ИНФОРМАТИКА

2015 **январь-март** **№ 1**

УДК 004.056.53

В.В. Сергейчик, А.А. Иванюк

ОБЗОР МЕТОДОВ РЕАЛИЗАЦИИ АППАРАТНЫХ ВОДЯНЫХ ЗНАКОВ В ЦИФРОВЫХ УСТРОЙСТВАХ ПРОГРАММИРУЕМОЙ ЛОГИКИ

Рассматривается применение технологии водяных знаков для защиты цифровых устройств и их проектных описаний. Приводятся основные определения, модели, категории атак, характеристики, классификация водяных знаков для данной области. Описываются примеры использования аппаратных водяных знаков.

Введение

Рост интереса к программируемым логическим устройствам (ПЛУ), и в частности к программируемым логическим интегральным схемам (ПЛИС), вызван сокращением разницы в производительности между заказными интегральными схемами (ИС) и ПЛУ, отсутствием в случае ПЛУ дорогой и продолжительной фазы изготовления проекта в кремнии (расходы на маску составляют 2 млн долл. для технологии 32 нм [1]), возможностями быстрого прототипирования. В 2010 г. были начаты около 110 000 проектов на базе ПЛИС и 2500 на базе заказных ИС [2].

Разрыв между конструированием и проектированием [3] осложняется высокими требованиями к производительности, функциональности, надежности цифровых устройств и скорости выхода на рынок. Одним из решений данной проблемы является методология повторного использования, основанная на применении готовых, оптимизированных и протестированных модулей, называемых компонентами интеллектуальной собственности (IP-компонентами) [4]. Механизмы приобретения и распространения IP-компонентов недостаточно отработаны: производители IP-компонентов оказываются уязвимыми перед пиратством, не имея возможности обнаружить и предотвратить неправомерное использование своих компонентов.

Угрозы включают: копирование, обратное проектирование, аппаратные трояны, извлечение секретной информации по сторонним каналам. В индустрии ущерб от пиратства и подделок оценивается в 169 млрд долл. в год [5]. Аппаратные трояны – это злонамеренные изменения схемотехники устройства с целью снижения уровня защиты, передачи секретной информации из работающего устройства, нарушения работоспособности. Извлечение секретной информации осуществляется путем измерения физических параметров схемы (тока, задержки, электромагнитного излучения), которые затем могут быть скоррелированы с внутренними вычислениями, секретными ключами. Обратное проектирование дает представление о внутреннем устройстве и функционировании схемы, облегчая внедрение трудно обнаружимых аппаратных троянов, проведение атак по сторонним каналам, клонирование и несанкционированное использование.

Существует несколько подходов к защите. При шифровании IP-компонент интегрируется в проект без раскрытия его внутренней структуры. Подход привязан к конкретным системам автоматизированного проектирования (САПР), что снижает гибкость процесса проектирования и открывает уязвимость перед атаками на САПР и ключи [2]. Примеры промышленного использования: Synopsis (DesignWare), Xilinx (Core Generator), Microsemi (Direct Cores). Второй вариант подхода – шифрование бит-образа – требует аппаратной поддержки в ПЛИС и уязвим к атакам на ключи по сторонним каналам. Аппаратные водяные знаки (АВЗ) – это цифровые водяные знаки (ЦВЗ), используемые для защиты цифровых устройств (далее в тексте термин ЦВЗ будет употребляться в определениях, в целом справедливых для водяных знаков). АВЗ скрывают информацию об авторстве внутри описания или схемы. Отпечатки пальцев – это АВЗ, уникальные для каждого пользователя, что позволяет отслеживать источник нелегального копирования. Суть идентификации состоит в доказательстве присутствия IP-компонента в про-

ектном описании путем сравнения характеристик проекта и характеристик компонента. Активное измерение использует уникальные неклонируемые различия, возникающие при производстве ИС. Компонент начинает работу в нефункциональном состоянии, определяемом модулем физически неклонируемой функции. Для перевода в стартовое состояние требуется подача входной последовательности, различной для каждой ИС. В отличие от АВЗ, помечающих проектное описание, а не ИС, при пассивном измерении помечаются конкретные ИС с целью последующего наблюдения или отслеживания. Обфускация скрывает смысл описания, структуры или функционирования схемы, усложняя понимание проектного описания и схемы, а также обратное проектирование. Примерами использования обфускации могут служить IP-компоненты Microsemi Direct Cores [6].

1. Модель АВЗ

Существуют различные определения ЦВЗ. Например, ЦВЗ – это технология, обеспечивающая безопасность, идентификацию и защиту авторского права для цифровых данных [7]. ЦВЗ – это данные, внедряемые в информационный объект с целью контроля его использования [8]. ЦВЗ представляет собой метод встраивания информации, применяемый с определенной целью, например для идентификации и защиты авторского права [3]. Для лучшего понимания текста вводятся следующие определения: контейнер – информационный объект, в котором скрыт ЦВЗ [8]; сообщение – встраиваемые данные; сторона – лицо или группа лиц, осуществляющих общую деятельность: владелец IP-компонента, пользователь, поставщик IP-компонентов.

В коммуникационных моделях технология ЦВЗ рассматривается как передача проверяющей стороне сообщения от встраивающей стороны. В зависимости от степени использования моделью свойств контейнера он рассматривается как шум, шум со вспомогательной информацией, другое информационное сообщение, передаваемое вместе с ЦВЗ путем мультиплексирования [9].

Процедура использования АВЗ состоит из двух этапов: встраивания и извлечения.

В ходе этапа встраивания из исходного проектного описания V с помощью метода Wm постановки АВЗ и сообщения K, идентифицирующего автора, получают описание V^*, содержащее некоторое свойство (инвариантное преобразованиям, производимым при переходе к более низким уровням абстракции и при оптимизациях), которое позволяет доказать авторство: $V^* = Wm(V, K)$. Под оптимизациями здесь и далее понимаются оптимизационные преобразования, осуществляемые САПР на различных этапах проектирования.

