ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2022 Управление, вычислительная техника и информатика № 60
Tomsk State University Journal of Control and Computer Science

ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

DESIGNING AND DIAGNOSTICS OF COMPUTER SYSTEMS

Научная статья
УДК 519.873:519.718.7
doi: 10.17223/19988605/60/11

Определение ключа аппаратной защиты цифровых устройств

Людмила Андреевна Золоторевич[1], Валерий Андреевич Ильинков[2]

[1, 2] Белорусский государственный университет информатики и радиоэлектроники, Минск, Республика Беларусь
[1] lazolotorevich@gmail.com
2 v.ilyinkov@gmail.com

Аннотация. Рассматриваются особенности и надежность логического кодирования комбинационных схем. Предлагается алгоритм взлома кода комбинационных схем, основанный на описании закодированной структуры функцией разрешения и сведении задачи к КНФ-выполнимости. Исходными данными для декодирования структуры цифрового устройства являются структурная реализация закодированной схемы, полученная, например, методом обратного проектирования (проектирования по прототипу), а также активированный физический образец интегральной схемы, в защищенную от несанкционированного доступа память которой загружено подлинное значение ключа. Этот образец может использоваться в виде модели черного ящика. Основная идея взлома ключа состоит в том, чтобы решить задачу, не прибегая к исследованиям на большом интервале значений входных и выходных переменных.

Ключевые слова: цифровое устройство; логическое кодирование; декодирование; функция разрешения; выполнимость КНФ-функции

***Для цитирования*:** Золоторевич Л.А., Ильинков В.А. Определение ключа аппаратной защиты цифровых устройств // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2022. № 60. С. 102–110. doi: 10.17223/19988605/60/11

Original article
doi: 10.17223/19988605/60/11

Determining security key the hardware for digital devices

Ludmila A. Zolotorevich[1], Valery A. Ilyinkov[2]

[1, 2] Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
[1] lazolotorevich@gmail.com
[2] v.ilyinkov@gmail.com

Abstract. The features and reliability of logical coding of combinational circuits are considered. An algorithm for breaking the code of combinational circuits is proposed, based on the description of the encoded structure by the resolution function and the reduction of the problem to CNF satisfiability. The initial data for decoding the structure of a digital device are the structural implementation of the encoded circuit, obtained, for example, by the reverse engi-

neering method (prototype design), as well as an activated physical sample of the integrated circuit, the correct key value is loaded into the memory protected from unauthorized access. This sample can be used as a black box model. The main idea ofkey breaking is to solve the problem without resorting to research on a large range of values of input and output variables.

Keywords: digital device; logical encoding; decoding; resolution function; CNF satisfiability

***For citation*:** Zolotorevich, L.A., Ilyinkov, V.A. (2022) Determining security key the hardware for digital devices. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitelnaja tehnika i informatika – Tomsk State University Journal of Control and Computer Science*. 60. pp. 102–110. doi: 10.17223/19988605/60/11

Серьезной проблемой для электронной и оборонной промышленности в последние годы стали пиратство, перепроизводство и контрафакция, что привело к необходимости защиты проектов СБИС, СнК от несанкционированного вмешательства в цикл проектирования и / или производства интегральных схем [1]. По оценкам Technology Information Handling Services (IHS), финансовый риск из-за контрафактных и несанкционированных микросхем оценивается в более чем 169 млрд долларов в год, что примерно в 10 раз превышает ущерб от пиратства в области ПО [2]. Для оборонной промышленности важнейшей проблемой является возможность использования контрафактных интегральных схем с модифицированными функциями, что в определенное время может деструктивно повлиять на функционирование структуры, ухудшить эксплуатационные характеристики, привести к раскрытию конфиденциальной информации и др. Кроме больших финансовых потерь существует реальная проблема обеспечения национальной безопасности, так как 15% интегральных схем в системах оборонной промышленности являются контрафактными.

В связи с этим стала очевидной необходимость защиты проектов на основе создания таксономии нарушений и отклонений, общего подхода к контролю СБИС, СнК, с моделями которых приходится работать при проектировании и организации контроля на всех этапах жизненного цикла цифровой системы с учетом злонамеренных внедрений в цикл проектирования и производства интегральных схем. Как развитие теории контролепригодного проектирования (Design-for-Testability; DfT) в работе [3] предлагается подход к проектированию Design-for-Trust (DfTr), который дополнительно включает средства для контроля и предотвращения аппаратных атак при проектировании и изготовлении СБИС.

В последние годы для защиты проектов интегральных схем применяются методы и средства аппаратного кодирования комбинационных блоков. Для обеспечения надежности подобной защиты проектов необходимы средства контроля эффективности применяемых методов кодирования, выявления внесенных троянов на основе создания общего подхода к контролю проектов на всех этапах проектирования и производства.

В работе рассматриваются некоторые особенности метода логического кодирования структурных схем цифровых устройств комбинационного типа. Предлагается метод взлома кода при наличии информации о структуре закодированного объекта и возможности доступа к физической модели. Задача решается на основе описания закодированной структуры в виде КНФ-функции разрешения, решения задачи выполнимости (SAT) и физического моделирования объекта.

