ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ
№ ФИЗИКА 2015

УДК 519.7

С.А. ОСТАНИН, И.Е. КИРИЕНКО**

МИНИМИЗАЦИЯ ЧАСТИЧНО ОПРЕДЕЛЕННЫХ БУЛЕВЫХ ФУНКЦИЙ, ПРЕДСТАВЛЕННЫХ BDD-ГРАФАМИ[1]

В работе предлагается модификация алгоритма построения минимального BDD-графа, являющегося реализацией частично определенной булевой функции. Алгоритм основан на использовании графа совместимости с последующим выделением клик графа с использованием дополнительных условий. Модификация алгоритма позволяет сократить количество операций при вычислении клик графа без потери точности решения.

Ключевые слова:*Binary Decision Diagram (BDD), частично определенные булевы функции, минимизация булевых функций.*

Полностью определенная булева функция f является реализацией частично определенной функции g если на всех наборах α, на которых функция g имеет определенные значения, $g(\alpha) = f(\alpha)$. В данной статье предлагается точный алгоритм нахождения функции f для g, BDD (Binary Decision Diagram) представление которой имеет минимальное число узлов. Хорошо известно, что для заданного порядка переменных BDD представление для функции f является единственным с точностью до изоморфизма [1].

Существует множество практических задача, где нужно найти минимальную, в смысле количества узлов BDD, реализацию частично определенной булевой функции, например, логический синтез в базисе FPGA[2], синтез маскирующих подсхем для повышения надежности частично программируемых схем [3], поиск представлений временных Шенноновских схем с низким потреблением энергии [4].

В общем случае задача поиска одной из реализаций частично определенной булевой функции, представленной BDD с минимальным числом узлов, является NP-полной [5], в связи с этим разработано множество эвристических алгоритмов, решающих эту задачу за приемлемое время [5, 6]. Для некоторых приложений точность решения данной задачи эвристическими алгоритмами не достаточна; нужно использовать алгоритмы, дающие точное решение. В работе [7] предложено построение минимального BDD-графа, представляющего реализацию частично определенной булевой функции при условии, что частично определенная функция задана двумя BDD-графами BDD(f_{off})и BDD(f_{on})для области нулей и единиц, соответственно. В данной работе предлагается модификация алгоритма построения минимального BDD-графа, предложенного в работе [7]. Алгоритм основан на использовании графа совместимости с последующим выделением клик графа с использованием дополнительных условий. Модификация алгоритма позволяет сократить количество операций при вычислении клик графа без потери точности решения.

Рассмотрим базовые определения.

BDD-граф – это направленный ациклический граф, каждый узел которого помечен переменной x_i, и где каждый нетерминальный узел n_i имеет исходящие 0- и 1-дуги, соединяющие узел n_i с дочерними узлами n_i^z и n_i^o, соответственно. Терминальные узлы обозначим n_z и n_o. Далее 0-дуга будет соответствовать левой исходящей дуге из узла n_i, 1-дуга – правой.

BDD-граф соответствует полностью определенной булевой функции $f(x_1,\ldots,x_n)$. Множество f_{on} содержит наборы, на которых функция $f(x_1,\ldots,x_n)$ принимает значение 1, множество f_{off} содержит наборы, на которых функция $f(x_1,\ldots,x_n)$ принимает значение 0.

BDD-граф называется сокращенным (ROBDD), если из любых двух узлов не существует одного и того же пути в терминальный узел. Так как ROBDD-граф с заданным порядком переменных является каноническим представлением полностью определенной булевой функции f, будем обозначать n_i как узел ROBDD-графа и как переменную булевой функции f.

