УДК 681.325

Л.Д. Черемисинова

МНОГОКРАТНАЯ СВЕРТКА РЕГУЛЯРНЫХ СТРУКТУР НА ОСНОВЕ РЕШЕНИЯ ЛОГИЧЕСКИХ УРАВНЕНИЙ

Рассматривается задача минимизации площади регулярных матричных структур заказных СБИС методом многократной свертки. Предлагается метод решения ключевой проблемы многократной свертки – проверки реализуемости множества свертки, который основывается на сведении задачи к решению логического уравнения и проверке выполнимости конъюнктивной нормальной формы.

Введение

Проектирование блоков управляющей логики заказных СБИС ориентировано, как правило, на использование схем с регулярной структурой. Их применение позволяет уменьшить стоимость разработки схемы в силу упрощения задачи генерации топологии по структурному (или функциональному) описанию реализуемого устройства. Регулярная матричная структура состоит из взаимно пересекающихся строк и столбцов, на пересечении которых находятся транзисторы (или схемы из транзисторов). Примерами двухмерных регулярных матричных структур являются структуры типа программируемых логических матриц, матриц Вайнбергера, транзисторных матриц, регулярных схем на базе последовательно соединенных МОП-транзисторов [1–3]. За основной критерий оптимальности при проектировании СБИС на основе регулярных структур принимается площадь кристалла, оцениваемая на логическом уровне произведением чисел столбцов и строк.

Существенным недостатком матричных структур является то, что они, имея регулярную организацию, проигрывают многоуровневым реализациям на основе произвольной логики по площади, занимаемой на кристалле, за счет неэффективного ее использования. Последнее выражается в сильной разреженности матричных структур. Одним из наиболее эффективных методов топологической оптимизации является свертка столбцов и строк матричной структуры, сокращающая число неиспользуемых транзисторов. История развития методов свертки берет начало в работах [4, 5], обзор основных подходов к свертке регулярных структур можно найти в [6, 7].

Свертка основана на разрыве шин матричной структуры и реализации на одной вертикальной (и (или) горизонтальной) шине двух (при простой свертке) или более (при многократной свертке) столбцов и (или) строк. При простой свертке вертикальная (и (или) горизонтальная) шина свернутой структуры разделяется на два сегмента, каждый из которых реализует один столбец (строку) исходной матричной структуры. При многократной свертке число свертываемых столбцов и (или) строк (число сегментов) не ограничивается. Очевидно, что преобразование матричной структуры с использованием свертки не изменяет функциональность схемы. В теоретическом плане задача свертки сводится к комбинаторной задаче поиска оптимального переупорядочивания и совмещения столбцов и (или) строк матричных структур и является, как доказано в [4], NP-трудной.

Центральной частью любой регулярной структуры (ее ядром) является двухмерная матрица. Ее столбцы представляют собой линии подвода входных сигналов, а строки реализуют конъюнкции входных переменных, соответствующих тем столбцам, в которых на пересечении со строками находятся транзисторы. В настоящей работе рассматривается задача сокращения площади такой двухмерной матрицы путем ее многократной свертки. Предлагается решение ключевой задачи многократной свертки – проверки множества свертки на реализуемость.

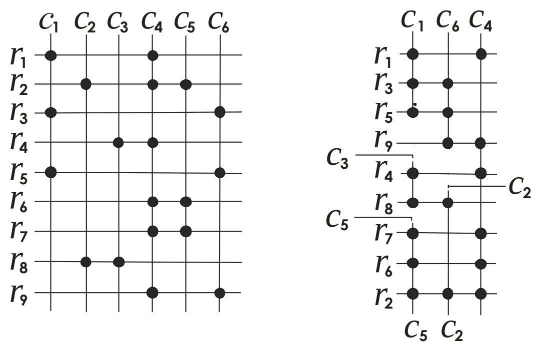
В работе [8] к задаче булевой выполнимости впервые была сведена проблема трассировки соединений. Позднее аналогичным образом к задаче выполнимости была сведена проблема простой столбцовой свертки программируемой логической матрицы [9]. В этих работах решаемые задачи сводились к проверке выполнимости булевой функции, вид которой определялся данными задачами. Для представления анализируемой булевой функции и анализа ее на выполнимость использовался аппарат BDD (Binary Decision Diagrams – диаграммы двоичных решений) [10].

В работе [11] к проверке выполнимости сведена, так же как и в [9], задача анализа множества простой свертки на реализуемость. В отличие от упомянутой работы решение этой задачи формулируется как поиск выполняющего набора для специальным образом формируемой конъюнктивной нормальной формы (КНФ), задающей условия, предъявляемые к реализуемому множеству свертки. Такое представление позволяет использовать для поиска выполняющего набора SAT-решатели (SAT-solver) [12–14], которые обеспечивают проверку выполнимости достаточно сложных КНФ, содержащих до тысячи переменных. Кроме того, задача простой свертки в работе [11] рассматривалась в более общей постановке – для случая неупорядоченных свертываемых наборов.

В настоящей работе рассмотрен более общий вид свертки – многократная свертка регулярных матричных структур заказных СБИС. Показано, как свести анализ множества свертки на реализуемость к задаче поиска решения логического уравнения и проверке выполнимости КНФ.

1. Формализация задачи многократной свертки

Во всех постановках задачи свертки математической моделью регулярной структуры является булева матрица $\boldsymbol{B}$, имеющая множества $C(\boldsymbol{B}) = \{c_1, c_2, \ldots\}$ столбцов и $R(\boldsymbol{B}) = \{r_1, r_2, \ldots\}$ строк. Каждый столбец $c_j \in C(\boldsymbol{B})$ (строка $r_j \in R(\boldsymbol{B})$) порождает множество $R(c_j)$ строк ($C(r_j)$ столбцов), имеющих единицы на пересечении с этим столбцом, т. е. $r_i \in R(c_j)$, если $b^j_i \in \boldsymbol{B}$ имеет значение 1. Площадь матричной структуры определяется как $S = |C(\boldsymbol{B})|\,|R(\boldsymbol{B})|$. Далее будем говорить о столбцовой свертке, имея в виду, что все сказанное может быть отнесено и к строчной свертке.