1. Логическое кодирование ИС как метод аппаратной защиты

В работе [2] проанализированы различные модели процесса злонамеренного искажения проекта, описывающие условия, при которых подобное искажение может внедриться в цифровую систему. В числе возможных источников искажений рассматриваются поставщики базовых функциональных блоков интеллектуальной собственности (IP's), которые приобретаются разработчиками СнК, собственно разработчики СнК, а также кремниевые фабрики – изготовители СнК. Методы несанкционированного доступа в проект могут быть различными, в том числе основываться на применении специальных средств САПР, способных исказить проект на RTL-уровне. В современных условиях наиболее уязвимым этапом может быть этап производства.

Одним из методов борьбы с вышеупомянутыми угрозами является логическое кодирование, которое обеспечивает доступ к объекту только авторизованным пользователям [4]. Метод предпола-

гает сокрытие функциональности проекта и использование ключа, применение которого выводит систему в область правильного функционирования.

Идея кодирования основана на том, чтобы изменить конструкцию ИС, добавив в нее дополнительные логические элементы и новые входы, называемые ключевыми, т.е. на применении метода обфускации структуры объекта. В такой постановке если злоумышленник не владеет ключом, то ему недоступна внутренняя реализация объекта. Задача структурной обфускации и логического кодирования заключается в том, чтобы затруднить или сделать невозможным получение подлинного ключа. Ключевые входы подсоединяются к защищенной от несанкционированного доступа памяти, а закодированная схема будет работать правильно только в том случае, если на ее ключевые входы поданы подлинные значения. Значения ключевых входов передаются после изготовления микросхем конечным пользователям (рис. 1).

Таким образом, логическое кодирование основывается на предположении, что производитель не знает и не может вычислить подлинные значения ключевых входов или, в противном случае, поиск подлинного ключа должен быть для злоумышленника затруднителен.

В литературе предложены различные методы кодирования комбинационной логики, в которых используются в качестве ключевых вентилей элементы XOR / XNOR [1, 5–7], AND / OR [8], мультиплексоры [9] или комбинации этих вентилей [10]. Выбор линии для включения вентиля, тип применяемого вентиля существенно влияют на эффективность кодирования. Воздействие неподлинного ключа можно сравнить с влиянием неисправности константного типа на данной линии (см. рис. 1). При выборе в качестве ключевых вентилей XOR или NXOR применение неподлинного ключа приводит к появлению неисправности константного типа в любом случае, при любом входном воздействии, в отличие от вентилей OR, NOR, AND, NAND, что влияет в целом на эффективность кодирования.

