УДК 004.023

Распределенная система и алгоритмы поиска минимальных и близких к ним контактных схем для булевых функций от малого числа переменных

С.А. Ложкин, М.С. Шуплецов, В.А. Коноводов,

Б.Р. Данилов, В.В. Жуков, Н.Ю. Багров

Московский государственный университет имени М.В. Ломоносова – факультет вычислительной математики и кибернетики, lozhkin@cs.msu.ru

***Аннотация* — В работе рассматривается задача построения каталогов схем, реализующих функции алгебры логики малого количества переменных. Эта задача рассматривается на примере синтеза контактных схем. Для решения задачи были разработаны алгоритмы синтеза схем и на их основе реализованы программные инструменты, с помощью которых для целого ряда функций алгебры логики пяти переменных получены новые более оптимальные схемы, а также установлена верхняя оценка сложности реализации указанных функций в классе контактных схем.**

***Ключевые слова* — контактная схема, сложность, методы синтеза схем, база данных схем, метод каскадов.**

I. Введение

Построение минимальных и близких к ним схем в заданной модели дискретных управляющих систем для функций от малого числа переменных является актуальной задачей математической кибернетики. Каталоги или библиотеки таких схем находят свое применение в различных алгоритмах логического синтеза цифровых схем (см., например, [1]). В свою очередь, анализ структуры оптимальных схем для функций малого числа переменных может быть полезным при разработке элементов технологической библиотеки для проектирования интегральных схем.

В данной работе исследуется традиционная модель реализации функций алгебры логики (ФАЛ) — модель контактных схем (КС). Серьезные математические исследования КС были начаты еще Шенноном [2, 3], а их результаты использовались для синтеза и моделирования переключательных схем, составленных из (электромеханических) реле и контактов. В настоящее время модель КС может быть использована для моделирования и синтеза на транзисторном уровне МОП схем различных типов (см., например, [4]).

С формальной точки зрения (p,q)-КС $\Sigma=\Sigma(x_1,\ldots,x_n;a'_1,\ldots,a'_p;a''_1,\ldots,a''_q)$ представляет собой неориентированный граф с выделенными вершинами – входами $a'_1,\ldots,a'_p$ и выходами $a''_1,\ldots,a''_q$, в котором все ребра помечены переменными $x_1,\ldots,x_n$ или их отрицаниями $\bar{x}_1,\ldots,\bar{x}_n$. При этом число контактов (ребер графа) называется сложностью КС Σ и обозначается $\mathrm{L}(\Sigma)$. Считается, что в каждой вершине v (1,m)-КС $\Sigma(x_1,\ldots,x_n;a_0;a_1,\ldots,a_m)$ реализуется ФАЛ проводимости от входа a_0 к этой вершине, которая равна 1 на наборе $\alpha=(\alpha_1,\ldots,\alpha_n)$ тогда и только тогда, когда в Σ имеется цепь из проводящих на наборе α контактов вида $x_1^{\alpha_1},\ldots,x_n^{\alpha_n}$ (полагаем, как обычно, что $x^0=\bar{x}$, а $x^1=x$), идущая из a_0 в v. При этом КС Σ реализует систему ФАЛ $F=(f_1,\ldots,f_m)$, где f_j — ФАЛ проводимости от a_0 к выходу a_j, $1\le j\le m$. Частным случаем КС являются параллельно-последовательные КС (π -схемы). Простейшей π -схемой считается любая (1,1)-КС, которая состоит из одного контакта, соединяющего полюса. Если π -схемы Σ_1 , Σ_2 уже определены, то (1,1)-КС, которая получается в результате их параллельного или последовательного соединения, тоже является π -схемой. Сложностью ФАЛ f (системы ФАЛ $F=(f_1,\ldots,f_m)$) в классе КС называется минимальная сложность (1,1)-КС (соответственно (1,m)-КС), ее реализующей.

Удобной математической моделью интегральных схем специального вида на дополняющих МОП-транзисторах является следующий класс (2,m)-КС. Пусть (2,m)-КС вида $\Sigma(x_1,\ldots,x_n;a',a'';a_1,\ldots,a_m)$ получается из двух непересекающихся (1,m)-КС вида $\Sigma'(x_1,\ldots,x_n;a';a'_1,\ldots,a'_m)$ и $\Sigma''(x_1,\ldots,x_n;a'';a''_1,\ldots,a''_m)$, которые реализуют наборы ФАЛ $F=(f_1,\ldots,f_m)$ и $\bar{F}=(\bar{f}_1,\ldots,\bar{f}_m)$ соответственно в результате отождествления вершин

$a_j = a'_j = a''_j$, $1 \le j \le m$. Заметим, что при этом на любом наборе $\alpha = (\alpha_1, \ldots, \alpha_n)$ значений переменных $x_1, \ldots, x_n$ и при любом указанном j в КС Σ имеется проводящая цепь, ведущая в вершину a_j либо из входа a', если $f_j(\alpha) = 1$, либо из входа a'', если $f_j(\alpha) = 0$.

