АППАРАТНЫЕ ТРОЯНЫ. ЧАСТЬ 4: ПРОГРАММНО-АППАРАТНЫЕ КОНТРМЕРЫ

HARDWARE TROJANS. PART 4: SOFTWARE AND HARDWARE COUNTERMEASURES

УДК 621.382, ВАК 05.27.01, DOI: 10.22184/1993-8578.2017.72.2.42.56

Е.Кузнецов, А.Сауров**
E.Kuznetsov, A.Saurov**

В завершающей статье, посвященной аппаратным троянам, рассматривается построение безопасных систем, которые надежно функционировали бы, в том числе, в присутствии аппаратной закладки произвольного типа. Хотя общий подход к такому способу противодействия на сегодняшний день не разработан и не предложен, рассмотрим некоторые аспекты такой защиты, детали ее реализации и проведем общий анализ применимости подобных контрмер.

In the final part of the cycle of review articles devoted to Hardware Trojans problems of maintaining system secure operation in the presence of Hardware Trojans are considered. To date no general countermeasures have yet been developed or proposed that would allow an IC to operate in a trustworthy manner in the presence of an arbitrary Hardware Trojan. Some aspects of such protection, details of its implementation and analyzing the general applicability of the countermeasures are discussed.

Как уже отмечалось ранее [1–3], использование превентивных подходов предупреждения и современных методов обнаружения аппаратных троянов не дают полной гарантии того, что изготовленная ИС не содержит аппаратных закладок. Большое разнообразие угроз безопасности, связанных с ними, а также огромное пространство состояний для размещения аппаратных закладок поставило перед разработчиками вопрос об обеспечении безопасной эксплуатации системы с "инфицированными" ИС и, в частности, задачу предотвращения активации троянов. Такой подход позволил бы использовать аппаратуру, не обращая внимания на внедренные аппаратные трояны, и даже использовать COTS-компоненты (коммерческие электронные компоненты, находящиеся в свободной продаже) для построения устойчивых к аппаратным закладкам, надежных вычислительных систем.

В большинство публикаций, связанных с такого рода противодействием аппаратным троянам, рассматривается защита только от одного класса или подкласса угроз. Большую безопасность, как предполагается, можно обеспечить с помощью многоуровневой защиты, в которой каждый уровень независимо от других ориентирован на определенные действия и механизмы активации троянов, с последующим объединением всех этих мер в общую стратегию защиты.

Предложенные и экспериментально проверенные механизмы контрмер можно разделить на следующие группы:

- охрана данных;
- новые архитектуры на RTL-уровне (на уровне регистровых передач);
- реконфигурируемая архитектура;
- стратегии репликации, фрагментации и мажоритарной выборки.

ОХРАНА ДАННЫХ

Охрана данных (включая команды процессора) предполагает предотвращение активации аппаратного трояна или/и блокирование прямого доступа троянского оборудования к любым уязвимым данным. Защитное устройство может контролировать выборку данных, хранимых или

* НПК "Технологический центр"/ SMC "Technological Centre".

передаваемых внутри или между ИС и логическими модулями, блокируя механизм, посредством которого троян взаимодействует с данными.

В работе [4] предлагается несколько методов охраны данных с предотвращением активации трояна. Некоторые из этих методов были реализованы на симуляторе Zesto ×86. Для предотвращения получения троянской закладкой активационного кода используется скремблирование (шифрование) информационного канала (шины). Оно применяется для блоков обработки данных, которые не задействованы в вычислениях. Авторы предлагают использовать простые доверенные схемы шифрования для скрытия данных (например, исключающее ИЛИ с псевдослучайными числами – Xorshif-шифрование), засекречивая их только на короткий период времени.

Эффективность такого скремблирования исследовалась путем введения параметризируемой задержки в кэше и контроллере памяти. Скремблирование шины предотвращало активацию простых троянов, при этом ее вероятность все же сохранялась. Так, например, при использовании простого 32-разрядного триггера, активация произойдет за 2^{32} циклов. Более контролируемый подход может быть реализован через переопределение всех входов в полностью функционально достоверное пространство состояний с использованием доверенной и устойчивой схемы шифрования.

Поскольку для вычислительных блоков такая обфускация данных приведет к неверным вычислениям, в этом случае предлагается использовать гомоморфное шифрование [5], позволяющее вычислительным блокам работать с зашифрованными данными. Шифрование определяется гомоморфным по вычислительной функции, и правильные результаты вычислительный блок получает только с зашифрованными значениями. Полученный в итоге результат может быть расшифрован.

Реализация таких гомоморфных функций – задача нетривиальная, а их вычисление требует затрат, при этом достаточно трудно построить

As it has been previously noted [1–3], the preventive approaches and modern methods for the detection of hardware Trojans do not give a complete assurance that manufactured IC does not contain hardware backdoors. A wide variety of security threats associated with them, as well as a huge state space for embedding hardware backdoors raised the question, how to ensure the safe operation of the system with the "infected" IC and, in particular, the goal to prevent the activation of Trojans. This approach would allow the use of the equipment, not paying attention to embedded hardware Trojans, and even use of COTS (Commercial off-the-shelf) components to build reliable computing systems that are resistant to hardware backdoors.

The most publications associated with this kind of counteraction to hardware Trojans, consider the protection against only one class or subclass of the threats. It is assumed that greater security will provided by a multilevel defense in which each level irrespective of others focused on specific actions and mechanisms of activation of Trojans, followed by combining all these measures into an overall strategy of protection.

Proposed and experimentally tested mechanisms of countermeasures can be divided into the following groups:

- data protection;
- new architecture at the RTL-level (register-transfer level);
- reconfigurable architecture;
- strategies of replication, fragmentation and voting.

DATA PROTECTION

Protection of data (including processor commands) involves the prevention of the activation of a hardware Trojan or/and blocking direct access of Trojan equipment to any vulnerable data. The protective device can control data stored or transmitted within or between ICs and logical modules, and block mechanism by which the Trojan interacts with the data.

In [4] several methods of protection of data by preventing activation of a Trojan are proposed. Some of these methods were implemented on the Zesto x86 simulator. To prevent the receiving an activation code by Trojan a scrambling (encryption) of information channel (bus) is used. It is used for data processing units that are not involved in the calculations. The authors propose to use simple trusted encryption schemes to hide data (for example, XOR with pseudo-random numbers – Xorshif encryption), secretive them only for a short time.