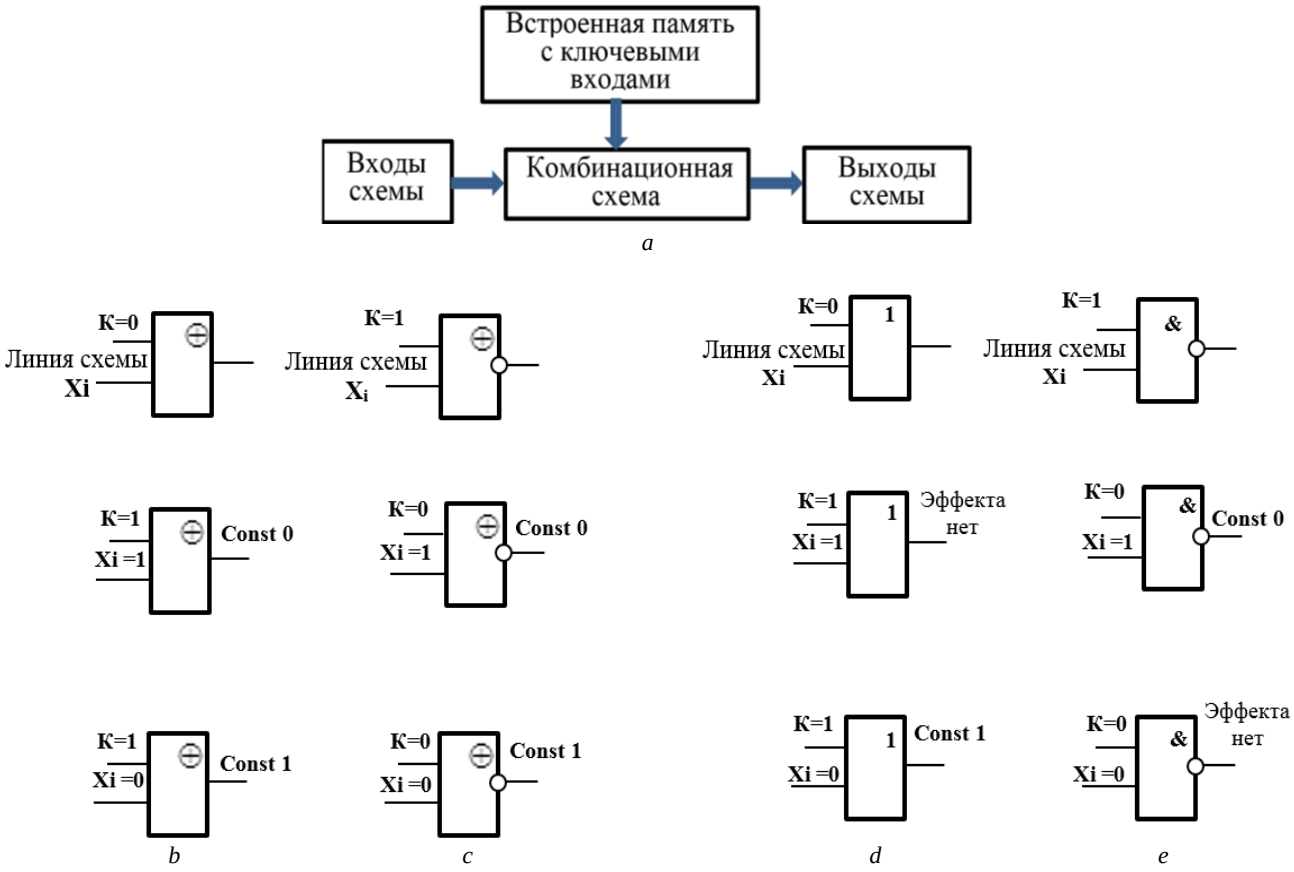

Рис. 1. Логическое кодирование цифровых устройств:
a – общая идея кодирования; *b* – эффекты применения ключевого вентиля XOR-типа; *c* – NXOR; *d* – OR; *e* – NAND
Fig. 1. Logic encoding of digital devices:
a – general idea of coding; b – effects of applying an XOR-type key gate; c – NXOR; d – OR; e – NAND

Кроме типа применяемого вентиля существует еще два основных способа увеличить влияние кодовых вентилей на значения выходов схемы. Один из них заключается в выборе линий, сигналы в которых влияют на максимально возможное количество выходов схемы, второй – в повышении чувствительности схемы в ответ на применение неподлинного ключа.

Выбор линии для включения вентиля в большой степени влияет на эффективность кодирования. Один из подходов основан на случайном выборе линии схемы [4]. В работе [1] показана недостаточная эффективность этого метода. Во-первых, вставка ключевого вентиля в случайно выбранную линию схемы не может гарантировать необходимое расстояние Хэмминга между истинным выходным вектором и полученным в случае применения неподлинного ключа.

В работе [11] для характеристики эффективности выбора линии в схеме для введения ключевого вентиля предложено использовать метрику $M = N_0P_0 * N_0O_0 + N_0P_1 * N_0O_1$, где N_0P_0 (N_0P_1) – количество входных наборов, которые обнаруживают неисправность типа *const* 0 (*const* 1), а N_0O_0 (N_0O_1) – количество ошибочных бит выходного вектора в результате появления неисправности *const* 0 (*const* 1). Данная метрика может быть усовершенствована для получения возможности отслеживать в динамике параметры N_0O_0 (N_0O_1) для анализа неактивированных выходов при кодировании. Использование метрики *M* при кодировании можно сформулировать как нахождение множества неисправностей кодируемой схемы, которые вместе будут влиять на 50% выходных линий при их активизации. Кодирование на основе использования метрики *M* требует моделирования схемы $Q = 2s \times 2^n$ раз, где *s* – общее количество линий схемы (переменных полного состояния схемы), *n* – количество входных переменных схемы. Для примера схемы, рассмотренного ниже, $M = 256 \times 34 = 8\,704$. Для реальных схем подобный подход практически не приемлем по причине высоких вычислительных затрат. В то же врмя с целью оптимизации вычислительных процедур предлагается эвристическое решение – сократить количество моделируемых входных наборов до 1 000 [11].

Основная задача, которая должна быть решена при практической реализации данной общей идеи, заключаются в том, чтобы определить оптимальное множество внутренних линий схемы и количество ключевых элементов для создания максимальных трудностей для злоумышленника по поиску подлинного ключа. При включении очередного вентиля при кодировании логических устройств необходимо проводить анализ на появление эффекта маскирования неисправностей, который способен блокировать эффект кодирования [1. Рис. 2]. При наличии избыточности некоторые линии схемы не могут быть активированы ни одним входным набором, поэтому вставка ключевого вентиля в данном случае может быть бесполезной [1. Рис. 1].

В работе [1] на примере рассмотрена возможность повышения эффективности кодирования за счет включения в структуру схемы вентилей управления, которые позволяют активизировать влияние неподлинного состояния каждого отдельного бита ключа на формирование выходного вектора закодированной схемы. Для того чтобы усилить влияние неподлинного бита кодового слова на результат функционирования схемы, управляющие вентили объединяют биты кодового слова в группы, использовав при этом их выходы в качестве входов ключевых вентилей. В таком случае реализуется групповое воздействие нескольких битов кодового слова на активацию ключевого вентиля. Если хотя бы один из ключевых входов, включенных в группу, принимает неподлинное значение, ключевой вентиль окажется активированным.

