ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

LOGICAL DESIGN

УДК 519.873:519.718.7

Поступила в редакцию 03.01.2019
Received 03.01.2019

Принята к публикации 11.03.2019
Accepted 11.03.2019

Обфускация комбинационных схем цифровых устройств от несанкционированного доступа

Л. А. Золоторевич

Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь
E-mail: zolotorevichLA@bsuir.by

Аннотация. Анализируются проблемы проектирования современных СБИС и систем на кристалле. Наиболее сложными из них являются проблемы верификации проектов на разных этапах проектирования. Наряду с задачами, которые возникают и решаются в режиме благоприятствующего проектирования, в последнем десятилетии возникла необходимость защиты и дополнительного контроля проектов с целью обнаружения несанкционированного стороннего вмешательства в проект.
Рассматриваются вопросы формирования общего подхода к решению задач контроля и верификации при проектировании современных интегральных схем, основанного на анализе моделей неисправностей структурных реализаций цифровых устройств комбинационного типа; ошибок, возникающих в процессе проектирования, а также преднамеренных искажений на этапах проектирования и изготовления, т. е. вопросы создания и развития таксономии возможных отклонений в проекте.
Предлагается алгоритм логической обфускации и кодирования цифровых устройств на основе применения методов и средств тестового диагностирования.

Ключевые слова: СБИС, таксономия отклонений, искажение функций проектов, моделирование неисправностей, кодирование устройства, обфускация

Для цитирования. Золоторевич, Л. А. Обфускация комбинационных схем цифровых устройств от несанкционированного доступа / Л. А. Золоторевич // Информатика. – 2019. – Т. 16, № 3. – С. 89–100.

Obfuscation of combination circuits of digital devices from unauthorized access

Lyudmila A. Zolotorevich

Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
E-mail: zolotorevichLA@bsuir.by

Abstract. The problems of designing modern VLSI and SoC are analyzed. The most difficult problems of design are problems of verification of projects at different stages of design. Along with the problems that arise and are solved in the mode of favorable design, in the last decade there was a problem of protection and additional control of projects in order to detect unauthorized third-party interference in the project with different fundamental goals.
We consider the formation of a common approach to solving problems of control and verification in the design of modern integrated circuits based on the analysis of fault models of structural realizations of digital devices, errors

arising in the design process, as well as deliberate distortions during the design and manufacturing stages, i. e. creation and development of taxonomy of possible deviations in the project.
The algorithm of logical obfuscation and coding of digital structures based on the use of methods and means of test diagnostics is proposed.

Keywords: VLSI, taxonomy of deviations, distortion of functions of projects, modeling of faults, devices coding, obfuscation

For citation. Zolotorevich L. A. Obfuscation of combination circuits of digital devices from unauthorized access. *Informatics*, 2019, vol. 16, no. 3, pp. 89–100 (in Russian).

Введение. Акцент на цифровую экономику, приоритетность разработок цифровых технологий требуют постоянного совершенствования теории и практики проектирования интегральных схем и систем на кристалле (СнК) как технической базы создания встраиваемых электронных систем. Совершенствование технологий СБИС и СнК существенным образом зависит от развития методов применяемых систем автоматизированного проектирования (САПР) и повышения их качества. Разработка САПР микроэлектроники началась вместе с появлением первых интегральных схем в 1958 г. Ежегодно проводится большое число международных симпозиумов и семинаров по разным аспектам теории и практики автоматизированного проектирования. Важнейшими из них являются задачи обеспечения контроля, верификации, построения тестов контроля функциональных блоков и систем. Наукоемкость решения указанных задач постоянно возрастает из-за увеличения сложности проектируемых объектов, отсутствия общего подхода к рассмотрению ошибок, вносимых в проект при проектировании, неисправностей реальных объектов, корреляции разного типа ошибок проектирования и неисправностей структурных реализаций. Все проблемы, связанные с разработкой методов и созданием средств верификации проектов и построения тестов контроля объектов в разных классах неисправностей, систем функционального контроля, являются достаточно сложными, но естественными. Они возникают непреднамеренно и должны решаться в режиме благоприятствующего проектирования. Вместе с тем в последние годы возникла потребность в дополнительном контроле проектов на предмет несанкционированного внедрения с целью их искажения с разными основополагающими целями. Подобные действия являются преднамеренными и тщательно скрываются, что препятствует прямому применению существующих методов тестирования и функционального контроля СБИС. В связи с этим стала очевидной необходимость выработки общего подхода к контролю СБИС и СнК на основе создания таксономии нарушений и отклонений, 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 [1] предлагается подход к проектированию Design for-Trust (DfTr), который дополнительно включает средства для контроля и предотвращения аппаратных атак при проектировании и изготовлении СБИС.

В настоящей работе предлагается метод кодирования цифровых устройств комбинационного типа на уровне их структурного представления с целью предотвращения хищения и злонамеренного искажения на основе использования методов и средств тестового диагностирования.

Современные СнК и особенности их проектирования и изготовления. Современная СнК объемом около 10 млрд транзисторов на кристалле содержит как цифровые, так и аналоговые функциональные блоки, различные датчики и исполнительные устройства. Впечатляющие достижения в области производства СнК (реально работающие цифровые и смешанного типа СнК на пластине размером порядка 450 мм) являются следствием больших успехов в области смешанной системной интеграции. При этом существенно увеличилась стоимость владения «кремниевой фабрикой», которая достигла в 2015 г. 5 млрд долл. Большинство проектных фирм не имеют собственных производственных мощностей. Они вынуждены использовать аутсорсинг и решать ряд возникающих в связи с этим экономических проблем и проблем безопасности.

