Предотвращение аппаратных атак при проектировании и изготовлении интегральных схем

Золоторевич Л. А.[1], Ильинков В. А.[2]
[1]*zolotorevichla@bsuir.by,* [2]*v.ilyinkov@gmail.com*
[12]*Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь*

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 является структурная реализация закодированной схемы, полученная методом обратного проектирования (проектирования по прототипу), а также активированный физический образец интегральной схемы, в защищенную от несанкционированного доступа память которой загружено правильное значение ключа. Этот образец используется 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

Prevention of hardware attacks in the design and manufacture of integrated circuits

Zolotorevich L. A.[1], Ilyinkov V. A.[2]
[12]*Belarussian State University of Informatics and Radioelectronics, Minsk, Belarus*

The features and reliability of logical coding of combinational circuits are considered. An algorithm for cracking the code of combinational schemes is proposed, based on the description of the encoded structure by the resolution function and reducing the problem to KNF-feasibility. The initial data for decoding the structure of a digital device is the structural implementation of the encoded circuit obtained by reverse engineering (prototyping), as well as an activated physical sample of an integrated circuit, in whose memory the correct key value is loaded, protected from unauthorized access. This sample is used as a black box model. The main idea of cracking a key 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, feasibility of the KNF-function

Введение

Серьезной проблемой для электронной и оборонной промышленности в последние годы стало пиратство, перепроизводство и контрафакция, что привело к необходимости защиты проектов СБИС и систем на кристалле (СнК) от несанкционированного вмешательства в цикл проектирования и (или) производства интегральных схем (ИС) [1]. По оценкам Technology Information Handling Services, финансовый риск из-за контрафактных и несанкционированных микросхем оценивается более чем в 169 млрд долл. в год, что примерно в 10 раз превышает ущерб от пиратства в области программного обеспечения [2]. Для

оборонной промышленности важнейшей проблемой является возможность использования контрафактных ИС с модифицированными функциями, что в определенное время может деструктивно повлиять на функционирование структуры, ухудшить ее эксплуатационные характеристики, привести к раскрытию конфиденциальной информации и др. Кроме больших финансовых потерь, существует реальная проблема обеспечения национальной безопасности, так как 15 % ИС 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 и 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 ИС. 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 ИС применяются методы и средства аппаратного проектирования и производства.

Рассматриваются вопросы аппаратной защиты на основе логического кодирования структурных схем цифровых устройств комбинационного типа. Для оценки качества защиты предлагается способ 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. Контроль надежности кодирования комбинационных схем

В работе [2] проанализированы различные модели процесса злонамеренного искажения проекта, описывающие условия, при которых подобное искажение может внедриться в цифровую систему. Одним из методов борьбы с вышеупомянутыми угрозами является логическое кодирование, которое обеспечивает доступ к объекту только авторизованным пользователям [4]. Метод предполагает сокрытие функциональности проекта и использование ключа, применение которого выводит систему в область правильного функционирования.

Основная идея кодирования состоит в том, чтобы изменить конструкцию ИС, добавив в нее дополнительные логические элементы и новые входы, называемые ключевыми, т. е. применить метод 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 правильного ключа. Ключевые входы подсоединяются к защищенной от несанкционированного доступа памяти, а закодированная схема будет работать правильно только в том случае, если поданы верные значения на ее ключевые входы. 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 правильные значения ключевых входов. В противном случае поиск правильного ключа должен быть для злоумышленника затруднителен.

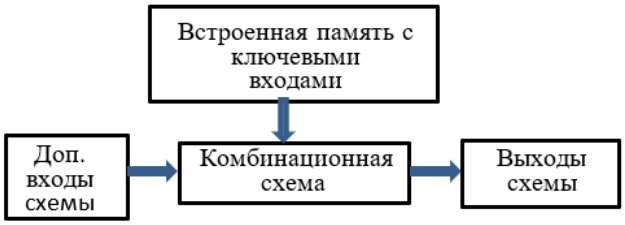

Рис. 1. Общая идея логического кодирования

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 правильного 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 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 не правильного ключа можно сравнить