2. Контроль надежности кодирования комбинационных схем

В работе [2] предлагается подход SAT-атаки для определения кода аппаратной защиты комбинационных схем цифровых устройств на структурном уровне. Подход основан на сведении задачи к определению выполнимости булевой функции.

Ниже приводится алгоритм декодирования комбинационных схем на основе описания закодированной структуры КНФ-функции разрешения схемы и решения задачи ее выполнимости. Алгоритм проиллюстрирован на примере фрагмента схемы.

Исходными данными для декодирования структуры цифрового устройства являются структурная реализация закодированной схемы, которая может быть получена методом обратного проектирования (проектирования по прототипу), а также активированный физический образец интегральной схемы, в защищенную от несанкционированного доступа память которой заказчик загрузил подлинное значение ключа. Этот образец может использоваться в виде модели черного ящика $Y = eval(X)$. Основная идея SAT-атаки взлома ключа состоит в том, чтобы определить подлинный ключ, не прибегая к исследованиям на большом интервале входно-выходных переменных [2].

Обозначим $\vec{Y} = f(\vec{X})$ – функцию, реализуемую комбинационной схемой с первичными входами $\vec{X}$ и выходами $\vec{Y}$. КНФ-функции разрешения исходной схемы $Cir_a(\vec{X},\vec{Y})$. Сведем задачу получения ключа к описанию закодированной схемы в виде КНФ-представления булевой функции разрешения $Cir_b(\vec{X},\vec{K},\vec{Y})$, где $\vec{X}$ – первичные входы схемы, $\vec{X} = (x_1, x_2, ..., x_n)$; $\vec{K}$ – ключевые входы схемы, $\vec{K} = (k_1, k_2, ..., k_r)$; $\vec{Y}$ – выходные линии схемы, $\vec{Y} = (y_1, y_2, ..., y_m)$.

Если $F = f(\vec{X},\vec{Y})$ – функция, реализуемая исходной схемой, то для любого $\vec{X}$ $F = Cir_b(\vec{X},\vec{K},\vec{Y})$, если применить к закодированной схеме подлинное значение ключа. Цель злоумышленника состоит в том, чтобы найти такой ключ $\vec{K} = (k_1, k_2, ..., k_r)$, чтобы $\forall \vec{X}$ $Cir_b(\vec{X},\vec{K},\vec{Y}) \wedge Cir_a(\vec{X},\vec{Y})$. Но злоумышленник не может получить формулу $Cir_a(\vec{X},\vec{Y})$, так как ему недоступно структурное описание исходной схемы. Не получив доступа к структуре исходной схемы и не имея, таким образом, возможности построить отношение $Cir_a(\vec{X},\vec{Y})$, злоумышленник может наблюдать реакцию схемы на требуемое входное воздействие на активированной ИС, выполнив функцию черного ящика *eval*:

$$\vec{X}_i = (x_1, x_2, ..., x_n) \rightarrow \vec{Y}_i = (y_1, y_2, ..., y_m).$$

Для заданного набора входных векторов $\vec{X}_1, \vec{X}_2, ..., \vec{X}_p$ и соответствующих выходных наблюдений $\vec{Y}_1, \vec{Y}_2, ...\vec{Y}_P$ определение ключевого значения, которое согласуется с этими p наблюдениями, является достаточно простым, если свести задачу к решению выполнимости формулы

$$(\text{SAT}) \ \wedge_{j=1}^{p} Cir_b(\vec{X}_j,\vec{K},\vec{Y}_j).$$

Однако если теперь выполнить новое наблюдение на физическом образце схемы $eval(\vec{X}_s) = \vec{Y}_s$, то нет гарантии, что удовлетворительное присваивание $\vec{K}$ для формулы $\wedge_{j=1}^{p} Cir_b(\vec{X}_j,\vec{K},\vec{Y}_j)$ также будет удовлетворительным присваиванием $\vec{K}$ для формулы $\wedge_{j=p+1}^{2^n} Cir_b(\vec{X}_j,\vec{K},\vec{Y}_j)$.

Для практической атаки при большом числе входных переменных функция *eval* может быть определена только на небольшом числе входных векторов $Cir_b(\vec{X},\vec{K},\vec{Y}) \Leftrightarrow eval(\vec{X}) = \vec{Y}$, в то время как $\exists \vec{K} : \forall \vec{X}$ $Cir_b(\vec{X},\vec{K},\vec{Y}) \wedge Cir_a(\vec{X},\vec{Y})$.

Решение проблемы заключается в том, что вместо поиска подлинного ключа выполняется определение ключа как члена класса эквивалентности ключей, который дает на выходах правильный результат для всех входных состояний.

Определение 1. Два ключа $\vec{K}_1$ и $\vec{K}_2$ являются эквивалентными ($\vec{K}_1 = \vec{K}_2$) тогда и только тогда, когда для входного значения $\vec{X}_i$ закодированная схема выдает одинаковое выходное значение $\vec{Y}_i$ для ключей $\vec{K}_1$ и $\vec{K}_2$.

