

## ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

УДК 519.7

П.Н. Бибило, Н.А. Кириенко

ОПТИМИЗАЦИОННЫЕ ПРЕОБРАЗОВАНИЯ ЛОГИЧЕСКОЙ СХЕМЫ  
НА ОСНОВЕ БЛОЧНОГО РАЗБИЕНИЯ

*Исследуется возможность использования метода разбиения логических сетей для минимизации сложности логических схем, синтезируемых в системе проектирования Leonardo. Описывается технология проведения эксперимента, полученные экспериментальные результаты представлены в таблицах. Делается вывод об эффективности применения процедуры разбиения для минимизации сложности многоуровневых схем, синтезируемых в библиотеке проектирования базовых матричных кристаллов (БМК).*

**Введение**

Задача разбиения логических схем на подсхемы известна давно и широко применяется при проектировании в различных элементных базисах. Например, при синтезе схем в базисе Field Programmable Gate Array (FPGA) возникает задача разбиения на подсхемы, удовлетворяющие некоторым ограничениям на число входных и выходных переменных. Решение этой задачи приводит к размещению каждой полученной подсхемы на одной FPGA. Алгоритм разбиения, предлагаемый в [1], позволяет минимизировать число блоков разбиения, а также число связей между блоками, что приводит к оптимизации площади кристалла.

В работе [2] утверждается, что реализация схем в виде сети программируемых логических матриц (ПЛИС) приблизительно одинаковой размерности позволяет получить более быстрые, компактные, помехоустойчивые решения, чем при реализации на одной ПЛИС. Выполнение процедуры сокращения числа соединений между отдельными ПЛИС позволяет еще больше сократить площадь кристалла [3]. В [4] предлагается алгоритм реализации логической схемы в виде сети ПЛИС-блоков, позволяющий минимизировать число блоков, что ведет к сокращению площади кристалла. Эти задачи также сводятся к задаче разбиения логических схем.

Логические схемы в процессе проектирования могут рассматриваться на различных уровнях детализации проекта, в частности на системном, логическом, схемотехническом. На любом из этих уровней моделью логической схемы может быть ориентированный граф. Элементам схемы соответствуют вершины графа, соединениям между элементами – дуги графа. По методам обработки графа алгоритмы разбиения делятся на три класса [5]. Первый класс алгоритмов разбиения не изменяет структуру графа, второй может модифицировать граф путем копирования вершин, которое позволяет сократить число дуг в разрезе графа. Третий класс служит для взвешенных графов, каждой вершине которых соответствует некоторая логическая функция. Разбиение графа в этом случае сопровождается некоторыми оптимизационными преобразованиями подграфов, в частности, может произойти «слияние» некоторых вершин подграфа и в связи с этим изменение логических функций вершин.

По типам алгоритмов методы делятся на конструктивные и итеративные [6]. С помощью конструктивных методов проводят разбиение исходного графа путем группирования вершин в блоки, удовлетворяющие некоторым ограничениям. В большинстве конструктивных процедур разбиения предпринимается попытка сгруппировать в один подграф (блок) сильно связанные элементы (имеющие много соединений друг с другом). С помощью итеративных методов пытаются улучшить существующее разбиение исходного графа путем перераспределения вершин между подграфами, следуя определенным эвристикам.

В настоящей работе исследуется возможность использования алгоритма [7] разбиения комбинационных логических сетей на блоки ограниченных размеров для минимизации сложности логических схем. Ограничениями на сложность каждого из блоков являются числа входных и

выходных переменных блока. После разбиения сети на блоки функции каждого блока представляются в виде дизъюнктивных нормальных форм (ДНФ) и минимизируются в классе ДНФ, затем осуществляется синтез логической схемы в библиотеке проектирования БМК. В качестве синтезатора используется Leonardo [8]. Проведенный эксперимент показывает, что такое предварительное оптимизационное преобразование логической сети, основанное на процедуре блочного разбиения, позволяет уменьшить сложность схем БМК.

### 1. Метод разбиения логической сети

Один из алгоритмов, реализующих рассматриваемый метод, представлен в [7]. Он относится к третьему классу и является конструктивным. Алгоритм реализован в виде процедуры в системе автоматизированного проектирования Custom Logic [9].

Исходная логическая сеть представлена на функциональном уровне, а именно в виде системы логических уравнений. Результирующая сеть представлена на структурном уровне – в виде сети из блоков, каждый из которых является подсистемой логических уравнений, полученных в результате разбиения исходной системы. Для удобства описания метода разбиения вводится графовая модель представления логической сети. Текущим представлением логической сети является взвешенный ориентированный граф  $G = (V, E)$ . Вершине  $v_i \in V$  соответствует логическое уравнение. Две вершины  $v_i, v_j$  графа  $G$  связывает дуга  $e_{ij}$ , принадлежащая множеству дуг  $E$  (дуга исходит из вершины  $v_i$  и заходит в вершину  $v_j$ ), в том и только в том случае, когда выходная переменная уравнения, соответствующего  $v_i$ , является входной переменной уравнения, соответствующего  $v_j$ . Пусть граф состоит из  $m$  вершин и  $k$  дуг.

Задачу разбиения исходной системы логических уравнений на подсистемы можно представить как задачу разбиения множества  $V$  вершин графа  $G$  на множество непересекающихся подмножеств вершин  $\{V_1, \dots, V_p\}$  графа (блоков), таких, что  $V_i \subset V, V_i \neq \emptyset, V_i \cap V_j = \emptyset, \bigcup_{i=1}^p V_i = V$

для  $i, j \in \{1, \dots, p\}, i \neq j$ .

Разбиение можно рассматривать как процесс преобразования исходного графа, представляющего систему логических уравнений, в результирующий граф, представляющий взаимосвязь полученных подсистем логических уравнений, или блоков. Вершина графа в предлагаемой модели может иметь числовые характеристики, которыми могут являться, например, числа входных, выходных переменных в подсистеме логических уравнений, соответствующих этой вершине.