The effectiveness of this scrambling was investigated by introducing parameterized delay in the cache and the memory controller. The scrambling of the bus prevented the activation of simple Trojans, however, this possibility

схему гомоморфного шифрования общего назначения. Так же, как и в случае скремблирования шины данных, блоки шифрования и дешифрования должны быть реализованы в полностью проверенном аппаратном обеспечении. В работе [4] не рассматривалось гомоморфное шифрование, но взамен предложено и проанализировано применение криптографического алгоритма с открытым ключом (RSA – шифрование). Использование схем, реализующих алгоритм "искаженных цепей" Яо (Yao's Garbled Circuit) [6], можно рассматривать в качестве альтернативного подхода для обфускации данных вычислительных блоков.

Также в работе [4] предложено использовать "защиту времени" для предотвращения активации аппаратного трояна в пределах подтвержденного пространства состояний. ИС проверяется на полную функциональную достоверность за заданное число циклов. В дальнейшем при работе ИС после наработки этого количества циклов схема отключается и включается, тем самым предотвращается временная активация трояна. Аппаратное решение позволяет сохранять контекст в период отключений, обеспечивая непрерывность вычислительного процесса. Авторы при этом делают допущение, что любой аппаратный троян, который находился в состоянии покоя при полном тестировании пространства состояний (в пространстве состояний времени и входов), будет находиться в состоянии покоя в течение этого же периода и в условиях эксплуатации.

Рассмотренный подход не подходит для триггеров, построенных на основах энергонезависимой памяти, накопительного механизма (например, накопления заряда на емкости [7]), побочных каналов, деградации, а также триггеров с внешним управлением (например, по радиоканалу). Авторы предлагают обойти первый из этих вариантов путем визуального контроля на наличие ячеек энергонезависимой памяти, либо специальным выжиганием таких ячеек при сборке. Другим решением может быть использование техпроцесса, исключающего реализацию энергонезависимой памяти. Любые аппаратные средства, использующиеся для перезапуска ИС, а также средства сохранения контекста в период выключения/включения должно быть поверенным.

Предложены также "охранники", которые случайным образом изменяют порядок событий и вносят фиктивные события во входные последовательности различных модулей, например в контроллер памяти, так что сохраняемые или загружаемые последовательности нарушаются. Такие "охранники" защищают от активации триггеров последовательностного типа.

В заключении, авторы [4] предлагают использовать несколько версий непроверенных аппаратных блоков ИС, разработанных различными проектировщиками. Выходы с модуля проверяются путем сравнения между собой, и истинное значение определяется путем мажоритарной выборки. Недостатки такого подхода – высокая стоимость из-за увеличения площади чипа и высокое энер-

still remained. For example, when using a simple 32-bit trigger, activation will occur for 2^{32} cycles. A more controlled approach could be implemented through the override of all inputs into a fully functionally reliable state space using the trusted and stable encryption schemes.

Since such data obfuscation will lead to wrong calculations in computing units, in this case, it is proposed to use homomorphic encryption [5], which allows computing units to work with encrypted data. Encryption is determined as homomorphic by the computing function and computing unit obtain the correct results only with encrypted values. The final result can be decrypted.

The implementation of such homomorphic functions is not a trivial task, and their computation is costly, and it is quite difficult to build a scheme of homomorphic encryption for general use. In the same way as in the case of scrambling of data bus, the encryption and decryption blocks must be implemented in a fully tested hardware. The authors of [4] did not consider homomorphic encryption, but instead have proposed and analyzed the use of a cryptographic algorithm with a public key (RSA-encryption). The use of schemas that implement the Yao's Garbled Circuit algorithm [6], can be considered as an alternative approach for obfuscation of data of the computing blocks.

It is also proposed in [4] to use a "time protection" to prevent activation of a hardware Trojan within a confirmed state space. The IC is checked for complete functional reliability for a specified number of cycles. In the future, when IC reaches this number of cycles, the circuit is deactivated and activated, thereby avoiding temporary activation of a Trojan. The hardware solution allows to save context during the period of the outage,

гопотребление ИС. Авторы работы попытались охватить все аспекты активации аппаратных троянов, однако полностью обошли вопросы, связанные с активацией по побочным каналам, а также с внешней триггерной активацией.

В работе [8] для чипов "системы на кристалле" SoC предложена защищенная архитектура системной шины. Ее особенность заключается в том, что шина изменяет свое состояние ведомой или ведущей (master/slave) с целью обнаружения аппаратного трояна, который пытается ее заблокировать, используя при этом стандартные команды управления шинами. Такая активность легко обнаруживается с помощью номинальных обновляемых счетчиков и простых эвристических алгоритмов, после чего формируется черный список подозрительных мастер/ведомых устройств на шине с представлением отчета. Предложенный подход был проверен на шинной архитектуре Advanced Micro-controller Bus Architecture (AMBA) от фирмы ARM. Подобные контрмеры ориентированы на конкретную архитектурную особенность и специальный класс аппаратных троянов, предотвращают вмешательство закладок в корректную работу шины SoC и, следовательно, влияют на производительность всей системы.

Ряд исследователей рассматривали вопросы размещения "охранников" шины памяти в архитектуре процессора, решая при этом задачи предотвращения активации трояна и утечки данных.

В работе [9] предлагается вставлять так называемые "теневые записи" – сопутствующие записи для всех инструкций памяти. Адреса этих теневых записей являются шифрованными версиями оригинальных записей. Ядро привратника (GateKeeper, аппаратная реализация), находящегося на шине памяти, проверяет, чтобы все записи в память происходили по соответствующим зашифрованным адресам. Таким образом, привратник гарантирует выполнение только законных записей, тем самым предотвращая утечку конфиденциальной информации через аппаратный троян. Схема привратника должна быть полностью поверенной. Авторами был разработан полнофункциональный прототип, выполняющий инструкции ×86 и содержащий схему привратника, которая детектировала все теневые записи. Однако этот подход основан на том, что вывод данных предполагает их запись в память (для отправки по сети, например), но на практике он может также происходить с использованием побочных каналов, например путем анализа изменения энергопотребления или изменения временных характеристик.

Двойная защита между центральным процессором и шиной данных предложена в [10]. Два охранных устройства имеют независимые ключи и проверяют друг друга на корректность. Исполняемые программы шифруются одновременно на двух "охранниках" с разными ключами, данные расшифровываются на пути к центральному процессору и шифруются на обратном пути в память. Система предполагает отсутствие взаимосвязи между охранниками. Аппаратный компилятор также составляет критическую часть системы, фор-

providing continuity of the computational process. The authors make the assumption that any hardware Trojan, which was in rest mode at the complete testing of the state space (the state space of time and inputs), will be in rest mode also during the same period of operation.