Заметим также, что ФАЛ проводимости между входами a' и a'' в КС Σ равна нулю и будем считать, что Σ реализует систему ФАЛ F как в прямом, так и в инверсном видах. Пример такой (2,2)-КС, реализующей систему ФАЛ $F_1 = (\bar{x}_1 \cdot \bar{x}_2, x_1 \oplus x_2)$, показан на рисунке 1. Отмеченные особенности, очевидно, позволяют использовать схемы указанного вида для построения схемы на дополняющих МОП-транзисторах, реализующей набор ФАЛ F. Пример такой схемы, реализующей систему ФАЛ F_1, показан на рис. 2.

Первые каталоги минимальных КС, реализующих ФАЛ от малого числа переменных, появились еще в 1950-х гг. Так, например, в статье [8], в книге [6] и работе [7] приведены таблицы верхних оценок контактной сложности для всех типовых ФАЛ четырех переменных (всего 402 функции).

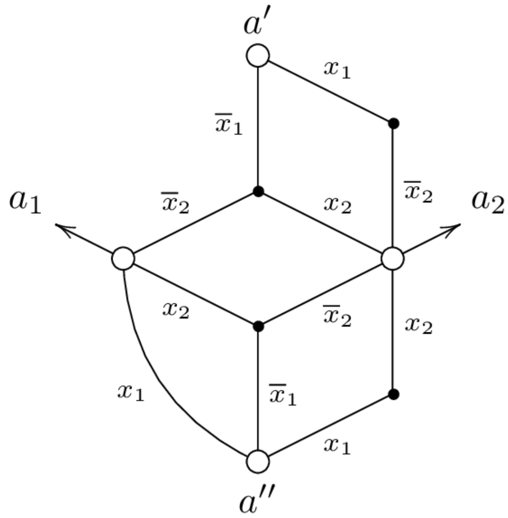

Рис. 1. (2,2)-КС $\Sigma(x_1, x_2; a', a''; a_1, a_2)$, реализующая систему ФАЛ $F_1 = (\bar{x}_1 \cdot \bar{x}_2, x_1 \oplus x_2)$

В [8] Г.Н. Поваровым с помощью построения схем было установлено, что одна ФАЛ из этих 402 требует не более 14 контактов, четыре ФАЛ – не более 13 контактов, а остальные – не более 12 контактов. Затем В.Л. ван дер Пуль [9] построил для первой из этих функций схему с 13 контактами. Ю.Л. Васильев [7], взяв за основу каталоги из работ Г.Н. Поварова [8], Игоннэ и Греа [10], указал минимальные значения сложности всех функций от четырех переменных. Три ФАЛ имели сложность 12, и две – 13. Позднее в работе В.Ю. Сусова [11] каталог Ю.Л. Васильева был уточнен; было доказано, что сложность 13 имеет только одна функция от четырех переменных.

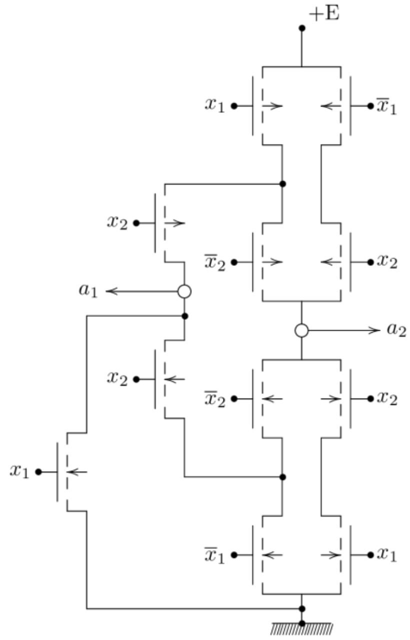

Рис. 2. Схема на дополняющих МОП-транзисторах, построенная по (2,2)-КС $\Sigma(x_1, x_2; a', a''; a_1, a_2)$

В работе К. Шеннона [3] было доказано, что для реализации любой ФАЛ от 5 переменных в классе контактных схем достаточно 30 контактов. Позднее Г.Н. Поваров [12] уточнил эту оценку до 28 контактов, используя метод каскадов и полученные ранее результаты для ФАЛ от четырех переменных. В.Ю. Сусовым [11] была найдена ФАЛ от 5 переменных, контактная сложность которой не меньше 19. Таким образом, остается открытым вопрос о нахождении более точных верхних оценок сложности реализации произвольной ФАЛ от 5 переменных в классе КС. Для решения этого вопроса был разработан ряд алгоритмов поиска оптимальных КС для функций малого числа переменных, а также система хранения, обработки и верификации найденных схем.

II. Основные классы алгоритмов поиска минимальных и близких к ним контактных схем

A. Алгоритмы переборного типа

Одним из очевидных способов решения задачи о минимизации сложности функций является полный перебор. Чтобы доказать, что заданная ФАЛ f имеет контактную сложность, равную L, необходимо, с одной стороны, предъявить (1,1)-КС данной сложности, реализующую эту функцию, а с другой стороны, показать, что никакой схемой меньшей сложности эта функция не реализуется. Последнее можно доказать, перебрав все различные схемы за исключением тех из них, которые заведомо не реализуют заданную функцию. Для этого используются дополнительные ограничения на структуру графа КС. Например, предположим, что функция f не представляется в виде $f = x_i \cdot g$, где g – некоторая ФАЛ, тогда степень как входной, так и выходной вершины любой реализующей ее КС не менее двух.