а) *б)*

Схематичное представление матричной структуры: *а)* исходная структура; *б)* ее многократная свертка

На рисунке точками обозначены транзисторы на пересечениях некоторых строк и столбцов. Формальной моделью этой матричной структуры является следующая булева матрица:

$$\boldsymbol{B} = \begin{array}{cccccc c} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & \\ 1 & 0 & 0 & 1 & 0 & 0 & r_1 \\ 0 & 1 & 0 & 1 & 1 & 0 & r_2 \\ 1 & 0 & 0 & 0 & 0 & 1 & r_3 \\ 0 & 0 & 1 & 1 & 0 & 0 & r_4 \\ 1 & 0 & 0 & 0 & 0 & 1 & r_5 \\ 0 & 0 & 0 & 1 & 1 & 0 & r_6 \\ 0 & 0 & 0 & 1 & 1 & 0 & r_7 \\ 0 & 1 & 1 & 0 & 0 & 0 & r_8 \\ 0 & 0 & 0 & 1 & 0 & 1 & r_9 \end{array} \,. \qquad (1)$$

Непересекающиеся столбцы c_k и c_l ($R(c_k) \cap R(c_l) = \varnothing$) не имеют активных транзисторов на пересечении с одними и теми же строками и являются строчно совместимыми. Совместимые столбцы c_k и c_l образуют свертываемую пару (c_k, c_l), т. е. могут быть свернуты и реализованы на одной вертикальной шине регулярной структуры. Аналогично набор $l_k = (c_{k1}, c_{k2},\ldots, c_{km})$, состоящий из попарно совместимых столбцов c_{ij}, называется свертываемым (если не наложены никакие дополнительные ограничения на тип свертки). Только такие наборы (или пары) столбцов могут быть свернуты. Неупорядоченный свертываемый набор $l_k = (c_{k1}, c_{k2},\ldots, c_{km})$ порождает m! различных упорядоченных наборов $l_k^o = <c_{k1}, c_{k2},\ldots, c_{km}>$. Под упорядочением набора l_k понимается выбор одного из этих упорядоченных наборов l_k^o. Любой упорядоченный свертываемый набор столбцов l_k^o может быть реализован на одной вертикальной шине свернутой двухмерной структуры, на которой c_{k1} помещается над c_{k2}, c_{k2} – над c_{k3} и т. д.: $c_{k(m-1)}$ – над c_{km}. Итак, набор l_k^o порождает такое переупорядочение строк матрицы, что $R(c_{k1}) > R(c_{k2})$ – строки из $R(c_{k1})$ располагаются выше всех строк из $R(c_{k2})$; $R(c_{k2}) > R(c_{k2})$ – строки из $R(c_{k2})$ выше всех строк из $R(c_{k3})$ и т. д.: $R(c_{k(m-1)}) > R(c_{km})$, порождая отношение на множестве $R(\boldsymbol{B})$:

$$T(l_k^o) = \bigcup_{i,j} (R(c_{ki}) \times R(c_{kj})), \quad \text{т. е.} \quad T(l_k^o) = \{ r_p \times r_q / r_p \in R(c_{ki}), r_q \in R(c_{kj}), i < j\}.$$

Это отношение является отношением частичного порядка на множестве $R(\boldsymbol{B})$, так как оно по определению иррефлексивно, асимметрично и транзитивно.

Неупорядоченный свертываемый набор $l_k = (c_{k1}, c_{k2},\ldots, c_{km})$ порождает m! возможных отношений частичного порядка на множестве $R(\boldsymbol{B})$. Для матричной структуры (рисунок, *а*), описываемой булевой матрицей $\boldsymbol{B}$ (1), существует, например, набор, состоящий из трех попарно совместимых столбцов, $l_1 = (c_1, c_3, c_5)$, который порождает шесть упорядоченных свертываемых наборов

$$l_1^o = <c_1, c_3, c_5>,\ l_2^o = <c_1, c_5, c_3>,\ l_3^o = <c_3, c_1, c_5>,\ l_4^o = <c_3, c_5, c_1>,\ l_5^o = <c_5, c_1, c_3>,\ l_6^o = <c_5, c_3, c_1>$$

и, соответственно, шесть частичных упорядочиваний на множестве строк. Первый упорядоченный свертываемый набор $l_1^o = <c_1, c_3, c_5>$ порождает, например, следующий частичный порядок на множестве строк:

$$\{r_1, r_3, r_5\} > \{r_4, r_8\},\ \{r_4, r_8\} > \{r_2, r_6, r_7\}.$$

Два свертываемых набора l_p и l_q совместимы, если они не пересекаются, т. е. если все c_{pi}, c_{qj} (для всех i, j) различны. Множество $L = \{l_1, l_2,\ldots, l_n\}$, состоящее из попарно совместимых свертываемых наборов столбцов, называется неупорядоченным множеством столбцовой свертки (НМС). Аналогично определяются упорядоченные множества свертки (УМС): $L_p^o = \{ l^o_{p1}, l^o_{p2},\ldots, l^o_{pn} \}$. Число столбцов, входящих во все свертываемые наборы множества свертки L (или L_p^o), называется ее размером. Задача свертки состоит в поиске наибольшего реализуемого множества свертки, которое порождает свернутую матричную структуру наименьшей площади.

