ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

2022 Управление, вычислительная техника и информатика № 61

Tomsk State University Journal of Control and Computer Science

# ДИСКРЕТНЫЕ ФУНКЦИИ И АВТОМАТЫ

# DISCRETE FUNCTION AND AUTOMATONS

Научная статья
УДК 519.7
doi: 10.17223/19988605/61/13

## Графовые представления множеств всех достижимых реакций комбинационной схемы

**Виктор Алексеевич Провкин[1], Анжела Юрьевна Матросова[2]**

*[1, 2] Томский государственный университет, Томск, Россия*
*[1] prowkan@mail.ru*
*[2] mau11@yandex.ru*

**Аннотация.** Рассматривается задача получения множества всех достижимых реакций комбинационной логической схемы. Предлагается алгоритм построения ROBDD-графа, представляющего все достижимые реакции схемы. Получаемый граф содержит внутренние вершины, которые помечены только выходными переменными схемы. Алгоритм может быть использован в тех случаях, когда ROBDD-граф, зависящий от входных и выходных переменных, не может быть построен из-за экспоненциального роста количества вершин.

**Ключевые слова:** комбинационные схемы; ROBDD-графы; булевы функции

***Для цитирования*:** Провкин В.А., Матросова А.Ю. Графовые представления множеств всех достижимых реакций комбинационной схемы // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2022. № 61. С. 128–138. doi: 10.17223/19988605/61/13

Original article
doi: 10.17223/19988605/61/13

## Graph representations of the sets of all reachable reactions of the combinational circuit

**Viktor A. Provkin[1], Anzhela Yu. Matrosova[2]**

*[1, 2] Tomsk State University, Tomsk, Russian Federation*
*[1] prowkan@mail.ru*
*[2] mau11@yandex.ru*

**Abstract.** The problem of obtaining the set of all achievable reactions of a combinational logic circuit is considered. An algorithm for constructing a ROBDD graph representing all the achievable reactions of the circuit is proposed. The resulting graph contains internal vertices, which are labeled only with schema output variables. The algorithm can be used in cases where an ROBDD graph that depends on input and output variables cannot be obtained because of the exponential growth of the number of vertices.

**Keywords:** combinational circuits; reduced ordered binary decision diagrams; Boolean functions

***For citation*:** Provkin V.A., Matrosova A.Yu. (2022) Graph representations of the sets of all reachable reactions of the combinational circuit. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitelnaja tehnika i informatika – Tomsk State University Journal of Control and Computer Science*. 61. pp. 128–138. doi: 10.17223/19988605/61/13

При решении ряда задач диагностики и тестирования логических схем полезными оказываются особые функции для внутренних полюсов, которые зависят от некоторого подмножества предшествующих полюсов схемы. Эти функции являются частично определенными. Область их определения состоит из таких наборов предшествующих полюсов, что смена значения сигнала на соответствующем набору внутреннем полюсе вызывает смену значений сигналов хотя бы на одном из выходов схемы. Такие функции могут использоваться для маскирования неисправностей и вредоносных подсхем [1, 2], верификации частичных реализаций схем [3, 4], структурного кодирования («обфускации») [5, 6], защиты схем со структурной избыточностью (троирование) [7, 8].

Такие функции могут вычисляться либо с помощью частично определенных функций внутренних полюсов, зависящих от входов схемы (область определения таких функций – множество тестовых наборов для одиночной константной неисправности в соответствующем полюсе) [9, 10], либо без использования таких функций. Второй способ более предпочтителен, так как построение частично определенной функции внутренних полюсов, зависящей от входов схемы, возможно далеко не всегда, поскольку в некоторых случаях размеры ROBDD-графов, требуемых для построения функции, растут экспоненциально. Алгоритм построения, реализующий этот способ, приведен в работе [11]. На первом шаге алгоритма вычисляется множество наборов, которые могут появиться на некотором подмножестве внутренних полюсов схемы. Эта задача эквивалентна задаче получения множества всех возможных выходных реакций некоторой известной комбинационной схемы.

В работе [11] данная задача решается путем перебора всех возможных наборов (выходных реакций на подмножестве полюсов) с проверкой их на достижимость. При маскировании неисправностей число переменных, от которых зависит частично определенная функция, обычно достигает одного-двух десятков, и поэтому полный перебор выполняется за приемлемое время. Однако при решении других задач, например при верификации частичных реализаций логических схем, число переменных может достигать нескольких десятков, а число возможных реакций, как известно, увеличивается экспоненциально с ростом числа переменных. Поэтому желательно иметь алгоритмы, которые определяют множество достижимых реакций схемы без полного перебора и компактно их представляют.

В данной работе предлагается алгоритм построения ROBDD-графа, представляющего множество всех достижимых реакций комбинационной логической схемы. Получаемый граф содержит внутренние вершины, которые помечены только выходными переменными схемы. Алгоритм предлагается использовать в тех случаях, когда ROBDD-граф, зависящий от входных и выходных переменных, не может быть построен из-за экспоненциального роста количества его внутренних вершин.

## 1. Основные определения и постановка задачи

Комбинационная логическая схема – модель устройства без памяти. У таких устройств состояния выходов однозначно определяются набором входных сигналов (булевым вектором на множестве входных переменных). Состояние выходов комбинационной схемы (булев вектор на множестве выходных переменных схемы) при подаче на входы определенного набора называется реакцией схемы на этот входной набор.

