно-технический журнал “Вестник Воронежского государственного университета”. Воронеж, ВГТУ, 2011. Т. 7. № 11.1. С. 85-88.

[9] Гаврилов Г.П., Сапоженко А.А. Задачи и упражнения по дискретной математике: Учеб. пособие. 3-е изд., перераб. М.: ФИЗМАТЛИТ, 2005. 416 с.

[10] Mozammel H.A. Khan. An ASIC Architecture for Generation Optimum Mixed Polarity Reed – Muller Expression // Engineering Letters, 13:3, EL_13_3_2 (Advance online pulication: 4 November 2006). 8 с.

[11] Акинин А.А., Акинина Ю.С., Тюрин С.В. Автоматизация полиномиального разложения булевых функций на основе метода конечных разностей // Системы управления и информационные технологии: Научно-технический журнал. Москва-Воронеж, ООО Издательство “Научная книга”, 2011. № 4 (46). С. 12-16.

[12] Saluga K.K., Ong E.H. Minimization of Reed-Muller canonic expansion functions // IEEE Trans. Comput. 1979. С. 535-537.

Работа выполнена при поддержке регионального гранта РФФИ № 09-07-97508 р_центр_а.