ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2020 Управление, вычислительная техника и информатика № 50

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 519.873:519.718.7
DOI: 10.17223/19988605/50/9

Л.А. Золоторевич

АППАРАТНАЯ ЗАЩИТА ЦИФРОВЫХ УСТРОЙСТВ

Рассматривается задача защиты проектов цифровых устройств на структурном уровне от вредоносного искажения и нарушения авторских прав. Предлагается алгоритм управляемого кодирования комбинационных структур на основе применения методов и средств тестового диагностирования. Алгоритм не требует моделирования неисправностей устройства в явном виде, что сокращает объем вычислительных процедур при кодировании схемы. Приводятся особенности проектирования современных СнК. Акцентируется внимание на необходимости создания и развития общего подхода к рассмотрению задач контроля и верификации проектов (таксономии отклонений). Таксономия отклонений включает анализ ошибок, возникающих непосредственно в процессе проектирования, и преднамеренных искажений на этапах проектирования и изготовления.
Ключевые слова: цифровая экономика; СБИС; защита авторских прав; кодирование логических схем.

Акцент на цифровую экономику, приоритетная разработка цифровых технологий требуют постоянного совершенствования теории и практики проектирования интегральных схем, систем на кристалле (СнК) как технической базы создания электронных систем различного назначения.

Развитие технологии СБИС, СнК определяющим образом зависит от развития методов и качества применяемых средств автоматизированного проектирования (САПР) [1], в особенности методов и средств контроля, верификации, построения тестов контроля функциональных блоков и систем. Сложность решения указанных задач постоянно возрастает из-за возрастания 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, возникающими непреднамеренно, и должны решаться в режиме благоприятствующего проектирования. Однако 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 скрываемыми, что препятствует прямому применению существующих методов тестирования и функционального контроля СБИС.

В связи с этим стала очевидной необходимость защиты проектов на основе создания общего подхода к контролю СБИС, СнК, 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 [2] предлагается подход к проектированию Design-for-Trust – DfTr, который дополнительно включает средства для контроля и предотвращения аппаратных атак при проектировании и изготовлении СБИС.

1. 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 [2]. Рост степени интеграции интегральных схем и вместе с этим высокая стоимость эксплуатации кремниевых производств расширяет аутсорсинг, который стал важной тенденцией в производстве интегральных схем.

Ущерб от пиратства и других угроз в области производства аппаратного обеспечения составляет около 4 млрд долларов в год, что примерно в 10 раз превышает ущерб от пиратства в области ПО [3]. Кроме пиратства появляются новые виды угроз [4]: 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.

В работе [5] 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.

В связи с тем, что искажения в проекте могут происходить на разных этапах проектирования, – на RTL-уровне, на уровне структурного описания схем (уровень netlist), в топологическом проекте, существует потребность в разработке методов обнаружения искажений на разных уровнях абстракции. Одной из известных методик защиты исходных кодов программ от обратного проектирования является функциональная обфускация. К сожалению, эффект от применения методов обфускации в случае языка VHDL ограничен, поскольку результаты их применения не приводят к изменению конечного результата синтеза, так как структурные реализации устройств до и после обфускации выглядят одинаково [3].

2. Обфускация и логическое кодирование цифрового устройства на структурном уровне

Одним из методов блокирования попыток внешнего вмешательства в проект цифровой системы на структурном уровне является логическое кодирование структурной реализации, которое обеспечивает доступ к объекту только авторизованным пользователям [7]. 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 [8].

Первый метод основан на включении в логическую сеть дополнительных вентилей, управляемых внешними логическими ключами, т.е. на применении обфускации структуры объекта. В такой постановке если злоумышленник не владеет ключом, то ему недоступна внутренняя реализация объекта. Поэтому 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 [7]. Выбор линии для включения вентиля, тип вентиля существенно влияют на эффективность кодирования. На рис. 1, *a* приведен фрагмент логической схемы, а на рис. 1, *b* 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 $К_1$ однобитового ключа. Схема будет работать в требуемом режиме только в том случае, если сигнал на входе $К_1$ будет равен 0. В противном случае на выходе элемента XOR $СС_1$ будет формироваться сигнал, инверсный правильному.