Подготовка встраиваемого сообщения часто включает хеширование, шифрование, генерацию псевдослучайной последовательности, добавление помехоустойчивых кодов. Результатом процедуры синтеза DD описаний V и V^* являются схемные представления Sch и Sch^*, различающиеся, но соответствующие одной и той же функциональной спецификации $func$:

$$DD(V) = Sch;\ DD(V^*) = Sch^*;\ \ func(Sch^*) = func(Sch).$$

В ходе извлечения АВЗ процедурой обнаружения D определяется присутствие или отсутствие сообщения K в описании и (или) синтезированной из него схеме:

$$D(V^*, K) = true;\ D(Sch^*, K) = true.$$

Доказательство авторства при использовании АВЗ опирается на аппарат теории вероятностей. Важнейшей характеристикой этапа встраивания является вероятность совпадения P_u, указывающая на возможность того, что незапланированный АВЗ будет обнаружен в проектном описании [10]. Длина последовательности сообщения АВЗ выбирается в соответствии с требуемым значением P_u. Фаза извлечения характеризуется двумя величинами: P_m – вероятностью не обнаружить существующий АВЗ и P_f – вероятностью ложного обнаружения.

Важнейшим требованием к АВЗ является сохранение функциональной корректности проектного описания. Выделяют следующие основные характеристики АВЗ [11]. *Стой-*

кость – устойчивость к искажениям (в том числе вызванным синтезом, оптимизациями) и атакам. Чтобы увеличить стойкость, АВЗ желательно сделать функциональной частью IP-компонента. *Емкость* – максимальное количество данных, которые можно внедрить в контейнер. *Затраты на встраивание (извлечение)* – вычислительная сложность встраивания (извлечения), необходимость наличия дополнительного оборудования, технологий, экспертов. *Вносимые издержки* – степень ухудшения качества помеченного проектного описания (и синтезированной из него схемы) по сравнению с проектным описанием (схемой) до внедрения. *Скрытность* – степень сходства свойств (в том числе статистических) АВЗ со свойствами окружающей области внедрения [12]. *Дополнительные характеристики*: доказательство происхождения (проектное описание было создано конкретной стороной, а не просто содержит АВЗ, указывающий на эту сторону), доказательство целостности (отсутствия вмешательства в проектное описание после создания), возможность опровержения (см. ниже атаку подделыванием авторства), невозможность отрицания передачи IP-компонента другим сторонам. Важной характеристикой АВЗ является *прозрачность* для средств проектирования [13]: этап внедрения АВЗ должен легко интегрироваться в процесс проектирования.

В модели оценки [14] производительность АВЗ вычисляется как функция пяти переменных: Em_Cost, Trace_Cost – стоимость встраивания, стоимость извлечения (время работы алгоритма на компьютере), Coin_Pro – вероятность совпадения, Security – устойчивость к конкретным видам атак, Overhead – увеличение аппаратных издержек.

Атаки на ЦВЗ делятся на четыре категории [9]:

1. К категории *неавторизованного удаления* относятся маскирующие атаки и атаки удалением. Маскирующие атаки затрудняют обнаружение и извлечение АВЗ, не изменяя его. Атаки удалением направлены на разрушение, ухудшение качества АВЗ до уровня, когда он больше не может использоваться для доказательства авторства.

2. *Неавторизованное встраивание* включает атаки встраиванием и атаки копированием. Атаки встраиванием направлены на добавление другого (не авторского) АВЗ в проектное описание. В атаках копированием (подделыванием авторства) злоумышленник внедряет АВЗ владельца, извлеченный из одного проектного описания, в другое проектное описание, утверждая, что это проектное описание создано владельцем, например, с целью дискредитации последнего.

3. Иногда возможность извлечения сообщения АВЗ или различения его частей (например, идентификаторов конкретных пользователей) должна быть доступна только ограниченному кругу лиц. Нарушение таких ограничений – это *атака неавторизованного обнаружения*.

4. *Атаки системного уровня* направлены на ошибки использования АВЗ, ошибки реализации, ключи или на саму концепцию как таковую.

2. Классификация методов АВЗ

Приведем классификацию методов АВЗ (рис. 1). АВЗ делят на статические и динамические [15]. Динамические строятся во время функционирования IP-компонента и представляют собой некоторое свойство его состояния. Эти методы особенно перспективны в силу того, что свойство проявляется только при подаче определенных входных данных. Статические представляют собой свойство проектного описания некоторого уровня абстракции.

АВЗ классифицируются по типу IP-компонента, используемого как контейнер. Выделяют программные, аппаратные и фирменные IP-компоненты [4]. Программные IP – это высокоуровневые описания, аппаратные IP – синтезированные описания после трассировки и размещения, фирменные IP – описания в форме списка соединений элементов для целевой ПЛИС до трассировки и размещения.

По устойчивости к изменению АВЗ делятся на устойчивые и хрупкие. В [16] предлагается метод встраивания символов хрупкого АВЗ в выходные последовательности КА. Хрупкие АВЗ не используются для доказательства авторства, а служат индикатором изменения или повреждения проектного описания. Изменение IP-компонента приводит к разрушению хрупких АВЗ.