УМС L_p^o порождает отношение порядка на множестве строк $R(\boldsymbol{B})$, которое представляет собой объединение отношений $T(l_k^o)$, порождаемых всеми свертываемыми наборами $l_i^o \in L_p^o$:

$$T(L_p^o) = \bigcup_{i=1}^{n} (T(l_{pi}^o)).$$

Отношение $T(L_p^o)$ в общем случае не есть отношение частичного порядка, так как оно иррефлексивно, асимметрично, но не обязательно транзитивно, но транзитивное замыкание $T^r(L_p^o)$ отношения $T(L_p^o)$ иррефлексивно, транзитивно, но не обязательно асимметрично.

В работе [5] показано, что упорядоченное множество L_p^o свертки реализуемо, если порождаемое им транзитивное замыкание $T^r(L_p^o)$ есть отношение частичного порядка на множестве $R(\boldsymbol{B})$. Это означает, что $T^r(L_p^o)$ асимметрично. Обобщив это определение реализуемости на неупорядоченное множество свертки, будем называть НМС L реализуемым, если существует реализуемое УМС L_p^o, получаемое из L путем упорядочения входящих в него неупорядоченных наборов l_{ki}.

Для рассматриваемого примера матричной структуры существуют четыре неупорядоченных максимальных свертываемых набора:

$$l_1 = (c_1, c_3, c_5),\ l_2 = (c_1, c_2),\ l_3 = (c_2, c_6),\ l_4 = (c_3, c_6).$$

Два набора совместимы и образуют множество свертки максимального размера: $L = \{(c_1, c_3, c_5), (c_2, c_6)\}$. НМС L порождают 12 УМС L_p^o, так как для l_1 возможно шесть упорядочиваний (как показано выше), а для l_2 – два: $<c_2, c_6>$ и $<c_6, c_2>$. Первое УМС $L_1^o = \{<c_1, c_3, c_5>, <c_2, c_6>\}$ порождает следующее отношение на $R(\boldsymbol{B})$:

$$T(L_1^o) = R(c_1) \times R(c_3) \cup R(c_3) \times R(c_5) \cup R(c_2) \times R(c_6) =$$

$$= \{r_1, r_3, r_5\} \times \{r_4, r_8\} \cup \{r_4, r_8\} \times \{r_2, r_6, r_7\} \cup \{r_2, r_8\} \times \{r_3, r_5, r_9\}.$$

Его транзитивное замыкание содержит конфликтные пары (r_3, r_2) и (r_2, r_3), откуда следует, что отношение $T^r(L_1^o)$ не является отношением частичного порядка на $R(\boldsymbol{B})$ и УМС L_1^o нереализуемо. Второе УМС $L_2^o = \{<c_1, c_3, c_5>, <c_6, c_2>\}$ порождает отношение

$$T(L_2^o) = R(c_1) \times R(c_3) \cup R(c_3) \times R(c_5) \cup R(c_6) \times R(c_2) =$$

$$= \{r_1, r_3, r_5\} \times \{r_4, r_8\} \cup \{r_4, r_8\} \times \{r_2, r_6, r_7\} \cup \{r_3, r_5, r_9\} \times \{r_2, r_8\}.$$

Его транзитивное замыкание $T^r(L_2^o)$ обладает свойством асимметричности. Следовательно, УМС L_2^o (а значит, и НМС L) является реализуемым множеством свертки.

Реализуемое множество свертки $L_p^o = \{l_{p1}, l_{p2},\ldots, l_{pn}\}$ определяет вид свернутой матричной структуры, а его размер – размер свертки: число наборов свертки соответствует числу вертикальных шин свернутой матрицы, которые заменят $\sum_{i=1}^{n} |l^o_{pi}|$ столбцов исходной матричной структуры. Например, множество свертки $L_2^o = \{<c_1, c_3, c_5>, <c_6, c_2>\}$ состоит из двух свертываемых наборов и имеет размер пять. Следовательно, пять столбцов исходной матричной структуры будут заменены двумя столбцами при ее свертке.

Задача свертки матричной структуры формулируется следующим образом: дана булева матрица, ее представляющая, требуется найти реализуемое множество свертки максимального размера. Существуют методы построения множеств многократной свертки. Например, в работах [5, 15] находятся упорядоченные множества свертки, в [16] – неупорядоченные (их существенно меньше). Задача свертки в [17] сведена к поиску максимальных клик графа отношения совместимости столбцов матрицы $\boldsymbol{B}$.

В настоящей работе рассматривается задача проверки на реализуемость и упорядочивание неупорядоченного множества свертки путем сведения этих задач к поиску решения логического уравнения с использованием известных SAT-решателей. При решении задачи поиска реализуемого множества свертки максимального размера предлагаемый метод используется для последовательной проверки реализуемости множеств свертки, упорядоченных в порядке убывания их размеров.

2. Проверка реализуемости упорядоченного множества свертки путем сведения к поиску корня логического уравнения

Задача проверки реализуемости множества свертки формулируется следующим образом: дано упорядоченное множество столбцовой свертки $L^o = \{l_1^o, l_2^o,\ldots, l_n^o\}$; требуется определить, реализуемо ли оно, и если да, то предъявить частичное упорядочение на множестве строк $R(\boldsymbol{B})$. Каждый свертываемый набор $l_k^o = <c_{k1}, c_{k2},\ldots, c_{km}> \in L^o$ порождает следующее частичное упорядочение строк из $R(\boldsymbol{B})$:

$$(R(c_{k1}) > R(c_{k2})) \cap (R(c_{k2}) > R(c_{k3})) \ \cap \ldots\ \cap (R(c_{k(m-1)}) > R(c_{km})). \qquad (2)$$

Условие (2) говорит о том, что строки из $R(c_{k1})$ должны располагаться в свернутой структуре выше строк из $R(c_{k2})$, а они – выше строк из $R(c_{k3})$ и т. д.

Задача проверки реализуемости УМС $L^o = \{l_1^o, l_2^o,\ldots, l_n^o\}$ должна установить, существует ли такое упорядочение строк, для которого выполняется условие (2). Покажем, как эту задачу можно свести к решению сначала логического уравнения, а затем к решению задачи выполнимости КНФ.