Вместо элемента $СС_1$ типа XOR может быть установлен элемент XNOR. В этом случае однобитовый правильный ключ, поступающий на вход $К_1$, равен 1. Заметим, что применение неправильного ключа равносильно появлению неисправности константного типа const 0 (const 1) на выходе элемента $С_1$ в зависимости от входного набора и истинного значения сигнала на $С_1$, равного 1 (0). Этот факт является важным, так как позволяет формализовать задачу обфускации на основе применения методов и средств тестового контроля цифровых устройств.

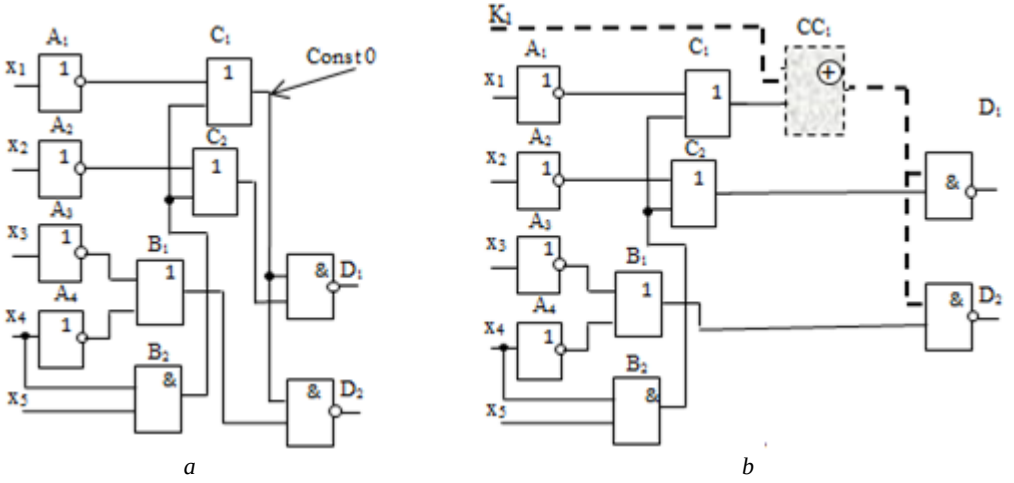


Рис. 1. Фрагмент логической сети: *a* – исходная комбинационная схема; *b* – схема с однобитовым ключом
Fig. 1. Fragment of a logical network: *a* – the original combinational circuit; *b* – scheme with a one-bit key

При воздействии входного набора X = (00000) и неправильного ключа $К_1 = 1$ (см. рис. 1) на выходах схемы D_1, D_2 формируются сигналы (11), в то время как при правильном ключе $К_1 = 0$ – (00). Так же поведет себя схема (см. рис. 1) при неисправности const 0 на выходе элемента $С_1$. То есть входной набор X = (00000) является тестом контроля данной неисправности и в то же время при отсутствии неисправности искажает выходное состояние схемы при подаче неправильного ключа.

Таким образом, для сокрытия функциональности схемы необходимо добавить в некоторые линии схемы 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 $К_1 = 1$ (см. 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 $С_1$.

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 для создания максимальных трудностей для злоумышленника по поиску 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:

1) при использовании действительного секретного ключа K функция производит правильные выходы для всех тестовых шаблонов ввода;

2) при использовании неправильных значений секретных ключей функция генерирует неправильные выходы соответственно:

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

здесь Z (Z') – правильный (неправильный) выходной вектор.