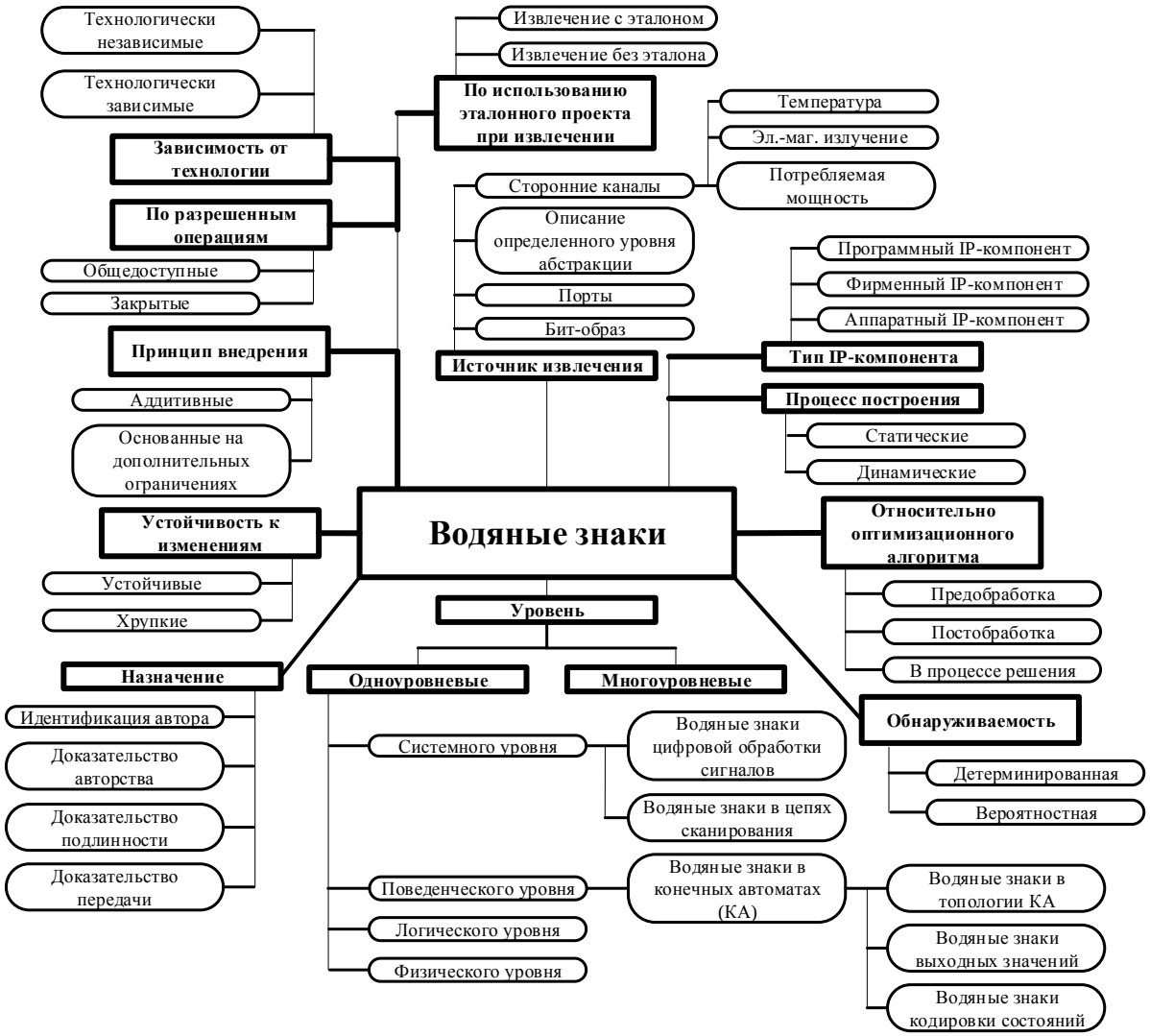


Рис. 1. Классификация методов АВЗ

По времени встраивания АВЗ относительно алгоритма решения оптимизационной задачи методы делят на три группы: предобработку, постобработку и методы в процессе решения [14, 17]. Методы предобработки внедряют АВЗ в описание до передачи алгоритму решения оптимизационной задачи. АВЗ, сохранившийся после решения оптимизационной задачи, труднее удалить. В методах постобработки сначала решается оптимизационная задача, затем в результат решения добавляется АВЗ. Легкость реализации таких методов компенсируется тем, что они в большей степени подвержены атакам. В методах третьей группы встраивание АВЗ является частью алгоритма оптимизации и осуществляется во время решения оптимизационной задачи. Сложность представляет интеграция в процесс разработки.

В зависимости от необходимости использования при извлечении эталона (проектного описания до встраивания водяного знака) АВЗ делятся на слепые и неслепые [9]. В слепых методах АВЗ может быть извлечен без эталона. В неслепых методах проводится сравнение, ищется корреляция свойств проверяемого и эталонного описаний.

По критерию обнаружения методы делят на детерминированные и вероятностные. В детерминированных методах при проверке АВЗ требуется побитовое равенство, также возможно использование пороговых значений. Вероятностные методы опираются на некоторое статистическое свойство.

АВЗ делят на общедоступные и закрытые в зависимости от доступности операции обнаружения. В закрытых АВЗ лишь привилегированная сторона (например, владелец IP-компонента) может извлечь и декодировать исходное сообщение.

Методы, зависящие от конструктивных особенностей конкретной платформы ПЛИС, называют технологически зависимыми, остальные – технологически независимыми.

По принципу внедрения выделяют две основные группы методов: аддитивные (основанные на добавлении новых элементов или свойств в проектное описание) и основанные на введении дополнительных ограничений.

По уровню абстракции проектного описания АВЗ для ПЛИС делятся на следующие группы: системного, поведенческого, логического, физического уровней и многоуровневые. Процесс проектирования и место АВЗ в нем показаны на рис. 2. Описания более низких уровней представляют собой детализацию более высоких уровней, поэтому АВЗ более высокого уровня сохраняются в описаниях низкого уровня, т. е. содержатся (в разной форме) в нескольких контейнерах. В случае обратного проектирования при достижении более высокого уровня абстракции АВЗ низших уровней не сохраняются, так как изначально на этом уровне не существует нужный контейнер.