Сформируем множество R_L строк, каждая из которых входит по крайней мере в одно из множеств $R(c_{ij})$ ($c_{ij} \in l_i^o$ из L^o), $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, i_m\}$. Расположение строк из $R(\boldsymbol{B}) \setminus R_L$ в свертываемой согласно L^o регулярной матричной структуре никак не влияет на свертываемость матричной структуры относительно столбцов из наборов множества L^o. Следовательно, допустимо рассматривать далее только строки из R_L. Более того, нет нужды рассматривать и те строки из R_L, каждая из которых входит только в одно из множеств $R(c_{ij})$, так как порядок их следования фиксируется относительно строк только одного множества $R(c_{ij})$ и, следовательно, исключаются конфликты с размещением строк из других множеств $R(c_{sq})$ ($s \neq i$). Таким путем можно сократить число переменных и термов формируемого логического уравнения.

Для задания отношений следования строки из R_L кодируются позиционным кодом $x_1x_2 \ldots x_q$, где $q = \lceil \log_2 |R_L| \rceil$ ($\lceil t \rceil$ – наименьшее сверху целое, не меньшее, чем t). Численное значение кода $x^i_1x^i_2\ldots x^i_q$ строки r_i задает ее порядковый номер, указывающий физическое расположение горизонтальной линии свернутой матричной структуры, выделяемой для строки r_i. Тогда условия, порождаемые упорядочениями строк $r_i > r_j$, представляются функциями

$$f_{ij} = (x^i_qx^i_{q-1}\ldots x^i_1) > (x^j_qx^j_{q-1}\ldots x^j_1). \tag{3}$$

С учетом (3) условие (2), порождаемое набором свертки $l_k^o = \langle c_{k1}, c_{k2},\ldots, c_{km}\rangle$ и накладываемое на порядок следования строк, представляется в виде

$$\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{kq})} \Big(\bigcap_{r_j \in R(c_{k(q+1)})} f_{ij} \Big)\Big) = 1. \tag{4}$$

УМС $L^o = \{l_1^o, l_2^o,\ldots, l_n^o\}$ является реализуемым, если и только если следующее уравнение имеет хотя бы одно решение:

$$\bigcap_{k=1}^{n} \Big(\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{kq})} \Big(\bigcap_{r_j \in R(c_{k(q+1)})} f_{ij} \Big)\Big)\Big) = 1. \tag{5}$$

Если уравнение (5) имеет корень (набор значений переменных x^i_l, выполняющих его), то соответствующее УМС L^o реализуемо. Этот корень определяет порядковые номера строк из R_L. В противном случае, если корня уравнения (5) не существует, УМС L^o нереализуемо.

Для примера рассмотрим УМС $L_2^o = \{\langle c_1, c_3, c_5\rangle, \langle c_6,c_2\rangle\}$ для приведенной выше матричной структуры. Так как $R(c_1) = \{r_1, r_3, r_5\}$, $R(c_3) = \{r_4, r_8\}$, $R(c_5) = \{r_2, r_6, r_7\}$, $R(c_2) = \{r_2, r_8\}$, $R(c_6) = \{r_3, r_5, r_9\}$, то множество строк, подлежащих упорядочиванию согласно УМС L_2^o, $R_L =\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}$, можно сократить до $R_L =\{r_2, r_3, r_5, r_8\}$. Соответственно сокращается и число рассматриваемых строк в множествах $R(c_i)$: $R'(c_1) = \{r_3, r_5\}$, $R'(c_3) = \{r_8\}$, $R'(c_5) = \{r_2\}$, $R'(c_2) = \{r_2, r_8\}$, $R'(c_6) = \{r_3, r_5\}$. Четыре строки из R_L можно закодировать двумя булевыми переменными x_2 и x_1, их значения в кодах строк r_i будем обозначать через x_2^i и x_1^i. После кодирования строк уравнение (5) принимает вид

$$\begin{gathered}(f_{3,8} \wedge f_{5,8} \wedge f_{8,2}) \wedge (f_{3,2} \wedge f_{3,8} \wedge f_{5,2} \wedge f_{5,8}) = f_{3,8} \wedge f_{5,8} \wedge f_{8,2} \wedge f_{3,2} \wedge f_{5,2} = \\ = (x^3_2x^3_1 > x^8_2x^8_1) \wedge (x^5_2x^5_1 > x^8_2x^8_1) \wedge (x^8_2x^8_1 > x^2_2x^2_1) \wedge (x^3_2x^3_1 > x^2_2x^2_1) \wedge (x^5_2x^5_1 > x^2_2x^2_1).\end{gathered} \tag{6}$$

3. Приведение уравнения для проверки реализуемости упорядоченного множества свертки к виду КНФ

Для решения уравнения (5) можно привлечь аппарат решения логических уравнений, но для этого необходимо представить функции $f_{ij} = ((x^i_px^i_{p-1}\ldots x^i_1) > (x^j_px^j_{p-1}\ldots x^j_1))$ в виде формул булевой алгебры. Для решения полученного уравнения можно привлечь известный аппарат проверки выполнимости КНФ, используемый в настоящее время в области автоматизации логического проектирования для решения задач большой размерности (например, Chaff [12], BerkMin [13]). Для этого необходимо представить левую часть уравнения (5), и в частности функции f_{ij}, в виде КНФ. КНФ представляет собой конъюнкцию дизъюнктов, каждый из которых задается дизъюнкцией литералов (булевых переменных и их инверсий).