Для определения степени защищенности устройства при его кодировании принимается расстояние Хэмминга (HD) – число, используемое для обозначения меры различия между двумя двоичными последовательностями. HD 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, что достигается при HD (Z, Z') = m/2, когда на каждом входном воздействии около 50% выходных сигналов в случае применения неправильного ключа принимает логические значения, инверсные правильным.

3. 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 [7] при кодировании логических устройств ключевые вентили помещались в схему случайным образом. При таком подходе применение неправильного ключевого бита не гарантирует появления неправильного выходного сигнала (рис. 2) и не может требуемым образом затруднить злоумышленнику доступ к структуре устройства. Во-первых, возможен эффект маскирования неисправностей, что показано на рис. 2. Схема, зашифрованная тремя битами ключа K_1, K_2, K_3 на рис. 2, на входном наборе 00000 как при подаче правильного ключа 000, так и при неправильном ключе 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.

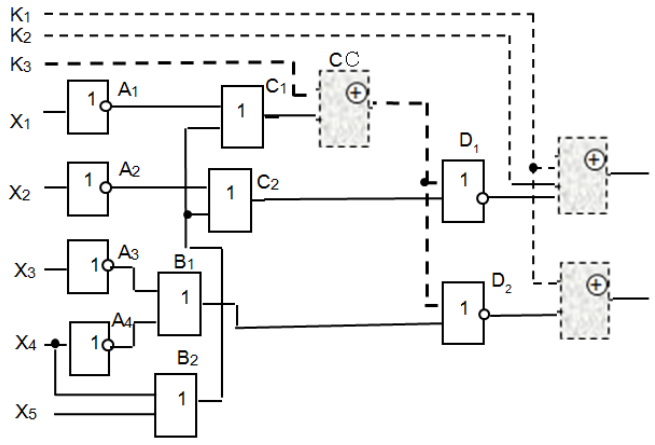


Рис. 2. Влияние маскирования неисправностей на результаты кодирования
Fig. 2. Impact of masking faults on coding results

На рис. 3 приведена структура цифрового устройства , реализующего систему булевых функций $D_1=\overline{x}_1x_3x_4x_5\vee\overline{x}_2x_3x_4x_5;F_1=\overline{\overline{x}_1\overline{x}_3}\vee\overline{\overline{x}_2\overline{x}_3}\vee\overline{\overline{x}_1\overline{x}_4}\vee\overline{\overline{x}_2\overline{x}_4}\vee x_6\vee x_7$. 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 элемента 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 изменит 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.

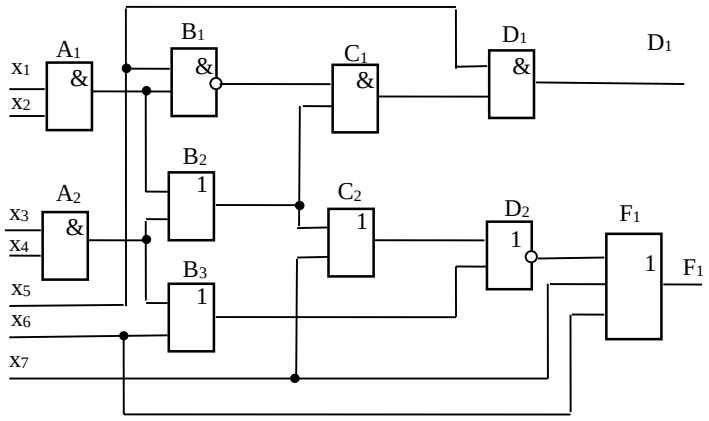


Рис. 3. Структурная реализация цифрового устройства
Fig. 3. Structural implementation of a digital device

В работе [9] предложен подход к определению множества линий структуры для кодирования, основанный на моделировании схемы с внесенной *i*-й неисправностью и вычислении признака $P_i = X_i \times Y_i$, характеризующего 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 $M = 2s \times 2^n$ раз, где s – общее количество линий схемы (переменных полного состояния схемы), n – количество входных переменных схемы. Для схемы на рис. 3 $M = 128 \times 34 = 4\,352$. 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 [9] (в этом случае M = 200k)