При построении минимальных и близких к ним КС число рассматриваемых ФАЛ можно существенно сократить, воспользовавшись отношением инверсной конгруэнтности. Будем говорить, что ФАЛ f *инверсно-конгруэнтна* ФАЛ g, если ФАЛ g может быть получена из ФАЛ f при помощи операций перестановки и инвертирования переменных. Заметим, что введенное отношение является отношением эквивалентности и разбивает множество всех функций на соответствующие классы эквивалентности, которые будем называть *инверсно-конгруэнтными классами*. Если ФАЛ f имеет сложность L в рассматриваемой модели КС, то сложность любой ФАЛ g, попадающей в инверсно-конгруэнтный класс, которому принадлежит ФАЛ f, также имеет сложность L.

Таким образом, при изучении сложности ФАЛ достаточно рассматривать только по одному представителю из каждого класса, поскольку любая полученная оценка сложности ФАЛ f справедлива для любой эквивалентной ей в указанном смысле ФАЛ. Выбор этого представителя среди всех функций класса может быть сделан определенным фиксированным способом. Например, таким «каноническим» представителем может быть функция класса, столбец значений которой лексикографически минимален среди всех ФАЛ этого класса. Быстрый алгоритм решения задачи нахождения «канонического» представителя для класса эквивалентности, которому принадлежит заданная функция, был предложен в [13]. Следует отметить, что в случае пяти переменных число указанных классов эквивалентности ФАЛ приблизительно равно 1 228 000, что значительно меньше числа всех таких функций, которое близко к 4,3 млрд.

Перебор КС для реализации выбранного множества ФАЛ ведется сначала по всем попарно неизоморфным графам с фиксированным числом ребер, удовлетворяющим заданному набору структурных ограничений, а затем по всем возможным разметкам их ребер переменными и их отрицаниями. Существуют различные программы для генерации графов. В данной работе для получения всех попарно неизоморфных графов, удовлетворяющих заданному набору ограничений, использовалась программа nauty [14, 15]. Предположим, что для заданного значения сложности L построены все попарно неизоморфные графы с L ребрами, допустимые для выбранного множества ФАЛ. Далее, построим множество всех возможных пометок для L ребер так, чтобы в этом множестве не содержалось наборов, получающихся друг из друга перестановкой переменных и/или навешиванием отрицаний. Для каждой пары из декартового произведения построенных множеств графов и пометок перебираются все возможные пары вершин, и вычисляются функции проводимости между ними. Если в текущей базе данных сложность результирующей функции проводимости больше, чем L, то полученная схема запоминается как наилучшая на текущий момент для данной функции (при этом определив вход и выход в соответствии с теми вершинами, между которыми реализуется данная функция).

Любая контактная схема, полученная в процессе перебора, дает верхнюю оценку сложности реализуемой ею ФАЛ. Для доказательства того факта, что полученная верхняя оценка является точной используются теоретические соображения и проведение полного перебора по всем КС, сложность которых меньше, чем полученная верхняя оценка. С использованием современных компьютеров полный перебор КС можно осуществить для схем, содержащих не более чем 12 или 13 контактов. Таким образом, для ряда ФАЛ на данный момент установить точную оценку сложности не представляется возможным.

Для таких ФАЛ первоначальные оценки их сложности были получены при помощи алгоритма генерации случайных КС. На каждом шаге такого алгоритма случайным образом строится КС заданной сложности L. Далее, вычисляется реализуемая ею функция. В случае, когда текущая оценка сложности этой функции в базе больше, чем L, эта оценка обновляется до L.

Существует множество различных методов построения случайных схем, в частности, можно генерировать случайный граф на заданном числе ребер и выбирать случайные пометки для контактов. Однако в этом случае слишком часто будут получаться схемы, имеющие функцию проводимости, равную нулю. В данной работе используется следующий способ для решения этой проблемы – построение схемы за счет последовательного добавления цепей, обладающих ненулевой проводимостью. Сначала строится схема S_0 из двух полюсов (вход и выход). Затем до тех пор, пока схема не будет содержать L контактов, на каждом i-м, $i = 1,\ldots,L$, шаге выбирается число t, $1 \le t \le \min\{5, L - L(S_i)\}$, и схема S_{i+1} получается из схемы S_i добавлением цепи из t контактов, соответствующих различным переменным.

Преимущество переборных алгоритмов в их простом распараллеливании. Для получения заметного ускорения достаточно запустить несколько параллельных процессов, занимающихся независимыми последовательными задачами перебора.

Результаты переборных алгоритмов дают хорошее приближение для дальнейшего решения задачи. Эти алгоритмы можно применить к различным подклассам схем и подмножествам функций для последующего уточнения верхних оценок их сложности.