[1]Работа выполнена при частичной поддержке грантом Российского научного фонда № 14-19-00218

Уровнем $L(n_i)$ узла n_i назовем индекс переменной в порядке разложения. Пусть корень BDD-графа имеет уровень 0. Если N – количество переменных, то N – уровень терминальных узлов BDD-графа. Максимальный уровень $L_{\max}(n_i)$ множества s узлов – это максимальный уровень среди узлов множества s. Уровнем $L(h)$ функции h называется уровень переменной BDD-графа, реализующего функцию h. Если n_i – узел BDD-графа, m – набор значений переменных, то $n_i(m)$ – терминальный узел, который достигается из вершины n_i при m (или значение функции n_i на наборе m). Поэтому, если m – набор значений переменных, F – BDD-граф для функции f, то $n_0(m)$ – значение функции f на наборе m.

BDD-граф называется полным, если каждое ребро BDD-графа соединяет узел уровня i с узлом уровня i+1.

Трехтерминальный BDD-граф (3TBDD) определяется аналогично BDD-графу с тем исключением, что 3TBDD имеет 3 терминальные вершины n_z, n_o и n_x. 3TBDD-граф соответствует частично определенной булевой функции f, которая имеет множества наборов f_{on}, f_{off} и f_{dc}, на которых функция f принимает значения n_o, n_z и n_x, соответственно.

В данной работе будем использовать 3TBDDF, который соответствует частично определенной булевой функции f. Будем полагать, что граф F полный (при необходимости можно ввести дополнительные узлы, 0- и 1-дуги которых ведут в тот же узел).

Введем определения совместимых узлов и узлов, совместимых в обобщенном смысле.

Определение 1. Два узла n_i и n_j графа F будем называть совместимыми ($n_i \sim n_j$), если и только если не существует такого набора m, для которого справедливы $n_i(m) = n_z \vee n_j(m) = n_o$ или $n_i(m) = n_o \vee n_j(m) = n_z$.

По определению терминальные узлы n_z и n_o не совместимы между собой, терминальный узел n_x совместим с любой вершиной 3TBDD.

Определение 2. Два узла n_i и n_j графа F будем называть совместимыми в обобщенном смысле ($n_i \approx n_j$), если и только если существует полностью определенная функция h такая, что $h \sim n_i$ и $h \sim n_j$ и $L(h) \geq \max(L(n_i), L(n_j))$.

По определению терминальные узлы n_z и n_o между собой не совместимы в обобщенном смысле, терминальный узел n_x совместим в обобщенном смысле с любой вершиной 3TBDD.

Важно отметить, что узел, соответствующий полностью определенной функции h, необязательно является некоторым узлом 3TBDD. В большинстве случаев это действительно так, поскольку многие узлы 3TBDD соответствуют частично определенным функциям.

Следующая лемма устанавливает отношение между совместимостью и совместимостью в обобщенном смысле.

Лемма 1. Если $n_i \approx n_j$, то $n_i \sim n_j$.

Лемма 2 формулирует условие, при котором верно и обратное утверждение.

Лемма 2. Если $L(n_i) = L(n_j)$, то $n_i \sim n_j$, следовательно $n_i \approx n_j$.

Совместимость в обобщенном смысле можно определить для множества узлов.

Определение 3. Узлы множества $s_i = (n_1, n_2, ..., n_s)$ совместимы в обобщенном смысле, если и только если существует полностью определенная булева функция h такая, что $h \sim n_i$, $\forall i \in \{1,...,s\}$, и $L(h) \geq L_{\max}(s_i)$.

Определение 4. Множество узлов, совместимых в обобщенном смысле, будем называть совместимым множеством или совместимостью.

Множество узлов является совместимостью, если любые два узла этого множества попарно совместимы в обобщенном смысле. В общем случае обратное утверждение неверно. Однако справедлива следующая лемма.

Лемма 3. Пусть s_i - это множество узлов, принадлежащих одному уровню в графе. Тогда s_i является совместимым множеством, если и только если все узлы в s_i попарно совместимы в обобщенном смысле.

Введем определение графа совместимости.

Определение 5. Граф совместимости $G=(V,E)$ - это не ориентированный граф, который показывает, какие узлы в 3TBDD-графе *F* могут быть совмещены. Кроме терминального узла n_x, каждому узлу графа *F* соответствует один узел в графе *G* с тем же значением уровня. Узлам n_i^z и n_i^o графа *F* соответствуют узлы g_i^z и g_i^o графа *G*.