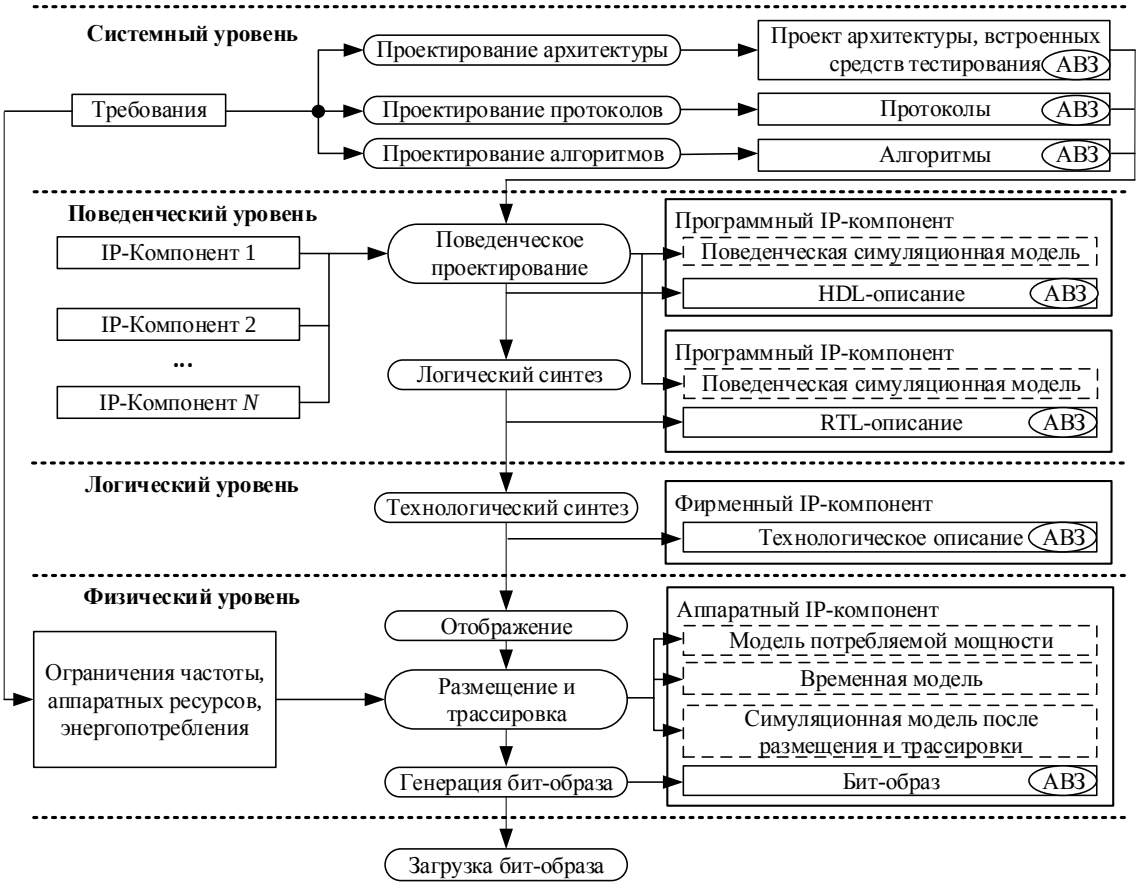


Рис. 2. Процесс проектирования цифровых устройств

2.1. Классификация методов по уровню абстракции

АВЗ системного уровня. Методы системного уровня используют особенности реализуемого алгоритма, предметной области, архитектуры разрабатываемого цифрового устройства. В [18] описывается метод встраивания АВЗ в спецификацию цифрового фильтра на алгоритмическом уровне путем разбиения полосы пропускания на отрезки с увеличением или уменьшением амплитуды в пределах отрезка на величину Δ в соответствии с битом АВЗ. Если бит АВЗ равен единице, то единица отнимается от верхней границы амплитуды отрезка; в противном случае единица прибавляется к нижней границе [19].

Следующая группа методов предоставляет доступ к АВЗ через встроенные средства тестирования и самотестирования. Метод [20] основан на NP-сложной проблеме упорядочивания сканирующих ячеек с целью минимизации времени или потребляемой мощности в ходе сканирующего теста. Перестановка π – это отображение один к одному набора сканирующих ячеек $R = \{r_i\}$ на набор позиций $P = \{p_j\}$ так, что j-й бит тестового вектора загружается в i-ю сканирующую ячейку, как только полный тестовый вектор установлен в сканирующую цепь, где $i, j = 1, 2, \ldots, N$. В процессе оптимизации вводится ограничение на значения определенных сканирующих ячеек для конкретных тестовых векторов, при подаче которых на выходе сканирующей цепи получается ответ, содержащий биты АВЗ на заданных позициях.

В работе [21] используются несколько сканирующих цепей. При внедрении вычисляется хэш-значение сообщения. Для расщепления хэш-значения на группы генерируется последовательность случайных чисел $R_n = \{r_1, r_2, \ldots, r_n\}$, представляющих число битов в группах. Биты групп представляются в виде целого числа – номера сканирующей ячейки в сканирующей цепи, значение которого будет инвертировано в режиме извлечения водяного знака (высокий уровень сигнала wmEn). При извлечении один и тот же вектор подается на вход сканирующей цепи в обоих режимах (wmEn = 1 и wmEn = 0). Определяются номера позиций, в которых ответы различаются. С помощью R_n из номеров восстанавливаются биты соответствующей группы водяного знака.

АВЗ поведенческого уровня. Среди АВЗ поведенческого уровня лучше изучены методы, связанные со свойствами конечных автоматов (КА). Методы рассматриваются на примере КА (рис. 3, *а*).

Идея метода [22] заключается в таком преобразовании графа передачи состояний (ГПС), что последовательность состояний r_i, посещаемая при подаче бит АВЗ a_i, имеет специальные топологические свойства: каждое r_i может быть достигнуто только из r_{i-1} и только при подаче на вход a_i. Для этого ГПС при построении дублируется, а между копиями создается единственный путь (рис. 3, *в*). Проверка присутствия АВЗ сводится к подтверждению с помощью техник тестирования наличия указанных топологических свойств для последовательности АВЗ.