This approach is not suitable for triggers, built on the bases of non-volatile memory, storage mechanism (for example, the accumulation of charge on the capacitance [7]), side channels, degradation, and triggers with external control (for example, via radio). The authors propose to circumvent the first of these variants by visual inspection for the presence of cells of non-volatile memory, or a special burning of such cells during assembly. Another solution is to use a manufacturing process, eliminating the realization of non-volatile memory. Any hardware used to restart the IC, as well as means of saving context during the period of on/off switching must be trusted.

Also "security guards" are proposed, which randomly changes the order of events and introduce a fictitious event into the input sequences of the various modules, for example, in the memory controller, so that saved or loaded sequences are violated. These "guards" protect from the activation of trigger of sequenced type.

In conclusion, the authors of [4] propose to use multiple versions of untested hardware blocks of IC developed by different designers. Outputs from the module are verified through comparison between themselves and the true value is determined by a voting. The disadvantages of this approach is high cost due to the increase in chip area and high power consumption of IC. The authors tried to cover all aspects of the activation of hardware Trojans, however, completely bypassed the issues related to the

мируя двоичный код, команды BIOS и образы операционной системы непосредственно перед выполнением. Двойная защита снимает необходимость доверия обоим охранникам, более важным становится отсутствие "сговора" между ними. Для исследования рассматриваемого подхода использовали симулятор компьютерной архитектуры с открытым кодом SimpleScalar.

Еще в 2003 году, в работе [11] была представлена AEGIS-архитектура процессора, в которой возможно использование непроверенных подключаемых периферийных устройств, а также запуск непроверенной операционной системы. Процессор при этом должен быть поверенным. Используя примитивное шифрование, он выступает в качестве охранника между собой и всеми ненадежными периферийными устройствами. Основная трудность такой реализации – уверенность, что ИС подобного процессора полностью свободна от аппаратных закладок.

Как видим, идея использовать минимальную проверенную вычислительную базу (TCB, Trusted Computing Base), или поверенные аппаратные средства для противодействия аппаратным закладкам, является основным условием каждой из приведенных контрмер. Рассмотренные механизмы защиты памяти от аппаратных троянов могут быть распространены на другие информационные каналы передачи данных и аппаратные модули внутри ИС. В работе [12] развили эту идею, введя понятие Silicon Security Harness – "ремни кремниевой безопасности" (по аналогии с ремнями безопасности в автомобиле). Предложенная концепция включает в себя несколько уровней защиты и слежения, которые обеспечиваются аппаратными средствами и системными компонентами или реализуются как часть архитектуры. Она призвана обеспечить защитные меры и увеличить стойкость к действию аппаратных закладок.

НОВЫЕ АРХИТЕКТУРЫ НА RTL-УРОВНЕ

В нескольких работах с целью защиты от аппаратных троянов предложено внесение специальных модификаций в процессор или архитектуру ИС. В основе такого подхода лежит добавление или изменение логических вентилей для идентификации присутствия или предупреждения активации аппаратных закладок.

В работе [13] предлагается комплексное программно-аппаратное обеспечение для противодействия аппаратным закладкам под названием BlueChip (голубой чип). Это оборонительная стратегия, включающая в себя компоненты безопасности как во время разработки, так и во время эксплуатации системы для противодействия на RTL-уровне аппаратным троянам с произвольной локализацией. Разработанный алгоритм под названием UCI (Untrusted Circuit Identification – идентификация ненадежных схем) и набор инструментальных средств автоматически определяет и удаляет потенциально опасные цепи в проекте на RTL-уровне для процессорных ИС. Во время верификации проекта обнаруживаются и удаляются все подозрительные схемы, которые включены

activation through side channels and external trigger activation.

In [8] a secure architecture of system bus is proposed for SoC chips. Its peculiarity lies in the fact that the bus changes its state (master/slave) for the detection of a hardware Trojan, which is trying to block it using the standard control commands. Such activity is easily detected using updated counters and simple heuristic algorithms, which then forms a black list of suspicious master/slave devices on the bus with the submission of the report. The proposed approach was tested on a Advanced Microcontroller Bus Architecture (AMBA) of ARM company. Such countermeasures are focused on a specific architectural feature and a special class of hardware Trojans, prevent the intervention of the backdoors into the correct operation of the SoC bus and therefore affect the performance of the entire system.

A number of researchers have considered the placement of the "guards" of the memory bus in the processor architecture, solving the task of preventing activation of the Trojan and data leakage.

In [9] it is proposed to insert so-called "shadow entries", a related entries for all memory instructions. The addresses of these shadow entries are encrypted versions of the original records. The core of the gatekeeper (hardware implementation) located on the memory bus, verifies that all memory entries have occurred for the respective encrypted addresses. Thus, the gatekeeper ensures execution of only legitimate entries, thereby preventing leakage of confidential information via a hardware Trojan. The circuit of the gatekeeper must be fully tested. The authors have developed a fully functional prototype that executes x86 instructions and included gatekeeper circuit, which has detected all of the shadow entries. However, this approach is

в проект, но не влияют на какой-либо из выходов во время тестирования. Удаленные аппаратные устройства заменяются логикой, вызывающей исключение (непредвиденную ошибку), если удаленный участок когда-либо активируется. Это может произойти вследствие потенциальной активации трояна или предоставления допуска к системе, инициализированного удаленным фрагментом. Программное обеспечение низкого уровня путем эмуляции старается восстановить и предсказать последствия, к которым может привести действие обнаруженных подозрительных участков, при этом предложено использовать малый набор "проверенных" инструкций.

Концепция BlueChip была опробована на процессоре Leon3 (Aeroflex Gaisler AB), разработанном на ПЛИС Xilinx Virtex5. В этой концепции безопасность в основном обеспечивается поверенными программными компонентами, и лишь частично поверенными аппаратными средствами, используемыми для эмуляции удаленных фрагментов. Стратегия лучше всего подходит для процессорных проектов, в которых возможно проведение программной эмуляции по исследованию исключений. Для распространения подхода на общий случай проектирования ИС можно использовать поверенный сопроцессор, с помощью которого будут выявляться исключения и выполняться эмуляция при удалении аппаратных устройств. Концепция BlueChip также нуждается в развитии, поскольку известны вредоносные аппаратные реализации [14], которые не обнаруживаются алгоритмом UCI и проходят верификацию.