Например, рассмотрим комбинационную схему (рис. 1), реализующую систему булевых функций:

$$y_1 = x_1x_2 \vee x_3,$$
$$y_2 = \overline{x_1x_2}x_3,$$
$$y_3 = \overline{x_3} \vee \overline{x_4 \vee x_5},$$
$$y_4 = \overline{x_3}(x_4 \vee x_5).$$

Если на входы схемы поступает набор 10101, то на выходах схемы достигается набор 1100, т.е. набор 1100 является реакцией схемы на входной набор 10101.

Булев вектор в пространстве выходных переменных называется достижимой реакцией, если существует набор, при подаче на входы которого заданный булев вектор появляется в качестве реакции схемы.

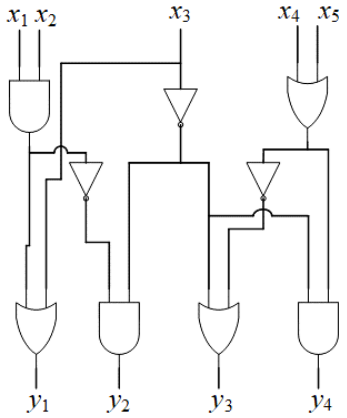


Рис. 1. Пример комбинационной схемы
Fig. 1. Example of combinational circuit

Формально задача может быть поставлена следующим образом. Имеется многовыходная комбинационная схема *C*, реализующая систему булевых функций $f_i(x_1, x_2 \ldots, x_n), \quad i = 1, \ldots, m$. Требуется найти такое множество $N = \{(\beta_1, \beta_2, \ldots, \beta_m)\} \subseteq B_2^m$, что для каждого $(\beta_1, \beta_2, \ldots, \beta_m) \in N$ существует непустое множество, такое что для каждого $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in M \quad f_i(\alpha_1, \alpha_2, \ldots, \alpha_n) = \beta_i, \quad i = 1, \ldots, m$. Элементы множества $N = \{(\beta_1, \beta_2, \ldots, \beta_m)\} \subseteq B^m$ будем называть достижимыми реакциями схемы. Множество $M = \{(\alpha_1, \alpha_2, \ldots, \alpha_n)\} \subseteq B_2^n$ будем называть полным прообразом реакции $(\beta_1, \beta_2, \ldots, \beta_m) \in N$. Отметим, что для различных наборов $(\beta_1^1, \beta_2^1, \ldots, \beta_m^1) \in N$ и $(\beta_1^2, \beta_2^2, \ldots, \beta_m^2) \in N$ соответствующие множества $M^1$ и $M^2$ не пересекаются (их пересечение означало бы, что подача одного набора на входы схемы приводит к двум разным реакциям одновременно, что невозможно).

## 2. Определение функции, содержащей информацию о достижимых реакциях комбинационной схемы, и ее свойства

Пусть задана многовыходная комбинационная схема *C*, реализующая систему булевых функций $y_1 = f_1(x_1, x_2, \ldots, x_n), \quad y_2 = f_2(x_1, x_2, \ldots, x_n), \quad \ldots, \quad y_m = f_m(x_1, x_2, \ldots, x_n)$. Рассмотрим следующую функцию, которая зависит от входных и выходных переменных комбинационной схемы:

$$f(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m) = \bigwedge_{i=1}^{m} \left[ y_i \sim f_i(x_1, x_2, \ldots, x_n) \right],$$

где ~ – логическая операция «эквивалентность». Эта функция обладает следующими свойствами:

1. Если вместо входных переменных подставить константы из {0, 1}, то получим функцию, которая принимает единичное значение на единственном наборе. Этот набор является реакцией схемы на данный входной набор. На остальных наборах полученная функция принимает нулевое значение. Пусть $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ – некоторый входной набор, тогда существует один и только один выходной набор $(\beta_1, \beta_2, \ldots, \beta_m)$ такой, что $f(\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_m) = 1$, причем $\beta_1 = f_1(\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\beta_2 = f_2(\alpha_1, \alpha_2, \ldots, \alpha_n)$, ..., $\beta_m = f_m(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Для всякого набора $(\gamma_1, \gamma_2, \ldots, \gamma_m) \neq (\beta_1, \beta_2, \ldots, \beta_m)$

$$f(\alpha_1, \alpha_2, \ldots, \alpha_n, \gamma_1, \gamma_2, \ldots, \gamma_m) = 0.$$

2. Если вместо выходных переменных подставить константы из {0, 1}, то получится функция, которая принимает единичное значение на всех наборах, которые обеспечивают проявление заданной реакции на выходах схемы. В частном случае, если заданная реакция недостижима, то полученная функция тождественно равна 0. Рассмотрим некоторый выходной набор $(\beta_1, \beta_2, \ldots, \beta_m)$. Тогда

$f(\alpha_1,\alpha_2,\ldots,\alpha_n,\beta_1,\beta_2,\ldots,\beta_m)=1$ для любого набора $(\alpha_1,\alpha_2,\ldots,\alpha_n)$ такого, что $\beta_1 = f_1(\alpha_1,\alpha_2,\ldots,\alpha_n)$, $\beta_2 = f_2(\alpha_1,\alpha_2,\ldots,\alpha_n)$, …, $\beta_m = f_m(\alpha_1,\alpha_2,\ldots,\alpha_n)$. Если таких наборов не существует, то $f(\alpha_1,\alpha_2,\ldots,\alpha_n,y_1=\beta_1,y_2=\beta_2,\ldots,y_m=\beta_m)=0$ для всех наборов $(\alpha_1,\alpha_2,\ldots,\alpha_n)$.