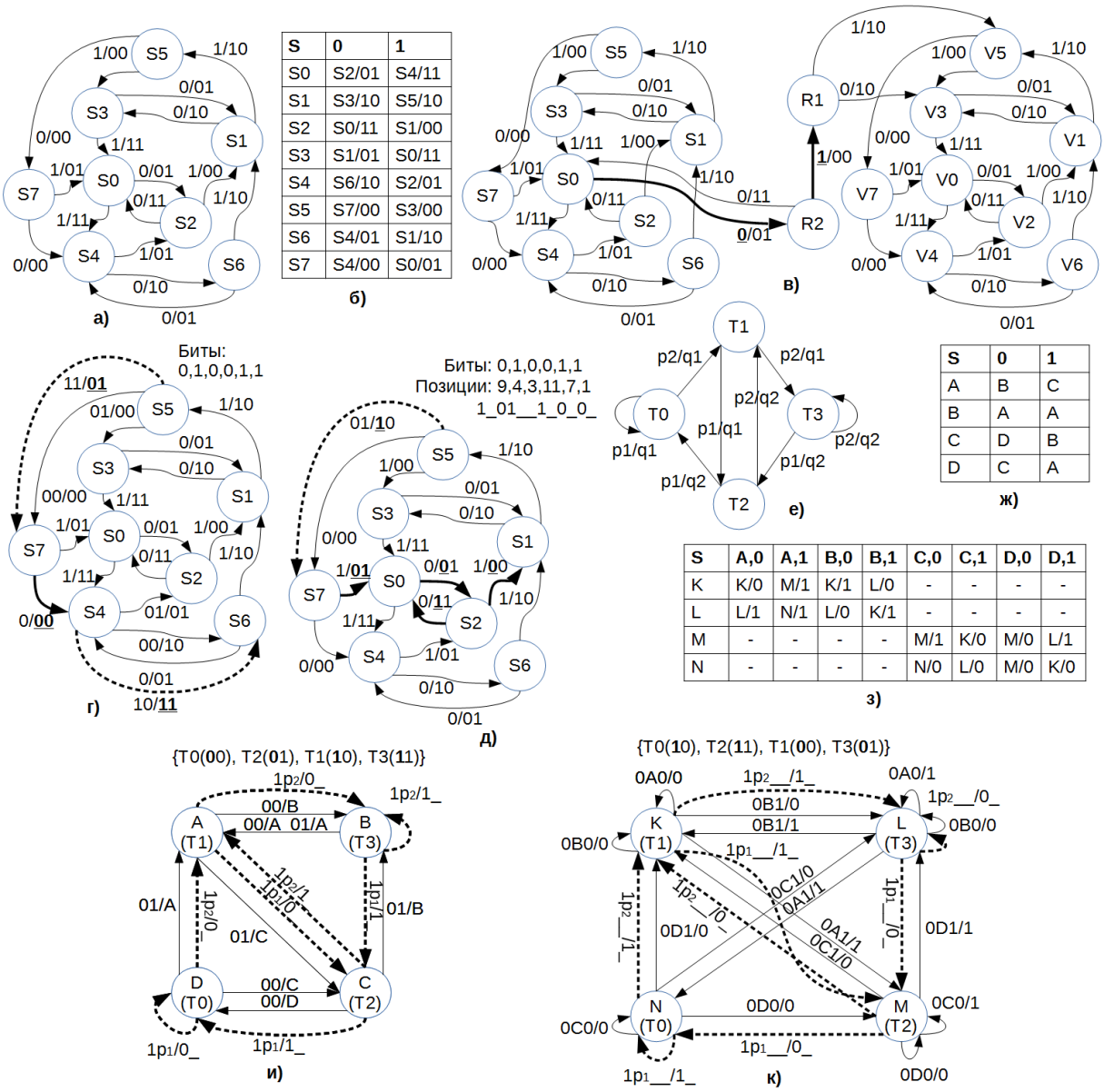
Метод, описанный в [10], опирается на поиск неиспользуемой последовательности входных и выходных символов, соответствующей АВЗ. Если пар входных и выходных символов недостаточно, то вводятся новые входные переменные. Авторы используют геномный поиск для вероятностного подтверждения присутствия АВЗ в случае его повреждения.

Подобно [10], в [15] в качестве АВЗ используется последовательность входных и выходных символов. Встраивание начинается с произвольного состояния, при этом не осуществляется ресурсоемкий поиск путей в ГПС, а выбираются переходы, выходные символы которых совпадают с битами АВЗ. Если совпадений нет, то добавляются новые переходы, если больше переходов добавить нельзя, то вводится новый входной символ. Символ принимает значение 0 для существующих состояний и 1 для добавляемых (рис. 3, *г*).

В [23] предусматривается несколько модификаций метода [15] (рис. 3, *д*). Генерируется псевдослучайная последовательность индексов позиций. Начиная с произвольного состояния, выбираются переходы, в которых биты выходных значений, заданных индексами, совпадают с битами АВЗ под теми же индексами. Это позволяет лучше защитить АВЗ, повысить вероятность нахождения существующего перехода с требуемыми выходными сигналами, снизить издержки и увеличить эффективность внедрения в КА с большим количеством выходных переменных. Добавленным входным переменным присваиваются произвольные значения, а в [15] требуются фиксированные, что может облегчить проведение атак.

Метод [24] внедрения АВЗ в кодировку состояний КА в методологии тестопригодного проектирования опирается на широкое использование существующей проектной логики для реализации тестовой логики, помеченной АВЗ. Последовательностная схема из N триггеров может быть представлена в виде множества связанных объектных автоматов (ОА). Каждому ОА из n триггеров можно поставить в соответствие обобщенный тестовый автомат (ТА) с двумя входными, двумя выходными переменными и 2^n состояниями (рис. 3, *е*). ТА устанавливается в любое состояние синхронизирующей последовательностью длины n, состояние на выходе идентифицируется с помощью разделяющей последовательности длины n. АВЗ представляет собой ограничения на кодировку состояний ТА, выбранных псевдослучайно среди всех ТА проектного описания. Например, значение первого триггера для первого состояния может быть равно значению

бита водяного знака. Устанавливая ТА в определенные состояния и проверяя выходные значения, можно определить кодировку, выбранную для конкретного автомата, а следовательно, и биты водяного знака. Декомпозиция автомата *а* на два ОА показана на рис. 3, *ж*, *з*, результирующие КА со встроенными ТА и АВЗ 01 – на рис. 3, *и*, *к*.