В настоящей работе рассматривается задача разбиения графа, представляющего систему логических уравнений, на минимальное число  $p$  блоков  $V_i$  при выполнении заданных ограничений на числа входных ( $n$ ) и выходных ( $m$ ) переменных блоков  $V_i$ . Под входными, выходными и внутренними переменными блоков будем понимать входные, выходные и внутренние переменные соответствующих подсистем уравнений. В настоящей работе предлагается последовательный метод формирования блоков.

Построение каждого нового блока включает следующие этапы:

- 1) выбор начальной вершины для блока и исключение ее из исходного множества вершин;
- 2) построение множества смежных вершин для блока;
- 3) просмотр множества смежных вершин и выбор очередной вершины для включения в блок, исключение выбранной вершины из исходного множества;
- 4) преобразование подсистемы логических уравнений, соответствующей данному текущему состоянию блока, с целью сокращения числа внутренних переменных блока.

Этапы 2–4 повторяются до тех пор, пока возможно включение уравнений в подсистему без нарушения заданных ограничений.

Следуя выбранной графовой модели, перечисленные этапы можно представить как выбор вершин из исходного графа и объединение их в подграф (блок). Сформированный подграф будет характеризоваться параметрами, равными числу входных ( $n$ ) и выходных ( $m$ ) переменных в соответствующей подсистеме. Значения параметров каждого из блоков разбиения не должны превышать заданных ограничений.

Описываемый метод разбиения рассмотрим на примере функционального описания логической схемы, заданной системой логических уравнений (1).

Через « $\vee$ » обозначена логическая операция дизъюнкция, через « $\bar{\phantom{x}}$ » – отрицание, обозначение операции конъюнкции « $\wedge$ » для простоты опускается. Эту систему можно представить с помощью графа (рис. 1), где каждому уравнению соответствует вершина. Входными переменными системы являются  $a, b, c, d, e, f, g, h, i, j, k$ ; выходными –  $l, m, n$ ; промежуточными –  $a_0, d_0, e_0, f_0, g_0, h_0, i_0, j_0, k_0, l_0, m_0, n_0, q_0, t_0, u_0, v_0, w_0$ . Чтобы не усложнять рис. 1, на нем показаны входные переменные лишь для некоторых уравнений.

Для данного примера задача ставится следующим образом. Требуется разбить исходную систему на минимальное число подсистем (блоков), для каждой из которых выполняются ограничения  $n \leq 5, m \leq 3$ .

Рассмотрим систему:

$$\begin{aligned}
 l &= \bar{n}_0 \vee \bar{v}_0 \bar{t}_0; & h_0 &= \bar{f}_0 \bar{g}_0; \\
 m &= \bar{v}_0 \bar{w}_0; & i_0 &= \bar{v} \vee \bar{e}_0 \bar{f}_0; \\
 n &= \bar{l}_0 \vee \bar{v}_0 \bar{u}_0; & j_0 &= \bar{x} \vee \bar{f}_0 \bar{d}_0; \\
 u &= \bar{e} d; & k_0 &= \bar{i} h; \\
 v &= \bar{b} \bar{c} \vee \bar{u} \bar{c}; & l_0 &= \bar{h}_0 \bar{i}_0 \vee \bar{k}_0 \bar{i}_0; \\
 w &= e \bar{d}; & m_0 &= \bar{i} \bar{h}; \\
 x &= \bar{b} \bar{a} \vee \bar{w} \bar{a}; & n_0 &= \bar{h}_0 \bar{j}_0 \vee \bar{m}_0 \bar{j}_0; \\
 a_0 &= d e \vee \bar{d} \bar{e}; & q_0 &= h i \vee \bar{h} \bar{i}; \\
 d_0 &= \bar{g} \vee f; & t_0 &= \bar{k} \vee j; \\
 e_0 &= g \vee \bar{f}; & u_0 &= k \vee \bar{j}; \\
 f_0 &= \bar{b} \vee \bar{a}_0; & v_0 &= \bar{h}_0 \vee \bar{q}_0; \\
 g_0 &= f \bar{g} \vee \bar{f} g; & w_0 &= \bar{j} k \vee j \bar{k}.
 \end{aligned}
 \tag{1}$$

В результате работы алгоритма [7] система (1) будет представлена в виде четырех подсистем, обозначенных пунктирными линиями на рис. 1. В процессе формирования подсистем выполняется элиминация внутренних переменных: в полученных подсистемах отсутствуют соответствующие уравнения ( $t_0, w_0, u_0, k_0$  и др.).

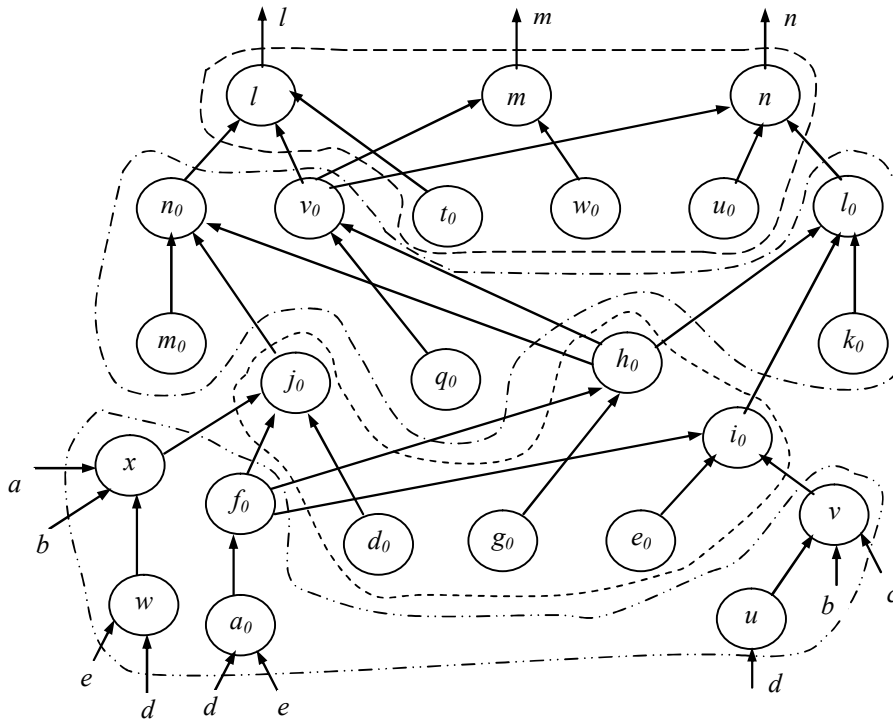


Рис. 1. Графовая модель системы логических уравнений (1)