С этой связи, в работе [15] утверждается, что не существует механизмов защиты от троянов, гарантирующих их обнаружение до непосредственной эксплуатации ИС. Авторы предлагают проводить обнаружение атак, связанных с наличием троянов в ИС, в процессе работы путем добавления дополнительных интегрированных логических блоков, осуществляющих самотестирование ИС и поиск аппаратных закладок. Такая дополнительная логика, называемая DEFENSE (DEsign-For-Enabling-Security), интегрируется в SoC для выполнения в режиме реального времени конфигурируемых проверок безопасности через мультиплексирование различных частей системы с проверочным блоком. Например, могут проверяться законность доступов и законность состояний; ситуации, связанные с DoS-ошибками, а также целостность системы. При обнаружении атаки в режиме реального времени предпринимаются контрмеры, например, отключаются подозрительные логические блоки. Авторы предлагают также использовать отказоустойчивые состояния, резервные логические блоки, копирование текущего состояния при обнаружении атаки. Обеспечение комплексного охвата всевозможных аппаратных атак в режиме реального времени является достаточно трудной задачей, поэтому прототип платформы DEFENSE не был создан. Не менее сложная задача – осуществление таких контрмер в режиме реального времени без прерывания функционирования ИС.

В работе [16] предложено снимать уникальную контрольную сумму аппаратного обеспечения

based on the fact that the data output supposes their writing to memory (to send over the network, for example), but in practice it may also occur with the use of side channels, for example, by analyzing changes in consumption or changes in time characteristics.

Double protection between the CPU and the data bus is proposed in [10]. Two security devices have independent keys and check each other for correctness. Executable programs are encrypted at the same time on two "guards" with different keys, the data is decrypted on the way to the CPU and encrypted on the way back to memory. The system assumes the absence of correlation between the guards. The hardware compiler is a critical part of the system, forming a binary code, BIOS commands, and images of operating system immediately before execution. Double protection eliminates the need to trust both guards, more important is the absence of "collusion" between them. To study this approach a simulator of computer architecture with open source SimpleScalar was used.

In 2003, in [11] was presented AEGIS processor architecture in which it is possible to use an untested plug-in peripheral devices as well as run an untested operating system. Using primitive encryption, it acts as a guard between itself and all unreliable peripheral devices. The main difficulty of such an implementation is that IC of such a CPU should be completely free from hardware backdoors.

As you can see, the idea of the use of a minimum proven computing base (TCB, Trusted Computing Base) or tested hardware to counter hardware backdoors, is the basic condition for each of these countermeasures. The discussed mechanisms of memory protection against hardware Trojans can be extended to other channels of data transmission and hardware modules inside the

за некий ограниченный промежуток времени с использованием доверенного оборудования. Такая аппаратная контрольная сумма вычисляется для задействованных низкоуровневых микроархитектурных элементов процессора. Аппаратное обеспечение опрашивается, и контрольная сумма определяется в течение ограниченного времени. Авторы полагают, что подлинная контрольная сумма не может быть эмулирована или сымитирована в ограниченное время, и только аутентичные аппаратные средства могут корректно откликаться. Механизм гарантирует отсутствие троянских закладок, внесенных после изготовления ИС. Для обеспечения такого контроля, названного функцией микроархитектурной подписи MSF (Micro-Architecture Signature Function), и генерации уникального ответа на запрос были разработаны новые инструкции для процессора. Однако, такой подход не позволяет обнаруживать трояны, вставленные в проект на стадиях спецификации, разработки, верификации или изготовления.

В работе [10] при рассмотрении угроз со стороны аппаратных троянов, связанных с ошибками типа "отказ в обслуживании" (DoS-атака), разработана специальная функция "сердцебиения" для проверки непрерывности работы ИС. Некэшируемые выборки из памяти добавляются в программное обеспечение, после чего эти сигналы проявляются на шине памяти, как регулярные, но случайные интервалы, которые используются для определения, подверглась ли ИС DoS-атаке.

РЕКОНФИГУРИРУЕМАЯ АРХИТЕКТУРА

Использование реконфигурируемой логики для противостояния аппаратным троянам имеет существенные преимущества, но ставит новые задачи и проблемы перед проектированием. Существует целый спектр реконфигурируемых логических устройств, в том числе:

- ПЛИС высокой логической плотности, где большая часть устройства является перепрограммируемой;
- платформенные целевые ПЛИС, содержащие фиксированные полупроводниковые элементы, например контроллеры памяти, и даже целые процессорные ядра;
- пользовательские (заказные) СИС (ASIC), которые могут содержать мелкие реконфигурируемые части для выполнения определенных функций, например реализации связующих логических схем или заказного элемента для совместной обработки.

Основное преимущество реконфигурируемой логики заключается в том, что при ее использовании становится возможным разделение между проектированием ИС и ее аппаратной реализацией. Если проект типовой ИС передается непосредственно в кристальное производство, то в случае реконфигурируемой логики программируемый логический блок или макроблок вставляется в ИС перед ее запуском в производство, а после изготовления программируется конфигурируемым битовым потоком, завершая аппаратную реализацию проекта. Такое разделение означает, что проект может быть разработан практически полностью

IC. The authors of [12] have developed this idea by introducing the concept of the Silicon Security Harness (by analogy with the seat belts in the car). The proposed concept includes several levels of protection and monitoring which are provided by the hardware and system components, or are implemented as part of the architecture. It is designed to provide protective measures and increase resistance to hardware backdoors.

NEW ARCHITECTURES AT RTL-LEVEL

In several works, to protect against hardware Trojans, it is proposed making special modifications to the processor architecture or IC. The basis of this approach is the addition or change of logical gates to identify the presence or the to prevent activation of hardware backdoors.

In [13] a comprehensive software and hardware, the BlueChip, is proposed to counter hardware Trojans. This defensive strategy includes safety components both during development and during system operation to counteract hardware Trojans with arbitrary localization, at RTL-level. The developed algorithm of UCI (Untrusted Circuit Identification) and the toolkit automatically detects and removes potentially malicious circuits in the project at RTL-level for processor ICs. All suspicious schemes, which are included in the project, but do not affect any outputs during testing are detected and removed during verification of the project. Removed hardware devices are replaced with logic that causes an exception (unexpected error) if the removed area is activated. This can occur due to the potential activation of a Trojan or permit access to the system, initialized by the removed fragment. Low-level software by emulation try to reconstruct and to predict the consequences of the action of suspicious areas, using a small set of proven instructions.

в доверяемой среде, за исключением некоторых периферийных функций, добавляемых к основным логическим элементам.

Подход обеспечивает полный контроль проекта на RTL-уровне, однако разработка и включение реконфигурируемой логики подвержены тем же угрозам со стороны аппаратных троянов, которые характерны для стандартных специализированных ИС (ASIC). Однако, злоумышленник может выполнять только общие атаки на архитектуру реконфигурируемой логики, что затрудняет беспрепятственное вмешательство или модификацию логической операции сконфигурированного проекта. При этом можно осуществлять полный спектр атак: изменять функциональность и спецификацию, проводить утечку конфиденциальной информации и выполнять DoS-атаки. Так, например, изменения в логических элементах могут привести к появлению дополнительных логических операций, потенциально опасных и ведущих к возникновению скрытых ошибок в проекте или утечке информации через периферийные устройства.