Наряду с несомненно высокими достижениями в области производства СБИС имеет место существенное отставание теоретической базы автоматизированного проектирования в области разработки САПР, отсутствует системный подход к решению задач проектирования с учетом дестабилизирующих внешних факторов.

Наиболее узким местом в решении задач проектирования СнК является анализ функциональной корректности проектов на каждом из этапов процесса иерархического проектирования. Следует заметить, что применение отработанных в плане проектной корректности многократно используемых блоков интеллектуальной собственности (IP-блоков) при проектировании современных СнК не решает и даже существенно не упрощает задачу верификации проекта в целом. Объединение отлаженных отдельных функциональных блоков не дает никакой гарантии корректности полученного функционала вследствие возникающих несогласованностей, которые должны быть найдены и устранены на этапе верификации проекта в целом. Включая в проект определенный IP-блок, необходимо иметь уверенность в полноте поставляемого теста контроля, но более сложной задачей является согласование условий корректного совместного взаимодействия блоков внутри системы в целом.

Имеющиеся теоретические и практические результаты в областях синтеза, верификации проектов, построения тестов и организации контроля, во-первых, не достигли требуемого уровня развития, а во-вторых, продолжают оставаться корпоративными достижениями, ориентированными на применение специалистами высокой квалификации. В связи с этим разработка методов и средств функциональной верификации, а также тестов и систем контроля с учетом новых вызовов остается наиболее наукоемкой задачей, непосредственно определяющей сроки выполнения и стоимость проектов, требующей дальнейшего внимания разработчиков.

Источники угроз в области производства аппаратного обеспечения. В связи с быстрыми темпами роста объемов производства цифровых устройств в настоящее время особую остроту приобретает проблема нарушения авторских прав [1, 2]. Рост степени интеграции и функциональной сложности интегральных схем и высокая стоимость эксплуатации кремниевых производств расширяют аутсорсинг, который стал важной тенденцией в производстве интегральных схем.

Ущерб от пиратства и других угроз в области производства аппаратного обеспечения составляет около 4 млрд долл. в год, что примерно в 10 раз превышает ущерб от пиратства в области ПО [2]. Кроме пиратства, появляются новые виды угроз [3]: внедрение в проект дополнительных вредоносных несанкционированных операций с различной основополагающей целью, изменяющих функциональное наполнение системы; внедрение механизмов деградации схемных решений с целью нарушения системы синхронизации, приводящих к нарушению временной согласованности путей распространения сигналов и, в конечном итоге, к сбою системы; включение средств для получения конфиденциальной информации (к примеру, получение криптографических ключей) через порты контроля и др.

Очевидно, что после изготовления интегральной схемы проверить ее на наличие внесенных искажений и дополненной функциональности можно путем перепроектирования «по прототипу» с поэтапным восстановлением логики устройства и сравнением схемы с правильным образцом. При этом восстанавливается проект, реализованный в схеме, и сравнивается с моделью исходного проекта. Данный метод обеспечивает высокую вероятность обнаружения искажений, но время и стоимость, необходимые для выполнения перепроектирования, непомерно высоки. Поэтому основные практические методы контроля развиваются в направлении создания общего подхода к функциональному и тестовому контролю, таксономии нарушений для обнаружения как разного вида ошибок и неисправностей, возникающих в рабочем режиме проектирования, так и злонамеренных искажений, производимых путем применения известных механизмов.

Различные модели процесса злонамеренного искажения проекта, описывающие условия, при которых подобное искажение может внедриться в цифровую систему, приведены в работе [4]. В числе возможных источников искажений названы поставщики базовых функциональных блоков интеллектуальной собственности (IP's), которые приобретаются разработчиками СнК (модель *A*), «кремниевые фабрики» – изготовители СнК (модель *B*), а также разработчики СнК (модель *C*). Так, в модели *A* вредоносным источником является поставщик IP's, который продает свои изделия

разработчикам СнК. Эта модель вполне реалистична, так как разработчики СнК с целью сокращения стоимости и сроков проектирования широко используют привлечение существующих проектов. Искажение проекта может происходить на RTL, функционально-логическом или топологическом уровнях. В модели *B* угроза исходит от «кремниевой фабрики» на этапе производства интегральной схемы. Поскольку при изготовлении имеется доступ к топологическому проекту, то возможно восстановление и перепроектирование проекта, добавление элементов аппаратных искажений. Жизненность такой модели очевидна в связи с тем, что со стороны проектировщиков практически отсутствует возможность контроля деятельности в случае, например, офшорного производства. В модели *C* искажения проекта могут произойти на этапе проектирования вследствие преднамеренных злоумышленных действий конкретного информированного лица, что может иметь место при использовании ненадежной САПР.

Рассмотрены также другие модели возможных аппаратных искажений в случае ненадежности любых двух или всех трех участников процесса [4]. В связи с тем что искажения в проекте могут происходить на разных этапах проектирования (на RTL-уровне, на уровне структурного описания схем netlist, в топологическом проекте), существует потребность в разработке методов обнаружения искажений на любых уровнях абстракции. В настоящее время подобные методы защиты аппаратных средств от внешних угроз и борьбы с пиратством находятся на начальном этапе развития по сравнению с методами защиты программных средств. Одной из известных методик защиты исходных кодов программ от обратного проектирования является функциональная обфускация, основная задача которой заключается в затруднении понимания функционирования программы. К сожалению, эффект от применения методов обфускации в случае языка VHDL ограничен, так как полученные результаты не приводят к изменению конечного результата синтеза, а структурные реализации устройств до и после обфускации выглядят одинаково [2].