Для приведения левой части уравнения (5) к виду КНФ необходимо преобразовать к виду КНФ функцию $f_{ij} = (x_q^i x_{q-1}^i \ldots x_1^i) > (x_q^j x_{q-1}^j \ldots x_1^j)$ (3). Найдем представление этой функции в виде минимальной КНФ $C(f_{ij})$. Для случая $q = 1$ (имеются две строки и для их кодирования используется только одна переменная) получается функция $f_{ij} = (x_1^i) > (x_1^j)$ от двух аргументов, ее минимальная КНФ имеет два дизъюнкта:

$$C^1(f_{ij}) = x_1^i\ \bar{x}_1^j.$$

Для случая $q = 2$ (имеются не более чем четыре строки и для их кодирования используются две переменные) получается функция $f_{ij} = (x_2^i x_1^i) > (x_2^j x_1^j)$ от четырех аргументов, ее минимальная КНФ имеет пять дизъюнктов:

$$C^2(f_{ij}) = (x_2^i \vee x_1^i) \wedge (\bar{x}_2^j \vee x_1^i) \wedge (x_2^i \vee \bar{x}_1^j) \wedge (\bar{x}_2^j \vee \bar{x}_1^j) \wedge (x_2^i \vee \bar{x}_2^j).$$

В общем случае КНФ $C^t(f_{ij})$ (для $q = t$) имеет $2t$ аргументов: на два аргумента (x_t^i и x_t^j) больше, чем КНФ $C^{t-1}(f_{ij})$. Если КНФ $C^{t-1}(f_{ij})$ имеет s^{t-1} дизъюнктов, то КНФ $C^t(f_{ij}) - s^t = 2s^{t-1}+1$ дизъюнктов. В силу регулярности функции f_{ij} относительно числа аргументов нетрудно показать, что КНФ $C^t(f_{ij})$ может быть получена из КНФ $C^{t-1}(f_{ij})$ путем преобразования каждого ее r-го дизъюнкта d_r^{t-1} в два дизъюнкта КНФ $C^t(f_{ij})$ – $(x_{t+1}^i \vee d_r^t)$ и $(\bar{x}_{t+1}^j \vee d_r^t)$ и добавления в полученную КНФ одного нового дизъюнкта $(x_{t+1}^i \vee \bar{x}_{t+1}^j)$, зависящего только от вновь введенных аргументов.

После представления функции f_{ij} в виде КНФ $C(f_{ij})$ уравнение (5) преобразуется в логическое уравнение, левая часть которого представляет собой КНФ:

$$\bigcap_{k=1}^{n} \Big(\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{kq})} \Big(\bigcap_{r_j \in R(c_{k(q+1)})} C(f_{ij}) \Big)\Big)\Big) = 1. \qquad (7)$$

Таким образом, задача проверки реализуемости УМС $L^o = \{l_1^o, l_2^o, \ldots, l_n^o\}$ сведена к задаче поиска корня уравнения (7) путем проверки выполнимости КНФ его левой части с предъявлением выполняющего набора значений переменных (если решение существует).

Для пояснения сути изложенного метода продолжим процесс проверки реализуемости УМС $L_2^o = \{<c_1, c_3, c_5>, <c_6, c_2>\}$. После преобразования к виду КНФ левой части уравнения (6) проверка реализуемости УМС L_2^o сводится к проверке выполнимости следующей КНФ:

$$\begin{aligned}
&(x^3{}_2x^3{}_1 > x^8{}_2x^8{}_1) \wedge (x^5{}_2x^5{}_1 > x^8{}_2x^8{}_1) \wedge (x^8{}_2x^8{}_1 > x^2{}_2x^2{}_1) \wedge (x^3{}_2x^3{}_1 > x^2{}_2x^2{}_1) \wedge (x^5{}_2x^5{}_1 > x^2{}_2x^2{}_1) = \\
&= ((x^3{}_2 \vee x^3{}_1) \wedge (\bar{x}^8{}_2 \vee x^3{}_1) \wedge (x^3{}_2 \vee \bar{x}^8{}_1) \wedge (\bar{x}^8{}_2 \vee \bar{x}^8{}_1) \wedge (x^3{}_2 \vee \bar{x}^8{}_2)) \wedge \\
&\wedge ((x^5{}_2 \vee x^5{}_1) \wedge (\bar{x}^8{}_2 \vee x^5{}_1) \wedge (x^5{}_2 \vee \bar{x}^8{}_1) \wedge (\bar{x}^8{}_2 \vee \bar{x}^8{}_1) \wedge (x^5{}_2 \vee \bar{x}^8{}_2)) \wedge \\
&\wedge ((x^8{}_2 \vee x^8{}_1) \wedge (\bar{x}^2{}_2 \vee x^8{}_1) \wedge (x^8{}_2 \vee \bar{x}^2{}_1) \wedge (\bar{x}^2{}_2 \vee \bar{x}^2{}_1) \wedge (x^8{}_2 \vee \bar{x}^2{}_2)) \wedge \\
&\wedge ((x^3{}_2 \vee x^3{}_1) \wedge (\bar{x}^2{}_2 \vee x^3{}_1) \wedge (x^3{}_2 \vee \bar{x}^2{}_1) \wedge (\bar{x}^2{}_2 \vee \bar{x}^2{}_1) \wedge (x^3{}_2 \vee \bar{x}^2{}_2)) \wedge \\
&\wedge ((x^5{}_2 \vee x^5{}_1) \wedge (\bar{x}^2{}_2 \vee x^5{}_1) \wedge (x^5{}_2 \vee \bar{x}^2{}_1) \wedge (\bar{x}^2{}_2 \vee \bar{x}^2{}_1) \wedge (x^5{}_2 \vee \bar{x}^2{}_2)).
\end{aligned} \qquad (8)$$

Набором значений переменных, выполняющим КНФ (8), является, например, $x^3{}_2 = x^3{}_1 = x^5{}_2 = x^8{}_1 = 1$; $x^5{}_1 = x^8{}_2 = x^2{}_2 = x^2{}_1 = 0$. Таким образом, строки из $R_L = \{r_3, r_5, r_8, r_2\}$ получают следующие коды: r_3 – 11, r_5 – 10, r_8 – 01, r_2 – 00. Это решение задает частичное упорядочение r_3, r_5, r_8, r_2 строк из R_P и, например, следующее полное упорядочение r_1, r_3, r_5, r_9 r_4, r_8, r_7, r_6, r_2, которое легло в основу варианта свертки (рисунок, *б*) регулярной матричной структуры (рисунок, *а*).