Таким образом, возникает новая задача: как наилучшим образом реализовать надежный проект, зная, что лежащая в его основе реконфигурируемая логика может быть заражена произвольным аппаратным трояном, и как защитить целостность проекта после его создания, то есть защитить битовый поток от искажения или заражения трояном. Общий трехступенчатый подход к обеспечению целостности битового потока ПЛИС предложен в работе [17]. Во-первых, целостность конфигурации проверяется ее обратным чтением, во-вторых, в случае обнаружения неправильной конфигурации, ПЛИС частично модифицируется (от подлинной части битового потока), в-третьих, в случае если система была скомпрометирована, ПЛИС использует протокол запроса-ответа для уведомления третьей стороны.

В работе [18] предложен обзор решений по защите битовых потоков ПЛИС и конфигурационной памяти от событийных отказов. Авторы делают следующие выводы:

- если изготовление ПЛИС и проектирование ИС на ПЛИС полностью разделены или в проекте используются IP-блоки третьей стороны, то злоумышленнику не представляет труда внести изменения в любой проект;
- конфигурационные битовые потоки достаточно трудно поддаются обратному проектированию как для злоумышленника, что делает для него невозможным понять назначение ИС, так и с целью поиска закладок;
- битовые потоки могут быть зашифрованы, что обеспечивает хорошую защиту, и тогда во многих ПЛИС возможно внутреннее аппаратное декодирование битового потока. Однако такое шифрование не позволяет реализовывать частичную реконфигурацию, поэтому не применимо для таких приложений, как адаптивные вычисления, являющихся существенной нишей для ПЛИС. Шифрование битового потока также не защищает от проникновения аппаратных троянов через IP-блоки сторонних разработчиков.

The BlueChip conception was tested on a Leon3 processor (Aeroflex Gaisler AB) developed on the base of Xilinx Virtex5 FPGA. Safety in this concept is mainly provided by trusted software components, and only partially by trusted hardware, which are used to emulate the remote fragments. The strategy is best suitable for CPU projects, where the software emulation for the study of exceptions is possible. For dissemination of the approach to the general case of the IC design a trusted coprocessor can be used, which will identify exceptions and emulate removed hardware devices. The BlueChip concept also needs to be improved, because the malicious hardware are known [14], which cannot be detected by the UCI algorithm and successfully passes the verification.

With this regard, in [15] it states that there are no mechanisms of protection against Trojans, guaranteeing their discovery before start of use IC. The authors propose to detect attacks related to the presence of Trojans in ICs in the process of their use by adding additional integrated logic blocks, which perform a self-test of IC and search of hardware backdoors. Such additional logic, called DEFENSE (DEsign-For-Enabling-Security) is integrated into the SoC to perform real-time configurable security checks through the multiplexing of the various parts of the system with the verification block. For example, the legality of access and the legitimacy of the state, situations associated with DoS error messages, and the system integrity can be checked. In the case of detection of an attack, the real-time countermeasures are taken, for example, suspicious logical blocks are disabled. The authors also propose to use a failover states, backup logical units, copying of the current state when an attack is detected. The comprehensive coverage of all

Для идентификации подлинности разработанного проекта предлагается использовать дополнительные конфигурационные логические блоки (CLB) в составе ПЛИС на основе кода с коррекцией ошибки (ECC), образующие группу проверки четности. Проверка на четность каждого элемента CLB позволяет вскрывать все внесенные в проект изменения. Двухступенчатая рандомизация, используемая для формирования сигнала четности, обеспечивает непредсказуемость результата CLB.

В работе [19] предлагается защита от аппаратных троянов, которая вносится на стадии изготовления. Если размещать реконфигурируемые логические блоки между важнейшими элементами в проекте, то на производственной стадии будет видима некоторая реконфигурируемая архитектура в некоторых областях кристалла. Эти блоки, названные авторами барьерами, после изготовления программируются с использованием секретного ключа, что приводит к разблокировке всего проекта и его логическому завершению. Если расположение и функциональность этих барьеров выбраны оптимально, то любые вставленные аппаратные трояны будет достаточно трудно активировать, и их влияние на ИС можно блокировать. Особое внимание авторы уделяют типу перепрограммируемой логики, управлению ключами, и различным эвристическим методам размещения барьеров. Комбинируя постоянную и программированную логику, можно выработать уникальные решения против потенциальных аппаратных троянов. При этом реконфигурируемая логика может использоваться и для реализации локальных защитных механизмов.

Использование реконфигурируемой логики в качестве защиты от аппаратных закладок ставит новые задачи для проектирования и верификации и перемещает основное внимание по защите с полупроводникового производства на RTL-проект. Для реализации эффективной аппаратной закладки на уровне ИС в этом случае злоумышленнику необходимо обеспечить взаимосвязь (сговор) между производителем и поставщиком инструментальных средств. Реализация сложного реконфигурируемого логического проекта, как правило, основывается на интеграции нескольких IP-блоков. В работе [20] предложена идея "рвов и подъемных мостов" как изоляционных примитивов, которые применяются в случае использования нескольких IP-блоков. Они позволяют блокировать нелегальные ответвления и нелегальные межсоединения внутри реконфигурируемых блоков.

Перечислим некоторые другие подходы с применением реконфигурируемой логики, которые могут быть использованы для борьбы с аппаратными троянами:

- частичное и динамическое перепрограммирование логических блоков [21];
- шифрование конфигурационных битовых потоков [18];
- репликация и жесткая конфигурация (lock-stepping) логики [22];

kinds of hardware attacks in real time is a difficult enough task, so the prototype of DEFENSE platform was not created. No less difficult is the implementation of such countermeasures in real time without interrupting the operation of the IC.

In [16] it is proposed to use the unique checksum of hardware for a limited period of time using trusted hardware. This hardware checksum is calculated for the involved low-level microelements of the processor. The hardware is queried, and the checksum is determined for a limited time. The authors believe that the original checksum may not be emulated or simulated in a limited time, and only the authentic hardware can correctly respond. The mechanism guarantees that Trojans haven't been added after manufacturing of the IC. To ensure such control, called MSF (Micro-Architecture Signature Function), and to generate a unique response to the request, the new instructions for the processor were developed. However, this approach does not allow to detect Trojans inserted in the project at stages of specification, development, verification or manufacturing.