B. Алгоритмы, основанные на модификациях метода каскадов

Метод каскадов [16] является довольно простым и в то же время достаточно эффективным методом синтеза КС. Он связан с последовательным разложением заданных ФАЛ по булевым переменным и рекурсивным построением схемы, реализующей эти ФАЛ. Отличительной особенностью КС, построенных при помощи этого метода, является ограниченность сверху длины проводящих цепей числом переменных, от которых зависит реализуемая ФАЛ. Указанное свойство является важным при проектировании библиотек стандартных ячеек, так как напрямую влияет на задержку распространения переключений и другие физические характеристики схемы.

Оригинальный метод каскадов предполагает единый глобальный порядок переменных, который используется в процессе последовательного разложения и рекурсивного построения схемы. При этом выбор порядка следования переменных в разложении может сильно влиять на сложность результирующей схемы. Так как число возможных порядков следования n переменных равно $n!$, то перебор всех возможных вариантов является вычислительно сложной задачей и может быть осуществлен только для n меньше 10.

В данной работе предлагается несколько модификаций метода каскадов, которые позволяют за приемлемое время для произвольной ФАЛ строить такие реализующие её КС, сложность которых близка к оптимальной, а длины проводящих путей ограничены сверху числом переменных. Одна из этих модификаций связана с тем, что вместо глобального порядка следования переменных разложения на каждом шаге разложения для каждой из разлагаемых на нем ФАЛ переменная, по которой оно производится, выбирается индивидуально, исходя из некоторых соображений оптимальности.

В частности, переменная, по которой производится разложение ФАЛ $f(x_1,..., x_n)$, может выбираться в результате решения следующей оптимизационной задачи:

$$\hat{x} = \underset{x \in X(n)}{\arg\min}\left(L\left(f\big|_x\right) + L\left(f\big|_{\bar{x}}\right)\right),$$

где $f\big|_x$ и $f\big|_{\bar{x}}$ – остаточные ФАЛ, получаемые в результате подстановки вместо переменной x константы 1 и 0 соответственно.

Данная оптимизационная задача может быть решена полным перебором, но этот подход неприменим для ФАЛ с большим числом переменных из-за его экспоненциальной сложности. Вместо него предлагаются две следующих стратегии для приближенного решения указанной оптимизационной задачи.

Для ФАЛ f её булевой разностью (производной) по переменной x будем называть ФАЛ

$$\frac{\partial f}{\partial x} = f\big|_x \oplus f\big|_{\bar{x}}.$$

В свою очередь, булевым градиентом $g(f,x)$ ФАЛ f по переменной x назовем следующую целочисленную функцию:

$$g(f,x) = \sum_{\tilde{\sigma} \in B^{n-1}} \frac{\partial f}{\partial x}(\tilde{\sigma}),$$

где B^{n-1} – множество всех наборов из 0 и 1 длины $(n-1)$. Таким образом, на каждом этапе разложения выбирается такая переменная, на которой достигается максимум функции $g(f,x)$ (если таких переменных несколько, то переменная выбирается случайно среди всех переменных, обладающих указанным свойством).

Другая стратегия заключается в том, что функция $g(f,x)$ может быть использована для того, чтобы задать вероятность выбора переменной:

$$P_f(x = \hat{x}) = \frac{g(f,\hat{x})}{\sum_{x \in X(n)} g(f,x)}.$$

В этом случае на каждом шаге разложения генерируется значение случайной величины, которая имеет дискретное распределение, задаваемое вероятностями $P_f(x=\hat{x})$, $\hat{x} \in X(n)$, выбора заданной переменной. Значение указанной случайной величины используется для выбора переменной, по которой производится разложение.

Экспериментально было показано, что указанные модификации позволяют получать КС, сложность которых сопоставима с минимальной сложностью КС, построенных с использованием классического метода каскадов при переборе всех возможных вариантов выбора глобального порядка следования переменных разложения. При этом для ряда ФАЛ построенные КС имели сложность меньше, чем сложность минимальных схем, полученных при помощи классического метода каскадов.

C. Алгоритмы построения контактных схем на основе тупиковых ДНФ и контактных деревьев

Пусть задана некоторая ФАЛ $f(x_1,..., x_n)$, а также пусть N_f, как обычно, – её характеристическое множество, то есть множество наборов, на котором данная ФАЛ принимает значение 1. Зафиксируем некоторый порядок $T = (x_{i_1}, \ldots, x_{i_n})$ переменных ФАЛ f и возьмём такую тупиковую ДНФ $\mathbf{A} = K_1 \vee ... \vee K_m$, реализующую ФАЛ f в каждой элементарной конъюнкции K_i, $i = 1, \ldots, m$, в которой порядок следования переменных (в прямом или инверсном виде) соответствует заданному порядку T.