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

В отличие от решения, принятого в работе [9], рассмотрим более эффективный подход, основанный на применении метода сквозного вычисления неисправностей, покрываемых рассматриваемым входным вектором (метод конкурентно-дедуктивного моделирования) вместо моделирования каждой неисправной модификации схемы на определенном множестве случайных входных наборов с целью оценки степени влияния неисправностей на выходы схемы [10]. Метод конкурентно-дедуктивного моделирования неисправностей основан на моделировании исправной схемы и позволяет за один проход моделирования определять 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 преимущественные линии схемы для вставки ключевых вентилей. В то же время численное ограничение количества моделируемых входных воздействий [12] ограничивает возможность поиска наиболее эффективного решения.

Здесь предлагается другой подход, основанный на построении теста в классе неисправностей константного типа [10] и его применении на первом этапе кодирования. В рамках данного подхода вместо использования заранее определенного числа случайных входных воздействий (как, например, 100 в работе [11]) применятся тестовая последовательность входных векторов, которая обеспечивает близкое к полному покрытию неисправностей константного типа кодируемой структуры.

В табл. 1 приведены результаты построения теста для схемы, приведенной на рис. 3, и соответствующие разностные неисправные функции. Первый столбец таблицы содержит входные наборы теста, последующие – идентификаторы неисправностей константного типа всех линий схемы и 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 (рис. 3).

Таблица 1

Таблица разностных неисправных функций для схемы на рис. 3

Неисправности / 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
Неисправности / 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**	

Первая строка таблицы содержит идентификаторы неисправностей, последующие – единичные значения разностных неисправных функций. Верхний индекс в обозначении разностной неисправной функции (1^2) указывает, что функция относится ко второму выходу схемы, т.е. F_1. Если неисправность обнаруживается не на одном, а, к примеру, на трех выходах, то верхний индекс может иметь вид $1^{2, 3, 5}$. Из табл. 1 видно, что размещение ключевого вентиля XOR на выходе элемента B_3 не имеет смысла, так как теста контроля неисправности const 0 на выходе элемента B_3 не найдено по причине его отсутствия. 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 приведена 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 большим 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.

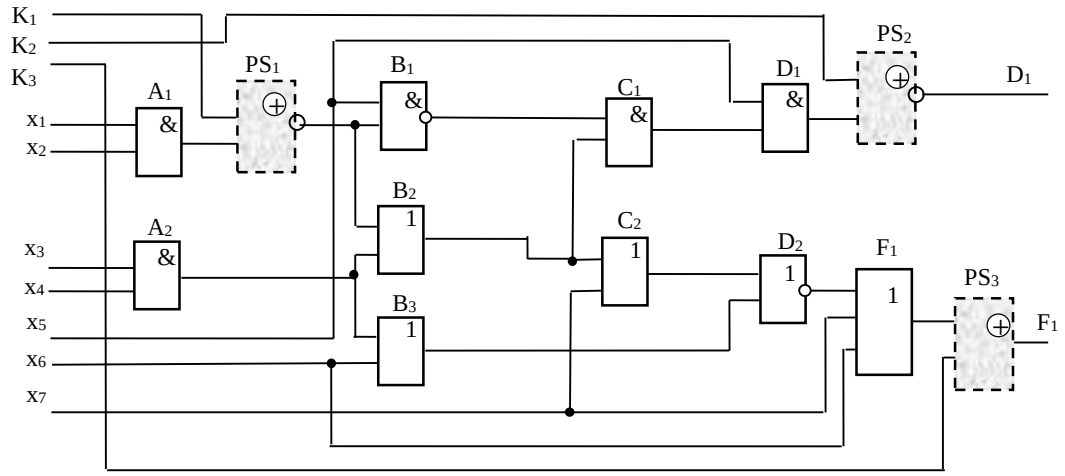


Рис. 4. Схема с вентилями PS_1, PS_2 и PS_3 для логического шифрования
Fig. 4. Circuit with PS_1, PS_2, and PS_3 gates for logical encryption