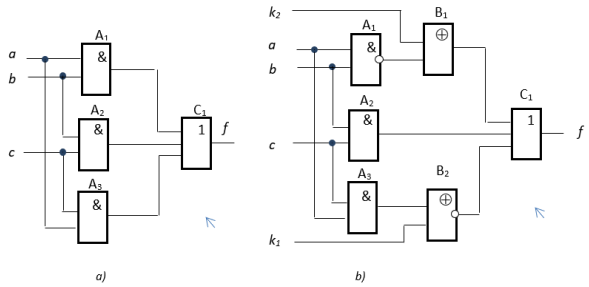

Рис. 2. Комбинационная схема для иллюстрации алгоритма взлома ключа: a) исходная; b) закодированная

с влиянием неисправности константного типа на данной линии (Рис. 1). При выборе в качестве ключевых вентилей XOR или NXOR применение не правильного ключа приводит к появлению неисправности константного типа в любом случае, на любом входном воздействии, в отличие от вентилей OR, NOR, AND, NAND, что влияет в целом на эффективность кодирования. 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 не правильного ключа.

Выбор линии для включения вентиля в большой степени влияет на эффективность кодирования. Один из подходов основан на случайном выборе линии схемы [11]. В работе [12] 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 $\mathrm{const}\,0(\mathrm{const}\,1)$, а $N_0O_0(N_0O_1)$— количество ошибочных бит выходного вектора в результате появления неисправности $\mathrm{const}\,0(\mathrm{const}\,1)$.

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.

Исходными данными для декодирования структуры цифрового устройства является структурная реализация закодированной схемы, полученной, например, методом обратного проектирования (проектирования по прототипу), а также активированный физический образец ИС, в защищенную от несанкционированного доступа память которой заказчик загрузил правильное 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 правильный ключ, не прибегая к исследованиям на большом интервале входных (выходных) переменных [2].

Обозначим $\boldsymbol{Y} = f(\boldsymbol{X})$ функцию, реализуемую комбинационной схемой с первичными входами $\boldsymbol{X}$ и выходами $\boldsymbol{Y}$, а $Cir_0(\boldsymbol{X}, \boldsymbol{Y})$— КНФ функции разрешения исходной схемы. Сведем задачу получения ключа к описанию закодированной схемы в виде КНФ-представления булевой функции разрешения $Cir_b(\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{Y})$, где $\boldsymbol{X}$— первичные входы схемы, $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$; $\boldsymbol{K}$— ключевые входы схемы,$\boldsymbol{K} = (k_1, k_2, \ldots, k_r)$; $\boldsymbol{Y}_i$— выходные линии схемы, $\boldsymbol{Y} = (y_1, y_2, \ldots, y_m)$.

Если $F = f(\boldsymbol{X}, \boldsymbol{Y})$ — функция, реализуемая исходной схемой, то для любого $\boldsymbol{X} F = Cir_b(\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{Y})$, если применить к закодированной схеме правильное значение ключа. Цель злоумышленника состоит в том, чтобы найти такой ключ $\boldsymbol{K} = (k_1, k_2, \ldots, k_r)$, при котором $\forall \boldsymbol{X} \quad Cir_b(\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{Y}) \wedge Cir_0(\boldsymbol{X}, \boldsymbol{Y})$. Однако злоумышленник не может получить формулу $Cir_0(\boldsymbol{X}, \boldsymbol{Y})$, так как для него недоступно структурное описание исходной схемы. Не получив доступ к структуре исходной схемы и не имея, таким образом, возможности построить отношение $Cir_0(\boldsymbol{X}, \boldsymbol{Y})$, злоумышленник может наблюдать реакцию схемы на требуемое входное воздействие по активированной ИС, выполнив функцию черного ящика $eval$:

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

Для заданного набора входных векторов $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_p$ и соответствующих выходных наблюдений $\boldsymbol{Y}_i$, $\boldsymbol{Y}_2$, $\ldots$, $\boldsymbol{Y}_P$ определение ключевого значения, которое согласуется с p наблюдениями, является достаточно простым, если свести задачу к решению выполнимости формулы $\wedge_{j=1}^{p} Cir_b(\boldsymbol{X}_j, \boldsymbol{K}, \boldsymbol{Y}_j)$. Однако если теперь выполнить новое наблюдение на физическом образце схемы $eval(\boldsymbol{X}_s) = \boldsymbol{Y}_s$, то нет гарантии, что удовлетворительное присваивание $\boldsymbol{K}$ для формулы $\wedge_{j=1}^{p} Cir_b(\boldsymbol{X}_j, \boldsymbol{K}, \boldsymbol{Y}_j)$ также будет удовлетворительным присваиванием $\boldsymbol{K}$ для формулы $\wedge_{j=p+1}^{2^n} Cir_b(\boldsymbol{X}_j, \boldsymbol{K}, \boldsymbol{Y}_j)$.