Результатом решения задачи являются следующие подсистемы, представленные в графовом виде на рис. 2:

$$\begin{aligned} l &= \bar{v}_0 \bar{j} k \vee \bar{n}_0; \\ m &= \bar{v}_0 \bar{k} \bar{j} \vee \bar{v}_0 k j; \\ n &= \bar{l}_0 \vee \bar{v}_0 \bar{k} j; \end{aligned} \quad (2)$$

$$\begin{aligned} l_0 &= \bar{i}_0 i \vee \bar{i}_0 \bar{h} \vee \bar{h}_0 \bar{i}_0; \\ v_0 &= \bar{h}_0 \vee h \bar{i} \vee \bar{h} i; \\ n_0 &= \bar{h}_0 \bar{j}_0 \vee \bar{i} \bar{j}_0 \vee h \bar{j}_0; \end{aligned} \quad (3)$$

$$\begin{aligned} j_0 &= g \bar{f} \bar{f}_0 \bar{x}; \\ h_0 &= g f \bar{f}_0 \vee \bar{g} \bar{f} \bar{f}_0; \\ i_0 &= \bar{v} \vee \bar{g} f \bar{f}_0; \end{aligned} \quad (4)$$

$$\begin{aligned} v &= \bar{c} e \vee \bar{c} \bar{d} \vee \bar{b} \bar{c}; \\ x &= \bar{b} \bar{a} \vee \bar{e} \bar{a} \vee d \bar{a}; \\ f_0 &= \bar{b} \vee d \bar{e} \vee \bar{d} e. \end{aligned} \quad (5)$$

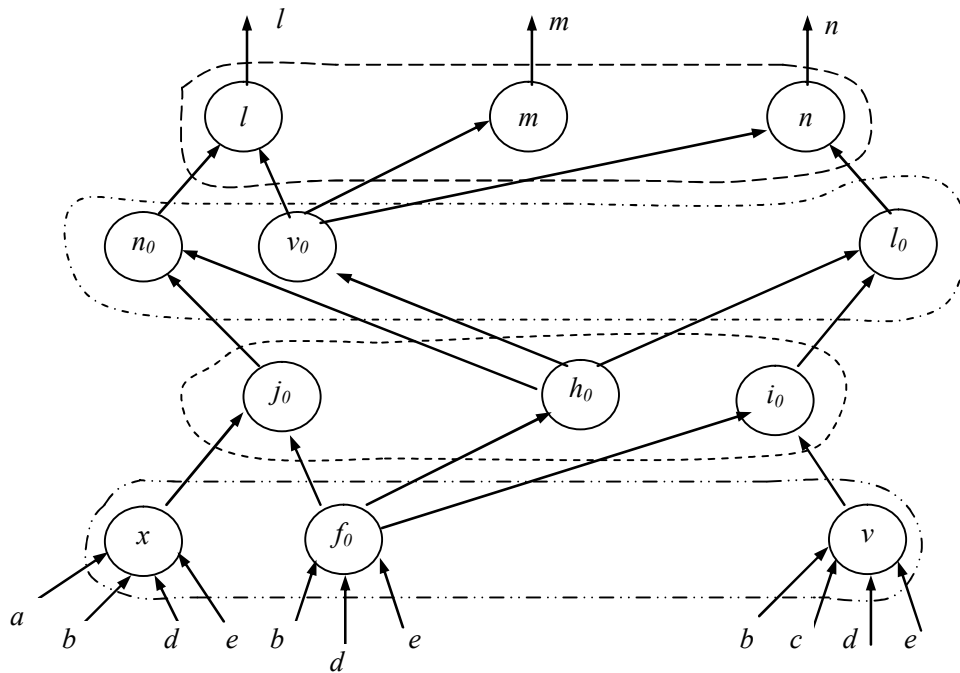


Рис. 2. Графовая модель решения задачи разбиения системы логических уравнений (1)

Как видим, уменьшилось общее количество конъюнкций и внутренних переменных в функциональном описании, что имеет важное значение для дальнейшего синтеза логической схемы. Таким образом, предлагаемый метод отличается от известных методов разбиения схем на блоки наличием этапа 4, производящего внутреннюю перестройку формируемых подсистем: элиминацию внутренних переменных и минимизацию в классе ДНФ подсистем булевых функций.

## 2. Особенности реализации этапов разбиения

В настоящей работе исследовались два алгоритма разбиения, первый назовем «по входам», второй – «полный». Для них этапы 1 и 4 одинаковы.

Рассмотрим алгоритм «по входам». На этапе 1 в качестве начальной вершины подсистемы выбиралась произвольная выходная вершина.

На этапе 2 алгоритм «по входам» формирует множество смежных вершин из уравнений, выходные переменные которых есть среди входных переменных подсистемы. Назовем это множество  $rA\_Inp$ .

На этапе 3 осуществляется выбор очередного уравнения из множества  $rA\_Inp$ , включение его в подсистему; при этом, естественно, не должны быть нарушены ограничения на число входных и выходных переменных подсистемы. Если ограничения нарушаются, то уравнение не включается, а выбирается следующее уравнение из множества  $rA\_Inp$ .

На этапе 4 выполняется преобразование полученной подсистемы. Осуществляется попытка выполнения элиминации внутренних переменных и минимизации полученной подсистемы. Если элиминация невозможна, то уравнение подключается без элиминации.

Для каждого уравнения из множества  $rA\_Inp$  выполняются последовательно этапы 3 и 4. Множество  $rA\_Inp$  просматривается, и, если хотя бы одно уравнение включается в подсистему, выполняется построение нового множества  $rA\_Inp$  для преобразованной подсистемы. Процесс формирования подсистемы заканчивается, когда ни одно из уравнений очередного множества  $rA\_Inp$  не включается в подсистему.