Следует заметить, что наряду с задачей защиты проектов от несанкционированного вмешательства весьма актуальна и другая задача – выявление вредоносных изменений и восстановление исходной структуры [5].

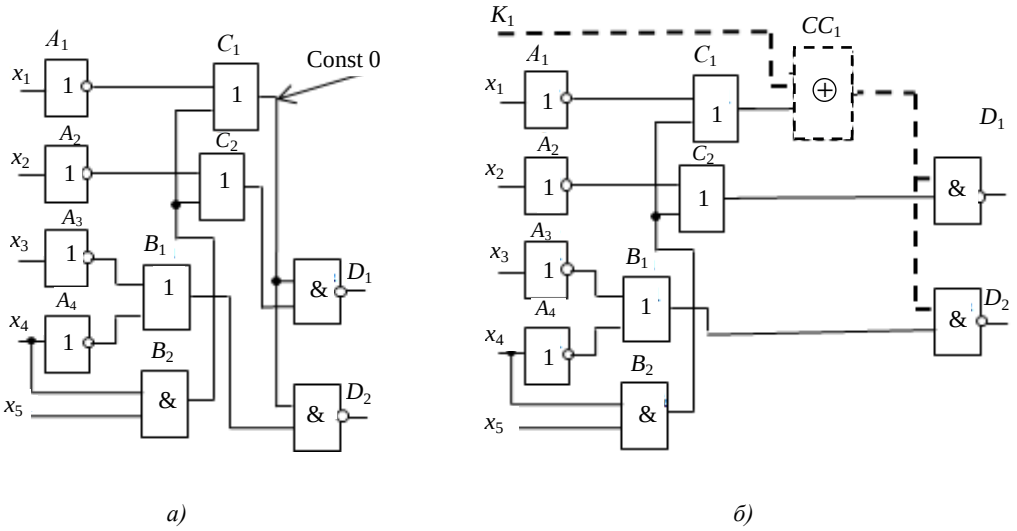
Обфускация и логическое кодирование цифрового устройства на структурном уровне. Для блокирования попыток внешнего вмешательства в проект цифровой системы на структурном уровне одним из методов является логическое кодирование структурной реализации, которое обеспечивает доступ к объекту только авторизованным пользователям [6]. Метод предполагает сокрытие функциональности проекта и использование ключа, применение которого выводит систему в область правильного функционирования. Кроме логического шифрования комбинационной схемы, известен метод внедрения новых внутренних состояний в граф перехода для последовательностных устройств, эффективность практического применения которого, к сожалению, не установлена [7].

Метод логического кодирования основан на включении в логическую сеть дополнительных вентилей, управляемых внешними логическими ключами, т. е. на применении обфускации структуры объекта. Таким образом, 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.

Чтобы защитить комбинационную схему с помощью *k*-разрядного ключа, предлагается простая процедура, которая требует включения в схему k дополнительных вентилей [6]. Во-первых, выбираются и сопоставляются с битами $\{y\}$ ключа k линий схемы $\{w_i\}$. Каждая выбранная линия w_i отключается от приемников сигнала, а на место обрыва подключается вентиль XOR или XNOR с выходной линией связи w'_i, на которой формируется сигнал, управляющий соответствующими приемниками сигнала вентиля w_i. При подключении вентиля XOR (XNOR) $w'_i = w_i \oplus y_i$ ($w'_i = w_i \overline{\oplus} y_i$), где y_i – соответствующий бит ключа. Выбор вентиля XOR или XNOR зависит от выбранного значения бита ключа: если выбранное значение $y_i = 0$, то $w'_i = w_i \oplus y_i$; если $y_i = 1$, то $w'_i = w_i \overline{\oplus} y_i$.

На рис. 1, *а* показан фрагмент логической схемы, а на рис. 1, *б* проиллюстрирована основная идея логического кодирования. Выход элемента C_1 отключен от нагрузки (элементы D_1

и D_2) и подключен к одному из входов дополнительного «ключевого» элемента типа XOR CC_1, на второй вход которого поступает внешний входной сигнал K_1 однобитового ключа. Схема будет работать в требуемом режиме только в том случае, если сигнал на входе K_1 будет равен 0. В противном случае на выходе элемента XOR CC_1 будет формироваться сигнал, инверсный правильному.



а) *б)*

Рис. 1. Фрагмент логической сети: *а)* исходная комбинационная схема; *б)* схема с однобитовым ключом

Вместо элемента CC_1 типа XOR может быть установлен элемент XNOR. В этом случае однобитовый правильный ключ, поступающий на вход K_1, равен 1. Заметим, что применение неправильного ключа равносильно появлению неисправности константного типа const 0 (const 1) на выходе элемента C_1 в зависимости от входного набора и истинного значения сигнала на C_1, равного 1 (0). Этот факт является важным, так как позволяет формализовать задачу обфускации на основе применения методов и средств тестового контроля цифровых устройств.

При подаче входного набора X = (00000) и неправильного ключа K_1 = 1 (рис. 1, *б*) на выходах схемы D_1, D_2 формируются сигналы (11), в то время как при правильном ключе K_1 = 0 – сигналы (00). Так же поведет себя схема при неисправности const 0 на выходе элемента C_1. Следовательно, входной набор X = (00000) является тестом контроля данной неисправности. В то же время при отсутствии неисправности он искажает выходное состояние схемы при подаче неправильного ключа.

Таким образом, для сокрытия функциональности схемы необходимо добавить в некоторые ее линии дополнительные элементы и определить правильный код, искажение которого выводит схему из области правильного функционирования. Заметим, что при воздействии входного набора X = (01110) и неправильного ключа K_1 = 1 (рис. 1) на выходах схемы D_1, D_2 появятся сигналы (11), как и при правильном ключе, так как входной набор X = (01110) не является тестом контроля неисправности const 0 на выходе элемента C_1.