На основе ДНФ $\mathbf{A}$ индукцией по i, $i = 0, 1, \ldots, m$, построим (1, i)-КС $D_i^{\mathbf{A}}$, представляющую собой контактное дерево, корень которого является её входом, а листья – выходами, причем в выходе с номером j, $1 \le j \le i$, реализуется ФАЛ K_j. Для этого положим, что дерево $D_0^{\mathbf{A}}$ состоит из одной вершины-корня, не являющейся выходом, и что переход от уже построенного на i-м шаге дерева $D_i^{\mathbf{A}}$ к дереву $D_{i+1}^{\mathbf{A}}$ происходит на основе операции добавления к нему элементарной конъюнкции K_{i+1} следующим (рекурсивным) способом:

1) если переменная x_k или её отрицание является первым сомножителем элементарной конъюнкции K_{i+1}, и существует ребро с соответствующей пометкой, соединяющее корень дерева $D_i^{\mathbf{A}}$ с некоторым поддеревом $\hat{D}$, то дерево $D_{i+1}^{\mathbf{A}}$ получается из дерева $D_i^{\mathbf{A}}$ заменой поддерева $\hat{D}$ деревом, которое является результатом добавления к нему элементарной конъюнкции K', получающейся в результате удаления переменной x_k или её отрицания из K_{i+1};

2) если такого ребра не существует, то присоединим к корню дерева $D_i^{\mathbf{A}}$ цепочку рёбер с пометками, соответствующими вхождениям переменных в элементарную конъюнкцию K_{i+1}, а концевую вершину этой цепочки объявим $(i+1)$ листом (выходом) дерева $D_{i+1}^{\mathbf{A}}$;

3) контактным деревом $D^{\mathbf{A}}$ для тупиковой ДНФ $\mathbf{A}$ назовём дерево $D_m^{\mathbf{A}}$.

Если все листья контактного дерева $D^{\mathbf{A}}$ для тупиковой ДНФ $\mathbf{A}$ объединить в одну вершину, то получится базовая КС Σ с одним входом и одним выходом, реализующая функцию f. В общем случае сложность полученной КС может быть достаточно велика, поэтому КС такого вида не представляют большого интереса.

Из описанной выше базовой КС Σ можно получить КС Σ' путём «склеивания», то есть отождествления по вершинам и контактам одинаковых контактных цепей, исходящих из выходной вершины схемы Σ. Операция склеивания цепей уменьшает сложность исходной схемы Σ ,1 и при этом результирующая схема Σ' может по-прежнему реализовывать ФАЛ f, если при склеивании в ней не образовались дополнительные проводящие цепи. Таким образом, в ряде случаев, перебирая контактные деревья для всех тупиковых ДНФ заданной ФАЛ f и меняя порядок переменных $T = (x_{i_1}, \ldots, x_{i_n})$, можно получить контактную схему Σ', реализующую функцию f с меньшей сложностью.

Если же после склеивания цепей схемы Σ получается схема Σ', которая реализует ФАЛ f', отличную от ФАЛ f, то значит в схеме Σ' появились некоторые дополнительные проводящие цепи и $f \to f' \equiv 1$. В этом случае для построения искомой КС можно рассмотреть ФАЛ f'', для которой $N_{f''} \subset N_f$ и которая реализуется ДНФ $\mathbf{A}''$, получающейся добавлением переменных или их отрицаний в отдельные конъюнкции ДНФ $\mathbf{A}$. Тогда после построения дерева $D^{\mathbf{A}''}$, объединения его выходов и последующего склеивания цепей может получиться так, что характеристическое множество функции, которую реализует полученная схема Σ'', совпадет с характеристическим множеством функции f, и, следовательно, схема Σ'' будет реализовывать функцию f.

Таким образом, можно указать следующие основные этапы работы алгоритма:

1) перебор всех тупиковых ДНФ функции f (включая порядок вхождения переменных и их отрицаний в элементарные конъюнкции);

2) перебор для каждой тупиковой ДНФ всевозможных сужений характеристического множества;

3) построение контактного дерева и объединение его листовых вершин для получения базовой КС;

4) перебор для полученной КС всех вариантов склеивания цепей;

5) выбор из всех вариантов пункта 4 той КС, которая реализует функцию f с минимальной сложностью.

Построение всех тупиковых ДНФ функции f можно осуществить из её сокращённой ДНФ, которую можно построить при помощи стандартного алгоритма Квайна. Для этого перебираются множества дизъюнктов, которые можно исключить из сокращённой ДНФ, не изменив при этом характеристическое множество.

Описанный алгоритм имеет очень высокую вычислительную сложность. Во-первых, число тупиковых ДНФ некоторых ФАЛ f может быть очень велико. Во-вторых, для каждой ДНФ имеется множество сужений характеристического множества. В-третьих, существует огромное число вариантов склеивания цепей в КС. Поэтому был реализован рандомизированный алгоритм, в котором случайным образом выбирались тупиковые ДНФ, которые необходимо рассматривать, случайно выбирались способы сужения характеристического множества и склеивания цепей в КС.

III. База данных оптимальных и близких к ним контактных схем

В целях решения задачи хранения актуальной базы схем была реализована база данных [17], которая содержит сведения о сложности реализации булевых функций пяти переменных в классе КС, а также о структуре соответствующих схем и минимальности некоторых из них. Для ее заполнения были разработаны алгоритмы синтеза (поиска) схем, основанные, в частности, на описанных выше подходах. Разработанные алгоритмы были программно реализованы и запущены на суперкомпьютере «Ломоносов» Московского государственного университета имени М.В. Ломоносова. Содержательная часть базы данных является результатом работы соответствующих программ.