В графе *G* два узла g_i и g_j соединены ребром, если узлы n_i и n_j графа *F* совместимы в обобщенном смысле. Каждое ребро в графе *G* имеет метки. Метка – это множество узлов. Существует три типа меток: типа *e*, *t* и *l*.

Следующие две леммы необходимы для построения графа совместимости.

Лемма 4. Если $L(n_i)=L(n_j)$ и $n_i \approx n_j$, то $n_i^o \approx n_j^o \vee n_i^z \approx n_j^z$.

Лемма 5. Если $L(n_i)<L(n_j)$ и $n_i \approx n_j$, то $n_i^o \approx n_j \vee n_i^z \approx n_j \vee n_i^o \approx n_i^z$.

Алгоритм построения графа совместимости.

1. Пусть граф *G* - полный. Удаляем одно ребро (g_z, g_o).

2. Если $L(g_i)=L(g_j)$, то согласно лемме 4, ребро (g_i, g_j) имеет две метки: $e:(g_i^0, g_j^0)$ и $t:(g_i^1, g_j^1)$.

3. Если $L(g_i)<L(g_j)$, то согласно лемме 5, ребро (g_i, g_j) имеет метку $l:(g_i^0, g_i^1, g_j)$.

4. Для всех пар узлов (g_i, g_j) проверяем: если ребро между узлами g_i и g_j имеет метку, которая содержит пару (g_a, g_b), при этом между вершинами g_a и g_b нет ребра, то удаляем ребро (g_i, g_j). Повторяем шаг 4 до тех пор, пока граф *G* не перестанет изменяться.

Следующая лемма устанавливает связь между существованием ребра в графе совместимости и понятиями совместимости и совместимости в обобщенном смысле.

Лемма 6. Если $n_i \approx n_j$, то в графе совместимости существует такое ребро *e*, что $e=(g_i, g_j)$. Следовательно, $n_i \sim n_j$.

Важно отметить, что в общем случае обратное утверждение неверно, то есть существование ребра между двумя узлами графа *G* не означает, что эти узлы совместимы в обобщенном смысле.

Рассмотрим 3TBDD на рис. 1. Для данного 3TBDD граф совместимости построен с помощью приведенного выше алгоритма.

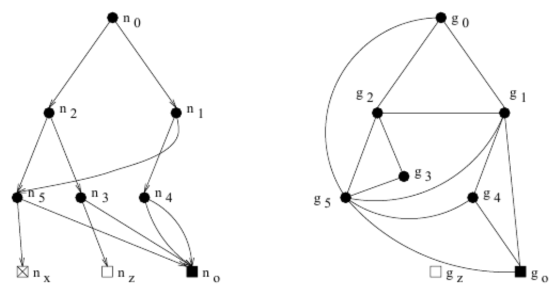

Рис. 1. 3TBDD и соответствующий граф совместимости

Кликой графа *G* называется его подграф, являющийся полным графом. Любому множеству *s* узлов, образующему клику графа *G*, соответствует класс множеств. Пусть $s_i=(g_{i_1},...,g_{i_w})$ – это множество узлов, которые образуют клику графа *G*. Для данного множества s_i определим *e*, *t* и *l* классы.

Определение 6. *e*-класс множества s_i (обозначим как $C_e(s_i)$) – это множество узлов из *e*-метки ребра между узлами g_j и g_k из s_i таких, что $L(n_k) = L(n_j) = L_{\max}(s_i)$.

Определение 7. *t*-класс множества s_i (обозначим как $C_t(s_i)$) – это множество узлов из *t*-метки ребра между узлами g_j и g_k из s_i таких, что $L(n_k) = L(n_j) = L_{\max}(s_i)$.

Определение 8. *l*-класс множества s_i (обозначим как $C_l(s_i)$) – это множество узлов из *l*-метки ребра между узлами g_j и g_k из s_i таких, что $L(g_j) \neq L(g_k)$.