Для определения подлинного ключа итеративно исключаются ключи из класса эквивалентности, которые выдают неправильные значения выходов по крайней мере для одного входного шаблона. Класс эквивалентных ключей определяется на некотором входно-выходном векторе решением выполнимости функции $Cir_b(\vec{X}_j,\vec{K},\vec{Y}_j)$ полным методом.

Определение 2. Входной вектор $\vec{X}^d$ называется различающим, если реакция схемы при использовании ключа $\vec{K}_1$ равна $\vec{Y}_1^d$ и отличается от реакции $\vec{Y}_2^d$ при использовании ключа $\vec{K}_2$.

При наличии различающего набора можно проверить реакцию активированной схемы для входа $\vec{X}^d$ и использовать ее, чтобы исключить ключ $\vec{K}_1$ или $\vec{K}_2$ как не являющийся подлинным ключом.

Приведем алгоритм для нахождения подлинного ключа из класса эквивалентности:

1 $i := 1$.

2 $F_i = Cir_b(\vec{X}, \vec{K}_1, \vec{Y}_1) \wedge Cir_b(\vec{X}, \vec{K}_2, \vec{Y}_2)$.

3 Если $F_i \wedge \vec{Y}_1 \neq \vec{Y}_2)$ не выполняется, переход к п. 8 – различающий набор не определен.

4 Решение $F_i = Cir_b(\vec{X}, \vec{K}_1, \vec{Y}_1) \wedge Cir_b(\vec{X}, \vec{K}_2, \vec{Y}_2) \wedge (\vec{Y}_1 \neq \vec{Y}_2)$, $\vec{X}_i{}^d := \vec{X}$. Входной набор $\vec{X}_i{}^d$ является различающим.

5 $\vec{Y}_i^d := eval(\vec{X}_i{}^d)$.

6 $i = i + 1$.

7 $F_i = F_{i-1} \wedge Cir_b(\vec{X}_i^d, \vec{K}_1, \vec{Y}_i^d) \wedge Cir_b(\vec{X}_i^d, \vec{K}_2, \vec{Y}_i^d)$. Переход к п. 3.

8 Выход.

Каждая итерация алгоритма исключает хотя бы один неверный член рассматриваемого класса. Это связано с тем, что поиск различающего входного набора ведется с условием $\vec{Y}_1 \neq \vec{Y}_2$, т.е. при одинаковых входных данных выходные должны отличаться для разных ключей. Следовательно, хотя бы один ключ окажется неправильным. Алгоритм завершается, когда определен подлинный ключ из класса эквивалентных ключей.

Для полноты изложения рассмотрим получение функции разрешения F^f. Функция F^f, называемая функцией разрешения для логической функции *f*, зависит не только от аргументов функции *f*, но и от самой *f* и принимает значение логической 1 при всех допустимых состояниях входных и выходной переменных [12]. Функция F^f принимает значение 0 при всех недопустимых состояниях входных и выходной переменных. Приведем функции разрешения F^f и запрета $\overline{F^f}$ в виде таблицы истинности для конъюнкции *f = a*b* (табл. 1).

В табл. 2 приведены КНФ-функций разрешения для некоторых типов вентильных элементов.

Таблица 1

Функции разрешения и запрета

a	***b***	***f***	F^f	$\overline{F^f}$
0	**0**	**0**	1	0
0	**1**	**0**	1	0
1	**0**	**0**	1	0
1	**1**	**1**	1	0
0	0	1	0	1
0	1	1	0	1
1	0	1	0	1
1	1	0	0	1

Таблица 2

Функции разрешения

№ п/п	Однобитовые арифметические и логические уравнения	КНФ-функций разрешения
1	$f = b \vee c$	$(\bar{b} \vee f)(\bar{c} \vee f)(b \vee c \vee \overline{f})$
2	$f = b * c$	$(b \vee \overline{f})(c \vee \overline{f})(\bar{b} \vee \bar{c} \vee f)$
3	$f = a \oplus b$	$(a \vee b \vee \overline{f})(a \vee \bar{b} \vee f)(\bar{a} \vee b \vee f)(\bar{a} \vee \bar{b} \vee \overline{f})$
4	$f = a \sim b$	$(a \vee b \vee f)(a \vee \bar{b} \vee \overline{f})(\bar{a} \vee b \vee \overline{f})(\bar{a} \vee \bar{b} \vee f)$

Рассмотрим пример. На рис. 2 приведены схема (рис. 2, *a*) и вариант ее кодирования, которое выполнено включением дополнительных вентилей XOR (B_1) и NXOR (B_2) (рис. 2, *b*).