In [10], during considering threats associated with "denial of service" attack (DoS attack), a special function of "heartbeat" is developed to check the continuity of operation of IC. Non-cacheable memory accesses are added to the software, after which these signals appear on the memory bus as a regular, but random intervals, which are used to detect fact of a DoS attack.

RECONFIGURABLE ARCHITECTURE

The use of reconfigurable logic to counter hardware Trojans has significant benefits, but poses new challenges at the stage of design. There is a spectrum of reconfigurable logic devices, including:

- использование функционально идентичных, но имеющих различную архитектуру логических блоков [23];
- генерация однозначных аппаратных модулей с использование случайных чисел [24].

РЕПЛИКАЦИЯ ЧАСТЕЙ, ФРАГМЕНТАЦИЯ И МАЖОРИТАРНАЯ ВЫБОРКА

Разработка эффективных аппаратных троянов связана с пониманием противником функционирования проекта ИС, начиная от уровня вентилей, RTL-уровня, уровня ИС и заканчивая макроуровнем всей системы. На всех этих уровнях для противодействия аппаратным троянам могут быть применены следующие общие подходы:

- репликация или удвоение логики или/и данных;
- разделение или фрагментация логики или/и данных;
- рассредоточение или распределение логики или/и данных;
- накопление и объединение логических функций или/и данных, например, используя мажоритарную выборку.

Эти общие контрмеры эффективны в трех случаях:

- при защите от аппаратных троянов, приводящих к утечке конфиденциальной информации путем разделения данных и обработки их независимыми логическими элементами;
- для защиты от функциональных или спецификационных модификаций элементов с помощью нескольких копий или дубликатов логических блоков;
- для защиты от DoS-атак путем задания избыточности работающих логических элементов в проекте.

Рассматриваемые контрмеры могут быть размещены на различных уровнях: вентильном, RTL, логического проектирования, функциональных модулей, IP-ядер, вплоть до уровня ИС и устройств на макроуровне. Механизмы защиты предполагают отсутствие "сговора" между репликационными или дублированными элементами в проекте.

Метод динамической оценки поверки оборудования во время его эксплуатации предложен в работе [23]. Его суть заключается в обнаружении аппаратного трояна при работе системы, после чего она продолжается с удалением или малым задействованием подозрительных элементов. Авторы предлагают использовать многоядерную систему обработки данных и воспользоваться присущим таким системам резервированием, отказываясь от ядер, которые не заслуживают доверия. Функционально эквивалентные варианты процессов проводятся на нескольких процессорах, после чего сравниваются их результаты. Различные варианты одинаковых процессорных обработок могут быть выполнены на основе разных компиляций или реализаций. Также могут использоваться различные алгоритмы обработки данных. Если результаты двух элементов отличаются, проводятся расчеты на третьем элементе и сравнение трех результатов. Этот процесс продолжается до тех пор, пока не будет достигнуто соответствие, по крайней

- FPGA of high logic density, where most part of the device is reprogrammable;
- target FPGAs containing fixed semiconductor elements, e.g., memory controllers, and even entire processor cores;
- custom ASIC that may contain small reconfigurable parts to perform certain functions, such as implementing a binding logic circuits or a custom element for joint processing.

The main advantage of reconfigurable logic is the possibility of separation between design of IC and its hardware implementation. If the design of a standard IC is passed directly to the chip production, then, in the case of reconfigurable logic, programmable logic unit or a macro block is inserted into the IC before start of manufacturing, and after production it is programmed by configurable bitstream completing the hardware implementation of the project. This separation means that the project can be developed almost entirely in the trusted environment, except for some peripheral functions that are added to the main logical elements.

The approach provides full control of the project at the RTL-level, but the development and inclusion of reconfigurable logic are subject to the same threats from hardware Trojans, which are characteristic of standard ASIC. However, the attacker can execute only a general attacks against the architecture of the reconfigurable logic, which hinders the smooth intervention or modification of logic operations of configured project. But it is possible to undertake the full range of attacks: to change functionality, to change the specification, to steal confidential information and to perform DoS attacks. So, for example, the changes in the logic elements can lead to additional logical operations that are potentially dangerous and leading

мере, между двумя элементами обработки данных. Процессорные элементы, которые дают противоречивые (ошибочные) результаты динамически "штрафуются", то есть "доверие" к ним становится меньше, и по возможности они используются меньше.

Этот метод может быть расширен с использованием случайной выборки вариантов функционально эквивалентных аппаратных средств. Он также может применяться на различных уровнях абстракции, например, инструкций, вентильном, программы или ИС. Если метод используется на уровне команд, то активность на нем может быть прозрачной для более высоких уровней, в том числе и для небольшой поверенной вычислительной аппаратной базы (TSB), на которой можно проверять командный уровень расписаний, выбора репликационных блоков, заданий вариантов и мажоритарной выборки.

В работе [22] предлагается выполненная на ПЛИС жесткая конфигурация двухпроцессорной архитектуры с непосредственными связями, которая на макроуровне представляет собой реализацию репликации и мажоритарной выборки. Оба процессора получают и обрабатывают одни и те же инструкции одновременно. Аппаратно реализованная логика проверяет и сравнивает все управляющие сигналы каждой транзакции на шине. Если обнаружена ошибка, система принудительно вводится в последовательность устранения ошибки. Для адекватного противодействия системы аппаратным троянам должна быть проведена полная верификация TSB – блоков проверки и устранения ошибок. Метод может быть расширен для большего числа процессоров, которые могут быть отдельными ИС или входить с состав одной ПЛИС.

В 1991 году в работе [25] исследовались вопросы обеспечения высокой надежности и работоспособности, а также сохранения конфиденциальности данных в больших распределенных системах. Угрозы, связанные с аппаратными троянами, не рассматривались, однако предложенные методы отказоустойчивости актуальны для противодействия несанкционированным закладкам. Общий подход предполагает разбиение данных на малые фрагменты, так что каждый из них содержит достаточно мало информации. Он может быть использован и для хранения данных, и для их обработки (соответственно, они группируются, как фрагменты для хранения и для обработки). Репликация (резервирование) фрагментов применяется для обеспечения надежности системы. Пороговые схемы, подобные схемам "тайного обмена" (например, [26]), предложены для перекомпоновки хранимых и обрабатываемых данных. При этом определение функций фрагментации общего назначения – достаточно сложный и дорогой вычислительный процесс. Подобный механизм может быть реализован в виде вычислительных элементов на дискретных аппаратных средствах. В этом случае необходимая TSB включает в себя входы и выходы обработки и хранения операций.

to the emergence of hidden errors in the project, or to the leakage of information through peripheral devices.