Для практической атаки при большом числе входных переменных функция $eval$ может быть определена только на небольшом числе входных векторов $Cir_b(\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{Y}) \Leftrightarrow eval(\boldsymbol{X}) = \boldsymbol{Y}$, в то время как $\exists \boldsymbol{K} : \ \forall \boldsymbol{X} \quad Cir_b(\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{Y}) \wedge Cir_0(\boldsymbol{X}, \boldsymbol{Y})$.

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

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

Для определения правильного 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 входном (выходном) векторе путем решения выполнимости функции $Cir_b(\boldsymbol{X}_j, \boldsymbol{K}, \boldsymbol{Y}_j)$полным методом.

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

При наличии различающего набора можно проверить реакцию активированной схемы для входа $\boldsymbol{X}^d$ и использовать ее, чтобы исключить ключ $\boldsymbol{K}_1$ или $\boldsymbol{K}_2$ как не входящий в класс эквивалентности правильных ключей.

Ниже приводится алгоритм нахождения входного различающего набора:

1. $i := 1$.
2. $F_i = Cir_b(\boldsymbol{X}, \boldsymbol{K}_1, \boldsymbol{Y}_1) \wedge Cir_b(\boldsymbol{X}, \boldsymbol{K}_2, \boldsymbol{Y}_2)$.
3. Если $F_i \wedge \boldsymbol{Y}_1 \neq \boldsymbol{Y}_2$ не выполняется, переход к п. 8.— различающий набор не определен.
4. Решение $F_i = Cir_b(\boldsymbol{X}, \boldsymbol{K}_1, \boldsymbol{Y}_1) \wedge Cir_b(\boldsymbol{X}, \boldsymbol{K}_2, \boldsymbol{Y}_2) \wedge (\boldsymbol{Y}_1 \neq \boldsymbol{Y}_2)$, $\boldsymbol{X}_i{}^d := \boldsymbol{X}$. Входной набор $\boldsymbol{X}_i{}^d$ является различающим.
5. $\boldsymbol{Y}_i^d := eval(\boldsymbol{X}_i{}^d)$.
6. $i = i + 1$.
7. $F_i = F_{i-1} \wedge Cir_b(\boldsymbol{X}_i^d, \boldsymbol{K}_1, \boldsymbol{Y}_i^d) \wedge Cir_b(\boldsymbol{X}_i^d, \boldsymbol{K}_2, \boldsymbol{Y}_i^d)$, переход к п. 3.
8. Выход.

Каждая итерация алгоритма исключает хотя бы один неверный член рассматриваемого класса эквивалентности ключей. Это связано с тем, что поиск различающего входного набора ведется с условием $\boldsymbol{Y}_1 \neq \boldsymbol{Y}_2$, т. е. при одинаковых входных данных выходные данные должны отличаться для разных ключей. Следовательно, хотя бы один ключ окажется неправильным. Алгоритм завершается, когда определен правильный ключ из класса эквивалентных ключей. В работах [13,14] рассматриваются некоторые вопросы реализации алгоритмов кодирования и декодирования структурных реализаций комбинационных схем, применяемых при аппаратной защите цифровых СБИС.

Покажем применение алгоритма на примере фрагмента схемы. На Рис. 2a изображена схема и вариант ее кодирования, которое выполнено путем включения дополнительных вентилей B_1 XOR и B_2 NXOR (Рис. 2b).