Для решения задачи хранения оптимальных или близких к ним схем для булевых функций пяти переменных необходим эффективный способ представления данных. С целью сокращения требуемой для хранения информации памяти осуществляется переход от булевых функций к их инверсно-конгруэнтным классам, в которых по схеме, реализующей любого представителя рассматриваемого класса функций, легко строится схема для любого другого представителя из того же класса, имеющая такую же структуру и сохраняющая все интересующие нас функционалы сложности.

Для 1 219 000 из около 1 228 000 инверсно-конгруэнтных типов ФАЛ от 5 переменных были построены нетривиальные контактные схемы, сложность которых сильно отличается от сложности схем для тех же функций, построенных известными методами синтеза (например, метод каскадов с оптимизацией порядка разложения переменных и построение случайных КС). Для всех функций четырех переменных были получены минимальные КС. Распределение сложностей оптимальных схем, хранящихся в базе данных, для всех возможных инверсно-конгруэнтных классов представлено на рисунке 3, из которого видно, что любая ФАЛ от пяти переменных может быть реализована [18] со сложностью, не превосходящей 22, причем 343 инверсно-конгруэнтных класса реализованы со сложностью 22, а остальные имеют меньшую сложность реализации.

Как уже было сказано ранее, КС, реализующая представителя данного инверсно-конгруэнтного класса, содержит всю необходимую информацию о любой ФАЛ этого класса, поэтому информация в базе данных хранится только для канонических представителей указанных классов ФАЛ, столбцы значений которых являются лексикографически минимальными (при фиксированном порядке следования переменных) среди ФАЛ, относящихся к рассматриваемому классу. Столбец значений канонического представителя класса называется *минкодом* этого класса.

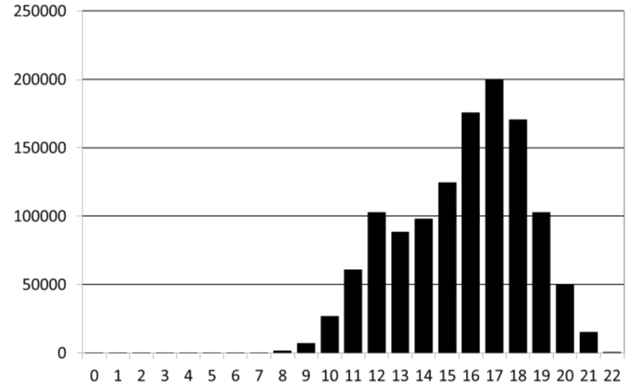

Рис. 3. Распределение сложностей инверсно-конгруэнтных типов в классе КС

В базе данных для каждой функции хранится ее минкод (ключ) и текущее оптимальное значение сложности схемы. В другой таблице хранятся представления схем, где для каждого представления также хранится минкод функции, которую оно реализует (внешний ключ). Такой подход позволяет хранить историю изменения базы и оценивать вклад новых алгоритмов в процесс оптимизации всей базы схем. Для выполнения распределенных вычислений по каждой из функций хранится количество запусков алгоритма на этой функции, что позволяет более равномерно распределить задачи по вычислительным узлам, избегая частых вычислений одинаковых функций.

IV. Структура и особенности функционирования распределенной среды для поиска минимальных и близких к ним контактных схем

В целях решения задачи наполнения базы данных и распределения вычислений была реализована клиент-серверная система. Сервер позволяет получить вычислительную задачу, которая состоит из набора функций для оптимизации. Результат вычисления этой задачи клиент отправляет на сервер в течение определенного времени. В это время другие клиенты получают другие наборы функций с целью избежать дублирования вычислений. Такая система является гибкой в плане использования различных алгоритмов и аппаратных средств. Вычисления возможны на любых устройствах, начиная с персональных компьютеров и заканчивая массивно-параллельными системами с общей памятью. Сервер также не привязан к виду и типу используемого алгоритма, поэтому возможно расширение методов оптимизации без модификации серверной части. Обмен данными основан на json/rest, что упрощает реализацию клиентской части за счет простого протокола и абстрагирования от способа хранения данных. Сторонние разработчики также могут применить свой алгоритм и протестировать его на базе контактных схем. Реализованная система является многопользовательской открытой системой, поэтому производится верификация поступающих данных.

Для решения задачи оптимизации базы схем переборными алгоритмами была реализована система для распределенных вычислений с использованием протокола передачи сообщений (англ. Message Passing Interface, MPI) версии 3.1 [19]. Главный узел распределенной системы хранит полную актуальную базу контактных схем, которая загружается на момент запуска из базы данных. На остальных вычислительных узлах хранится локальная база минкодов уже обработанных функций, что позволяет сократить объем требуемой памяти и не проводить повторного вычисления минкода. Также на каждом вычислительном узле хранится актуальная база сложностей реализаций ФАЛ, что снижает объем передаваемых по сети данных. При найденном улучшении сложности какой-либо функции на главный узел отправляется уведомление. Для обновления информации о сложностях схем всем вычислительным узлам рассылается широковещательное сообщение, которое обновляет локальную базу сложностей на каждом из узлов. Такой подход позволяет проводить эффективные вычисления с использованием относительно медленной локальной сети между MPI-узлами.