Для получения всех достижимых реакций схемы можно построить ROBDD-граф функции $f(x_1,x_2,\ldots,x_n,y_1,y_2,\ldots,y_m)$, причем выбираем следующий порядок переменных при использовании формулы разложения Шеннона: $y_1,y_2,\ldots,y_m,x_1,x_2,\ldots,x_n$. Полученный граф будет обладать следующими свойствами.

**Утверждение 1.** Каждая простая цепь, заканчивающаяся в вершине со значением 0 и не содержащая вершин, помеченных входными переменными, соответствует недостижимому набору (возможно, некоторому множеству наборов).

***Доказательство.*** Корневой вершине графа соответствует функция $f(x_1,x_2,\ldots,x_n,y_1,y_2,\ldots,y_m)$, и этой вершине приписана переменная $y_1$. При переходе из корневой вершины по дуге, помеченной значением 1 (0), попадаем в вершину, которой соответствует функция $f(x_1,x_2,\ldots,x_n,1,y_2,\ldots,y_m)$ ($f(x_1,x_2,\ldots,x_n,0,y_2,\ldots,y_m)$). Эта вершина помечена переменной $y_2$. Перейдя из нее по дуге, помеченной значением 1 (0), получаем функцию, которая построена из функции, соответствующей предыдущей вершине, подстановкой значения 1 (0) в переменную $y_2$. Продолжим этот процесс до тех пор, пока не дойдем до 0-концевой вершины. Так как в цепи присутствуют только вершины, помеченные выходными переменными, то это значит, что подстановки значений выполнялись только в те аргументы функции, которые соответствуют выходным переменным схемы. Поскольку цепь заканчивается в 0-концевой вершине, то $f(\alpha_1,\alpha_2,\ldots,\alpha_n,y_1=\beta_1,y_2=\beta_2,\ldots,y_m=\beta_m)=0$ для всех наборов $(\alpha_1,\alpha_2,\ldots,\alpha_n)$ и, следовательно, набор $(\beta_1,\beta_2,\ldots,\beta_m)$ недостижим.

**Утверждение 2.** Каждая простая цепь, содержащая вершины, помеченные входными переменными схемы, и заканчивающаяся в 1-терминальной вершине графа, соответствует некоторой достижимой реакции, представленной всеми выходными переменными схемы, за счет отрезка цепи, связывающего рассматриваемую простую цепь с корнем графа.

***Доказательство.*** Если пройти по отрезку цепи из корневой вершины до первой вершины *v* простой цепи, помеченной входной переменной, то этой вершине сопоставляется функция $f(x_1,x_2,\ldots,x_n,y_1=\beta_1,y_2=\beta_2,\ldots,y_m=\beta_m)$, которая принимает единичное значение на таких наборах $(\alpha_1,\alpha_2,\ldots,\alpha_n)$, для которых справедливы равенства $\beta_1 = f_1(\alpha_1,\alpha_2,\ldots,\alpha_n)$, $\beta_2 = f_2(\alpha_1,\alpha_2,\ldots,\alpha_n)$, …, $\beta_m = f_m(\alpha_1,\alpha_2,\ldots,\alpha_n)$. Иными словами, граф с корнем в вершине *v* представляет функцию, являющуюся полным прообразом реакции $(\beta_1,\beta_2,\ldots,\beta_m)$.

Пусть задана комбинационная схема, поведение которой описывается следующей таблицей истинности (табл. 1).

Таблица 1

**Пример таблицы истинности, описывающей поведение комбинационной схемы**

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y_1$ | $y_2$ | $y_3$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |

Окончание табл. 1

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y_1$ | $y_2$ | $y_3$ |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |

ROBDD-граф функции $f(x_1, x_2, x_3, x_4, y_1, y_2, y_3)$ представляется следующим образом (рис. 2).

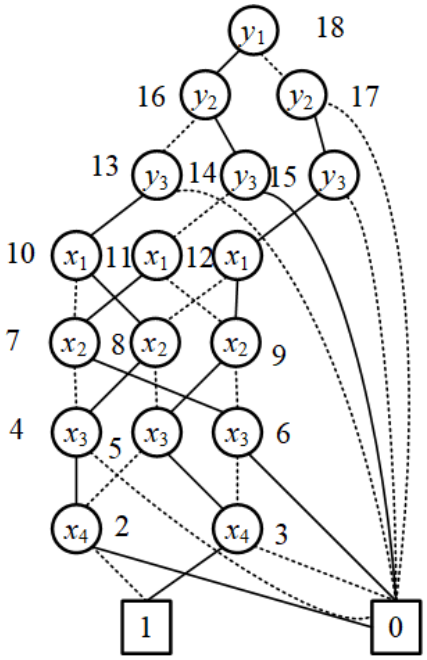


Рис. 2. ROBDD-граф функции $f(x_1, x_2, x_3, x_4, y_1, y_2, y_3)$

Fig. 2. ROBDD of function $f(x_1, x_2, x_3, x_4, y_1, y_2, y_3)$

Будем искать в ROBDD-графе пути, начинающиеся в корневой вершине, до первой вершины, помеченной входной переменной схемы. На рис. 2 видно, что таких путей три: 18–16–13–10, 18–16–15–11 и 18–17–15–12. Им сопоставляются двоичные векторы 101, 110 и 011 соответственно. Выделив подграфы с корневыми вершинами 10, 11 и 12, получаем входные наборы, поступление которых на входы схемы приводит к появлению соответствующей реакции на выходах. Так, для реакции 101 это наборы 0010, 0101, 1000, 1011 и 1110, для реакции 110 – наборы 0001, 0100, 0111, 1010 и 1101, а для реакции 011 – наборы 0000, 0011, 0110, 1001, 1100 и 1111.