4. Проверка реализуемости неупорядоченного множества свертки

Для случая неупорядоченного множества свертки $L = \{l_1, l_2, \ldots, l_n\}$, где $l_k = (c_{k1}, c_{k2}, \ldots, c_{km})$, рассматриваемая проблема формулируется следующим образом: проверить, реализуемо ли НМС L, и, если реализуемо, найти соответствующее реализуемое УМС и частичное упорядочение строк из $R(\boldsymbol{B})$. Этот случай отличается от рассмотренного выше только тем, что для проверки реализуемости необходимо просмотреть все возможные упорядочения наборов свертки $l_i \in L$, т. е. необходимо сгенерировать все возможные перестановки столбцов для каждого набора $l_i \in L$. Это требование следует из того факта, что неупорядоченный набор $l_k = (c_{k1}, c_{k2}, \ldots, c_{km})$ порождает множество $P(l_k)$ из $m!$ возможных упорядоченных наборов $l_{ki}^o = <c_{i1}^k, c_{i2}^k, \ldots, c_{im}^k>$ ($m = 1, 2, \ldots, m!$), соответствующих различным перестановкам столбцов из l_k. Таким образом, неупорядоченный набор l_k порождает следующие возможные отношения частичного порядка на $R(\boldsymbol{B})$, представляющие собой обобщение представления (2):

$$\bigcup_{l_{ks}^o \in P(l_k)} ((R(c_{s1}^k) > R(c_{s2}^k)) \cap (R(c_{s2}^k) > R(c_{s3}^k)) \cap \ \ldots \cap (R(c_{sm-1}^k) > R(c_{sm}^k))). \qquad (9)$$

После кодирования строк из соответствующего множества R_L, формируемого аналогично тому, как это делалось для упорядоченных наборов свертки l_k^o, (4) принимает следующую форму, которая более сложна, чем соответствующее выражение (4) для l_k^o:

$$\bigcup_{l_{ks}^o \in P(l_k)} \Big(\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{sq}^k)} \Big(\bigcap_{r_j \in R(c_{s(q+1)}^k)} f_{ij} \Big)\Big)\Big) = 1. \qquad (10)$$

Исходя из (10), по аналогии с (5) НМС $L = \{l_1, l_2, \ldots, l_n\}$ реализуемо, если и только если следующее уравнение имеет хотя бы одно решение:

$$\bigcap_{k=1}^{n} \Big(\bigcup_{l_{ks}^o \in P(l_k)} \Big(\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{sq}^k)} \Big(\bigcap_{r_j \in R(c_{s(q+1)}^k)} f_{ij} \Big)\Big)\Big)\Big) = 1. \qquad (11)$$

Если уравнение (11) имеет корень (набор значений переменных x^i_l, выполняющих уравнение), соответствующее НМС L реализуемо. Корень определяет порядковые номера строк из R_L и, соответственно, упорядочение наборов свертки $l_k = (c_{k1}, c_{k2}, \ldots, c_{km}) \in L$. Если уравнение (11) не имеет корня, НМС L нереализуемо.

Для рассмотренного ранее неупорядоченного множества свертки $L = \{(c_1, c_3, c_5), (c_2, c_6)\}$ аналогично упорядоченному множеству свертки L^o получаются множества $R_L = \{r_2, r_3, r_5, r_8\}$, $R'(c_1) = \{r_3, r_5\}$, $R'(c_3) = \{r_8\}$, $R'(c_5) = \{r_2\}$, $R'(c_2) = \{r_2, r_8\}$, $R'(c_6) = \{r_3, r_5\}$. НМС L порождает 12 возможных УМС L_k^o. После кодирования четырех строк из R_L булевыми переменными x_2 и x_1 уравнение (11) принимает вид

$$\begin{gathered}((f_{3,8} \wedge f_{5,8} \wedge f_{8,2}) \vee (f_{3,2} \wedge f_{5,2} \wedge f_{2,8}) \vee (f_{8,3} \wedge f_{8,5} \wedge f_{3,2} \wedge f_{5,2}) \vee (f_{8,2} \wedge f_{2,3} \wedge f_{2,5}) \vee \\ \vee (f_{2,3} \wedge f_{2,5} \wedge f_{3,8} \wedge f_{5,8}) \vee (f_{2,8} \wedge f_{8,3} \wedge f_{8,5})) \wedge ((f_{2,3} \wedge f_{2,5} \wedge f_{8,3} \wedge f_{8,5}) \vee (f_{3,2} \wedge f_{3,8} \wedge f_{5,2} \wedge f_{5,8})).\end{gathered} \qquad (12)$$

5. Приведение уравнения для проверки реализуемости неупорядоченного множества свертки к виду КНФ

Для случая неупорядоченного множества свертки после представления в (11) функций f_{ij} в виде КНФ $C(f_{ij})$ получается следующее булево уравнение, левая часть которого в общем случае не есть КНФ:

$$\bigcap_{k=1}^{n} \Big(\bigcup_{l_{ks}^o \in P(l_k)} C_{ks} \Big) = 1. \qquad (13)$$

Здесь C_{ks} представляет собой следующую КНФ:

$$C_{ks} = \bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c^k_{sq})} \Big(\bigcap_{r_j \in R(c^k_{s(q+1)})} C(f_{ij}) \Big)\Big). \tag{14}$$