Приведем функцию разрешения закодированной схемы $Cir_b(\vec{X},\vec{K},\vec{Y})$. При формировании соответствующей функции разрешения схемы переменные *a*, *b*, *c* – входные переменные, a_1, a_2, a_3, b_1, b_2, c_1 – выходы соответствующих элементов.

$$Cir_b = (a \vee b \vee a_1)(a \vee \overline{b} \vee a_1)(\overline{a} \vee b \vee a_1)(\overline{a} \vee \overline{b} \vee \overline{a_1}) \times$$
$$\times (b \vee c \vee \overline{a_2})(b \vee \overline{c} \vee \overline{a_2})(\overline{b} \vee c \vee \overline{a_2})(\overline{b} \vee \overline{c} \vee a_2) \times$$
$$\times (a \vee c \vee \overline{a_3})(a \vee \overline{c} \vee \overline{a_3})(\overline{a} \vee c \vee \overline{a_3})(\overline{a} \vee \overline{c} \vee a_3) \times$$
$$\times (a_3 \vee k_1 \vee b_2)(a_3 \vee \overline{k_1} \vee \overline{b_2})(\overline{a_3} \vee k_1 \vee \overline{b_2})(\overline{a_3} \vee \overline{k_1} \vee b_2) \times$$
$$\times (k_2 \vee a_1 \vee \overline{b_1})(k_2 \vee \overline{a_1} \vee b_1)(\overline{k_2} \vee a_1 \vee b_1)(\overline{k_2} \vee \overline{a_1} \vee \overline{b_1}) \times$$
$$\times (b_1 \vee a_2 \vee b_2 \vee \overline{c_1})(b_1 \vee a_2 \vee \overline{b_2} \vee c_1)(b_1 \vee \overline{a_2} \vee b_2 \vee c_1)(b_1 \vee \overline{a_2} \vee \overline{b_2} \vee c_1) \times$$
$$\times (\overline{b_1} \vee a_2 \vee b_2 \vee c_1)(\overline{b_1} \vee a_2 \vee \overline{b_2} \vee c_1)(\overline{b_1} \vee \overline{a_2} \vee b_2 \vee c_1)(\overline{b_1} \vee \overline{a_2} \vee \overline{b_2} \vee c_1).$$

1. В качестве входного вектора для поиска ключей используем случайный вектор $\vec{X} = 110$ для которого определяем $\vec{Y}$ с помощью активированной схемы: $eval(\vec{X}) = 1$.

2. Найдем решение задачи SAT для функции $F = Cir_b ab\overline{c}\overline{c_1}$ на основе полного алгоритма решения выполнимости:

$$F = \overline{a_1}\,\overline{a_2}\,\overline{a_3}(k_1 \vee b_2)(\overline{k_1} \vee \overline{b_2})(k_2 \vee \overline{b_1})(\overline{k_2} \vee b_1)ab\overline{c}\overline{c_1}.$$

Функция выполнима при следующих условиях:

а) $F = k_1 k_2 \overline{a_1}\,\overline{a_2}\,\overline{a_3} b_1 \overline{b_2} c_1$; $\vec{K}_1 = 11$;

б) $F = \overline{k_1} k_2 \overline{a_1}\,\overline{a_2}\,\overline{a_3} b_1 b_2 c_1$; $\vec{K}_2 = 01$;

в) $F = \overline{k_1}\,\overline{k_2}\,\overline{a_1}\,\overline{a_2}\,\overline{a_3}\,\overline{b_1} b_2 c_1$; $\vec{K}_3 = 00$.

Таким образом, определены три ключа, $\vec{K}_1 = 11$, $\vec{K}_2 = 01$, $\vec{K}_3 = 00$, которые составляют класс эквивалентных на данном этапе декодирования.

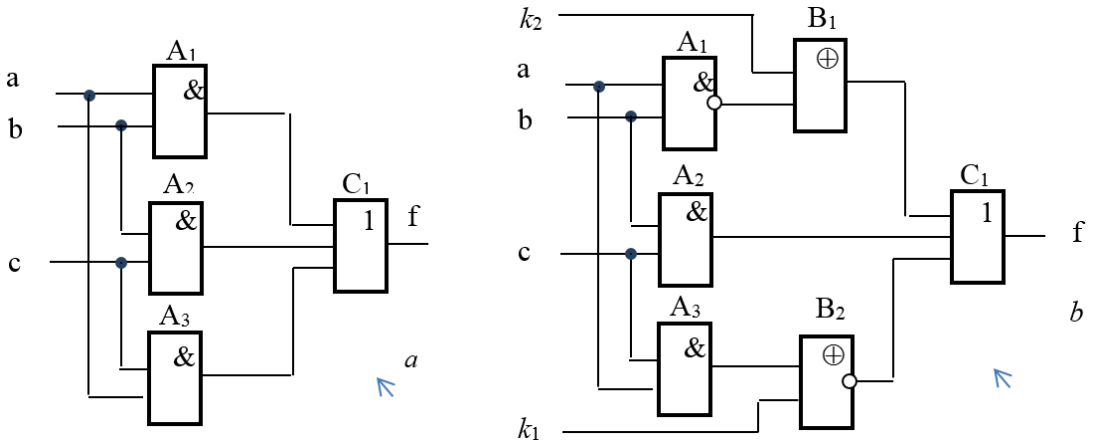

Рис. 2. Комбинационная схема для иллюстрации алгоритма взлома ключа:
a – исходная схема; *b* – закодированная схема
Fig. 2. Combination scheme to illustrate the key cracking algorithm:
a – original schema; b – coded scheme