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 с целью создания максимальных трудностей для злоумышленника при поиске правильного ключа.

Положим, что цифровое устройство состоит из n первичных входов, m первичных выходов и k бит ключа шифрования. При воздействии входного вектора $X \in 2^n$ на выходах устройства формируется соответствующий правильный выходной вектор $Z \in 2^m$. Пусть $K \in 2^k$ – правильные значения ключевых сигналов (правильный ключ). Возможны два сценария функционирования устройства при разных значениях переменных шифрования. Функция производит правильные выходы для всех тестовых шаблонов ввода при использовании действительного секретного ключа K либо неправильные – при неправильных значениях секретного ключа:

$$F(x, k) = \begin{cases} Z \ \forall X \in 2^n, \ \ Z \in 2^m; \\ Z' \ \forall X \in 2^n, \ \ Z' \in 2^m, \ Z' \neq Z, \end{cases}$$

где Z – правильный выходной вектор, Z' – неправильный.

Для определения степени защищенности устройства при его кодировании принимается расстояние Хэмминга (HD), которое для кодовых комбинаций булевых векторов A и B определяется как вес $V(C)$ такой третьей кодовой комбинации C, которая получается сложением по mod 2 исходных комбинаций A и B: A = 011011100, B = 100111001, C = 111100101, $V(C = A + B) = 6$ (расстояние Хэмминга).

Таким образом, расстояние Хэмминга – это число, используемое для обозначения меры различия между двумя двоичными строками. При кодировании структурных реализаций цифровых устройств расстояние Хэмминга позволяет количественно определить степень отличия правильной реакции устройства от ошибочной. Если HD (Z, Z') = 0, то это означает, что реакция закодированной схемы не зависит от ключа блокировки. При HD (Z, Z') = m Z' дополняет Z, что упрощает злоумышленнику поиск правильного ключа. Для того чтобы затруднить восстановление правильного ключа, необходимо обеспечить наименьшую корреляцию между правильными и неправильными выходными векторами. Это достигается при HD (Z, Z') = m/2, когда на каждом входном воздействии около 50 % выходных сигналов в случае применения неправильного ключа принимают логические значения, инверсные правильным.

Применение методов и средств тестового диагностирования для защиты цифровых устройств от вредоносных искажений. При включении очередного вентиля при кодировании логических устройств необходимо проводить анализ на появление эффекта маскирования неисправностей, который способен блокировать эффект кодирования. В работе [6] при кодировании логических устройств ключевые вентили помещались в схему случайным образом. При таком подходе использование неправильного ключевого бита не гарантирует появления неправильного выходного сигнала и не может должным образом затруднить злоумышленнику доступ к структуре устройства. Во-первых, возможен эффект маскирования неисправностей (рис. 2). Схема, зашифрованная тремя битами ключа K_1, K_2, K_3, на входном наборе 00000 при подаче как правильного ключа 000, так и неправильного 111 вырабатывает одинаковую выходную реакцию 00. Это происходит по причине маскирования неисправностей const 0, которые одновременно возникают на выходах элементов C_1, D_1 и D_2. Во-вторых, для некоторых линий отсутствует возможность активизации пути от данной линии к выходам устройства.

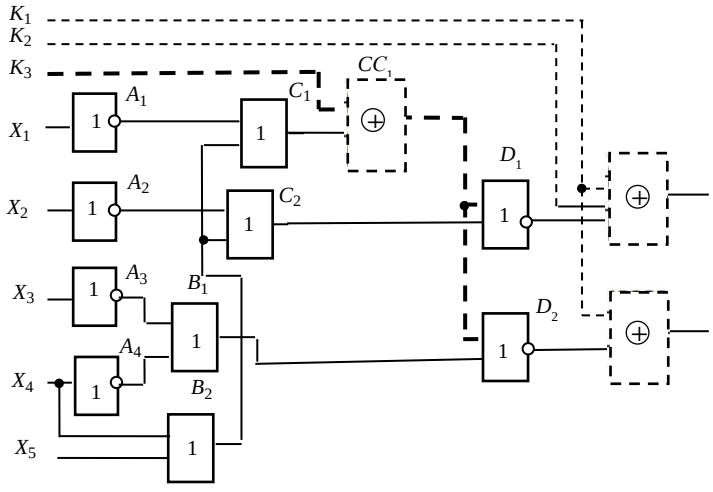


Рис. 2. Влияние маскирования неисправностей на результаты кодирования

На рис. 3 изображена структура цифрового устройства, реализующего систему булевых функций $D_1 = \overline{x}_1 x_3 x_4 x_5 \vee \overline{x}_2 x_3 x_4 x_5$, $F_1 = \overline{x_1 x_3} \vee \overline{x_2 x_3} \vee \overline{x_1 x_4} \vee \overline{x_2 x_4} \vee x_6 \vee x_7$. Как было сказано выше, кодирование схемы путем случайного подбора мест вставки в структуру ключевых вентилей оказывается недостаточно эффективным. К примеру, добавление вентиля XOR на выходе эле-

мента B_3 не принесет ожидаемого эффекта, так как для неисправности const 0 на выходе B_3 не существует проверяющего теста и применение неправильного ключа, равного 1, не приведет к изменению реакции схемы при подаче любой входной последовательности. Поэтому при кодировании структуры устройства необходимо отслеживать эффективность каждого шага. При решении основной задачи – затруднить злоумышленнику доступ к структурной реализации устройства – необходимо обеспечить оптимизацию объема необходимого дополнительного оборудования, учесть влияние задержек дополнительно включенных элементов на функционирование устройства.