S	0	1
S0	S2/01	S4/11
S1	S3/10	S5/10
S2	S0/11	S1/00
S3	S1/01	S0/11
S4	S6/10	S2/01
S5	S7/00	S3/00
S6	S4/01	S1/10
S7	S4/00	S0/01

S	0	1
A	B	C
B	A	A
C	D	B
D	C	A

S	A,0	A,1	B,0	B,1	C,0	C,1	D,0	D,1
K	K/0	M/1	K/1	L/0	-	-	-	-
L	L/1	N/1	L/0	K/1	-	-	-	-
M	-	-	-	-	M/1	K/0	M/0	L/1
N	-	-	-	-	N/0	L/0	M/0	K/0

Рис. 3. Примеры методов АВЗ: *а), б)* исходный КА и его таблица переходов и выходов; *в)* метод [22]; *г)* метод [15]; *д)* метод [23]; *е)* ТА; *ж), з)* результат декомпозиции *а*; *и), к)* декомпозированные компоненты со встроенными ТА и АВЗ [24]

В [25] используется концепция иерархических КА, в соответствии с которой выделяют главный и подчиненный КА. Входной и выходной алфавиты подчиненного КА выступают подмножествами соответствующих алфавитов главного. Подчиненный КА отвечает на реакцию главного. Реакция иерархического КА определяется следующим образом: если данное состояние простое, то иерархический КА ведет себя как обычный КА, в противном случае реагируют и подчиненный и главный КА, т. е. переключаются два состояния и выполняются два действия. Для постановки АВЗ некоторые состояния расщепляются, создается подчиненный КА, добавляются ложные переходы. АВЗ извлекают из выходных значений переходов аналогично [15].

Среди немногочисленных не связанных с КА методов следует упомянуть внедрение АВЗ в поведенческое описание в ходе высокоуровневого синтеза [17]. В процессе синтеза устройство управления изменяется так, чтобы транслировать выбранные внутренние промежуточные значения на выходные порты в течение выбранных свободных временных окон.

АВЗ логического уровня. В работе [13] АВЗ внедряется в неиспользуемые и частично занятые LUT (Look Up Table) на уровне списка соединений элементов (при этом LUT преобразуются в сдвиговые регистры SRL16 для предотвращения минимизации в процессе синтеза). К неиспользуемым входам SRL16 подключается сигнальный источник 0 для уменьшения его динамически адресуемого пространства. Свободные биты SRL16 заполняются битами АВЗ. Метод затрагивает функциональные элементы проектного описания, что затрудняет удаление АВЗ. При этом снижается скрытность: заземленные выводы SRL16 достаточно заметны. АВЗ извлекается путем поиска по содержимому LUT в бит-образе или путем поиска по заданным позициям.

АВЗ физического уровня. В работе [13] предлагается внедрение АВЗ в неиспользуемые LUT бит-образа, что требует знания элементов формата бит-образа. В [26] АВЗ внедряется в неиспользуемые LUT технологического описания, затем осуществляется размещение элементов проекта вокруг LUT с АВЗ. В [27] биты АВЗ внедряются в ILUT (ILUT – нефункциональный, пустой LUT в используемом CLB (Configurable Logic Block)). Для обеспечения дополнительной скрытности ILUT соединяются с неиспользуемыми входами функциональных LUT.

В [28] биты АВЗ и отпечатка пальца встраиваются в неиспользуемые LUT. Группы символов отпечатка и АВЗ размещаются в одинаковой позиции относительно друг друга: АВЗ – в первом, а отпечаток – во втором LUT неиспользуемого CLB. Тем самым облегчается извлечение АВЗ и отпечатка пальца. Концепция плиток применяется для преодоления уязвимости к атаке сговором. Плитка – это секция из CLB, среди которых хотя бы один неиспользуемый. Плитка 2x2 имеет четыре конфигурации (по положению неиспользуемого CLB). САПР размещает такой CLB в произвольной позиции, соединяя его выводы с безразличными выводами соседних элементов, а затем вокруг него – остальные CLB, входящие в плитку. Для каждого пользователя генерируется уникальный вариант проектного описания. Плитки сокращают время генерации с помощью САПР.

В [29] биты АВЗ вносятся в неиспользуемые позиции бит-образа. АВЗ не связан с исходным проектом, что упрощает его удаление. Приводится метод внедрения АВЗ в биты конфигурации мультиплексоров в неиспользуемых CLB.

Многоуровневые, иерархические методы. После того как IP-компонент интегрирован в систему на кристалле и упакован, извлечение информации об авторстве непосредственно в рабочих условиях затруднено. Многие системы снабжены средствами тестирования или самотестирования, связывающими порты с внутренними компонентами. Постановка АВЗ непосредственно на цепи сканирования выглядит недостаточно надежной мерой из-за того, что цепи добавляются на конечных стадиях разработки и могут быть удалены или заменены сравнительно легко. Поэтому появляются гибридные, или иерархические, методы, оперирующие на различных уровнях абстракции и стадиях разработки.

В работе [30] АВЗ вносятся на нескольких уровнях абстракции: на поведенческом уровне помечаются КА путем встраивания в выходные сигналы существующих и неспецифицированных переходов [23], в ходе тестопригодного проектирования помечается сканирующая цепь [20]. Подключение КА к сканирующей цепи для тестирования и пометка этой цепи АВЗ позволяют легко извлекать водяной знак в ходе функционирования. Здесь сочетаются два метода: более уязвимый для атак, но и более простой для извлечения метод встраивания АВЗ в сканирующую цепь и более устойчивый к атакам, но и более сложный для извлечения метод встраивания АВЗ в переходы КА. Менее надежный АВЗ дополнительно используется в качестве хрупкого, указывая на попытки повреждения проектного описания.

2.2. Источники извлечения АВЗ

В случае бит-образа возникают две проблемы: формат бит-образа держится в секрете производителями ПЛИС; в некоторых ПЛИС предусматривается шифрование бит-образа с запретом чтения. Пример извлечения АВЗ из LUT в бит-образе ПЛИС Xilinx приведен в [13].