### 3. Модификация общего алгоритма построения ROBDD-графа для построения графа реакций

С помощью ROBDD-графа, построенного по функции $f(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)$, можно получить все достижимые реакции схемы. Однако часто в задачах требуется найти только достижимые реакции схемы, и не требуется знать, какие входные наборы дают соответствующие реакции. Например, для предыдущего примера граф, который представляет только достижимые реакции схемы, выглядит следующим образом (рис. 3).

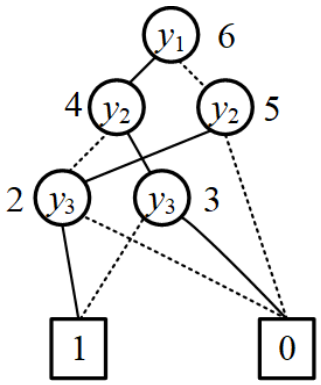


Рис. 3. ROBDD-граф достижимых реакций схемы
Fig. 3. ROBDD of reachable circuit patterns

Этот граф, назовем его в дальнейшем графом достижимых реакций, получается из исходного следующим образом: если 1(0)-дуга из вершины, помеченной выходной переменной схемы, ведет в вершину, помеченную входной переменной схемы, то из вершины, помеченной выходной переменной схемы, проводится 1(0)-дуга в 1-терминальную вершину графа. Затем из графа удаляются все вершины, помеченные входными переменными схемы. Далее в графе выполняются соответствующие для ROBDD операции упрощения.

Очевидно, что граф достижимых реакций содержит существенно меньше внутренних вершин, чем граф функции $f\left(x_1,x_2,\ldots,x_n,y_1,y_2,\ldots,y_m\right)$ (7 вершин вместо 19). Хотя граф достижимых реакций может быть получен из графа функции $f\left(x_1,x_2,\ldots,x_n,y_1,y_2,\ldots,y_m\right)$, желательно иметь алгоритм его построения без предварительного получения графа функции *f*. Такой алгоритм может быть использован в тех случаях, когда, например, построение графа функции *f* невозможно из-за экспоненциального роста количества его вершин.

Предлагаемый алгоритм построения ROBDD-графа, представляющего все достижимые реакции схемы, основан на алгоритме построения ROBDD-графа произвольной булевой функции. В графе сначала выполняется разложение по выходным переменным схемы, а затем по ее входным переменным. Модификация этого алгоритма основана на следующем правиле.

Пусть в графе дуга из некоторой вершины, помеченной выходной переменной схемы, ведет в вершину, помеченную входной переменной схемы. Тогда вершина, помеченная входной переменной схемы, представляет функцию $f\left(x_1,x_2,\ldots,x_n\right)=f\left(x_1,x_2,\ldots,x_n,y_1=\beta_1,y_2=\beta_2,\ldots,y_m=\beta_m\right)$, причем эта функция не равна тождественно ни константе 0, ни константе 1 (так как в противном случае эта вершина была бы 1- или 0-терминальной вершиной графа). Эта функция принимает значение 1 на таких двоичных векторах, подача которых на входы схемы приводит к появлению на ее выходах реакции $\left(\beta_1,\beta_2,\ldots,\beta_m\right)$. Если важна только достижимость реакции, но нет необходимости знать, на каких входных наборах она достигается, то значение функции

$$f\left(x_1,x_2,\ldots,x_n\right)=f\left(x_1,x_2,\ldots,x_n,y_1=\beta_1,y_2=\beta_2,\ldots,y_m=\beta_m\right)$$

можно положить равным 1. Это значит, что в соответствующем ROBDD-графе достижимых реакций дуга из вершины, помеченной выходной переменной схемы, должна вести в 1-терминальную вершину. На основе вышесказанного предлагается поступать следующим образом.

Если в процессе построения графа на множестве входных и выходных переменных обнаружено, что функция *f* принимает значение 1 на некотором наборе, то соответствующая 0- или 1-дуга из последней вершины *v*, помеченной выходной переменной схемы, ведет в 1-терминальную вершину графа реакций, а граф с корнем в вершине *v*, состоящий из вершин, помеченных входными переменными схемы, удаляется.

Если для функции *f* такого набора не существует, т.е. она тождественно равна нулю, то соответствующая 0- или 1-дуга из вершины *v* ведет в 0-терминальную вершину строящегося графа.

Далее в полученном графе, состоящем из вершин, помеченных только выходными переменными, выполняются операции упрощения с целью получения ROBDD-графа достижимых реакций.

**Утверждение 3**. Если для двух схем при одном и том же порядке разложения выходных переменных графы достижимых реакций различны, то схемы реализуют различные системы булевых функций.