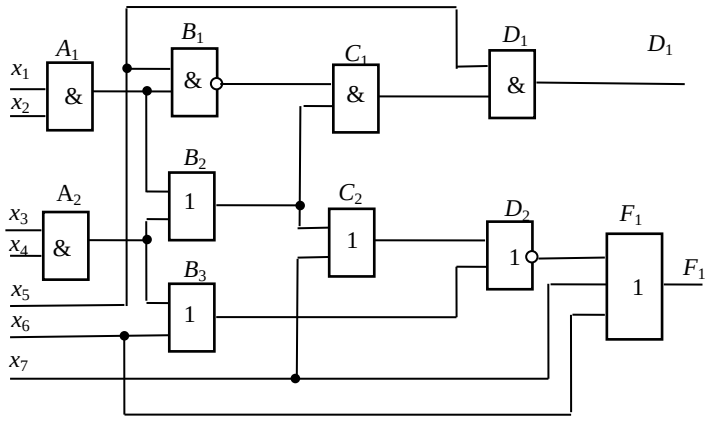


Рис. 3. Структурная реализация цифрового устройства

В работе [8] предложен подход к определению множества линий структуры для кодирования, основанный на моделировании схемы с внесенной *i*-й неисправностью и вычислении признака $P_i = X_i \cdot Y_i$, который характеризует линию с точки зрения эффективности ее выбора при кодировании схемы. Здесь X_i – количество входных наборов, которые покрывают анализируемую неисправность, Y_i – количество выходных переменных, которые искажаются при появлении данной неисправности. По результатам анализа полученных признаков определяется множество внутренних линий схемы для кодирования.

Очевидно, что данный подход требует моделирования схемы $M = 2s \cdot 2^n$ раз, где s – общее количество линий схемы (переменных полного состояния схемы), n – количество входных переменных схемы. Для схемы на рис. 3 $M = 128 \cdot 34 = 4352$. Для реальных схем подобный подход практически неприемлем по причине высоких вычислительных затрат. С целью оптимизации вычислительных процедур предлагается эвристическое решение – сократить количество моделируемых входных наборов до 100 [8] (в этом случае $M = 200k$).

Сведем задачу кодирования к поиску неисправностей константного типа кодируемой структуры, обнаруживаемых на большем количестве выходных линий и на максимальном количестве входных векторов.

В отличие от решения, принятого в работе [8], рассмотрим более эффективный подход, который основан на применении метода сквозного вычисления неисправностей, покрываемых входным вектором, т. е. конкурентно-дедуктивного моделирования вместо моделирования каждой неисправной модификации схемы на определенном множестве случайных входных наборов с целью оценки степени влияния неисправностей на выходы схемы [9]. Метод конкурентно-дедуктивного моделирования неисправностей основан на моделировании исправной схемы и позволяет за один проход моделирования определить все неисправности константного типа, обнаруживаемые на моделируемом входном наборе. За счет того что моделируется только исправная схема, эффективность решения существенно повышается по сравнению с моделированием одиночной неисправности на множестве входных векторов.

Вначале вычисляются неисправности, обнаруживаемые на моделируемом ограниченном множестве случайных входных наборов. Затем по результатам анализа определяются те неисправности, которые обнаруживаются наибольшим числом наборов и указывают преимуще-

ственные линии схемы для вставки ключевых вентилей. В то же время численное ограничение количества моделируемых входных воздействий [10] сужает возможность поиска наиболее эффективного решения.

В настоящей работе предлагается другой подход, основанный на построении теста в классе неисправностей константного типа [9] и его применении на первом этапе кодирования. В рамках данного подхода вместо заранее определенного числа случайных входных воздействий (как, например, 100 в работе [10]) применяется тестовая последовательность входных векторов, которая обеспечивает близкое к полному покрытие неисправностей константного типа кодируемой структуры.

В табл. 1 приведены результаты построения теста для схемы на рис. 3 и единичные значения разностных неисправных функций. Первый столбец таблицы содержит входные наборы теста, последующие (согласно идентификаторам неисправностей константного типа всех линий схемы) – единичные значения разностных неисправных функций, реализуемых на соответствующем выходе схемы. Здесь X_1^0 – неисправность типа const 0 на входе X_1, а A_1^1 – неисправность типа const 1 на выходе элемента A_1. Верхний индекс при единичном значении разностной неисправной функции указывает, на каком выходе схемы реализуется данная функция. В данном случае значение 1^1 относится к функции, реализуемой на первом выходе схемы, т. е. на выходе элемента D_1. Верхний индекс в обозначении разностной неисправной функции (1^2) указывает, что функция относится ко второму выходу схемы, т. е. F_1. (Если неисправность обнаруживается, к примеру, на трех выходах, то верхний индекс может иметь вид $1^{2,3,5}$.)