Сторонние каналы интересны тем, что открывают возможность извлечения АВЗ из упакованного устройства даже в случае зашифрованных списков соединений элементов и за-

шифрованных бит-образов [13]. Среди сторонних каналов можно выделить потребляемую мощность, температуру, электромагнитное излучение. Необходимость применения дополнительного оборудования (например, высокоточного осциллографа) является недостатком таких источников извлечения.

Извлечение АВЗ в случае использования стороннего канала потребляемой мощности осуществляется путем модуляции ее профиля в соответствии с битами АВЗ. В [13] в качестве модулятора используется потребитель мощности (сдвиговый регистр). Если бит водяного знака равен единице, то сдвиговый регистр выполняет сдвиг, создавая пик потребляемой мощности; если бит равен нулю, то сдвига не происходит. В [31] предлагается усовершенствованная по ресурсам ПЛИС модификация метода, в которой в качестве модулятора используются буферы синхронизации, потребляющие 50 % динамической мощности ПЛИС. Эксперименты [13] показывают, что потребляемая мощность как источник извлечения водяных знаков имеет достаточно высокий уровень шума, поэтому требуется поиск способов модуляции и кодирования АВЗ в сигнале.

В случае электромагнитного излучения извлечение осуществляется так же, как и в случае извлечения АВЗ из потребляемой мощности [13], однако дополнительная информация о пространственном расположении источника АВЗ повышает точность извлечения. Извлечение через такой источник может осложняться наличием металлического корпуса, не пропускающего излучение, а также необходимостью дополнительного оборудования.

Проблема применения температуры как источника извлечения водяных знаков слабо освещена в литературе. В [32] описан метод использования пассивной температурной метки. Имеется внешний управляемый источник тепла, например лампа накаливания, а в цифровом устройстве находится компонент пассивной тепловой отметки, который при повышении температуры включает остальную схему. Синхронизатор определяет период передаваемого сообщения, декодер в зависимости от относительного изменения температуры на каждом такте передачи возвращает бит 0 или 1. При равенстве переданной и хранимой меток схема генерирует ответ, например отключает компонент. Недостатки данного метода: необходимость дополнительного оборудования, высокое время извлечения (16 мин для 64 битов), необходимость знания метки до извлечения для устройства (или полный перебор меток).

Порты ввода-вывода не требуют дополнительного оборудования или специфических знаний в отличие от бит-образа. Однако использовать порты не всегда возможно, потому что IP-компонент может оказаться полностью внутренним и непосредственно не контактировать с ними. В таком случае можно применить методы тестопригодного проектирования, предполагающие использование встроенных средств тестирования с доступом к тестовым портам.

Источником извлечения многих статических АВЗ являются проектные описания. Этот источник характеризуется простотой извлечения и легкостью проведения атак на АВЗ при условии, что описание незашифровано.

3. Актуальные проблемы в области АВЗ

В настоящее время важнейшей проблемой является отсутствие методологии оценки качества АВЗ. Сравнение АВЗ необходимо не только для выбора более эффективного метода, но и для разрешения споров об авторстве, когда в проектном описании содержатся АВЗ нескольких сторон. В [33] описаны разрозненные критерии оценки, в [14] предложена методология оценки, однако практически отсутствуют методы измерения конкретных критериев. Например, в [14] стоимость встраивания приравнивается к стоимости извлечения и измеряется в процессорном времени работы алгоритмов внедрения и извлечения без учета необходимости дополнительного оборудования и экспертов. Метрикой оценки издержек выступает количество проводников и новых элементов, при этом остается в стороне возможное изменение производительности, энергопотребления, надежности и других свойств проекта. Не исследован вопрос оценки стоимости проведения атак: может оказаться, что атака вполне достижима и реализуема, однако дороже, чем законное приобретение IP-компонента. Не изучены способы оценки скрытности АВЗ в проектных описаниях. Скрытность увеличивает стоимость осуществления атаки.

Важное значение имеет проблема извлечения АВЗ из упакованных IP-компонентов. Первые шаги в этом направлении представлены методами АВЗ, основанными на тестировании,

а также извлечении по сторонним каналам. Практически не развиты многоуровневые, гибридные методы, обеспечивающие комплексную защиту на разных уровнях абстракции.

Заключение

ЦВЗ широко применяются для защиты авторских прав на мультимедийные данные. Использование ЦВЗ для защиты цифровых устройств ставит ряд новых задач: внедрение ЦВЗ без нарушения функционирования, обеспечение устойчивости перед алгоритмами синтеза и оптимизациями, гарантирование высокой достоверности и низкого уровня проектных издержек. Рассмотренные методы успешно решают некоторые из этих задач, однако они все еще далеко не универсальны.

Список литературы

1. Architecture and Design Flow for a Highly Efficient Structured ASIC / H. Man-Ho [et al.] // IEEE Transactions on VLSI Systems. – 2012. – Vol. 21, iss. 3. – P. 424–433.

2. Majzoobi, M. Introduction to hardware security and trust / M. Majzoobi, F. Koushanfar, M. Potkonjak. – N.Y. : Springer, 2011. – 427 p.

3. Qu, G. Intellectual Property Protection in VLSI Design Theory and Practice / G. Qu, M. Potkonjak. – Dordrecht : Kluwer Publishing, 2003. – 203 p.

4. System-on-Chip: Reuse and Integration / R. Saleh [et al.] // Proceedings of the IEEE. – 2006. – Vol. 94, no. 6. – P. 1050–1069.

5. Can EDA Combat the Rise of Electronic Counterfeiting? / F. Koushanfar [et al.] // Design Automation Conference. – San Francisco, USA, 2012. – P. 133–137.

6. IP DirectCores. Microsemi [Electronic Resource]. – Mode of access : http://www.microsemi.com/products/fpga-soc/design-resources/ip-cores/direct-cores. – Date of access : 5.01.2015.

7. Singh, P. A Survey of Digital Watermarking Techniques, Applications and Attacks / P. Singh, R. Chadha // Intern. J. of Engineering and Innovative Technology. – 2013. – Vol. 2, iss. 9. – P. 165–175.