Далее, если в исходной системе осталось хотя бы одно уравнение, начинается формирование следующей подсистемы. Иначе алгоритм заканчивает работу.

Алгоритм «полный» предполагает формирование на втором этапе нескольких множеств смежных вершин. Множество  $rA\_Inp$  имеет прежний смысл. Множество  $rV\_Inp$  содержит те же уравнения, что и  $rA\_Inp$ , только на них накладывается условие запрета элиминации их выходной переменной. Множество  $rAV\_Out$  содержит те уравнения, у которых в качестве хотя бы одной входной переменной используется выходная переменная формируемой подсистемы.

На этапе 3 в «полном» алгоритме необходимо выбрать стратегию для перебора построенных множеств смежных вершин с целью выбора очередного уравнения для включения. В эксперименте, описываемом в настоящей работе, множества просматриваются в следующем порядке:  $rAV\_Out$ ,  $rA\_Inp$ ,  $rV\_Inp$ . В остальных этапах 3 и 4 для алгоритма «полный» совпадают с такими же этапами алгоритма «по входам».

### 3. Организация экспериментов по исследованию эффективности процедуры разбиения

В качестве исходных данных для проведения экспериментов (рис. 3) было отобрано 30 примеров из библиотеки [10]: 6 примеров схем ПЛМ из класса «математические» и 24 примера ПЛМ из класса «промышленные».

Результующими данными были логические схемы в библиотеке проектирования БМК. Описание элементов библиотеки приведено в [11]. Сложность  $S_{БМК}$  схемы в библиотеке проектирования БМК (далее просто схемы БМК) подсчитывалась как сумма площадей входящих в данную схему элементов, а площадь элемента подсчитывалась в числе элементарных ячеек БМК. Элементарная ячейка БМК соответствует одному транзистору, входящему в сеть транзисторов, из которых состоит логический элемент.

Описания данных для синтеза в системе Custom Logic представлялись на языке SF [9], а для синтеза в системе Leonardo – на языке VHDL [8].

#### Эксперимент 1. Синтез схем в библиотеке проектирования БМК без процедуры разбиения

Этап 1.1 (совместная минимизация и пять итераций повторного синтеза в библиотеке БМК).

Сначала проводилась совместная минимизация системы ДНФ булевых функций в системе Custom Logic, затем проводился повторный итерационный синтез [12] в системе Leonardo.

Этап 1.2 (пять итераций повторного синтеза с поочередной сменой целевой библиотеки).

Первая итерация синтеза: сначала синтезировалась схема в базисе программируемых логических схем типа FPGA; затем к полученной логической схеме в библиотеке программируемых элементов FPGA применялась команда unpar, после чего по полученному RTL-описанию синтезировалась схема БМК; полученная логическая схема с помощью команды unpar преобразовывалась в RTL-описание, являющееся исходным для второй ите-

рации повторного синтеза. Всего проводилось пять итераций повторного синтеза. RTL-описания представляют системы логических уравнений, в которых в каждом из уравнений используется только одна из логических операций (дизъюнкция, конъюнкция, сумма по модулю 2, отрицание).

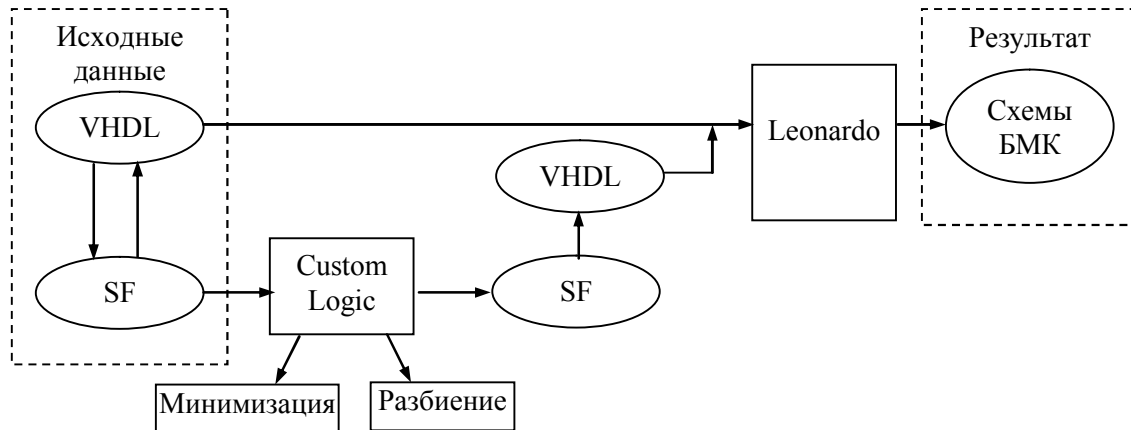


Рис. 3. Общая схема выполнения экспериментов

*Этап 1.3* (совместная минимизация и пять итераций повторного синтеза с поочередной сменой целевой библиотеки).

Сначала выполнялась совместная минимизация, затем этап 1.2 для минимизированных описаний. По результатам этапов 1.1, 1.2, 1.3 выбирались *лучшие решения* (логические схемы БМК), сложность  $S_{\text{БМК}}$  полученных схем БМК представлена в столбце *best of iterations* (табл. 1 и 2).

### **Эксперимент 2. Синтез схем в библиотеке проектирования БМК с процедурой разбиения**

Исходными данными в эксперименте 2 были RTL-описания схем, полученных в эксперименте 1.

Целью эксперимента 2 была проверка возможности улучшения решений (с помощью алгоритмов разбиения), полученных в эксперименте 1. Дополнительно в этом эксперименте была исследована эффективность применения процедур редукции при выполнении операций устранения иерархии описаний: *с редукцией* и *без редукции* уравнений связи. Дело в том, что процедура разбиения порождает структурное иерархическое описание. Чтобы получить RTL-описание, требуется устранить иерархию, а это может породить появление таких уравнений (назовем их уравнениями связи), которые имеют в правой части только одну переменную. Для устранения этих уравнений может быть выполнена операция редукции системы логических уравнений путем элиминации некоторых промежуточных переменных, тождественных константам или другим промежуточным или входным переменным. Операция редукции уравнений связи значительно сокращает размер системы уравнений.