Лемма 7. Если множество узлов s_i образует клику графа *G* и $C_l(s_i) \subseteq s_i$, то s_i является совместимым множеством.

Важно отметить, что некоторая клика графа *G*, не удовлетворяющая условию леммы 7, необязательно будет совместимым множеством.

Алгоритм поиска минимальной BDD, совместимой с исходной частично определенной функцией, основан на выборе таких узлов графа *G*, которые могут быть заменены одним узлом в финальной BDD. Если множество *s* узлов графа *G* могут быть объединены в один узел, то *s* должно быть совместимым множеством. Следовательно, множество *s* должно быть кликой графа *G*, удовлетворяя условию определения 7. Таким образом, необходимо найти такое множество клик, что каждый узел графа *G* будет покрыт по крайней мере одной кликой. Данный поиск требует дополнительных ограничений.

Определение 9. Множество $S = \{s_1, \ldots, s_n\}$ множеств узлов графа *G* называется замкнутым покрытием кликами для графа *G*, если справедливы следующие условия:

1. *s*является покрытием*G*: $\forall g_i \in G \exists s_j \in S : g_i \in s_j$.

2. Все s_k являются кликами графа *G*: $\forall g_i, g_j \in s_k : (g_i, g_j) \in edges(G)$.

3. *s* замкнуто относительно меток типа *e* и *t*: $\forall s_i \in S \exists s_j \in S : C_e(s_i) \subseteq s_j \wedge \forall s_i \in S \exists s_j \in S : C_t(s_i) \subseteq s_j$.

4. Все множества из *s* замкнуты относительно метки типа *l*: $\forall s_i \in S : C_l(s_i) \subseteq s_i$.

Ниже приведена предлагаемая модификация алгоритма построения минимального BDD-графа *R*.

1. Строим замкнутое покрытие кликами $S = \{s_1, \ldots, s_n\}$ для графа *G*. На каждой итерации проверяем замкнутость относительно класса C_l, если замкнутость отсутствует, то переходим к другому подмножеству.
2. Для каждого $s_i \in S$ создаем узел r_i нового графа *R*, причем уровень этого узла $L(r_i) = L_{\max}(s_i)$.

3. Узел r_i, соответствующий множеству $s_i \in S$, будет терминальным в графе *R*, если множество S_i содержит узел, который соответствует терминальному узлу в графе *F*.

4. 0-ребро узла r_i соединяет его с узлом r_j, соответствующим множеству $s_j \in S$, если $C_e(s_i) \subseteq s_j$.

5. 1-ребро узла r_i соединяет его с узлом r_j, соответствующим множеству $s_j \in S$, если $C_t(s_i) \subseteq s_j$.

Лемма 8. Построенный граф *R* является OBDD-графом, совместимым с исходным графом *F*.

Пусть $\mathfrak{R}$ – это множество всех BDD-графов, представляющих функции, каждая из которых совместима с исходной частично определенной булевой функцией *f*. Тогда справедлива следующая лемма.

Лемма 9.BDD-граф, построенный на основе минимального замкнутого покрытия для *G*, является BDD-графом из $\mathfrak{R}$ с минимальным числом узлов.

Проведенный нами анализ показал, что модифицированный алгоритм требует меньше времени для построения минимального BDD-графа по сравнению с базовым алгоритмом.