***Доказательство*** следует из свойств ROBDD-графов, полученных при одном и том же порядке разложения переменных.

Покажем работу алгоритма на примере. Рассмотрим комбинационную схему с четырьмя входами и тремя выходами, поведение которой описывается системой булевых функций:

$$y_1 = f_1(x_1,x_2,x_3,x_4) = \left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3} \vee x_4\right) \vee x_1\overline{x_2}\left(x_3 \vee \overline{x_4}\right),$$

$$y_2 = f_2(x_1,x_2,x_3,x_4) = \left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3} \vee x_4\right) \vee \overline{x_1}x_2\left(x_3 \vee \overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right),$$

$$y_3 = f_3(x_1,x_2,x_3,x_4) = \left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(x_3 \vee \overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3} \vee x_4\right).$$

Тогда функция

$$f(x_1,x_2,x_3,x_4,y_1,y_2,y_3) = \left(y_1 \sim f_1(x_1,x_2,x_3,x_4)\right)\left(y_2 \sim f_2(x_1,x_2,x_3,x_4)\right)\left(y_3 \sim f_3(x_1,x_2,x_3,x_4)\right) =$$

$$= \left[y_1 \sim \left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3} \vee x_4\right) \vee x_1\overline{x_2}\left(x_3 \vee \overline{x_4}\right)\right)\right] \wedge$$

$$\wedge \left[y_2 \sim \left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3} \vee x_4\right) \vee \overline{x_1}x_2\left(x_3 \vee \overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\right)\right] \wedge$$

$$\wedge \left[y_3 \sim \left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(x_3 \vee \overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3} \vee x_4\right)\right)\right].$$

Пусть переменные $y_1$, $y_2$ и $y_3$ имеют номера 1, 2 и 3 соответственно, переменные $x_1$, $x_2$, $x_3$ и $x_4$ – номера 4, 5, 6 и 7. Число входных переменных $n = 4$, число выходных переменных $m = 3$.

Номер текущей переменной $i = 1$. Подставляем в переменную $y_1$ значение 0:

$$f(x_1,x_2,x_3,x_4,0,y_2,y_3) = \overline{\left[\left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3} \vee x_4\right) \vee x_1\overline{x_2}\left(x_3 \vee \overline{x_4}\right)\right)\right]} \wedge$$

$$\wedge \left[y_2 \sim \left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3} \vee x_4\right) \vee \overline{x_1}x_2\left(x_3 \vee \overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\right)\right] \wedge$$

$$\wedge \left[y_3 \sim \left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(x_3 \vee \overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3} \vee x_4\right)\right)\right].$$

Переходим к переменной $y_2$ ($i = i + 1 = 2$) – подставляем в эту переменную значение 0:

$$f(x_1,x_2,x_3,x_4,0,0,y_3) = \overline{\left[\left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3} \vee x_4\right) \vee x_1\overline{x_2}\left(x_3 \vee \overline{x_4}\right)\right)\right]} \wedge$$

$$\wedge \overline{\left[\left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3} \vee x_4\right) \vee \overline{x_1}x_2\left(x_3 \vee \overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\right)\right]} \wedge$$

$$\wedge \left[y_3 \sim \left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(x_3 \vee \overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3} \vee x_4\right)\right)\right].$$

Переходим к переменной $y_3$ ($i = i + 1 = 3$) – подставляем в эту переменную значение 0:

$$f(x_1,x_2,x_3,x_4,0,0,0) = \overline{\left[\left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3} \vee x_4\right) \vee x_1\overline{x_2}\left(x_3 \vee \overline{x_4}\right)\right)\right]} \wedge$$

$$\wedge \overline{\left[\left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3} \vee x_4\right) \vee \overline{x_1}x_2\left(x_3 \vee \overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\right)\right]} \wedge$$

$$\wedge \overline{\left[\left(\left(\overline{x_1}\,\overline{x_2} \vee x_1x_2\right)\left(x_3 \vee \overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3} \vee x_4\right)\right)\right]}.$$

Переходим к переменной $x_1$ ($i = i + 1 = 4$) – подставляем в эту переменную значение 0:

$$f(0,x_2,x_3,x_4,0,0,0) = \overline{\left[\left(\overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee x_2\left(\overline{x_3} \vee x_4\right)\right)\right]} \wedge$$

$$\wedge \overline{\left[\left(\overline{x_2}\left(\overline{x_3} \vee x_4\right) \vee x_2\left(x_3 \vee \overline{x_4}\right)\right)\right]\left[\left(\overline{x_2}\left(x_3 \vee \overline{x_4}\right) \vee x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\right)\right]}.$$

Переходим к переменной $x_2$ ($i = i + 1 = 5$) – подставляем в эту переменную значение 0:

$$f(0,0,x_3,x_4,0,0,0) = \overline{\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\left(\overline{x_3} \vee x_4\right)\left(x_3 \vee \overline{x_4}\right)}.$$

Переходим к переменной $x_3$ ($i = i + 1 = 6$) – подставляем в эту переменную значение 0:

$$f(0,0,0,x_4,0,0,0) = 0.$$

При подстановке в переменную $x_4$ и значения 0, и значения 1 значение функции равно 0. Поэтому функции $f(0,0,0,x_4,0,0,0)$ соответствует 0-терминальная вершина.

Подставляем в переменную $x_3$ значение 1: $f(0,0,1,x_4,0,0,0) = \overline{x_4}x_4 = 0$.