Рассмотрим этапы эксперимента 2.

*Этап 2.1.* Редукция уравнений связи в исходных RTL-описаниях.

*Этап 2.2.* Выполнение алгоритма разбиения.

*Этап 2.3.* Устранение иерархии описания (с редукцией либо без редукции уравнений связи).

*Этап 2.4.* Синтез схемы БМК в синтезаторе Leonardo, получение значений  $S_{\text{БМК}}$  сложности схем БМК.

Эксперимент 2 состоит из множества экспериментов, каждый из них имеет свое имя (название), составленное из слов, характеризующих режимы эксперимента на каждом из этапов. Рассмотрим, как именуются конкретные эксперименты.

Если на этапе 2.1 выполнялась редукция уравнений связи, то в названии эксперимента присутствует слово «e». На этапе 2.2 разбиение на блоки выполнялось с помощью двух вышеописанных алгоритмов: «по входам» и «полный». В обозначении эксперимента использование алгоритма «по входам» кодируется словом «inp», использование алгоритма «полный» – словом

«full». Процедура разбиения предполагает задание ограничений:  $n$  – числа входных,  $m$  – числа выходных переменных в блоке. Значения параметров  $n$ ,  $m$  разбиения записываются в виде слова  $n_m$  в названии эксперимента. На этапе 2.3 выполняется процедура устранения иерархии: если при этом выполнялась редукция уравнений связи, то в названии эксперимента присутствует слово «red», иначе – слово «no\_red».

Таким образом, название эксперимента образуется путем объединения вышеперечисленных слов и однозначно определяет его суть. Например, эксперимент «e\_full\_8\_8\_no\_red» означает использование:

- предварительной редукции уравнений связи (e);
- алгоритма «полный» разбиения с параметрами  $n = 8$ ,  $m = 8$  (full\_8\_8);
- устранения иерархии без редукции уравнений связи (no\_red).

#### 4. Результаты экспериментов

Результаты экспериментов приведены в табл. 1–4. В табл. 1 представлены сложности схем БМК, полученных в результате эксперимента 2 с алгоритмом «по входам», в табл. 2 – с алгоритмом «полный». В столбце « $n$ ,  $m$ » табл. 1 и 2 представлены параметры схемы:  $n$  – число входных переменных,  $m$  – число выходных переменных. В столбце *best of iterations* (табл. 1 и 2) представлены лучшие решения, полученные по эксперименту 1.

Таблица 1

Результаты эксперимента 2 для алгоритма «по входам»: сложность  $S_{AiE}$  схем БМК

N	Имя схемы	$n, m$	$inp_{8_8}$ red	$inp_{8_8}$ no_red	$e_{inp_{8_8}}$ red	$e_{inp_{8_8}}$ no_red	$inp_{8_4}$ red	$inp_{8_4}$ no_red	$e_{inp_{8_4}}$ red	$e_{inp_{8_4}}$ no_red	$inp_{8_2}$ red	$inp_{8_2}$ no_red	$e_{inp_{8_2}}$ red	$e_{inp_{8_2}}$ no_red	best of inp	best of iterations
1	add6	12, 7	187	177	187	177	187	<b>159</b>	187	177	187	<b>159</b>	187	177	<b>159</b>	175
2	addm4	9, 8	1148	<b>1125</b>	1148	1220	1152	1143	1148	1220	1152	1143	1148	1220	<b>1125</b>	1136
3	b12	15, 9	195	208	195	208	189	190	195	208	189	190	195	208	189	188
4	b2	15, 17	2102	2063	2102	2060	2055	2059	2102	2060	2049	2048	2152	2108	2048	2048
5	b9	16, 5	<b>291</b>	323	<b>291</b>	323	324	309	<b>291</b>	323	324	309	<b>291</b>	336	<b>291</b>	297
6	bc0	21, 11	2565	2715	2565	2566	2655	2636	2565	2566	2694	2779	2706	2556	2556	2547
7	chkn	29, 7	1046	1091	1046	1038	1154	1028	1046	1038	1154	1028	1046	1038	1028	1010
8	dc2	8, 7	334	348	334	328	327	326	334	328	327	326	334	<b>298</b>	<b>298</b>	359
9	dk48	15, 17	398	398	398	405	403	408	398	405	403	408	398	405	398	398
10	fbm	48, 17	533	533	535	535	535	535	535	535	535	535	535	535	533	514
11	in0	15, 11	1621	1389	1621	1370	1543	1510	1621	1370	1438	1604	1621	1370	1370	1363
12	m2	19, 10	<b>1098</b>	1122	<b>1098</b>	1115	1119	1118	<b>1098</b>	1115	1109	1122	<b>1098</b>	1115	<b>1098</b>	1135
13	m6	33, 23	823	773	823	812	804	835	823	812	804	835	797	812	773	738
14	mtb	15, 7	5302	4905	5302	4897	6120	5415	5302	4897	6120	5415	5302	4897	4897	3594
15	m2	8, 16	673	683	673	683	659	<b>653</b>	673	683	659	<b>653</b>	673	683	<b>653</b>	682
16	m3	8, 16	969	934	969	973	988	963	969	973	988	963	941	973	934	899
17	misg	56, 23	199	181	<b>171</b>	<b>171</b>	<b>171</b>	181	<b>171</b>	<b>171</b>	<b>171</b>	181	<b>171</b>	<b>171</b>	<b>171</b>	181
18	newcpla1	9, 16	260	242	257	252	261	230	257	252	261	230	257	252	230	229
19	newtpla	15, 5	138	142	138	142	138	142	138	142	138	142	138	142	138	126
20	p82	5, 14	271	<b>269</b>	277	272	284	286	277	272	284	286	277	272	<b>269</b>	276
21	prom2	29, 27	11706	11436	11706	11347	10673	11412	11706	11347	10247	11412	11706	11347	10247	7029
22	root	8, 5	367	<b>351</b>	367	<b>351</b>	362	363	367	<b>351</b>	362	363	367	<b>351</b>	<b>351</b>	390
23	shift	19, 16	575	585	575	585	520	508	575	585	520	508	575	585	508	507
24	signet	39, 8	744	640	744	644	661	667	744	644	661	667	744	644	640	632
25	sqn	7, 3	322	316	305	316	315	330	305	316	319	330	322	312	305	297
26	sym10	10, 1	556	567	556	567	<b>550</b>	621	556	567	<b>550</b>	621	556	567	<b>550</b>	567
27	tial	14, 8	4785	4910	4785	4696	4333	4851	4785	4696	4448	4356	4755	4834	4333	4094
28	ts10	22, 16	913	913	913	913	791	780	913	913	791	780	913	913	791	751
29	Z5xp1	7, 10	296	296	296	296	296	289	296	296	296	289	296	296	289	289
30	Z9sym	9, 1	732	612	709	611	641	589	709	611	651	589	709	611	589	463