8. Защелкин, К. Метод внедрения цифровых водяных знаков в аппаратные контейнеры с LUT-ориентированной архитектурой / К. Защелкин, Е. Иванова // Информатика и математические методы в моделировании. – 2013. – Т. 3, № 4. – С. 369–384.

9. Digital Watermarking and Steganography / I. Cox [et al.]. – Burlington : Elsevier, 2008. – 587 p.

10. Torunoglu, I. Watermarking-Based Copyright Protection of Sequential Functions / I. Torunoglu, E. Charbon // IEEE J. of Solid-State Circuits. – 2000. – Vol. 35. – P. 434–440.

11. A Survey of Techniques for VLSI Protection / W. Liang [et al.] // Information Technology Journal. – 2013. – Vol. 12. – P. 2324–2332.

12. Collberg, C. Software Watermarking: Models and Dynamic Embeddings / C. Collberg, C. Thomborson // Proc. of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages. – N.Y., 1999. – P. 311–324.

13. Ziener, D. Techniques for Increasing Security and Reliability of IP Cores Embedded in FPGA and ASIC Designs / D. Ziener. – Erlangen, 2010. – 325 p.

14. Watermarking / T. Nie [et al.]. – Rijeka : InTech, 2012. – 276 p.

15. A Public-Key Watermarking Technique for IP Designs / A. Abdel-Hamid [et al.] // Design, Automation and Test in Europe, Proceedings. – 2005. – Vol. 1. – P. 330–335.

16. Abdel-Hamid, A. Fragile IP Watermarking Techniques / A. Abdel-Hamid, S. Tahar // NASA Conference on Adaptive Hardware and Systems. – Noordwijk, Netherlands, 2008. – P. 513–519.

17. Bossuet, L. Automatic low-cost IP watermarking technique based on output mark insertions / L. Bossuet, B. Gal // Design Automation for Embedded System. – 2012. – Vol. 16. – P. 71–92.

18. Rashid, A. Hierarchical Watermarking for Protection of DSP Filter Cores / A. Rashid, W. Mangione-Smith, M. Podkonjak // IEEE Custom Integrated Circuits Conference. – San Diego, USA, 1999. – P. 39–42.

19. Chapman, R. IP Protection of DSP Algorithms for System on Chip Implementation / R. Chapman, T. Durrani // IEEE Transactions on Signal Processing. – 2000. – Vol. 48. – P. 854–861.

20. Cui, A. An Improved Publicly Detectable Watermarking Scheme Based on Scan Chain Ordering / A. Cui, C.H. Chang // IEEE ISCAS, 2009. – Taipei, 2009. – P. 29–32.

21. Sequential Circuit-Based IP Watermarking Algorithm for Multiple Scan Chains in Design-for-Test / W. Liang [et al.] // Radioengineering. – 2011. – Vol. 20. – P. 533–539.

22. Oliveira, A. Robust Techniques for Watermarking Sequential Circuit Designs / A. Oliveira // Design Automation Conference. – New Orleans, USA, 1999. – P. 837–842.

23. A Robust FSM Watermarking Scheme for IP Protection of Sequential Circuit Design / A. Cui [et al.] // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2011. – Vol. 30, no. 5. – P. 678–690.

24. Zhang, L. State Encoding Watermarking for Field Authentication of Sequential Circuit Intellectual Property / L. Zhang, C. H. Chang // IEEE ISCAS, 2012. – Seoul, 2012. – P. 3013–3016.

25. Meenakumari, M. Improving the Protection of FPGA Based Sequential IP Core Designs Using Hierarchical Watermarking Technique / M. Meenakumari, G. Athisha // J. of Theoretical and Applied Information Technology. – 2014. – Vol. 63, no. 3. – P. 701–708.

26. Lach, J. Signature Hiding Techniques for FPGA Intellectual Property Protection / J. Lach, W. Mangione-Smith, M. Podkonjak // IEEE/ACM Intern. Conf. on Computer-Aided Design. – San Jose, USA, 1998. – P. 186–189.

27. Watermarking FPGA Bitfile for Intellectual Property Protection / J. Zhang [et al.] // Radioengineering. – 2012. – Vol. 21, iss. 2. – P. 764–771.

28. Lach, J. Fingerprinting Techniques for Field-Programmable Gate Array Intellectual Property Protection // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2001. – Vol. 20, no. 10. – P. 1253–1261.

29. Constraint-Based Watermarking Techniques for Design IP Protection / A. Kahng [et al.] // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2001. – Vol. 20, no. 10. – P. 1236–1252.

30. Cui, A. A Hybrid Watermarking Scheme for Sequential Functions / A. Cui, C. H. Chang, L. Zhang // IEEE ISCAS, 2011, Rio de Janeiro. – 2011. – P. 2333–2336.

31. Clock-Modulation Based Watermark for Protection of Embedded Processors / J. Kufel [et al.] // Design, Automation & Test in Europe Conference and Exhibition. – Dresden, 2014. – P. 1–6.

32. Marsh, C. Protecting Designs with a Passive Thermal Tag / C. Marsh, T. Kean, D. McLaren // 15th IEEE Intern. Conf. on Electronics, Circuits and Systems. – Malta, 2008. – P. 218–221.

33. Abdel-Hamid, A. IP Watermarking Techniques: Survey and Comparison / A. Abdel-Hamid, S. Tahar, E. Aboulhamid // The 3rd IEEE Intern. Workshop on System-on-Chip for Real-Time Applications. – Calgary, Canada, 2003. – P. 60–65.

Поступила 27.01.2015

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, ул. П. Бровки, 6
e-mail: vovasq@mail.ru
ivaniuk@bsuir.by

V.V. Sergeichik, A.A. Ivaniuk

A SURVEY OF HARDWARE WATERMARKING FOR PROGRAMMABLE LOGIC DEVICES PROTECTION

Application of watermarking technology for the protection of digital devices and their descriptions is considered. Primary definitions, models, categories of attacks, characteristics and classification of watermarks are described. Hardware watermarking examples are shown.