Thus, there is a new problem: how best to implement reliable project, knowing that the underlying reconfigurable logic can be infected by arbitrary hardware Trojan, and how to protect the integrity of the project after it is created, that is, to protect the bitstream from distortion or infection with a Trojan. General three-step approach to ensuring the integrity of the bitstream of the FPGA is proposed in [17]. First, the integrity of the configuration is checked by the reverse reading, secondly, in case of detection of incorrect configuration, the FPGA is partially modificeres (from original parts of the bitstream), and thirdly, if the system was compromised, the FPGA uses the request-response protocol for notifying a third party.

In [18] a review of the solutions on the protection of the bitstream of the FPGA and configuration memory from the event failures is proposed. The authors make the following conclusions:

- if the manufacturing of the FPGA and design of IC on an FPGA are completely separated, or third party's IP blocks are used, the attacker can easy make changes to any project;
- reverse engineering of configuration bitstreams is quite difficult both for the attacker, making impossible for him to understand the purpose of IC, and for search of backdoors;
- the bitstreams can be encrypted, which provides good protection, and then the internal hardware decoding of the bitstream is possible in many of the FPGA's. However, this encryption does not allow partial reconfiguration, therefore, is not applicable for applications such as adaptive

ЗАКЛЮЧЕНИЕ

В настоящее время пока нет единого решения, которое может обеспечить полную защиту от всего спектра угроз и механизмов активации аппаратных закладок при работе системы. Маловероятно, что такое решение когда-нибудь будет найдено, сочетание же контрмер необходимо для борьбы с конкретными классами аппаратных троянов в конкретных прикладных областях. Эти контрмеры должны разрабатываться с учетом систем, в которых они будут применены, а также с учетом обеспечиваемого уровня защиты. Как показано в работе [14], при разработке новых контрмер естественным образом возникают способы их обхода. Такая "гонка вооружений" в области аппаратных закладок диктует необходимость использования комплексных "глубоко эшелонированных" подходов к обеспечению безопасности электронных систем.

Статья подготовлена при финансовой поддержке Минобрнауки в рамках выполнения государственного задания 16.9021.2017/БЧ.

ЛИТЕРАТУРА

1. **Кузнецов Е., Сауров А.** Аппаратные трояны. Часть 1: Новые угрозы кибербезопасности // НАНОИНДУСТРИЯ. 2016. № 7 (69). С. 16–25.
2. **Кузнецов Е., Сауров А.** Аппаратные трояны. Часть 2: Способы предупреждения и обнаружения // НАНОИНДУСТРИЯ. 2017. № 1 (71). С. 30–40.
3. **Кузнецов Е., Сауров А.** Аппаратные трояны. Часть 3: Примеры реализации, способы внедрения и активации // НАНОИНДУСТРИЯ. 2016. № 8 (70). С. 12–21.
4. **Waksman A.**, Sethumadhavan S. Silencing hardware backdoors // Security and Privacy (SP), 2011 IEEE Symposium. IEEE, 2011. С. 49–63.
5. **Gentry C.** Computing arbitrary functions of encrypted data // Communications of the ACM. 2010. Т. 53. № 3. С. 97–105.
6. **Järvinen K.** et al. Garbled circuits for leakage-resilience: Hardware implementation and evaluation of one-time programs // Cryptographic Hardware and Embedded Systems, CHES 2010. Springer Berlin Heidelberg, 2010. С. 383–397.
7. **Chakraborty R.S.**, Narasimhan S., Bhunia S. Hardware Trojan: Threats and emerging solutions // High Level Design Validation and Test Workshop, 2009. HLDVT 2009. IEEE International. IEEE, 2009. С. 166–171.
8. **Kim L.W.**, Villasenor J.D., Koç C.K. A Trojan-resistant system-on-chip bus architecture // Military Communications Conference, 2009. MILCOM 2009. IEEE, 2009. С. 1–6.
9. **Das A.** et al. Detecting/preventing information leakage on the memory bus due to malicious hardware // Proceedings of the Conference on Design, Automation and Test in Europe. European Design and Automation Association, 2010. С. 861–866.
10. **Bloom G.** et al. Providing secure execution environments with a last line of defense against trojan circuit attacks // Computers & security. 2009. Т. 28. № 7. С. 660–669.

computing, which are an important niche for FPGA. Encryption of the bitstream does not protect against the penetration of hardware Trojans through IP blocks of third-party developers.

To identify the authenticity of the developed project it is proposed to use the additional configuration logic blocks (CLB) in the FPGA based on error correction code (ECC), which form an group for parity checking. Checking the parity of each element of the CLB allows to reveal all changes in the project. Two-stage randomization that is used to generate the parity signal, ensures the unpredictability of the result of the CLB.

In [19] the protection against hardware Trojans at the manufacturing stage is proposed. If to place a reconfigurable logic blocks between critical elements in the project, then some reconfigurable architecture in some areas of the chip is visible at the production stage. These blocks that are called "the barriers" by the authors are programmed after manufacture with use of a secret key that leads to the unlock of the project and its logical completion. If the location and functionality of these barriers are chosen optimally, then it will be difficult to activate any inserted hardware Trojans, and we can block their impact on IC. The authors pay special attention to the type of reprogrammable logic, key management, and various heuristic methods of placing barriers. Combining permanent and programmed logic, we can develop unique solutions against potential hardware Trojans. At the same time, reconfigurable logic can be used for the implementation of local protective mechanisms.

The use of reconfigurable logic as a protection against hardware backdoors poses new challenges