В столбце *best of inp* (табл. 1) даны лучшие решения для алгоритма «по входам», в столбце *best of full* (табл. 2) – для алгоритма «полный».

В строках табл. 1 даны оценки сложности схем БМК, синтезированных в системе Leonardo. Для каждого из 30 примеров было получено 12 эквивалентных вариантов функциональных описаний с помощью различных алгоритмов разбиения. В столбцах табл. 1 представлены наименования алгоритмов, с помощью которых получены функциональные описания. По наименованию алгоритма можно определить, используется ли предварительная редукция уравнений связи (*e*) и выполняется ли процедура устранения иерархии с редукцией уравнений связи (*\_red*) или без нее (*no\_red*). В качестве параметров разбиения выбраны числа (*n\_m*): *8\_8*, *8\_4*, *8\_2*. Полужирным шрифтом в каждой строке выделяются минимальные значения оценки. Для некоторых схем минимальные оценки достигаются сразу несколькими алгоритмами, как, например, для схемы *misg* или *dk48*. Итак, анализируя столбец *best of inp* табл. 1, видим улучшение результатов синтеза в 10 случаях из 30 (они выделены полужирным шрифтом).

Анализируя табл. 2, видим, что при использовании алгоритма «полный» улучшение результатов синтеза схем наблюдается в 22 случаях из 30 (столбец *best of full*). В общем, алгоритмы разбиения выигрывают в 22 случаях из 30, что видно при выборе лучших результатов из столбцов *best of inp* (см. табл. 1) и *best of full* (табл. 2). В большинстве случаев улучшение достигается как алгоритмом «по входам», так и алгоритмом «полный», они конкурируют друг с другом. Результаты сравнения лучших решений по эксперименту 1 и алгоритмов разбиения «по входам» и «полный» представлены в табл. 3. В строке *absolutely best* даны абсолютно лучшие результаты, значительная часть которых получена по эксперименту 2.

Таблица 2

Результаты эксперимента 2 для алгоритма «полный»: сложность  $S_{Ai \in}$  схем БМК

N	Имя схемы	<i>n, m</i>	<i>full_8_8_red</i>	<i>full_8_8_no_red</i>	<i>e_full_8_8_red</i>	<i>e_full_8_8_no_red</i>	<i>full_8_4_red</i>	<i>full_8_4_no_red</i>	<i>e_full_8_4_red</i>	<i>e_full_8_4_no_red</i>	<i>full_8_2_red</i>	<i>full_8_2_no_red</i>	<i>e_full_8_2_red</i>	<i>e_full_8_2_no_red</i>	<i>best of full</i>	<i>best of iterations</i>
1	add6	12, 7	187	186	173	191	201	165	187	178	182	<b>159</b>	181	173	<b>159</b>	175
2	addm4	9, 8	1086	1100	1145	1185	1160	1130	<b>1052</b>	1075	1113	1126	1141	1152	<b>1052</b>	1136
3	b12	15, 9	202	202	201	201	191	191	201	201	190	190	190	190	190	188
4	b2	15, 17	<b>2008</b>	2062	2078	2085	2088	2142	2075	2060	2096	2012	2099	2071	<b>2008</b>	2048
5	b9	16, 5	308	306	320	320	317	310	305	320	306	293	313	312	293	297
6	bc0	21, 11	2449	2519	2370	2472	2602	2696	2494	2614	2356	<b>2349</b>	2519	2511	<b>2349</b>	2547
7	chkn	29, 7	1028	1018	1066	1019	<b>999</b>	1012	1025	1037	1049	1004	1009	1064	<b>999</b>	1010
8	dc2	8, 7	322	362	<b>298</b>	343	341	318	<b>298</b>	344	345	310	325	<b>298</b>	<b>298</b>	359
9	dk48	15, 17	<b>398</b>	<b>398</b>	416	415	404	404	405	<b>398</b>	406	400	<b>398</b>	<b>398</b>	398	398
10	ibm	48, 17	532	532	<b>520</b>	532	534	532	534	522	534	535	535	534	520	514
11	in0	15, 11	1432	1376	1389	1367	1439	1405	1465	1428	1359	<b>1353</b>	1469	1447	<b>1353</b>	1363
12	in2	19, 10	1085	1085	1110	1106	<b>1084</b>	1092	1119	1106	<b>1084</b>	1107	1096	1106	<b>1084</b>	1135
13	in6	33, 23	747	<b>736</b>	824	766	767	767	782	761	739	745	825	789	<b>736</b>	738
14	intb	15, 7	4722	4614	4467	4769	5163	4617	5042	4865	5009	4716	4819	4607	4467	3594
15	m2	8, 16	<b>655</b>	669	673	674	676	686	682	683	676	661	693	679	<b>655</b>	682
16	m3	8, 16	897	915	<b>879</b>	931	943	948	931	923	895	897	919	949	<b>879</b>	899
17	misg	56, 23	186	186	183	192	<b>171</b>	183	186	186	198	196	178	178	<b>171</b>	181
18	newcpla1	9, 16	240	237	255	241	243	255	<b>208</b>	248	247	229	230	237	<b>208</b>	229
19	newtpla	15, 5	155	142	145	157	139	166	140	139	157	157	143	158	138	126
20	p82	5, 14	272	272	272	272	272	<b>270</b>	271	272	286	285	271	272	<b>270</b>	276
21	prom2	29, 27	9594	9295	9366	9179	9090	9070	9245	9218	9532	9707	9451	9335	9070	7029
22	root	8, 5	365	365	377	385	388	391	379	379	402	<b>360</b>	374	385	<b>360</b>	390
23	shift	19, 16	508	508	594	575	510	510	555	580	543	<b>490</b>	524	561	<b>490</b>	507
24	signet	39, 8	634	640	622	633	664	629	627	698	703	638	635	<b>618</b>	<b>618</b>	632
25	sqn	7, 3	<b>281</b>	322	327	326	298	325	327	311	307	284	298	329	<b>281</b>	297
26	sym10	10, 1	560	560	583	544	549	537	549	544	595	596	579	<b>529</b>	<b>529</b>	567
27	tial	14, 8	4390	4652	4305	4108	5051	4366	4371	4100	4795	4816	4333	4158	4100	4094
28	ts10	22, 16	785	740	913	802	701	913	913	802	<b>676</b>	701	913	913	<b>676</b>	751
29	Z5xp1	7, 10	296	296	<b>285</b>	<b>285</b>	289	289	289	289	<b>285</b>	<b>285</b>	289	289	<b>285</b>	289
30	Z9sym	9, 1	495	520	514	510	478	539	703	531	500	583	503	507	478	463