Приложения для оптимизации схем методом каскадов и методами, основанными на тупиковых ДНФ, запускаются на любых системах пользовательского уровня и обмениваются данными с сервером по специальному протоколу.

Реализованные система и база данных позволяет также работать и со схемами других типов, таких как, например, схемы из функциональных элементов.

Поддержка

Работа выполнена при финансовой поддержке РФФИ – грант № 15-01-07474-а.

Литература

[1] Mishchenko A., Chatterjee S., Brayton R. DAG-aware AIG rewriting: A fresh look at combinational logic synthesis // Proc. DAC'06. 2006. P. 532-536.

[2] Shannon C.E. A symbolic analysis of relay and switching circuits // Trans. Amer. Inst. Electrical Engineers. — Vol. 57. — 1938. — P. 713-723.

[3] Shannon C.E. The synthesis of two-terminal switching circuits // Bell System Technical Journal. — Vol. 28, No. 1. — 1949. — P. 59-98.

[4] Ложкин С.А., Сапоженко А.А. Методы логического проектирования и оценки сложности схем на дополняющих МОП-транзисторах // Журнал Микроэлектроника. — 1983. — Т. 12, № 1. — С. 42–47.

[5] Polya G.J. Sur les Types des Propositions Composees // Symb. Logik. 1940. Vol. 5, №3. P. 98-103.

[6] Синтез электронных вычислительных и управляющих систем. Пер. с англ. под ред. Шестакова В.И. М.: ИЛ, 1954.

[7] Васильев Ю.Л. Минимальные контактные схемы для булевых функций четырёх переменных // ДАН СССР. 1959. Т. 127, №2. С. 242-245.

[8] Поваров Г.Н. Исследование контактных схем с минимальным числом контактов. Диссертация, ИАТ АН СССР, 1954.

[9] Van der Poel W.L. Engine bijzondere onderwerpen uit de schakelalgebra // De Ingenieur (Utrecht). 1955. V. 67, No. 1. P. E.9-E.14.

[10] Higonnet R., Grea R. Etude logique des circuits electriques et des systemes binairs. Paris, 1955.

[11] Сусов В.Ю. Два алгоритма переборного типа для синтеза минимальных контактных схем и их реализация. Дипломная работа. ВМК МГУ. М. 1981.

[12] Поваров Г.Н. Математико-логическое исследование синтеза контактных схем с одним входом и k выходами // Сб. Логические исследования, ИАН СССР. М.: Наука, 1959. С. 379-405.

[13] Huang, Z., Wang, L., Nasikovskiy, Y. and Mishchenko, A., 2013, December. Fast Boolean matching based on NPN classification. In FPT (pp. 310-313).

[14] McKay B. «Nauty and Traces» [Электронный ресурс] // URL: http://pallini.di.uniroma1.it/ (дата обращения: 04.04.2016).

[15] McKay B.D., Piperno A. Practical Graph Isomorphism II // J. Symbolic Computation. 2013. Vol. 60. P. 94-112.

[16] Поваров Г.Н. Метод синтеза вычислительных и управляющих контактных схем // Автоматика и телемеханика. 1957. Т. 18, №2. С. 145-162.

[17] Ложкин С.А., Шуплецов М.С., Коноводов В.А., Данилов Б.Р. База данных «Значения функционалов сложности и известные оптимальные схемы из функциональных элементов и контактные схемы для булевых функций» [Электронный ресурс] // URL: http://mks2.cmc.msu.ru/ (дата обращения: 04.04.2016).

[18] Ложкин С.А., Шуплецов М.С., Коноводов В.А., Данилов Б.Р., Жуков В.В., Багров Н.Ю. Об уточнении значений функционала сложности контактных схем для булевых функций от пяти переменных // Материалы X молодежной научной школы по дискретной математике и ее приложениям (Москва, 5–11 октября 2015 г.). М.: ИПМ им. М.В. Келдыша, 2015. С. 42-46.

[19] MPI: A Message-Passing Interface Standard, Version 3.1. High Performance Computing Center Stuttgart, 2015. 868 p.

Distributed system and switching circuits optimization methods for Boolean functions of small number of variables

S.A. Lozhkin, M.S. Shupletsov, V.A. Konovodov,

B.R. Danilov, V.V. Zhukov, N.Yu. Bagrov

Lomonosov Moscow State University – Faculty of Computational Mathematics and Cybernetics, lozhkin@cs.msu.ru

***Keywords* — switching circuit, size-complexity, circuits synthesis methods, database, cascade circuit synthesis.**

ABSTRACT

The synthesis of optimal or suboptimal switching circuits is an actual problem of theory of discrete control systems. Libraries of such circuits could be used in various algorithms of logic synthesis (e.g., see [1]). The structure analysis of optimal circuits for functions of few variables may be useful in the development of standard cell libraries.

The first catalogs of minimal switching circuits that implement Boolean functions (BF) of small number of variables appeared [5,6,7] in the 1950s. They contained the upper bounds for complexities of all typical 4-variables BF (total 402 functions).