Таблица 1

Единичные значения разностных неисправных функций
для структурной реализации цифрового устройства

Тест-векторы	X_1^0	X_1^1	X_2^0	X_2^1	X_3^0	X_3^1	X_4^0	X_4^1	X_5^0	X_5^1	X_6^0	X_6^1	X_7^0	X_7^1	A_1^0	A_1^1	A_2^0
1010100				1^2				1^1								1^2	
1011100				1^1	1^1		1^1		1^1			1^2		1^2		1^1	1^1
1101100	1^2		1^2									1^2		1^2	1^2		
0101111		1^2				1^2										1^2	
0111010										1^1	1^2						
0011101					1^1		1^1		1^1				1^2			1^1	1^1
Тест-векторы	A_2^1	B_1^0	B_1^1	B_2^0	B_2^1	B_3^0	B_3^1	C_1^0	C_1^1	C_2^0	C_2^1	D_1^0	D_1^1	D_2^0	D_2^1	F_1^0	F_1^1
1010100	1^1				1^1		1^2		1^1		1^2		1^1	1^2		1^2	
1011100		1^1		1^1				1^1				1^1			1^2		1^2
1101100			1^1	1^2					1^1	1^2			1^1		1^2		1^2
0101111	1^2				1^2		1^2				1^2		1^1	1^2		1^2	
0111010													1^1			1^2	
0011101		1^1		1^1				1^1				1^1				1^2	

Из табл. 1 видно, что размещение ключевого вентиля XOR на выходе элемента B_3 не имеет смысла, так как тест контроля неисправности const 0 на выходе элемента B_3 отсутствует. Наиболее целесообразно выбрать вначале для последующего кодирования выходы элементов A_1, D_1, F_1, так как столбцы, соответствующие неисправностям A_1^1, D_1^1, F_1^0 данных элементов, содержат большее число единичных значений разностных неисправных функций. Это свидетельствует о том, что большее число входных векторов в случае применения неправильного ключа приведет к искажению реакции схемы.

На рис. 4 показана схема с внесенными ключевыми элементами PS_1, PS_2, PS_3 и ключевыми входами K_1, K_2, K_3. В схеме ключевой элемент PS_3 имеет тип XOR, так как неисправность const 0 на выходе элемента F_1 обнаруживается бóльшим числом входных сигналов по сравнению с неисправностью const 1. Ключевые элементы PS_1 и PS_2 имеют тип XNOR, так как соответствуют столбцам с неисправностями типа const 1.

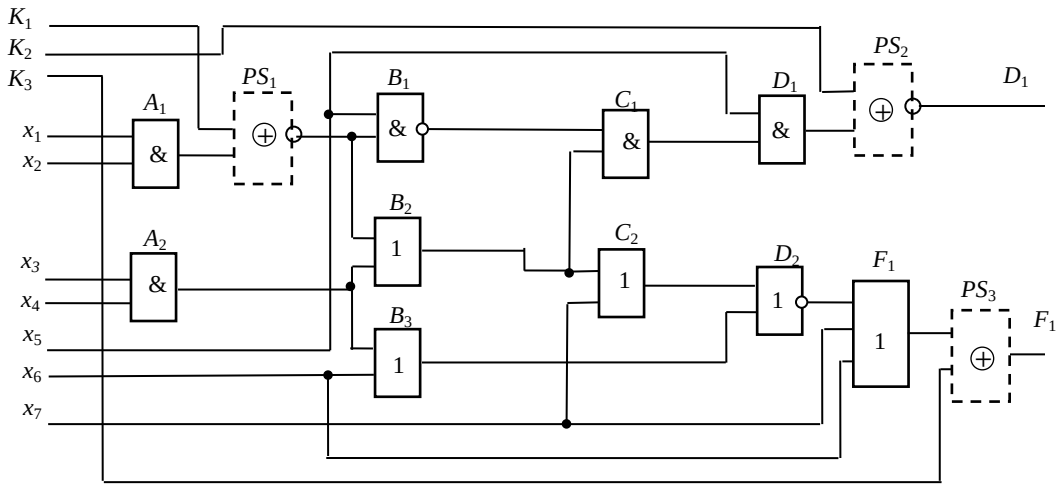


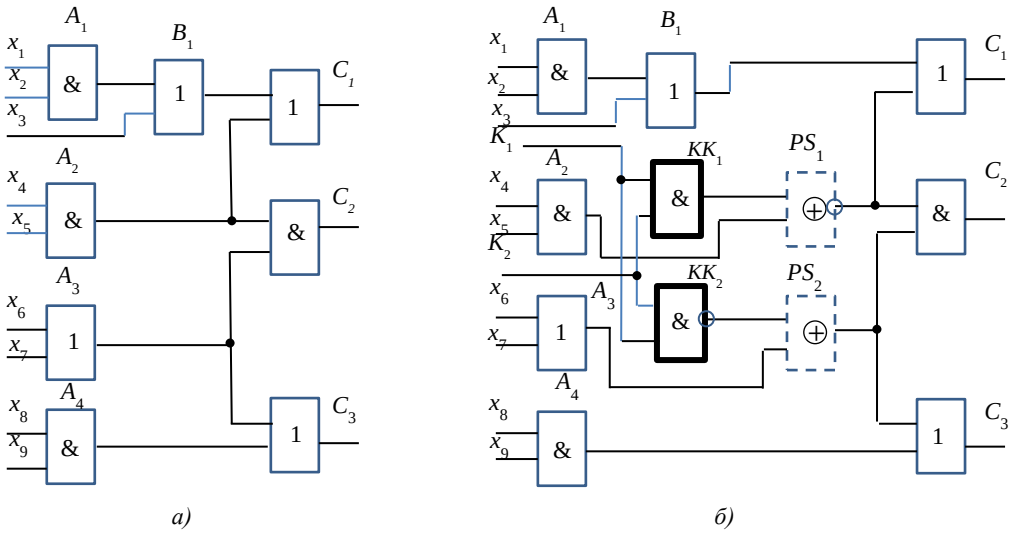
Рис. 4. Схема с вентилями PS_1, PS_2 и PS_3 для логического шифрования

После добавления ключевых элементов в структуру необходимо проанализировать полученные результаты кодирования, используя моделирование имеющейся частично закодированной структуры на наборах теста на всем булевом интервале множества ключевых входов и сравнение в каждом случае выходных реакций схемы с результатами моделирования исходной схемы. Как было показано выше, для максимального затруднения доступа к получению структуры схемы необходимо обеспечить кодовое расстояние Хэмминга между выходными состояниями схемы в условиях применения правильных и ошибочных ключевых кодов, близкое к 0,5 числа переменных выходного состояния [6].