Таблица 3  
Сравнение результатов эксперимента 2: сложность  $S_{il \dot{e}}$  схем БМК

Номер схемы	Имя схемы	<i>best of iterations</i>	<i>best of inp</i>	<i>best of full</i>	<i>absolutely best</i>
1	add6	175	<b>159</b>	<b>159</b>	159
2	addm4	1136	<b>1125</b>	<b>1052</b>	1052
3	b12	188	189	189	188
4	b2	2048	2048	<b>2008</b>	<b>2008</b>
5	b9	297	<b>291</b>	<b>293</b>	<b>291</b>
6	bc0	2547	2556	<b>2349</b>	<b>2349</b>
7	chkn	1010	1028	<b>999</b>	<b>999</b>
8	dc2	359	<b>298</b>	<b>298</b>	<b>298</b>
9	dk48	398	398	398	398
10	ibm	514	533	520	514
11	in0	1363	1370	<b>1353</b>	<b>1353</b>
12	in2	1135	<b>1098</b>	<b>1084</b>	<b>1084</b>
13	in6	738	773	<b>736</b>	<b>736</b>
14	intb	3594	4897	4467	3594
15	m2	682	<b>653</b>	<b>655</b>	<b>653</b>
16	m3	899	934	<b>879</b>	<b>879</b>
17	misg	181	<b>171</b>	<b>171</b>	<b>171</b>
18	newcpla1	229	230	<b>208</b>	<b>208</b>
19	newtpla	126	138	138	126
20	p82	276	<b>269</b>	<b>270</b>	<b>269</b>
21	prom2	7029	10247	9070	7029
22	root	390	<b>351</b>	<b>360</b>	<b>351</b>
23	shift	507	508	<b>490</b>	<b>490</b>
24	signet	632	640	<b>618</b>	<b>618</b>
25	sqn	297	305	<b>281</b>	<b>281</b>
26	sym10	567	550	<b>529</b>	<b>529</b>
27	tial	4094	4333	4100	4094
28	ts10	751	791	<b>676</b>	<b>676</b>
29	Z5xp1	289	289	<b>285</b>	<b>285</b>
30	Z9sym	463	589	478	463

В табл. 4 даны результаты исследования влияния алгоритмов разбиения и процедур редукции уравнений связи на количество достигаемых минимальных значений  $S_{БМК}$ .

Таблица 4  
Результаты анализа алгоритмов разбиения и процедур редукции уравнений связи

Группы методов	Методы	Общее количество минимальных значений	Количество выигрышей у методов эксперимента 1
Распределение минимальных значений $S_{БМК}$ по типу алгоритма разбиения	«По входам»	10	7
	«Полный»	24	18
Распределение минимальных значений $S_{БМК}$ по параметрам разбиения	Параметры $\delta\_8$	13	9
	Параметры $\delta\_4$	15	8
	Параметры $\delta\_2$	17	14
Распределение минимальных значений $S_{БМК}$ в зависимости от выполнения операции предварительной редукции уравнений связи ( $e$ )	С редукцией уравнений связи ( $e$ )	15	10
	Без редукции уравнений связи	20	15
Распределение минимальных значений $S_{БМК}$ в зависимости от выполнения операции редукции уравнений связи при устранении иерархии описания	С редукцией уравнений связи ( $red$ )	18	12
	Без редукции уравнений связи ( $no\_red$ )	16	13

Распределение минимумов подсчитывается внутри групп методов, представленных в строках столбца 1 табл. 4. Методы (или приемы), которые включены в каждую группу, представлены в столбце 2; в столбце 3 приведено число минимальных значений  $S_{БМК}$ , получаемых каждым из методов группы; в столбце 4 представлено количество выигрышей у итерационных методов, испытанных в эксперименте 1.

Данные табл. 4 помогают оценить эффективность применения различных алгоритмов разбиения, а также приемов дополнительной обработки описаний для оптимизации проектируемых схем. Так, например, табл. 4 показывает, что применение алгоритма «полный» значительно выигрывает у алгоритма «по входам» как по общему числу минимальных значений, так и по количеству выигрышей у метода итераций.

На рис. 3 представлены значения выигрышей для каждого из 30 исследованных примеров. По оси  $x$  представлен номер примера, по оси  $y$  – наибольший процент сокращения площади схемы  $S_{БМК}$ , полученный в результате применения одного из алгоритмов разбиения логических схем.