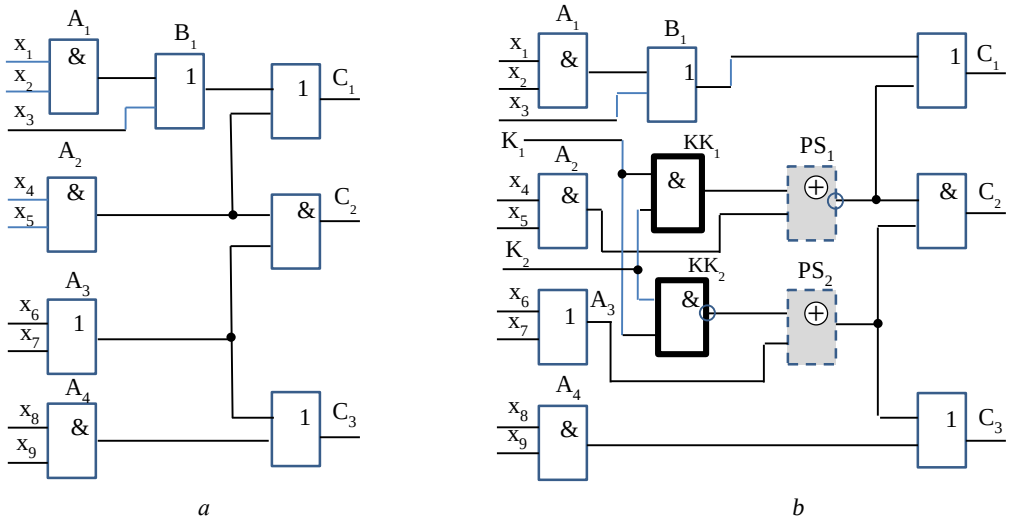
После добавления ключевых элементов в структуру необходимо проанализировать полученные результаты кодирования моделированием полученной частично закодированной структуры на наборах теста на всем булевом интервале множества ключевых входов и сравнением в каждом случае выходных реакций схемы с результатами моделирования исходной схемы. Как было указано 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 [7].

4. Управляемое кодирование цифровых устройств на структурном уровне

Очевидно, что результат кодирования проявляется на выходах схемы в зависимости от числа неправильных битов кода [9]. Если ключевой вентиль управляется одним битом ключевого кода, то 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. При этом, 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 при пятивходовом – 0,97 (только один ключевой вектор из 32 векторов данной группы является правильным).

На рис. 5, *a* приведена структура схемы с тремя выходами, а в табл. 2 – тестовая последовательность и соответствующие разностные неисправные функции. На рис. 5, *b* приведен пример двухуровневого кодирования. В соответствии с результатами табл. 2 в качестве линий для первоочередного включения ключевых вентилей для кодирования выбраны выходы элементов A2 (вентиль PS_1 типа XNOR) и A3 (вентиль PS_2 типа XOR). Тип ключевого вентиля XNOR на выходе элемента A2 выбирается в соответствии с неисправностью const 1, которая покрывается четырьмя из семи входными векторами 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 очувствляет (приводит к изменению) два выхода.

Дополнительно в схему включены управляющие двухвходовые вентили $КК_1$ и $КК_2$, которые усилили влияние на функционирование схемы каждого бита ключевого входа.



a *b*

Рис. 5. Пример схемы с двухуровневым кодированием:
a – логическая структура с тремя выходами; *b* – двухуровневое кодирование схемы
Fig. 5. An example of a scheme with two-level coding: *a* – logical structure with three outputs; *b* – two-level coding scheme

Таблица 2

Таблица разностных неисправных функций для схемы на рис. 5

Неисправности / 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											

Неисправности / 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	

Ниже приводятся основные этапы алгоритма управляемого логического кодирования комбинационных структур при использовании двухвходовых управляющих вентилей.

