УДК 004.056.53

ПОИСК ЦЕЛЕВЫХ БЛОКОВ LUT В ИНФОРМАЦИОННОЙ МОДЕЛИ FPGA-УСТРОЙСТВА В РАМКАХ ЗАДАЧИ КОНТРОЛЯ ЦЕЛОСТНОСТИ ПРОГРАММНОГО КОДА

К. В. Защелкин, Е. Н. Иванова

Одесский национальный политехнический университет

Аннотация. *Рассмотрена проблема обеспечения целостности программного кода FPGA-базированных компонентов компьютерных систем. Отмечено, что перспективным направлением контроля целостности компонентов такого рода является встраивание контрольного хэша в программный код в виде цифрового водяного знака. Предложена формализованная процедура поиска в информационной модели схемы FPGA-устройства целевых блоков LUT, предназначенных для непосредственного внедрения цифрового водяного знака.*

Ключевые слова: *FPGA, контроль целостности, LUT-ориентированная архитектура, программный код, цифровой водяной знак, программируемые логические интегральные схемы.*

Введение

Для построения компьютерных систем используются аппаратные компоненты, как с жесткой логикой функционирования, так и программно-управляемые устройства. Возможность программного управления позволяет менять поведение устройства в динамике его жизненного цикла. Среди программно-управляемых устройств принято выделять два класса, существенно различающихся принципами программирования: 1) микропроцессоры; 2) программируемые логические интегральные схемы (ПЛИС). Различие принципов программирования этих двух классов состоит в том, что микропроцессоры последовательно выполняют команды программы при помощи неизменяемой аппаратной структуры, а ПЛИС меняют свою структуру под воздействием конфигурирующего программного кода. По показателям производительности микропроцессоры уступают ПЛИС [1]. В силу этого ПЛИС применяются для решения задач, требующих уровня производительности, который в данных условиях не могут обеспечить микропроцессоры.

Наиболее часто используемой на текущий момент разновидностью ПЛИС является микросхемы FPGA (Field Programmable Gate Array) [2]. FPGA представляют собой матрицу элементарных программируемых блоков, обеспеченную системой программируемых межсоединений. Множество элементарных блоков FPGA представлено вычислительными блоками общего назначения, блоками ввода-вывода и специализированными блоками (блоками сосредоточенной памяти, блоками умножения, блоками PLL [3], а для отдельных серий FPGA, еще и рядом специфических блоков). Наиболее массовыми в структуре FPGA являются вычислительные блоки общего назначения, содержащие в своем составе программируемый вычислитель LUT [4] (Look-Up Table) и программируемый элемент памяти. Блок LUT выполняет вычисление логической функции от n аргументов. Настройка такого блока на реализацию конкретной логической функции выполняется при помощи 2^n разрядного программного кода.

Функции всех блоков и их соединения управляются программным кодом, размещаемым в конфигурационной памяти FPGA. Изменение содержимого конфигурационной памяти (программного кода) приводит к изменению поведения микросхемы FPGA, т.е. к ее перепрограммированию.

Для микросхемы FPGA характерна проблема обеспечения целостности программного кода. Поскольку программный код микросхем такого рода полностью определяет их функционирование, то нарушение целостности программного кода может приводить к некорректному функционированию FPGA-базированных систем. Это особенно критично для систем, которые управляют техническими объектами повышенного риска [5]. В силу этого обеспечение целостности программного кода микросхем FPGA составляет важную техническую задачу, являющуюся частью комплексной проблемы обеспечения надежности FPGA-базированных компьютерных систем.

1. Анализ проблемы и постановка цели работы

Методы контроля целостности программного кода, хорошо себя зарекомендовавшие для микропроцессоров [6], не могут быть в полной мере использованы для микросхем FPGA в силу

различий архитектур и принципов функционирования микросхем этих классов.