Булева функция $\bigcup\limits_{l^o_{ks} \in P(l_k)} C_{ks}$ в левой части уравнения (13) представляется дизъюнкцией нескольких КНФ. Для того чтобы преобразовать ее к виду КНФ, не производя сложных вычислений, введем, как это предложено в [17], $k_m!$ булевых переменных z_{ks} с целью закодировать унитарным кодом $m!$ КНФ C_{ks}. Нетрудно показать, что условие выполнимости уравнения (13) не изменится, если формулу $\bigcup\limits_{l^o_{ks} \in P(l_k)} C_{ks}$ заменить на

$$\Big(\bigcap_{l^o_{ks} \in P(l_k)} (C_{ks} \vee z_{ks}) \Big) \wedge \Big(\bigcup_{s=1}^{m!} \bar{z}_{ks} \Big). \tag{15}$$

Здесь $(C_{ks} \vee z_{ks})$ представляется в форме КНФ, которая достаточно просто получается из КНФ C_{ks} применением закона (булевой алгебры) дистрибутивности конъюнкции относительно дизъюнкции (кодирующая переменная z_{ks} добавляется в каждый дизъюнкт $d_i \in C_{ks}$: $d_i \vee z_{ks}$). Для кодирования всех наборов свертки $l_k \in L$ понадобится ввести $\sum\limits_{k=1}^{n} (k_m!)$ кодирующих переменных z_{ks}.

Таким образом, с учетом ранее введенных формул (7), (13)–(15) конструируется уравнение, левая часть которого представлена в форме КНФ:

$$\bigcap_{k=1}^{n} \Big(\bigcap_{l^o_{ks} \in P(l_k)} (C_{ks} \vee z_{ks}) \Big) \wedge \Big(\bigcup_{s=1}^{m!} \bar{z}_{ks} \Big) = 1.$$

Это уравнение решает ключевую задачу свертки регулярных матричных структур: проверяет реализуемость неупорядоченного множества свертки.

Для пояснения процедуры преобразования уравнений к виду КНФ продолжим приведение формулы (12) к виду КНФ для неупорядоченного множества свертки $L = \{(c_1, c_3, c_5), (c_2, c_6)\}$, анализируемого на реализуемость:

$$\begin{gathered}
((x^3{}_2 \vee x^3{}_1 \vee z^1{}_1) \wedge (\bar{x}^8{}_2 \vee x^3{}_1 \vee z^1{}_1) \wedge (x^3{}_2 \vee \bar{x}^8{}_1 \vee z^1{}_1) \wedge (\bar{x}^8{}_2 \vee \bar{x}^8{}_1 \vee z^1{}_1) \wedge (x^3{}_2 \vee \bar{x}^8{}_2 \vee z^1{}_1)) \wedge \\
\wedge ((x^5{}_2 \vee x^5{}_1 \vee z^1{}_1) \wedge (\bar{x}^8{}_2 \vee x^5{}_1 \vee z^1{}_1) \wedge (x^5{}_2 \vee \bar{x}^8{}_1 \vee z^1{}_1) \wedge (\bar{x}^8{}_2 \vee \bar{x}^8{}_1 \vee z^1{}_1) \wedge (x^5{}_2 \vee \bar{x}^8{}_2 \vee z^1{}_1)) \wedge \\
\wedge ((x^8{}_2 \vee x^8{}_1 \vee z^1{}_1) \wedge (\bar{x}^2{}_2 \vee x^8{}_1 \vee z^1{}_1) \wedge (x^8{}_2 \vee \bar{x}^2{}_1 \vee z^1{}_1) \wedge (\bar{x}^2{}_2 \vee \bar{x}^2{}_1 \vee z^1{}_1) \wedge (x^8{}_2 \vee \bar{x}^2{}_2 \vee z^1{}_1)) \wedge \ldots \\
\wedge ((x^2{}_2 \vee x^2{}_1 \vee z^6{}_1) \wedge (\bar{x}^8{}_2 \vee x^2{}_1 \vee z^6{}_1) \wedge (x^2{}_2 \vee \bar{x}^8{}_1 \vee z^6{}_1) \wedge (\bar{x}^8{}_2 \vee \bar{x}^8{}_1 \vee z^6{}_1) \wedge (x^2{}_2 \vee \bar{x}^8{}_2 \vee z^6{}_1)) \wedge \\
\wedge ((x^8{}_2 \vee x^8{}_1 \vee z^6{}_1) \wedge (\bar{x}^3{}_2 \vee x^8{}_1 \vee z^6{}_1) \wedge (x^8{}_2 \vee \bar{x}^3{}_1 \vee z^6{}_1) \wedge (\bar{x}^3{}_2 \vee \bar{x}^3{}_1 \vee z^6{}_1) \wedge (x^8{}_2 \vee \bar{x}^3{}_2 \vee z^6{}_1)) \wedge \\
\wedge ((x^8{}_2 \vee x^8{}_1 \vee z^6{}_1) \wedge (\bar{x}^5{}_2 \vee x^8{}_1 \vee z^6{}_1) \wedge (x^8{}_2 \vee \bar{x}^5{}_1 \vee z^6{}_1) \wedge (\bar{x}^5{}_2 \vee \bar{x}^5{}_1 \vee z^6{}_1) \wedge (x^8{}_2 \vee \bar{x}^5{}_2 \vee z^6{}_1)) \wedge \\
\wedge (\bar{z}^1{}_1 \vee \bar{z}^2{}_1 \vee \bar{z}^3{}_1 \vee \bar{z}^4{}_1 \vee \bar{z}^5{}_1 \vee \bar{z}^6{}_1) \wedge ((x^2{}_2 \vee x^2{}_1 \vee z^1{}_2) \wedge (\bar{x}^3{}_2 \vee x^2{}_1 \vee z^1{}_2) \wedge \\
\wedge (x^2{}_2 \vee \bar{x}^3{}_1 \vee z^1{}_2) \wedge (\bar{x}^3{}_2 \vee \bar{x}^3{}_1 \vee z^1{}_2) \wedge (x^2{}_2 \vee \bar{x}^3{}_2 \vee z^1{}_2)) \wedge \ldots \wedge (\bar{z}^1{}_2 \vee \bar{z}^2{}_2) = 1.
\end{gathered}$$

Одним из наборов значений переменных, выполняющих эту КНФ, является набор, найденный ранее для КНФ (8). Он допускает следующее кодирование строк из $R_L = \{r_3, r_5, r_8, r_2\}$: r_3 – 11, r_5 – 10, r_8 – 01, r_2 – 00, что согласуется с вариантом свертки рассматриваемой матричной структуры, приведенным на рисунке, *б*.