Исходные данные: описание кодируемой структуры схемы. **Результаты:** описание закодированной структуры схемы; правильный ключ.

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 элемент XNOR,

если неисправность const 1); включить управляющий вентиль с ключевым входом k_j; на второй вход управляющего вентиля подключить случайным образом дополнительный ключевой вход;

5) моделировать полученную структуру на всех наборах теста при всех возможных комбинациях значений ключа;

6) анализировать 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.

Заключение

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

Предложен алгоритм управляемого кодирования описаний цифровых устройств комбинационного типа на структурном уровне на основе применения средств тестового контроля. Предложенный алгоритм по сравнению с известными в литературе алгоритмами требует меньших вычислительных затрат и времени и проявляет устойчивость к восстановлению правильного ключа на основе «атаки SAT» [11]. Это обусловлено тем, что ключевые входы 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.

ЛИТЕРАТУРА

1. Zolotorevich L.A. Project verification and construction of superchip tests at the RTL level // Automation and Remote Control. 2013. V. 74, is. 1. P. 113–122.
2. 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.
3. Сергейчик В.В., Иванюк А.А. Методы лексической обфускации VHDL-описаний // Information Technologies and Systems 2013 (ITS 2013) : Proc. of The Int. Conference. BSUIR. Minsk, 2013. С. 198–199.
4. Shakya B., Salmani T.H., Forte D., Bhunia S., Tehranipoor M. Benchmarking of hardware Trojans and maliciously affected circuits // J. Hardw. Syst. Secur. (HaSS). 2017. V. 1 (1). P. 85–102.
5. 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. V. 22, No. 1. P. 1–23.
6. Dupuis S., Rouzeyre B., Flottes M.-L., Natale G.D., Ba P.-S. New Testing Procedure for Finding Insertion Sites of Stealthy Hardware Trojans // DATE: Design, Automation and Test in Europe. Grenoble, 2015. P. 776–781.
7. Roy J.A., Koushanfar F., Markov I.L. EPIC: Ending Piracy of Integrated Circuits // IEEE Computer. 2010. V. 43, No. 10. P. 30–38.
8. Chakraborty R.S., Bhunia S. Security against Hardware Trojan through a Novel Application of Design Obfuscation // IEEE/ACM Int. Conference on Computer-Aided Design. 2009. P. 113–116.
9. Karousos N., Pexaras K., Karybali I.G., Kalligeros E. Weighted Logic Locking: a New Approach for IC Piracy Protection // IEEE 23rd Int. Symposium on On-Line Testing and Robust System Design (IOLTS). 2017. P. 221–226.
10. Золоторевич Л.А. Исследование методов и средств верификации проектов и генерации тестов МЭС // Проблемы разработки перспективных микроэлектронных систем (МЭС-2006) : сб. науч. тр. всерос. науч.-техн. конф. / под общ. ред. А.Л. Стемпковского. М. : ИППМ РАН, 2006. С. 163–168.
11. Yasin M., Rajendran J., Sinanoglu O., Karri R. On Improving the Security of Logic Locking // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2016. V. 35, No. 9. P. 1411–1424.

Поступила в редакцию 29 мая 2019 г.

Zolotorevich L.A. (2020) HARDWARE PROTECTION OF DIGITAL DEVICES. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitelnaja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 50. pp. 69–78

DOI: 10.17223/19988605/50/9

In the last decade, the problem of protection and additional control of VLSI projects with the aim of detecting the consequences of unauthorized third-party interference in a project with different fundamental goals became urgent. Such actions are deliberate and carefully hidden, which prevents the direct application of existing methods for testing and functional control of VLSI.