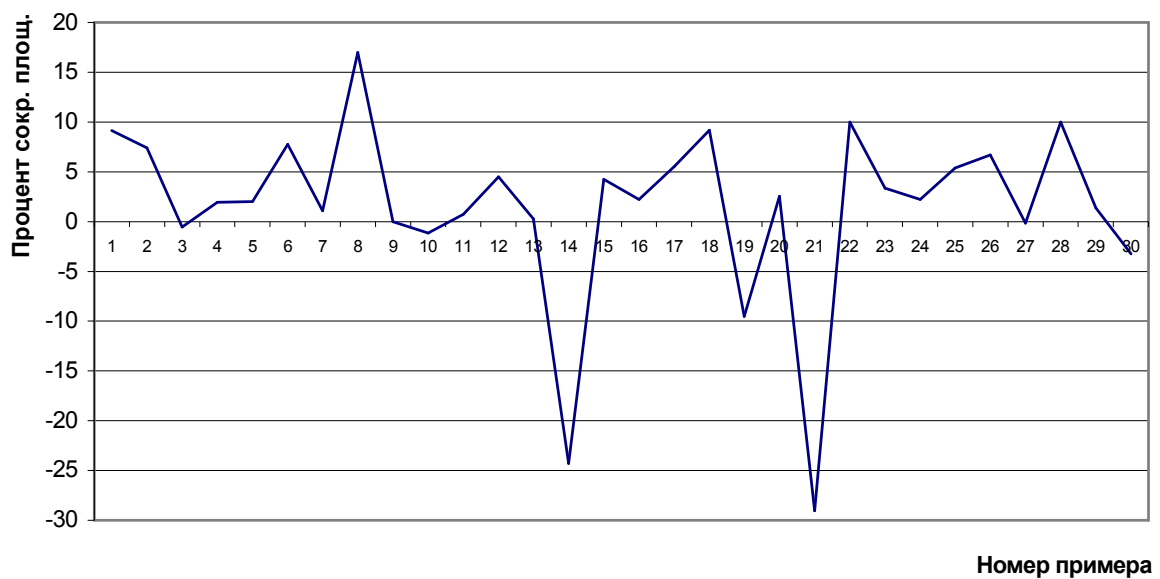


Рис. 3. Оценка величины выигрыша по площади синтезируемых схем при использовании алгоритмов разбиения

Анализируя результаты эксперимента 2, можно сделать вывод, что наиболее вероятно получить лучшую схему БМК, если использовать алгоритм «полный» с параметрами разбиения  $\delta_2$  без предварительной и без заключительной редукции уравнений связи.

### Заключение

В статье описан метод разбиения многоуровневых описаний логических схем на блоки и даны результаты широкого экспериментального исследования программы, реализующей предложенный метод. Показано, что использование процедуры блочного разбиения схемы во многих случаях является эффективным приемом минимизации сложности схем БМК, синтезируемых в системе Leonardo.

### Список литературы

1. Kuter, J. Architecture Driven Partitioning / J. Kuter, E. Barke // Proc. of the conf. on Design, automation and test in Europe. – Munich, Germany, 2001. – P. 479–487.
2. Khatri, S.P. Cross-talk immune VLSI design using a network of PLAs embedded in a regular layout fabric / S.P. Khatri, R.K. Brayton, A. Sangiovanni-Vincentelli // Proc. of the 2000 IEEE/ACM Int. conf. on Computer-aided design. – San Jose, California, 2000. – P. 412–418.

3. Binary and Multi-Valued SPFD-Based Wire Removal in PLA Networks / S. Sinha [et al.] // Proc. of the IEEE Int. conf. on Computer Design: VLSI In Computers & Processors (ICCD '00). – Austin, Texas, 2000. – P. 494–503.
4. Yan, K. Practical logic synthesis for CPLDs and FPGAs with PLA-style logic blocks / K. Yan // Proc. of ASP-DAC 2001, Asia and South Pacific Design Automation Conf. – Yokohama, Japan, 2001. – P. 231–234.
5. Cheung C.-C. Further Improve Circuit Partitioning using GBAW Logic Perturbation Techniques / C.-C. Cheung, Y.-L. Wu, D.I. Cheng // Proc. of the conf. on Design, automation and test in Europe. – Munich, Germany, 2001. – P. 233–239.
6. Фридман А. Теория и проектирование переключательных схем / А. Фридман, П. Мено. – М. : Мир, 1978. – 580 с.
7. Кириенко, Н.А. Разбиение системы логических уравнений на подсистемы с заданными ограничениями / Н.А. Кириенко // Новые информационные технологии в исследовании дискретных структур : докл. Третьей Всерос. конф. с международным участием. – Томск : ТИЦ СО РАН, 2000. – С. 236–241.
8. Бибило, П.Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum / П.Н. Бибило. – М. : СОЛОН-Пресс, 2005. – 384 с.
9. Система «Custom Logic» автоматизированного проектирования управляющей логики заказных цифровых СБИС / П.Н. Бибило [и др.] // Микроэлектроника. – 2004. – Т. 32, № 5. – С. 310–327.
10. Berkeley PLA test set [Electronic resource]. – Mode of access : <http://www1.cs.columbia.edu/~cs4861/sis/espresso-examples/ex/>. – Date of access : 03.05.2006.
11. Бибило, П.Н. Синтез логических схем с использованием языка VHDL / П.Н. Бибило. – М. : СОЛОН-Р, 2002. – 384 с.
12. Бибило, П.Н. Оптимизация логических схем заказных СБИС в синтезаторе Leonardo / П.Н. Бибило // Современная электроника. – 2007. – № 5. – С. 50–52.

Поступила 22.04.09

*Объединенный институт проблем  
информатики НАН Беларуси,  
Минск, Сурганова, 6  
e-mail: bibilo@newman.bas-net.by*

**P.N. Bibilo, N.A. Kirienko**

### **OPTIMIZING CONVERSIONS OF A LOGIC CIRCUIT BY PARTITIONING INTO BLOCKS**

The problem of optimizing conversions of a logic circuit by partitioning into blocks is considered. The results of synthesis for different descriptions of a search logic circuit in CAD systems Leonardo and Custom Logic are described. Synthesis is performed in the library gate basis. The results of experimental investigations are presented. Conclusions regarding the efficiency of the partitioning procedure reducing the complexity of synthesized circuit are drawn.