3. Определим различающий входной набор для первых двух ключей $\vec{K}_1 = 11$ и $\vec{K}_2 = 01$ из найденного класса. Для этого необходимо определить выполнимость функции

$$F_1 = Cir_b(\vec{X},\vec{K}_1,\vec{Y}_1) \wedge Cir_b(\vec{X},\vec{K}_2,\vec{Y}_2). \quad (1)$$

Для решения (1) определяем один из выполнимых входно-выходных векторов для первого ключа $\vec{K}_1 = 11$. Задача решается на основе неполного алгоритма выполнимости функции

$$F = Cir_b k_1 k_2 . \quad (2)$$

Получим $F = a\bar{b}ck_1k_2a_1\overline{a_2}a_3\overline{b_1}b_2c_1$. Таким образом, определены новые входной $\vec{X} = 101$ и выходной $\vec{Y}$ = 1 векторы. Проверим выполнимость функции $F = Cir_b a\bar{b}c\overline{k_1}k_2\overline{c_1}$ на полученном входном наборе при значении выходного вектора, отличном от полученного в (2), для второго ключа $\vec{K}_2 = 01$. В результате $F = a\bar{b}c\overline{k_1}k_2a_1\overline{a_2}a_3\overline{b_1}\overline{b_2}\overline{c_1}$.

Следовательно, $\vec{X} = 101$ является различающим входным набором, так как разным ключам соответствуют разные выходы.

4. Определим вектор $\vec{Y} \Rightarrow \vec{X} = 101$, $\vec{Y} = eval(\vec{X}) = 1$ с помощью активированной схемы.

5. Вычислим функцию (1) для $\vec{K}_1 = 11$: $F_{K_1} = Cir_b a\bar{b}ck_1k_2c_1$; для $\vec{K}_2 = 01$: $F_{K_2} = Cir_b a\bar{b}c\overline{k_1}k_2c_1$. Функция $F_{K_2} = Cir_b a\bar{b}c\overline{k_1}k_2c_1$ не выполняется. Следовательно, ключ $\vec{K}_2 = 01$ исключается из класса эквивалентности.

6. Вычислим функцию (1) для $\vec{K}_3 = 00$: $F_{K_3} = Cir_b a\bar{b}c\overline{k_1}\overline{k_2}c_1$. Функция F_{K_3} не выполняется, так как ключ $\vec{K}_3 = 00$ не является подлинным.

Поскольку ключи $\vec{K}_2 = 01$ и $\vec{K}_3 = 00$ оказались неверными, подлинным ключом является единственный ключ, оставшийся в классе эквивалентности ключей, – $\vec{K} = 11$.

Заключение

В работе рассмотрены некоторые особенности кодирования структурной реализации проекта интегральной схемы на основе использования средств тестового диагностирования.

Для оценки надежности кодирования предлагается алгоритм декодирования, основанный на решении SAT КНФ-функции разрешения, описывающей закодированную структуру. Проиллюстрированный на примере алгоритм нахождения правильного ключа из множества ключей класса эквивалентности направлен на решение проблемы декодирования схем практических размеров. Эффективность практического применения данного алгоритма зависит от эффективности этапа поиска различающего входного набора. При выполнении данного этапа целесообразно использовать неполные алгоритмы выполнимости, которые осуществляют поиск выполняющего набора неполным перебором пространства возможных решений. Основное достоинство таких алгоритмов – высокая скорость работы. Однако если алгоритм не находит решения за приемлемое время, применяется полный алгоритм решения выполнимости.

Список источников

1. Золоторевич Л.А. Аппаратная защита цифровых устройств // Вестник Томского государственного университета. Управление, вычислительная техника, информатика. 2020. № 50. С. 69–78.
2. Subramanyan P., Ray S., Malik S. Evaluating the security of logic encryption algorithms // IEEE International Symposium on Hardware Oriented Security and Trust (HOST). 2015. P. 137–143.
3. Rajendran J., Sam M., Sinanoglu O., Karri R. Security analysis of integrated circuit camouflaging // ACM SIGSAC Conference on Computer & Communications Security. Germany, Berlin. 04–08 November 2013. P. 709–720.
4. Roy J.A., Koushanfar F., Markov I.L. EPIC: Ending piracy of integrated circuits // IEEE Computer. 2010. V. 43, № 10. P. 30–38.
5. Yasin M., Rajendran J., Sinanoglu O., Karri R. On improving the security of logic locking // IEEE TCAD. 2016. V. 35, № 9. P. 1411–1424.
6. Rajendran J., Pino Y., Sinanoglu O., Karri R. Logic encryption: a fault analysis perspective // Proc. IEEE/ACM DATE. 2012. P. 953–958.
7. Rajendran J. et al. Fault analysis-based logic encryption // IEEE Trans. Comput. 2015. V. 64, № 2. P. 410–424.
8. Dupuis S., Ba P., Natale G.D., Flottes M., Rouzeyre B. A novel hardware logic encryption technique for thwarting illegal overproduction and hardware trojans // IEEE 20th International On-Line Testing Symposium (IOLTS). 2014. P. 49–54.
9. Plaza S.M., Markov I.L. Solving the third-shift problem in IC piracy with test-aware logic locking // IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 2015. V. 34, № 6. P. 961–971.