The task of protecting projects of digital devices at a structural level from malicious misrepresentation and copyright infringement is considered. The algorithm of controlled coding of combinational structures, based on the use of methods and tools for test diagnostics, is proposed. The algorithm does not require the simulation of device malfunctions in an explicit form, which reduces the number of computational procedures for encoding the circuit. The features of SoC design are considered. Attention is focused on the need to create and develop a unified approach to reviewing the tasks of monitoring and verification of projects (taxonomy of deviations). Taxonomy deviations include the analysis of errors that occur directly in the design process, and deliberate distortion during the design and manufacturing stages.

Keywords: digital economy; VLSI design; copyright protection; coding logic circuits.

ZOLOTOREVICH Ludmila Andreevna (Candidate of Technical sciences, docent, Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus).
E-mail: zolotorevichla@bsuir.by

REFERENCES

1. Zolotorevich, L.A. (2013) Project verification and construction of superchip tests at the RTL level. *Automation and Remote Control*. 74(1). pp. 113–122. DOI: 10.1134/S0005117913010104
2. Rajendran, J., Sam, M., Sinanoglu, O. & Karri, R. (2013) Security analysis of integrated circuit camouflaging. *ACM SIGSAC conference on Computer & communications security*. Germany. Berlin. pp. 709–720.
3. Sergeychik, V.V. & Ivanyuk A.A. (2013) Metody leksicheskoy obfuskatsii VHDL-opisaniy [Methods of lexical obfuscation of VHDL-Descriptions]. *Information Technologies and Systems (ITS 2013): Proc. of The Int. Conference. BSUIR.* Minsk. pp. 198–199.
4. Shakya, B., Salmani, T.H., Forte, D., Bhunia, S. & Tehranipoor, M. (2017) Benchmarking of hardware Trojans and maliciously affected circuits. *Journal of Hardware and System Securuty (HaSS)*. 1(1). pp. 85–102. DOI: 10.1007/s41635-017-0001-6
5. Xiao, K, Forte, D, Jin, Y, Karri, R., Bhunia, S. & Tehranipoor, M. (2016) Hardware Trojans: Lessons learned after one decade of research. *ACM transactions on design automation of electronic system.* 22(1). pp.1–23. DOI: 10.1145/2906147
6. Dupuis, S., Rouzeyre, B., Flottes, M.-L., Natale, G.D. & Ba, P.-S. (2015) New Testing Procedure for Finding Insertion Sites of Stealthy Hardware Trojans. *DATE: Design, Automation and Test in Europe*. France. Grenoble. March 9–13, 2015. pp. 776–781.
7. Roy, J.A., Koushanfar, F. & Markov, I.L. (2010) EPIC: Ending Piracy of Integrated Circuits. *IEEE Computer*. 43(10). pp. 30–38.
8. Chakraborty, R.S. & Bhunia, S. (2009) Security against Hardware Trojan through a Novel Application of Design Obfuscation. *IEEE/ACM International Conference on Computer-Aided Design*. pp. 113116.
9. Karousos, N., Pexaras, K., Karybali, I.G. & Kalligeros, E. (2017) Weighted Logic Locking: A New Approach for IC Piracy Protection. *IEEE 23rd Int. Symposium on On-Line Testing and Robust System Design (IOLTS)*. pp. 221–226.
10. Zolotorevich, L.A. (2006) Issledovanie metodov i sredstv verifikatsii proektov i generatsii testov MES [Research of methods and means of project verification and generation of MES tests]. In: Stempkovsky, A.L. (ed.) *Problemy razrabotki perspektivnykh mikroelektronnykh sistem (MES-2006)* [Problems of Microelectronic System Development (MIC-2006)]. Moscow: RAS. pp. 163–168.
11. Yasin, M., Rajendran, J., Sinanoglu, O. & Karri, R. (2016) On Improving the Security of Logic Locking. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 35(9). pp. 1411–1424. DOI: 10.1109/TCAD.2015.2511144