Большинство известных методов контроля целостности основано на вычислении хэша (дайджеста) [7] для информационного объекта, целостность которого контролируется. Хэш представляет собой двоичную последовательность фиксированной длины, которая получается в результате преобразования информационного объекта (в рассматриваемом случае, этот информационный объект – программный код FPGA) при помощи одной из специальных хэш-функций [8]. Основные свойства хэша, которые позволяют выполнить контроль целостности информационного объекта состоят в следующем: а) фиксированная длина хэша, не зависящая от величины контролируемого информационного объекта; б) необратимость – крайне высокая вычислительная сложность нахождения значения информационного объекта по его хэшу; в) значительное изменение хэша при незначительном изменении информационного объекта.

Одним из принципиальных атрибутов для методов контроля целостности является место хранения хэша. В классе микропроцессоров широкое распространение получили методы [9] контроля целостности, в рамках которых хэш программного кода: а) храниться отдельно от файлов программного кода; б) присоединяется к программному коду путем конкатенации; в) помещается в специально выделенное в формате программного кода поле.

Использование методов такого рода для микросхем FPGA затруднительно в связи с тем, что [10]: а) конфигурационная память FPGA обычно не предусматривает хранение дополнительной информации (отличной от конфигурационной); б) конфигурационные файлы FPGA обычно не предусматривают наличия дополнительных полей, в которые можно было бы поместить контролирующий целостность хэш.

Однако даже если указанные сложности удается устранить, используя искусственные решения, открытое хранение хэша и даже сам факт его наличия, создают возможность осуществления попыток фальсификации целостности.

Известен подход к контролю целостности программного кода, в рамках которого контролирующий хэш внедряется в программный код FPGA в виде цифрового водяного знака (ЦВЗ) [11]. Преимущества такого подхода состоят в том, что:

а) не требуется наличие дополнительных объемов памяти или полей конфигурационного файла для хранения хэша;

б) информация, контролирующая целостность, скрыта от стороннего наблюдателя;

в) сам факт того, что контроль целостности выполняется, не очевиден стороннему наблюдателю.

Особенностью указанного подхода является то, что функционирование устройства после внедрения ЦВЗ в программный код не претерпевает изменений по сравнению с функционированием, которое имело место до внедрения ЦВЗ. При использовании такого подхода хэш вычисляется для всего программного кода FPGA, а приемником ЦВЗ (включающего этот хэш) является программный код заданного подмножества блоков LUT. Контроль целостности при этом обеспечивается возможностью восстановления первоначального состояния программного кода при извлечении из него ЦВЗ [12], [13]. Под первоначальным состоянием здесь понимается то состояние, которое программный код имел до внедрения в него ЦВЗ.

Внедрение ЦВЗ в программный код FPGA обеспечивается путем эквивалентного изменения программного кода целевых блоков LUT в соответствии с принципами, изложенными в работах [14], [15]. Для этого в структуре FPGA-проекта должны быть выделены целевые пары последовательно подключенных блоков LUT – приемников ЦВЗ. Встраивание ЦВЗ осуществляется путем управляемого инвертирования программного кода первых блоков LUT каждой из таких пар с последующей перестановкой разрядов программного кода вторых блоков LUT пар для компенсации инверсии.

Для выполнения встраивания ЦВЗ в программный код необходимо выполнить несколько подготовительных этапов:

а) получить низкоуровневую информацию о размещении схемы устройства в пространстве FPGA. Под низкоуровневой информацией здесь понимается программный код элементарных блоков FPGA и списки их соединений друг с другом, а также с внешними выводами микросхемы;

б) на основании полученной информации, построить информационную модель схемы устройства, размещенной в FPGA;

в) выполнить приведение этой модели к виду, пригодному для выполнения встраивания ЦВЗ;

г) выбрать (с учетом заданных естественных ограничений и ограничений секретного ключа встраивания ЦВЗ) из информационной модели целевые блоки LUT, в которые будет непосредственно встраиваться ЦВЗ.

Цель данной работы состоит в формализации процедуры поиска целевых блоков LUT, в

информационной модели схемы, размещенной в пространстве FPGA. Указанная процедура необходима для обеспечения встраивания цифрового водяного знака с контрольной информацией в программный код FPGA-базированного устройства. Процедура является частью процесса обеспечения целостности программного кода для устройств такого вида.