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

$$\begin{aligned} Cir_b =& (a \vee b \vee a_1)(a \vee \overline{b} \vee a_1)(\overline{a} \vee b \vee a_1)(\overline{a} \vee \overline{b} \vee \overline{a_1}),\\ & (b \vee c \vee \overline{a_2})(b \vee \overline{c} \vee \overline{a_2})(\overline{b} \vee c \vee \overline{a_2})(\overline{b} \vee \overline{c} \vee a_2),\\ & (a \vee c \vee \overline{a_3})(a \vee \overline{c} \vee \overline{a_3})(\overline{a} \vee c \vee \overline{a_3})(\overline{a} \vee \overline{c} \vee a_3),\\ & (a_3 \vee k_1 \vee b_2)(a_3 \vee \overline{k_1} \vee \overline{b_2})(\overline{a_3} \vee k_1 \vee \overline{b_2})(\overline{a_3} \vee \overline{k_1} \vee b_2),\\ & (k_2 \vee a_1 \vee \overline{b_1})(k_2 \vee \overline{a_1} \vee b_1)(\overline{k_2} \vee a_1 \vee b_1)(\overline{k_2} \vee \overline{a_1} \vee \overline{b_1}),\\ & (b_1 \vee a_2 \vee b_2 \vee \overline{c_1})(b_1 \vee a_2 \vee \overline{b_2} \vee c_1)(b_1 \vee \overline{a_2} \vee b_2 \vee c_1)(b_1 \vee \overline{a_2} \vee \overline{b_2} \vee c_1),\\ & (\overline{b_1} \vee a_2 \vee b_2 \vee c_1)(\overline{b_1} \vee a_2 \vee \overline{b_2} \vee c_1)(\overline{b_1} \vee \overline{a_2} \vee b_2 \vee c_1)(\overline{b_1} \vee \overline{a_2} \vee \overline{b_2} \vee c_1). \end{aligned}$$

Для декодирования выполним следующие действия:

1. В качестве входного вектора для поиска ключей используем случайный вектор $\boldsymbol{X} = 110$, для которого определим $\boldsymbol{Y}$ с помощью активированной схемы: $eval(\boldsymbol{X}) = 1$.
2. Найдем решение задачи SAT для функции $F = Cir_b ab\overline{c}c_1$ на основе полного алгоритма решения выполнимости:

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

Функция выполнима при следующих условиях:

$$\begin{aligned} F &= k_1 k_2 \overline{a_1 a_2 a_3} b_1 \overline{b_2} c_1, \boldsymbol{K}_1 = 11;\\ F &= \overline{k_1} k_2 \overline{a_1 a_2 a_3} b_1 b_2 c_1, \boldsymbol{K}_2 = 01;\\ F &= \overline{k_1 k_2} \overline{a_1 a_2 a_3} \overline{b_1} b_2 c_1, \boldsymbol{K}_3 = 00. \end{aligned}$$

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

3. Найдем различающий входной набор для первых двух ключей $\boldsymbol{K}_1 = 11$ и $\boldsymbol{K}_2 = 01$ из найденного класса. Для этого необходимо вычислить булеву функцию

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

Для решения равенства (1) определим один из выполнимых входных (выходных) векторов для первого ключа$\boldsymbol{K}_1 = 11$. 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 новый входной $\boldsymbol{X} = 101$ и выходной $\boldsymbol{Y} = 1$ векторы. Проверим выполнимость функции $F = Cir_b a\bar{b}c\overline{k_1}k_2\overline{A_1}$ на входном наборе $\boldsymbol{X} = 101$ при значении выходного вектора, отличного от полученного в равенстве (2) для второго ключа $\boldsymbol{K}_2 = 01$. В результате $F = a\bar{b}c\overline{k_1}k_2a_1\overline{a_2}a_3\overline{b_1}b_2\overline{c_1}$.

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

4. Определим вектор $\boldsymbol{Y} \Rightarrow \overrightarrow{X} = 101$, $\overrightarrow{Y} = eval(\boldsymbol{X}) = 1$ с помощью активированной схемы.
5. Вычислим функцию (1): для $\boldsymbol{K}_1 = 11$ $F_{K_1} = Cir_b a\bar{b}ck_1k_2c_1$, для$\boldsymbol{K}_2 = 01$ $F_{K_2} = Cir_b a\bar{b}c\overline{k_1}k_2c_1$. Функция $F_{K_2} = Cir_b a\bar{b}c\overline{k_1}k_2c_1$ не выполняется. Следовательно, ключ $\boldsymbol{K}_2 = 01$ исключается из класса эквивалентности.
6. Вычислим функцию (1) для $\boldsymbol{K}_3 = 00$: $F_{K_3} = Cir_b a\bar{b}c\overline{k_1k_2}c_1$. Функция F_{K_3} не выполняется, так как ключ $\boldsymbol{K}_3 = 00$ неправильный.