При подстановке в переменную $x_4$ и значения 0, и значения 1 значение функции равно 0. Поэтому функции $f(0,0,1,x_4,0,0,0)$ соответствует 0-терминальная вершина.

Следовательно, и функции $f(0,0,x_3,x_4,0,0,0)$ соответствует 0-терминальная вершина.

Возвращаемся к переменной $x_2$ ($i = 5$) и подставляем в эту переменную значение 1:

$$f(0,1,x_3,x_4,0,0,0) = \overline{\left(\overline{x_3} \vee x_4\right)\left(x_3 \vee \overline{x_4}\right)\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)}.$$

Переходим к переменной $x_3$ и подставляем в нее значения 0 и 1:

$$f(0,1,0,x_4,0,0,0) = 0, \quad f(0,1,1,x_4,0,0,0) = 0.$$

Получаем, что функции $f(0,1,x_3,x_4,0,0,0)$ соответствует 0-терминальная вершина.

Возвращаемся к переменной $x_2$: получаем, что $f(0,0,x_3,x_4,0,0,0) = f(0,1,x_3,x_4,0,0,0) = 0$. Поэтому функции $f(0,x_2,x_3,x_4,0,0,0)$ также соответствует 0-терминальная вершина.

Возвращаемся к переменной $x_1$ ($i = 4$) и подставляем в эту переменную значение 1:

$$f(1,x_2,x_3,x_4,0,0,0) = \overline{\left[\left(x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee \overline{x_2}\left(x_3 \vee \overline{x_4}\right)\right)\right]} \wedge$$

$$\wedge \overline{\left[\left(x_2\left(\overline{x_3} \vee x_4\right) \vee \overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\right)\right]\left[\left(x_2\left(x_3 \vee \overline{x_4}\right) \vee \overline{x_2}\left(\overline{x_3} \vee x_4\right)\right)\right]},$$

$$f(1,0,x_3,x_4,0,0,0) = \overline{\left(x_3 \vee \overline{x_4}\right)\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\left(\overline{x_3} \vee x_4\right)} = 0,$$

$$f(1,1,x_3,x_4,0,0,0) = \overline{\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\left(\overline{x_3} \vee x_4\right)\left(x_3 \vee \overline{x_4}\right)} = 0.$$

В результате получили, что $f(x_1,x_2,x_3,x_4,0,0,0) = 0$, т. е. функции $f(x_1,x_2,x_3,x_4,0,0,0)$ соответствует 0-терминальная вершина.

Вернувшись к переменной $y_3$ ($i = 3$) и подставив в нее значение 1, получим, что

$$f(x_1,x_2,x_3,x_4,0,0,1) = 0.$$

Таким образом, имеем

$$f(x_1,x_2,x_3,x_4,0,0,y_3) = 0.$$

Теперь вернемся к переменной $y_2$ ($i = 2$) и подставим в нее значение 1. Далее подставим значение 0 в переменную $y_3$. Получим, что $f(x_1,x_2,x_3,x_4,0,1,0) = 0$. Подставим значение 1 в переменную $y_3$:

$$f(x_1,x_2,x_3,x_4,0,1,1) = \overline{\left[\left(\left(\overline{x_1}\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3} \vee x_4\right) \vee x_1\overline{x_2}\left(x_3 \vee \overline{x_4}\right)\right)\right]} \wedge$$

$$\wedge \left[\left(\left(\overline{x_1}\overline{x_2} \vee x_1x_2\right)\left(\overline{x_3} \vee x_4\right) \vee \overline{x_1}x_2\left(x_3 \vee \overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\right)\right] \wedge$$

$$\wedge \left[\left(\left(\overline{x_1}\overline{x_2} \vee x_1x_2\right)\left(x_3 \vee \overline{x_4}\right) \vee \overline{x_1}x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee x_1\overline{x_2}\left(\overline{x_3} \vee x_4\right)\right)\right].$$

Переходим к переменной $x_1$ – подставляем в эту переменную значение 0:

$$f(0,x_2,x_3,x_4,0,1,1) = \overline{\left[\left(\overline{x_2}\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right) \vee x_2\left(\overline{x_3} \vee x_4\right)\right)\right]}\left[\left(\overline{x_2}\left(\overline{x_3} \vee x_4\right) \vee x_2\left(x_3 \vee \overline{x_4}\right)\right)\right] \wedge$$

$$\wedge \left[\left(\overline{x_2}\left(x_3 \vee \overline{x_4}\right) \vee x_2\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)\right)\right].$$

Подставим значение 0 в переменную $x_2$: $f(0,0,x_3,x_4,0,1,1)=\overline{\left(\overline{x_3}x_4 \vee x_3\overline{x_4}\right)}\left(\overline{x_3} \vee x_4\right)\left(x_3 \vee \overline{x_4}\right)$.

Подставим значение 0 в переменную $x_3$: $f(0,0,0,x_4,0,1,1)=\overline{x_4}$.