Управляемое кодирование цифровых устройств на структурном уровне. Очевидно, что результат кодирования проявляется на выходах схемы в зависимости от числа неправильных битов кода [8]. Если ключевой вентиль управляется одним битом ключевого кода, вероятность того, что данный вентиль будет приведен в действие, $P = 0{,}5$. Это означает, что только половина ключевых вентилей повлияет на результат функционирования схемы при применении неправильного ключа. Для того чтобы увеличить вероятность P и усилить влияние неправильного бита кодового слова на результат функционирования схемы, применим управляющие вентили, с помощью которых можно объединить биты кодового слова в группы, использовав при этом их выходы в качестве входов ключевых вентилей. В таком случае будет реализовано групповое воздействие нескольких битов кодового слова на активизацию ключевого вентиля. Если хотя бы один из ключевых входов, включенных в группу, принимает неправильное значение, ключевой вентиль окажется активированным. Для этого с каждым ключевым вентилем используется управляющий вентиль. Если применяется двухвходовый управляющий вентиль, то вероятность активизации ключевого вентиля возрастает с 0,5 до 0,75; в случае трехвходового вентиля вероятность составляет 0,88, а пятивходового – 0,97 (только один ключевой вектор из 32 векторов данной группы является правильным).

На рис. 5, *а* показан фрагмент схемы с тремя выходами, на рис. 5, *б* – пример двухуровневого кодирования. В соответствии с полученными результатами (табл. 2) в качестве линий для первоочередного ввода ключевых вентилей для кодирования выбраны выходы элементов A_2 (вентиль PS_1 типа XNOR) и A_3 (вентиль PS_2 типа XOR). Тип ключевого вентиля XNOR на выходе элемента A_2 выбирается в соответствии с неисправностью const 1, которая покрывается четырьмя из семи входных векторов и обнаруживается на двух из трех выходов. Выбор неисправности A_3^0 обусловлен тем, что по сравнению с неисправностью C_2^1 неисправность A_3^0 приводит к изменению логического состояния двух выходов.

Дополнительно в схему включены управляющие двухвходовые вентили KK_1 и KK_2, которые усилили влияние на функционирование схемы каждого бита ключевого входа.



а) *б)*

Рис. 5. Пример схемы с двухуровневым кодированием: *а)* логическая структура с тремя выходами; *б)* двухуровневое кодирование схемы

Таблица 2

Единичные значения разностных неисправных функций для схемы с двухуровневым кодированием

Тест-векторы	X_1^0	X_1^1	X_2^0	X_2^1	X_3^0	X_3^1	X_4^0	X_4^1	X_5^0	X_5^1	X_6^0	X_6^1	X_7^0	X_7^1	X_8^0	X_8^1	X_9^0
100100101				1^1		1^1				$1^{1,2}$			1^3				
110010001	1^1		1^1									1^3		1^3		1^3	
001111001							1^2		1^2		$1^{2,3}$						
000010011						1^1		1^1							1^3		1^3
000110010							1^1		1^1			$1^{2,3}$		$1^{2,3}$			
101000110					1^1								1^3				
010001100		1^1				1^1											
Тест-векторы	X_9^1	A_1^0	A_1^1	A_2^0	A_2^1	A_3^0	A_3^1	A_4^0	A_4^1	B_1^0	B_1^1	C_1^0	C_1^1	C_2^0	C_2^1	C_3^0	C_3^1
100100101			1^1		$1^{1,2}$	1^3					1^1		1^1		1^2	1^3	
110010001		1^1					1^3		1^3	1^1		1^1			1^2		1^3
001111001				1^2		$1^{2,3}$						1^1		1^2		1^3	
000010011			1^1		1^1			1^3			1^1		1^1		1^2	1^3	
000110010	1^3			1^1			$1^{2,3}$		1^3			1^1			1^2		1^3
101000110					1^2	1^3				1^1		1^1			1^2	1^3	
010001100			1^1		$1^{1,2}$	1^3					1^1		1^1		1^2	1^3	

Рассмотрим основные этапы управляемого логического кодирования комбинационных структур при использовании двухвходовых управляющих вентилей.

Исходные данные: описание кодируемой структуры схемы. Результаты: описание закодированной структуры схемы, правильный ключ. Алгоритм управляемого кодирования цифровых устройств включает следующие шаги:

1. Построить тест контроля структуры в классе неисправностей константного типа методом случайного поиска на основе применения метода конкурентно-дедуктивного моделирования неисправностей.

2. Упорядочить множество *FN* обнаруживаемых на наборах теста неисправностей по убыванию числа покрывающих входных наборов и активизированных выходов схемы.

3. $J := 1$.

4. Из множества *FN* выбрать *j*-ю неисправность, в соответствии с типом неисправности включить в структуру схемы ключевой элемент (типа XOR, если неисправность const 0, и типа XNOR, если неисправность const 1), включить управляющий вентиль с ключевым входом k_j, на второй вход управляющего вентиля подключить дополнительный ключевой вход.

5. Смоделировать полученную структуру на всех наборах теста при всех возможных комбинациях значений ключа.

6. Проанализировать кодовое расстояние Хэмминга между реакциями исходной схемы и частично закодированной при неправильных битах ключа.

7. Если результат анализа кодирования неудовлетворителен, то $J := J + 1$, перейти к п. 4.

8. Выход.

В приведенном алгоритме отсутствуют этапы анализа закодированной схемы на предмет влияния включенных дополнительных аппаратных средств на временные параметры и алгоритмическую устойчивость схемы.