11. **Suh G.E.** et al. AEGIS: architecture for tamper-evident and tamper-resistant processing // Proceedings of the 17th annual international conference on Supercomputing. ACM, 2003. С. 160–171.
12. **Anderson M.**, North C. & Yiu K. Towards Countering the Rise of the Silicon Trojan // DSTO Technical Report DSTO-TR-2220. 2008. DSTO Information Sciences Laboratory.
13. **Hicks M.** et al. Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically // Security and Privacy (SP), 2010 IEEE Symposium. IEEE, 2010. С. 159–172.
14. **Sturton C.** et al. Defeating UCI: Building stealthy and malicious hardware // Security and Privacy (SP), 2011 IEEE Symposium. IEEE, 2011. С. 64–77.
15. **Abramovici M., Bradley P.** Integrated circuit security: new threats and solutions // Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies. ACM, 2009. С. 55.
16. **Deng D.Y., Chan A.H., Suh G.E.** Hardware authentication leveraging performance limits in detailed simulations and emulations // Proceedings of the 46th Annual Design Automation Conference. ACM, 2009. С. 682–687.
17. **Webb J.B.** Methods for securing the integrity of FPGA configurations. Дис. Virginia Polytechnic Institute and State University, 2006.
18. **Trimberger S.** Trusted design in FPGAs // Design Automation Conference, 2007. DAC'07. 44th ACM/IEEE. IEEE, 2007. С. 5–8.
19. **Baumgarten A., Tyagi A., Zambreno J.** Preventing IC piracy using reconfigurable logic barriers // IEEE Design and Test of Computers. 2010. Т. 27. № 1. С. 66–75.
20. **Huffmire T.** et al. Moats and drawbridges: An isolation primitive for reconfigurable hardware based systems // Security and Privacy, 2007. SP'07. IEEE Symposium. IEEE, 2007. С. 281–295.
21. **Silva M.L., Ferreira J.C.** Creation of partial FPGA configurations at run-time // Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference. IEEE, 2010. С. 80–87.
22. **Newgard B., Hoffman C.** Using multiple processors in a single reconfigurable fabric for high-assurance applications // Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on. IEEE, 2010. С. 25–29.
23. Mc**Intyre D.** et al. Dynamic evaluation of hardware trust // Hardware-Oriented Security and Trust, 2009. HOST›09. IEEE International Workshop. IEEE, 2009. С. 108–111.
24. **Kumar S.S.** et al. The butterfly PUF protecting IP on every FPGA // Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop. IEEE, 2008. С. 67–70.
25. **Trouessin G.** et al. Improvement Of Data Processing Security By Means Of Fault Tolerance // Proceedings Of The 14th National Computer Security Conference (NCSC'14). 1991. С. 295–304.
26. **Shamir A.** How to share a secret // Communications of the ACM. 1979. Т. 22. № 11. С. 612–613.

for design and verification, and moves the focus from the semiconductor manufacturing to the RTL project. To implement an effective hardware backdoor on the level of IC in this case, an attacker would have to provide the relationship (agreement) between the manufacturer and software supplier. Implementation of a complex reconfigurable logical project, usually is based on the integration of multiple IP blocks. In [20] the idea of "moats and drawbridges" as isolation primitives is proposed, which are used in case of multiple IP blocks. They help block illegal arms and interconnects inside the reconfigurable blocks.

Let's list some other approaches with the use of reconfigurable logic that can be used for combating hardware Trojans:

- partial and dynamic reprogramming of the logic blocks[21];
- encrypting the configuration bitstreams [18];
- replication and lock-stepping of logic [22];
- use of functionally identical, but having different architecture logic blocks [23];
- generation of unambiguous hardware modules with the use of random numbers [24].

REPLICATION OF PARTS, FRAGMENTATION AND VOTING

Development of effective hardware Trojans requires understanding of the operation of the IC project from the level of gates, RTL level, IC level, and ending with the macro level of the entire system. At all these levels to combating hardware Trojans the following general approaches can be applied:

- replication or doubling of logic and/or data;
- separation or fragmentation of logic and/or data;
- dispersion or distribution of logic and/or data;

- accumulation and merging of logical functions and/or data, for example, using the voting.

These general countermeasures are effective in three cases:

- for protection against hardware Trojans, leading to the leaking of confidential information by separating the data and its processing by independent logical elements;
- for protection against functional or specificational modifications of elements using multiple copies or duplicates of the logical blocks;
- for protection against DoS attacks by setting a redundancy operating logical elements in the project.

These countermeasures may be placed at different levels: gate, RTL, logic design, functional modules, IP cores, up to the level of IC and devices on the macro level. Protection mechanisms assume that there is no "collusion" between replicated or duplicated elements in the project.

Method of dynamic evaluation of verification of the equipment during its operation is proposed in [23]. Its essence lies in the detection of hardware Trojan in the system, after which it continues to operate with the removal or small use of suspicious items. The authors propose to use a multi-core data processing system with redundancy, rejecting the cores that are not credible. Functionally equivalent versions of the processes are performed on multiple processors with comparison of their results. Different variants of the same CPU processings can be performed based on different compilations or implementations. Different data processing algorithms can also be used. If the results of the two elements differ, the calculations are carried out at the third element with comparison of the three results. This process continues until you reach the conformity at least between two elements of data processing. Processor elements that give inconsistent (incorrect) results are dynamically "penalized", i.e., "trust" to them is less and they are used less.

This method can be extended with the use of random sampling of variants of functionally equivalent hardware. It can also be applied at different levels of abstraction, for example, of instructions, gate, software, or IC. If the method is used on the command level, the activity on it can be transparent to higher levels, including for small trusted computing base (TSB), which can verify command level schedules, selection of the replication blocks, options specification, and voting.

In [22] it is proposed a FPGA-based configuration of dual-processor architecture with direct connections, which at the macrolevel is an implementation of replication and voting. Both processors receive and process the same instructions at the same time. Hardware-implemented logic checks and compares all of the control signals of each transaction on the bus. If an error is detected, the system is forcibly introduced into the sequence to fix the error. For adequate counteraction to hardware Trojans a complete verification of TSB-blocks of detecting and fixing errors must be carried out. The method can be extended for a larger number of processors, which can be single ICs or parts of the same FPGA.

In 1991, in [25] the issues of high reliability and efficiency, as well as the preservation of data privacy in large distributed systems was studied. The threats associated with hardware Trojans, have not been considered, however, the proposed methods of fault tolerance are relevant to counter unauthorized backdoors. The general approach involves dividing the data into small fragments, so that each of them contains rather little information. It can be used both for data storage and processing (accordingly, they are grouped as fragments for the storage and processing). Replication (redundancy) of fragments is used to ensure the reliability of the system. Threshold schemes like a "secret sharing" (e.g. [26]) are proposed for rearranging the stored and processed data. In this case, determination of fragmentation functions of general purpose is quite difficult and expensive computational process. Such a mechanism can be implemented in the form of computational elements on discrete hardware. In this case, the TSB includes the inputs and outputs for processing and storage of operations.

CONCLUSION

Currently, there is no single solution that can provide complete protection against the entire spectrum of threats and mechanisms of activation of the hardware backdoors during operation of the system. It is unlikely that such a solution will ever be found, and a combination of countermeasures is necessary to counter specific classes of hardware Trojans in specific application areas. These countermeasures should be developed taking into account systems in which they are applied, and also given the provided level of protection. As it is shown in [14], the development of new countermeasures in a natural way creates ways to work around them. This "arms race" in the field of hardware Trojans necessitates the use of integrated "deeply layered" approaches to security of electronic systems. ■

This paper was created with the financial support of the Ministry of Education and Science of the Russian Federation within the framework of the state order 16.9021.2017/БЧ.