Подставим значение 1 в переменную $x_4$ и получим, что $f(0,0,0,0,0,1,1)=1$, т.е. функции $f(0,0,0,0,0,1,1)$ соответствует 1-терминальная вершина. Это значит, что функциям $f(0,0,0,x_4,0,1,1)$, $f(0,0,x_3,x_4,0,1,1)$, $f(0,x_2,x_3,x_4,0,1,1)$ и $f(x_1,x_2,x_3,x_4,0,1,1)$ также соответствует 1-терминальная вершина. Ранее мы получили, что функции $f(x_1,x_2,x_3,x_4,0,1,0)$ соответствует 0-терминальная вершина. Поэтому функции $f(x_1,x_2,x_3,x_4,0,1,y_3)$ соответствует внутренняя вершина, помеченная переменной $y_3$, из которой 0-дуга ведёт в 0-терминальную вершину, а 1-дуга – в 1-терминальную вершину. Добавим в ROBDD-граф такую вершину (рис. 4).

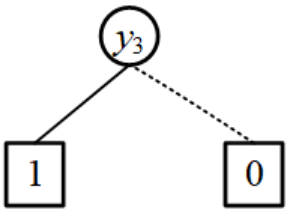


Рис. 4. ROBDD-граф функции $f(x_1,x_2,x_3,x_4,0,1,y_3)$

Fig. 4. ROBDD of function $f(x_1,x_2,x_3,x_4,0,1,y_3)$

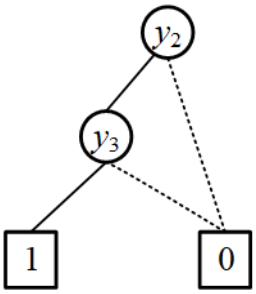


Рис. 5. ROBDD-граф функции $f(x_1,x_2,x_3,x_4,0,y_2,y_3)$

Fig. 5. ROBDD of function $f(x_1,x_2,x_3,x_4,0,y_2,y_3)$

Ранее мы получили, что $f(x_1,x_2,x_3,x_4,0,0,y_3)=0$. Поэтому функции $f(x_1,x_2,x_3,x_4,0,y_2,y_3)$ соответствует внутренняя вершина, помеченная переменной $y_2$, из которой 0-дуга ведет в 0-терминальную вершину, а 1-дуга – в ранее добавленную внутреннюю вершину. Теперь граф примет вид, представленный на рис. 5.

Далее подставим значение 1 в переменную $y_1$ и продолжим работу алгоритма. В результате получаем ROBDD-граф, изображенный на рис. 6.

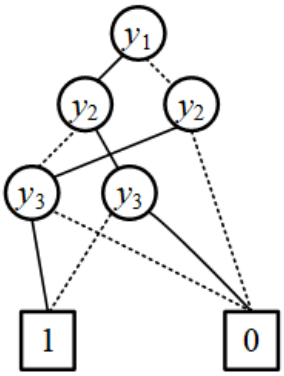


Рис. 6. ROBDD-граф, представляющий все достижимые реакции

Fig. 6. ROBDD representing all reachable patterns

### 4. Экспериментальное сравнение размеров полного и сокращённого графов

Разработана программа, реализующая алгоритм. Проведено сравнение количества вершин графа функции $f$ и графа достижимых реакций схемы на контрольных примерах схем из набора LGSynth89. Результаты сравнения приведены в табл. 2.

Таблица 2

**Экспериментальные результаты: сравнение числа вершин в полном и сокращенном графах**

| Название схемы | Число входов | Число выходов | Число достижимых реакций | Число вершин в графе функции *f* | Число вершин в графе реакций |
|---|---|---|---|---|---|
| alu2 | 10 | 6 | 38 | 736 | 10 |
| alu4 | 14 | 8 | 146 | 8 174 | 12 |
| C432 | 36 | 7 | 128 | 2 608 | 1 |
| x2 | 10 | 7 | 14 | 107 | 24 |

Из таблицы видно, что предлагаемый алгоритм позволяет строить ROBDD-графы достижимых реакций схемы, содержащие существенно меньше, иногда на порядки, внутренних вершин, чем графы соответствующих функций *f*, зависящих от входных и выходных переменных схемы. В частности, если все возможные двоичные наборы достижимы на выходах схемы, то получается граф реакций, состоящий из одной терминальной вершины 1 (схема C432).

## Заключение

Предложен алгоритм построения ROBDD-графа, представляющего множество всех достижимых реакций комбинационной логической схемы, зависящего только от выходных переменных схемы. При построении графа реакций не требуется получения предложенного в работе более громоздкого графа функции *f*, представляющей входные наборы схемы вместе с порождаемыми ими реакциями. Проведено сравнение количества вершин в графе функции *f* и соответствующем графе реакций схемы. Эксперименты показали, что предлагаемый алгоритм позволяет строить графы достижимых реакций схемы, значительно более компактные, чем графы функций *f*.