In [8] G.N. Povarov showed that only one of these 402 BF’s requires at most 14 contacts, four BF’s require at most 13 contacts, and the rest require at most 12 contacts. Then various authors [7-11] showed that only one of them has the complexity 13. In [3] Shannon showed that upper bound for 5-variables BF in the class of switching circuits is 30 contacts. Later, G.N. Povarov [12] lowered this estimate to 28. V.Yu. Susov [11] has found the BF of 5 variables, complexity of which is at least 19. Thus, the question of finding more precise upper bounds for the complexity of 5-variable BF in the class of switching circuits remains relevant.

A number of circuit synthesis algorithms have been developed to approach the problem for 5-variable BF’s. These algorithms have been implemented in a number of tools that have been used to improve known circuits for a large number of 5-variable BF’s. As a result, it was shown that upper bound for 5-varibale BF’s size-complexity is 22.

REFERENCES

[1] Mishchenko A., Chatterjee S., Brayton R. DAG-aware AIG rewriting: A fresh look at combinational logic synthesis // Proc. DAC’06. 2006. P. 532-536.
[2] Shannon C.E. A symbolic analysis of relay and switching circuits // Trans. Amer. Inst. Electrical Engineers. — Vol. 57. 1938. P. 713-723.
[3] Shannon C.E. The synthesis of two-terminal switching circuits // Bell System Technical Journal. — Vol. 28, No. 1. 1949. P. 59-98.
[4] Lozhkin S.A., Sapozhenko A.A. Metodi logicheskogo proektirovanija i ozenki slojnosti shem na dopolnjayushih MOP-tranzistorah // Jurnal Mikroelektronika. 1983. Vol. 12, No. 1. P. 42–47 (in Russian).
[5] Polya G.J. Sur les Types des Propositions Composees // Symb. Logik. 1940. Vol. 5, No. 3. P. 98-103.
[6] Sintez elektronnyh vychislitel'nyh i upravljajushih sistem. Per. s angl. pod red. Shestakova V.I. Moscow, IL, 1954 (in Russian).
[7] Vasil'ev Ju.L. Minimal'nye kontaktnye shemy dlja bulevyh funkcij chetyrjoh peremennyh // DAN SSSR. 1959. Vol. 127, No. 2. P. 242-245 (in Russian).
[8] Povarov G.N. Issledovanie kontaktnyh shem s minimal'nym chislom kontaktov. Dissertacija, IAT AN SSSR, 1954 (in Russian).
[9] Van der Poel W.L. Engine bijzondere onderwerpen uit de schakelalgebra // De Ingenieur (Utrecht). 1955. V. 67, No. 1. P. E.9-E.14.
[10] Higonnet R., Grea R. Etude logique des circuits electriques et des systemes binairs. Paris, 1955.
[11] Susov V.Ju. Dva algoritma perebornogo tipa dlja sinteza minimal'nyh kontaktnyh shem i ih realizacija. Diplomnaja rabota. VMK MGU. Moscow, 1981 (in Russian).
[12] Povarov G.N. Matematiko-logicheskoe issledovanie sinteza kontaktnyh shem s odnim vhodom i k vyhodami // Sb. Logicheskie issledovanija, IAN SSSR. Moscow, Nauka, 1959. P. 379-405 (in Russian).
[13] Huang, Z., Wang, L., Nasikovskiy, Y. and Mishchenko, A., 2013, December. Fast Boolean matching based on NPN classification. In FPT (pp. 310-313).
[14] McKay B. «Nauty and Traces» [Jelektronnyj resurs] // URL: http://pallini.di.uniroma1.it/ (accessed: 04.04.2016).
[15] McKay B.D., Piperno A. Practical Graph Isomorphism II // J. Symbolic Computation. 2013. Vol. 60. P. 94-112.
[16] Povarov G.N. Metod sinteza vychislitel'nyh i upravljajushhih kontaktnyh shem // Avtomatika i telemehanika. 1957. T. 18, No. 2. P. 145-162 (in Russian).
[17] Lozhkin S.A., Shupletsov M.S., Konovodov V.A., Danilov B.R. Baza dannyh «Znachenija funkcionalov slozhnosti i izvestnye optimal'nye shemy iz funkcional'nyh jelementov i kontaktnye shemy dlja bulevyh funkcij» [Jelektronnyj resurs] // URL: http://mks2.cmc.msu.ru/ (accessed: 04.04.2016) (in Russian).
[18] Lozhkin S.A., Shuplecov M.S., Konovodov V.A., Danilov B.R., Zhukov V.V., Bagrov N.Ju. Ob utochnenii znachenij funkcionala slozhnosti kontaktnyh shem dlja bulevyh funkcij ot pjati peremennyh // Materialy X molodezhnoj nauchnoj shkoly po diskretnoj matemati- ke i ee prilozhenijam (Moskva, 5–11 oktjabrja 2015 g.). Moscow, IPM im. M.V. Keldysha, 2015. P. 42-46 (in Russian).
[19] MPI: A Message-Passing Interface Standard, Version 3.1. High Performance Computing Center Stuttgart, 2015. 868 p.