Заключение. В работе обоснована необходимость развития таксономии отклонений, возникающих по разным причинам в проектах СБИС типа СнК на разных этапах проектирования и изготовления.

Предложенный алгоритм управляемого кодирования описаний цифровых устройств комбинационного типа на структурном уровне на основе применения средств тестового контроля требует меньших вычислительных затрат и времени, проявляет устойчивость к восстановлению правильного ключа на основе «атаки SAT» [10]. Это обусловлено тем, что ключевые входы устройства не связаны напрямую с ключевыми вентилями, а ключевые вентили активизируются не одним ключевым входом. Применение метода сквозного вычисления множества покрываемых неисправностей на основе моделирования исправной схемы существенно сокращает объем вычислительных процедур.

Список использованных источников

1. Security analysis of integrated circuit camouflaging / J. Rajendran [et al.] // ACM SIGSAC Conf. on Computer & Communications Security (CCS'13). – Berlin, 2013. – P. 709–720.

2. Сергейчик, В. В. Методы лексической обфускации VHDL-описаний / В. В. Сергейчик, А. А. Иванюк // Information Technologies and Systems 2013 (ITS 2013) : Proc. of the Intern. Conf. – Minsk, 2013. – С. 198–199.

3. Benchmarking of hardware Trojans and maliciously affected circuits / B. Shakya [et al.] // J. of Hardware and Systems Security. – 2017. – Vol. 1(1). – P. 85–102.

4. Hardware Trojans: lessons learned after one decade of research / K. Xiao [et al.] // ACM Transactions on Design Automation of Electronic System. – 2016. – Vol. 22, no. 1. – P. 1– 23.

5. New testing procedure for finding insertion sites of stealthy hardware Trojans / S. Dupuis [et al.] // Design, Automation & Test in Europe Conference & Exhibition (DATE'2015), Grenoble, France, 9–13 Mar. 2015. – Grenoble, 2015. – P. 776–781.

6. Roy, J. A. EPIC: Ending Piracy of Integrated Circuits / J. A. Roy, F Koushanfar, I. L. Markov // IEEE Computer. – 2010. – Vol. 43, no. 10. – P. 30–38.

7. Chakraborty, R. S. Security against hardware Trojan through a novel application of design obfuscation / R. S. Chakraborty, S. Bhunia // IEEE/ACM Intern. Conf. on Computer-Aided Design. – San Jose, 2009. – P. 113–116.

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

9. Золоторевич, Л. А. Исследование методов и средств верификации проектов и генерации тестов МЭС / Л. А. Золоторевич // 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.

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

References

1. Rajendran J., Sam M., Sinanoglu O., Karri R. Security analysis of integrated circuit camouflaging. *ACM SIGSAC Conference on Computer & Communications Security*. Berlin, 2013, pp. 709–720.

2. Sergejchik V. V., Ivanjuk A. A. Metody leksicheskoj obfuskacii VHDL-opisanij [Methods of lexical obfuscation of VHDL descriptions]. *Information Technologies and Systems 2013 (ITS 2013) : Proceedings of the International Conference*. Minsk, 2013, pp. 198–199 (in Russian).

3. Shakya B., Salmani T. H., Forte D., Bhunia S., Tehranipoor M. Benchmarking of hardware Trojans and maliciously affected circuits. *Journal of Hardware and Systems Security*, 2017, vol. 1(1), pp. 85–102.

4. Xiao K, Forte D, Jin Y, Karri R, Bhunia S., Tehranipoor M. Hardware Trojans: lessons learned after one decade of research. *ACM Transactions on Design Automation of Electronic System*, 2016, vol. 22, no. 1, pp. 1–23.

5. Dupuis S., Rouzeyre B., Flottes M.-L., Natale G. D., Ba P.-S. New testing procedure for finding insertion sites of stealthy hardware Trojans. *Design, Automation & Test in Europe Conference & Exhibition (DATE'2015), Grenoble, France, 9–13 March 2015*. Grenoble, 2015, pp. 776–781.

6. Roy J. A., Koushanfar F., Markov I. L. EPIC: Ending Piracy of Integrated Circuits. *IEEE Computer*, 2010, vol. 43, no. 10, pp. 30–38.

7. Chakraborty R. S., Bhunia S. Security against hardware Trojan through a novel application of design obfuscation. *IEEE/ACM International Conference on Computer-Aided Design*. San Jose, 2009, pp. 113–116.

8. 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)*. Thessaloniki, 2017, pp. 221–226.

9. Zolotorevich L. A. Issledovanie metodov i sredstv verifikacii proektov i generacii testov MJeS [Research of methods and means of project verification and test generation of MES]. Sbornik nauchnyh trudov Vserossijskoj nauchno-tehnicheskoj konferencii "Problemy razrabotki perspektivnyh mikrojelektronnyh sistem (MJeS–2006)" [*Collection of scientific papers of the all-russian scientific and technical conference "Problems of Development of Promising Microelectronic Systems" (MES–2006)*], Moscow, Institut problem proektirovanija v mikrojelektronike Rossijskoj akademii nauk, 2006, pp. 163–168 (in Russian).

10. 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, vol. 35, no. 9, pp. 1411–1424.

Информация об авторе

Золоторевич Людмила Андреевна, кандидат технических наук, доцент, Белорусский государственный университет радиоэлектроники и информатики, Минск, Беларусь.
E-mail: zolotorevichLA@bsuir.by

Information about the author

Lyudmila A. Zolotorevich, Cand. Sci. (Eng.), Assoc. Prof., Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus.
E-mail: zolotorevichLA@bsuir.by