### Список источников

1. Jiang J.-H.R., Kravets V.N., Lee N.-Z. Engineering Change Order for Combinational and Sequential Design Rectification // Design, Automation & Test in Europe Conference & Exhibition (DATE) : Proc. 2020. P. 726–731.
2. Dhar T., Roy S.K., Giri C. Hardware Trojan Detection by Stimulating Transitions in Rare Nets // Proc. 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID). 2019. P. 537–538.
3. Wimmer R., Wimmer K., Scholl C., Becker B. Analysis of Incomplete Circuits Using Dependency Quantified Boolean Formulas // Advanced Logic Synthesis / A. Reis, R. Drechsler (eds). Cham : Springer, 2018. P. 151–168.
4. Becker B., Scholl C., Wimmer R. Verification of Incomplete Designs. Formal System Verification / R. Drechsler (eds). Cham : Springer, 2018. P. 37–72.
5. Золоторевич Л.А. Аппаратная защита цифровых устройств // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2020. № 50. С. 69–78.
6. Yasin M., Rajendran J., Sinanoglu O., Karri R. On Improving the Security of Logic Locking // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2016. V. 35 (9). P. 1411–1424.
7. Gunti N.B., Lingasubramanian K. Fault Sensitive Neutralization of Hardware Trojans Using Multi-level Triple Modular Redundancy Scheme // Proc. 2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS). 2017. P. 105–110.
8. Gunti N.B., Lingasubramanian K. Neutralization of the Effect of Hardware Trojan in SCADA System Using Selectively Placed TMR // Proc. 2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS). 2017. P. 99–104.
9. Matrosova A., Provkin V., Nikolaeva E. Masking Internal Node Faults and Trojan Circuits in Logical Circuits // Proc. of 2019 IEEE East-West Design & Test Symposium (EWDTS), 13–16 September 2019, Batumi. Kharkov : IEEE, 2019. P. 416–419.
10. Matrosova A., Provkin V. Masking Internal Node Logical Faults and Trojan Circuits Injections with Using SAT Solvers // Proc. 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2020), Cluj-Napoca, Romania 21–23 May 2020. New York : IEEE, 2020. P. 1–4.
11. Matrosova A., Provkin V. Applying Incompletely Specified Boolean Functions for Patch Circuit Generation // Proc. 2021 IEEE East-West Design & Test Symposium (EWDTS). Batumi, Georgia, 10–13 September 2021. Red Hook : IEEE, 2021. P. 238–241.

### References

1. Jiang, J.-H.R., Kravets, V.N. & Lee, N.-Z. (2020) Engineering Change Order for Combinational and Sequential Design Rectification. *Design, Automation & Test in Europe Conference & Exhibition (DATE).* Proceedings of the Conference. pp. 726–731.

2. Dhar, T., Roy, S.K. & Giri, C. (2019) Hardware Trojan Detection by Stimulating Transitions in Rare Nets. *Proc. of the 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*. pp. 537–538.
3. Wimmer, R., Wimmer, K., Scholl, C. & Becker, B. (2018) Analysis of Incomplete Circuits Using Dependency Quantified Boolean Formulas. In: Reis, A. & Drechsler, R. (eds) *Advanced Logic Synthesis*. Springer, Cham.
4. Becker, B., Scholl, C. & Wimmer, R. (2018) Verification of Incomplete Designs. In: Drechsler, R. (eds) *Formal System Verification*. Springer, Cham.
5. Zolotorevich, L.A. (2020) Hardware protection of digital device. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 50. pp. 69–78. DOI: 10.17223/19988605/50/9
6. Yasin, M., Rajendran, J., Sinanoglu, O. & Karri, R. (2016) On Improving the Security of Logic Locking. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 35(9). pp. 1411–1424. DOI: 10.1109/TCAD.2015.2511144
7. Gunti, N.B. & Lingasubramanian, K. (2017) Fault Sensitive Neutralization of Hardware Trojans Using Multi-level Triple Modular Redundancy Scheme. *IEEE International Symposium on Nanoelectronic and Information Systems (INIS)*. pp. 105–110.
8. Gunti, N.B. & Lingasubramanian K. (2017) Neutralization of the Effect of Hardware Trojan in SCADA System Using Selectively Placed TMR. *Proceedings 2017 IEEE International Symposium on Nanoelectronic and Information Systems (INIS)*. pp. 99–104.
9. Matrosova, A., Provkin, V. & Nikolaeva, E. (2019) Masking Internal Node Faults and Trojan Circuits in Logical Circuits. *Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS).* 13–16 September. Batumi, Kharkov: IEEE. pp. 416–419.
10. Matrosova, A. & Provkin, V. (2020) Masking Internal Node Logical Faults and Trojan Circuits Injections with Using SAT Solvers. *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2020)*. Cluj-Napoca, Romania, May 21–23, 2020. Proceedings. New York, USA: IEEE. pp. 1–4.
11. Matrosova, A. & Provkin, V. (2021) Applying Incompletely Specified Boolean Functions for Patch Circuit Generation. *Proceedings IEEE East-West Design & Test Symposium (EWDTS).* Batumi, Georgia, September 10–13, 2021. [Red Hook]: IEEE. pp. 238–241.

***Информация об авторах*:**

**Провкин Виктор Алексеевич** – аспирант кафедры компьютерной безопасности Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета (Томск, Россия). E-mail: prowkan@mail.ru

**Матросова Анжела Юрьевна** – профессор, доктор технических наук, профессор кафедры компьютерной безопасности Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета (Томск, Россия). E-mail: mau11@yandex.ru

***Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации. Авторы заявляют об отсутствии конфликта интересов.***

***Information about the authors:***

**Provkin Viktor A.** (Post-graduate Student, National Research Tomsk State University, Tomsk, Russian Federation). E-mail: prowkan@mail.ru

**Matrosova Anzhela Y.** (Doctor of Technical Sciences, Professor, National Research Tomsk State University, Tomsk, Russian Federation). E-mail: mau11@yandex.ru

***Contribution of the authors: the authors contributed equally to this article. The authors declare no conflicts of interests.***

*Received 14.07.2022; accepted for publication 29.11.2022*

*Поступила в редакцию 14.07.2022; принята к публикации 29.11.2022*