В связи с тем что ключи $\boldsymbol{K}_2 = 01$ и $\boldsymbol{K}_3 = 00$ оказались неверными, правильным является единственный ключ, оставшийся в классе эквивалентности правильных ключей, $\boldsymbol{K} = 11$.

Заключение

В работе рассмотрены некоторые особенности кодирования структурной реализации ИС на основе использования средств тестового диагностирования.

Для оценки надежности кодирования предлагается алгоритм декодирования, который проиллюстрирован на примере. Анализ надежности кодирования основан на решении SAT КНФ-функции разрешения, описывающей закодированную структуру.

Метод нахождения правильного ключа из класса эквивалентности предназначен для решения проблемы декодирования схем практических размеров без необходимости исследовать всю область возможных решений.

Список литературы

[1] *Золоторевич Л. А.* Аппаратная защита цифровых устройств // Вестник Томского гос. ун-та. Управление, вычислительная техника, информатика.— 2020.— № 50. — С. 69–78.— https://doi.org/10.17223/19988605/50/9.

[2] *Subramanyan P., Ray S., Malik S.* Evaluating the security of logic encryption algorithms // 2015 IEEE Intern. Symp. on Hardware Oriented Security and Trust (HOST), Washington, DC, USA, 5–7 May 2015.— Washington, 2015.— P. 137–143.

[3] *Rajendran J.* Security analysis of integrated circuit camouflaging // ACM SIGSAC Conf. on Computer & Communications Security, Berlin, Germany, 04–08 Nov. 2013.— Berlin, 2013.— P. 709–720.

[4] *Roy J. A., Koushanfar F., Markov I. L.* EPIC: Ending piracy of integrated circuits // IEEE Computer.— 2010.— Vol. 43, no. 10.— P. 30–38.

[5] *Yasin M.* On improving the security of logic locking // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.— 2016.— Vol. 35, no. 9.— P. 1411–1424.

[6] *Rajendran J.* Logic encryption: a fault analysis perspective // DATE–12 : Proc. of the Conf. on Design, Automation and Test in Europe, Dresden, Germany, March, 2012.— Dresden, 2012.— P. 953–958.

[7] *Rajendran J.* Fault analysis-based logic encryption // IEEE Transactions on Computers.— 2015.— Vol. 64, no. 2.— P. 410–424.

[8] *Dupuis S.* A novel hardware logic encryption technique for thwarting illegal overproduction and hardware trojans // 20th IEEE Intern. On-Line Testing Symp., Platja d'Aro, Catalunya, Spain, 7–9, July 2014.— Platja d'Aro, 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems.— 2015.— Vol. 34, no. 6.— P. 961–971.

[10] *Lee Y. W., Touba N.* Improving logic obfuscation via logic cone analysis // Proc. Latin-American Test Symp., Puerto Vallarta, Mexico, 25–27 March 2015.— Puerto Vallarta, 2015.— P. 1–6.

[11] *Roy J. A., Koushanfar F., Markov I. L.* Ending piracy of integrated circuits // IEEE Computer.— 2010.— Vol. 43, no. 10.— P. 30–38.

[12] *Karousos N* Weighted logic locking: a new approach for IC piracy protection //IEEE 23rd Intern. Symp. on On-Line Testing and Robust System Design (IOLTS), Thessaloniki, Greece, 3–5 July 2017.— Thessaloniki. 2017.— P. 221–226.

[13] *Золоторевич Л. А.* Исследование методов и средств верификации проектов и генерации тестов МЭС // Сб. науч. тр. Всерос. науч.-техн. конф. «Проблемы разработки перспективных микроэлектронных систем» (МЭС–2006) / под общ. ред. А. Л. Стемпковского. — М. : ИППМ РАН, 2006. — С. 163–168.

[14] *Zolotorevich L. A.* Project verification and construction of superchip tests at the RTL levelh // Automation and Remote Control.— 2013.— Vol. 74, Iss. 1.— P. 113–122.