СПИСОК ЛИТЕРАТУРЫ

1. Bryant R. Graph Based Algorithm for Boolean Function Manipulation // IEEE Trans. Computers. – 1986. – V. 35. – No. 8. – P. 667–691.
2. Matrosova A., Loukovnikova E., Ostanin S., Zinchuk A., Nikolaeva E. Test Generation for Single and Multiple Stuck-at Faults of a Combinational Circuit Designed by Covering Shared ROBDD with CLBs // Proc.IEEE Intl.Symp. on Defect and Fault-Tolerance in VLSI Systems (DFT 2007). – 2007. – P. 206–214.
3. Matrosova A.,Ostanin S., Kirienko I. Increasing Manufacturing Yield Using Partially Programmable Circuits with CLB implementation of Incompletely Specified Boolean Function of the Corresponding Sub-circuit // Proc. IEEE Intl. Symp. on Design and Diagnostics of Electronic Circuits and Systems (DDECS 2015). – 2015. – P. 267–270.
4. Lavagno L., McGeer P., Saldanha A., and Sangiovanni-Vincentelli A. Timed Shannon Circuits: A Power-Efficient Design Style and Synthesis Tool// Proc. 32nd Design Automation Conf. – 1995. – P. 254–260.
5. Shiple T., Hojati R., Sangiovanni-Vincentelli A., and Bryaton R. Heuristic Minimization of BDDs Using Don't Cares// Proc. Design Automation Conf. – 1994. – P. 225–231.
6. Chang S., Cheng D., Marek-Sadowska M. Minimizing ROBDD Size of Incompletely Specified Multiple Output Functions // Proc. European Design and Test Conf. – 1994. – P. 620–624.
7. Oliveira A., Carloni L., Villa T., and Sangiovanni-Vincentelli A. Exact minimization of Binary Decision Diagrams Using Implicit Techniques// IEEE Trans. Computers. – 1998. – V. 47. – No. 11. – P. 1282–1296.

*Национальный исследовательский Томский государственный университет, г. Томск, Россия

E-mail: sergeiostanin@yandex.ru

Останин Сергей Александрович, к.т.н., доцент;
Кириенко Ирина Александровна, магистрант.

S.A. OSTANIN, I.E. KIRIENKO

MINIMIZING INCOMPLETELY SPECIFIED BOOLEAN FUNCTIONSIMPLEMENTED BY BDD

The modification of the algorithm for the exact minimization of BDD size for incompletely specified Boolean functions is proposed. A proposed algorithm is based on use of the compatibility graph with the subsequent allocation of cliques of the graph using of additional conditions. The proposed modification allows to reduce the number of operations when determining cliques of the graph without the accuracy loss.

Keywords: *Binary Decision Diagram (BDD), incompletely specified Boolean functions, minimizing BDD size.*

REFERENCES

1. Bryant R. *Graph Based Algorithm for Boolean Function Manipulation* // IEEE Trans. Computers. – 1986. – V. 35. – No. 8. – P. 667–691.
2. Matrosova A., Loukovnikova E., Ostanin S., Zinchuk A., Nikolaeva E. *Test Generation for Single and Multiple Stuck-at Faults of a Combinational Circuit Designed by Covering Shared ROBDD with CLBs* // Proc. IEEE Intl. Symp. on Defect and Fault-Tolerance in VLSI Systems (DFT 2007). – 2007. – P. 206–214.
3. Matrosova A., Ostanin S., Kirienko I. *Increasing Manufacturing Yield Using Partially Programmable Circuits with CLB implementation of Incompletely Specified Boolean Function of the Corresponding Sub-circuit* // Proc. IEEE Intl. Symp. on Design and Diagnostics of Electronic Circuits and Systems (DDECS 2015). – 2015. – P. 267–270.
4. Lavagno L., McGeer P., Saldanha A., and Sangiovanni-Vincentelli A. *Timed Shannon Circuits: A Power-Efficient Design Style and Synthesis Tool* // Proc. 32nd Design Automation Conf. – 1995. – P. 254–260.
5. Shiple T., Hojati R., Sangiovanni-Vincentelli A., and Bryaton R. *Heuristic Minimization of BDDs Using Don't Cares* // Proc. Design Automation Conf. – 1994. – P. 225–231.
6. Chang S., Cheng D., Marek-Sadowska M. *Minimizing ROBDD Size of Incompletely Specified Multiple Output Functions* // Proc. European Design and Test Conf. – 1994. – P. 620–624.
7. Oliveira A., Carloni L., Villa T., and Sangiovanni-Vincentelli A. *Exact minimization of Binary Decision Diagrams Using Implicit Techniques* // IEEE Trans. Computers. – 1998. – V. 47. – No. 11. – P. 1282–1296.