10. Lee Y.-W., Touba N. Improving logic obfuscation via logic cone analysis // Proc. Latin-American Test Symposium. 2015. P. 1–6.
11. Karousos N., Pexaras K., Karybali I.G., Kalligeros E. Weighted logic locking: A new approach for ic piracy protection // IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS). 2017. P. 221–226.
12. Zolotorevich, L.A. Issledovaniye metodov i sredstv verifikatsii proyektov i generatsii testov MES / L.A. Zolotorevich // Sbornik nauchnykh trudov vserossiyskoy nauchno-tekhnicheskoy konferentsii "Problemy razrabotki perspektivnykh mikroelektronnykh sistem – MES-2006". Red. A.L. Stempkovsky. M.: IPPM RAN. 2006. P. 163–168.

References

1. Frolov, I. (2018) *ER-diagramma – eto... Opisanie, vidy, pravila postroeniya* [ER diagram is... Description, types, construction rules]. [Online] Available from: https://fb.ru/article/436948/er-diagramma---eto-description-vidyi-pravila-postroeniya (Accessed: 12th January 2022).
2. Efimenko, V. (2019) *Uchimsya proektirovaniyu Entity Relationship – diagramm* [Learning to design Entity Relationship diagrams]. [Online] Available from: https://habr.com/ru/post/440556/ (Accessed: 12th January 2022).
3. Watt, A. & Eng, N. (2014) *Database Design*. 2nd ed. BCcampus.
4. Shvetsov, V. (n.d.) *Bazy dannykh. Pervaya stadiya kontseptual'nogo proektirovaniya bazy dannykh (kontseptual'noe modelirovanie)* [Databases. The first stage of the conceptual design of the database (conceptual modeling)]. [Online] Available from: https://intuit.ru/studies/courses/508/364/lecture/8647 (Accessed: 12th January 2022).
5. Kara-Ushanov, V.Yu. (2017) *Model' "Sushchnost'–Svyaz'"*[Model "Entity - Relationship"]. Yekaterinburg: Ural Federal University.
6. Inf-teh-lotos.ru. (n.d.) *Sozdanie ER-Diagramm. Informatsionnye tekhnologii* [Information technology. Creation of ER Diagrams]. [Online] Available form: http://inf-teh-lotos.ru/sozdanie-er-diagramm (Accessed: 12th January 2022).
7. Chen, P. (1976) The entity-relationship model – towards a unified view of data. *ACM Transactions on Database Systems*. 1(1). pp. 9–36. DOI: 10.1145/320434.320440
8. Khrustalev, E.Yu. & Baranova, N.M. (2013) Intelligent semantic models for improving the quality of educational and research processes. *Ekonomicheskiy analiz: teoriya i praktika – Economic Analysis: Theory and Practice*. 35(338). pp. 2–10.
9. Khusainova, G.Ya. (2017) Metodika postroeniya ER-diagrammy dlya bazy dannykh [Technique for constructing an ER diagram for a database]. *NovaInfo*. 76-1. pp. 322–327.
10. Nguyen Kim Anh. (2009) *Data Modeling Using Entity-Relationship Model*. [Online] Available from: https://cnx.org/contents/aM2VU eRT@1/Data-Modeling-Using-Entity-Relationship-Model (Accessed: 12th January 2022).
11. Tsichritzis, D. & Lochovsky, F. (1982) *Data Models*. Prentice Hall Inc.
12. Babanov, A.M. (2011) Semantic method of database designing and its prospects opening with application of the erm data model. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science.* 16(3). pp. 58–66.

***Информация об авторах*:**

Золоторевич Людмила Андреевна – доцент, кандидат технических наук, доцент кафедры электронных вычислительных машин Белорусского государственного университета информатики и радиоэлектроники (Минск, Республика Беларусь). E-mail: lazolotorevich@gmail.com

Ильинков Валерий Андреевич – доцент, кандидат технических наук, доцент кафедры инфокоммуникационных технологий Белорусского государственного университета информатики и радиоэлектроники (Минск, Республика Беларусь). E-mail: v.ilyinkov@gmail.com

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

***Information about the authors*:**

Zolotorevich Ludmila A. (Candidate of Technical Sciences, Associate Professor, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus)**.** E-mail: lazolotorevich@gmail.com

Ilyinkov Valery A. (Candidate of Technical Sciences, Associate Professor, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus)**.** E-mail: v.ilyinkov@gmail.com

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

Поступила в редакцию 31.01.2022; принята к публикации 30.08.2022

Received 31.01.2022; accepted for publication 30.08.2022