Заключение

Разработан метод, решающий ключевую задачу многократной свертки – анализ множества свертки на реализуемость. Показано, как свести задачу анализа множества свертки на реализуемость к решению логического уравнения, а поиск корней уравнения – к известной задаче проверки выполнимости КНФ, для решения которой разработаны эффективные методы и программы. В отличие от известных методов проверки на реализуемость, формулируемых относительно упорядоченных множеств свертки, предлагаемый метод ориентирован на неупорядоченные множества, что позволяет снизить размерность задач построения множеств свертки. Описываемый метод попутно решает задачу нахождения такого упорядочения наборов множества свертки (если оно существует), которое обеспечивает реализуемость этого множества.

Список литературы

1. Ульман, Дж. Вычислительные аспекты СБИС / Дж. Ульман. – М. : Радио и связь, 1990. – 480 с.

2. Бибило, П.Н. Кремниевая компиляция заказных СБИС / П.Н. Бибило. – Минск : Ин-т техн. кибернетики АН Беларуси, 1996. – 268 с.

3. Biswas, N.N. Logic design theory / N.N. Biswas. – Prentice-Hall International, 1993. – 306 p.

4. Hachtel, G.D. An Algorithm for optimal PLA Folding / G.D. Hachtel, A.R. Newton, A.L. Sangiovanni-Vincentelli // IEEE Trans. Computer-Aided Design of Integrated Circuit Syst. – 1982. – Vol. CAD-1, no. 2. – P. 63–77.

5. DeMicheli, G.A. Multiple Constrained Folding of Programmable Logic Arrays: Theory and Applications / G.A. DeMicheli, A.L. Sangiovanni-Vincentelli // IEEE Trans. Computer-Aided Design. – 1983. – Vol. CAD-2, no. 3. – P. 151–167.

6. Черемисинова, Л.Д. Минимизация площади регулярных матричных структур заказных СБИС / Л.Д. Черемисинова // Информатика. – 2004. – № 1. – С. 121–131.

7. Минимизация площади заказных СБИС на этапе топологического проектирования цифровых схем / Л.Д. Черемисинова [и др.] // Управляющие системы и машины. – 2011. – № 4 (240). – С. 42–50.

8. Devadas, S. Optimal Layout via Boolean Satisfiability / S. Devadas // Proc. of Intern. Conf. on Computer-Aided Design (ICCAD ’89). – Santa Clara, CA, USA, 1989. – P. 294–297.

9. Optimum PLA Folding through Boolean Satisfiability / J.M. Quintana [et al.] // Asian South Pacific Design Automation Conference (ASP DAC’95). – Chiba, Japan, 1995. – P. 289–293.

10. Bryant, R.E. Graph-based algorithms for Boolean function manipulation / R.E. Bryant // IEEE Trans. Computers. – 1986. – Vol. C-35, no. 8. – P. 677–691.

11. Черемисинова, Л.Д. Свертка регулярных структур на основе решения логических уравнений / Л.Д. Черемисинова // Танаевские чтения : доклады Четвертой Междунар. науч. конф. (29–30 марта 2010, Минск). – Минск : ОИПИ НАН Беларуси, 2010. – С. 129–134.

12. Mahajan, Y. Zchaff2004: An Efficient SAT Solver / Y. Mahajan, Z. Fu , S. Malik // Theory and Applications of Satisfiability Testing (2004 SAT Solver Competition and QBF Solver Evaluation (Invited Papers)). – Berlin, Heidelberg : Springer, 2005. – P. 360–375.

13. Goldberg, E. BerkMin: A Fast and Robust SAT-Solver / E. Goldberg , Y. Novikov // Design, Automation, and Test in Europe. – Paris, 2002. – P. 142–149.

14. Eén, N., MiniSat / N. Eén, N. Sörensson [Electronic resource]. – Mode of access : http://www.cs.chalmers.se/Cs/Research/FormalMethods/ MiniSat. – Date of access : 09.02.2015.

15. Lecky, I.E. Graph theoretic flgorithms for the PLA folding problem / I.E. Lecky, O.I. Murphy, R.G. Abshe // IEEE Trans. Computer-Aided Design. – 1989. – Vol. 8, no. 9. – P. 1014–1021.

16. Cheremisinova, L.D. Some results in optimal PLA folding / L.D. Cheremisinova // Proc. of the Third Intern. Conf. on Computer-Aided Design of Discrete Devices (CAD DD’99). – Minsk : UIIP NAS B, 1999. – Vol. 1. – P. 59–64.

17. Cheremisinova, L. SAT-Based Approach to Verification of Logical Descriptions with Functional Indeterminacy / L. Cheremisinova, D. Novikov // 8th Intern. Workchop on Boolean problems. – Freiberg (Sachsen), 2008. – P. 59–66.

Поступила 09.02.15

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

L.D. Cheremisinova

MULTIPLE FOLDING OF REGULAR STRUCTURES VIA SOLVING LOGIC EQUATIONS

The problem under consideration is to reduce the area of the layout of regular VLSI structures by means of their multiple folding. The method of solving the key problem of multiple folding, which is implementability checking of the folding set, is suggested. The method is based on the task reduction to solving a logic equation and checking Boolean satisfiability of a conjunctive normal form.