2. Процедура поиска целевых блоков LUT для внедрения контролирующего целостность ЦВЗ

Стоит задача формализации процедуры поиска целевых блоков LUT для внедрения разрядов ЦВЗ. Поиск заключается в следующем. На множестве связанных между собой блоков LUT необходимо найти пары последовательно соединенных блоков при наличии ограничений, регламентирующих выбор результирующих пар.

Предлагаемая в данной работе процедура рассматривается на примере микросхем и программного обеспечения компании Altera [3], [4] (подразделение корпорации Intel), которая является одними из основных производителей FPGA. Предложения данной работы в силу унификации архитектуры справедливы и для микросхем других производителей FPGA.

Компания Altera обеспечивает процесс проектирования FPGA-базированных устройств при помощи собственной САПР Altera Quartus. Информация о размещении программируемых блоков в пространстве микросхемы FPGA храниться во внутренней базе данных Quartus-проекта. САПР Altera Quartus имеет программный интерфейс API (Application Program Interface) через который может быть решена, в том числе, и задача получения низкоуровневой информации о размещении программного кода. В базе данных Quartus-проекта информация о размещении схемы представлена в виде совокупность узлов (nodes) и связей между ними. Каждому из узлов соответствует программируемый блок FPGA. Указанные узлы подразделяются на следующие типы: lcell – логические ячейки (вычислительные блоки общего назначения); io – блоки ввода-вывода; pll – phase locked loop блоки; dsp – digital signal processing блоки; ram – блоки сосредоточенной оперативной памяти. Каждый из узлов типа lcell может включать в себя пару узлов, относящихся к подтипам LCCOMB (блок LUT) и FF (программируемый триггер).

В работе [16] предложена методика получения информационной модели LUT-схемы, размещенной в пространстве микросхемы FPGA. Эта методика формализует: получение низкоуровневых данных о схеме; построение информационной модели FPGA-базированного устройства; редукцию (сокращение) информационной модели путем удаления избыточной информации. Предлагаемая в данной работе процедура формализует этапы обработки проектной информации, которые следуют после действий указанной методики [16].

На рис. 1 представлен граф потока данных, который описывает структуру программного обеспечения, обеспечивающего внедрение ЦВЗ с информацией, контролирующей целостность в программный код FPGA-базированного устройства. Функционирование программных модулей M_1–M_3 формализовано в методике [16]. В данной работе предлагается формализованная процедура, определяющая функционирование модуля M_4. Предлагаемая процедура в качестве входных данных получает информационную модель схемы FPGA-базированного устройства и описания ограничений встраивания ЦВЗ. Результатом выполнения процедуры, предлагаемой в данной работе, является упорядоченный список целевых блоков LUT, в программный код которых на последующих этапах будет производиться встраивание ЦВЗ.

Информационная модель, являющаяся элементом входных данных предлагаемой процедуры представлена в виде JSON-массива. Каждый элемент этого массива содержит JSON-объект (объект типа «ключ-значение»), описывающий отдельный узел схемы устройства. Формат такого объекта представлен в табл. 1. Атрибуты, помеченные знаком «*» присутствуют только в объектах, соответствующе узлам подтипа LCCOMB (блоки LUT), для узлов подтипа FF эти атрибуты отсутствуют.

Таблица 1

Формат JSON-объекта, описывающего узел схемы в информационной модели

Ключ	**Описание значения**	
nodeId	Идентификатор узла в матрице FPGA	
nodeName	Символьное имя узла в базе Quartus	
location	Строка, содержащая координаты узла в матрице FPGA	
mode*	Режим функционирования узла (normal/arithmetic)	
code*	Шестнадцатеричное представление программного кода	
inputPorts	Массив входных портов узла	
	name	имя порта
	source	идентификатор узла-источника сигнала для данного входного порта
outputPorts	Массив выходных портов узла	

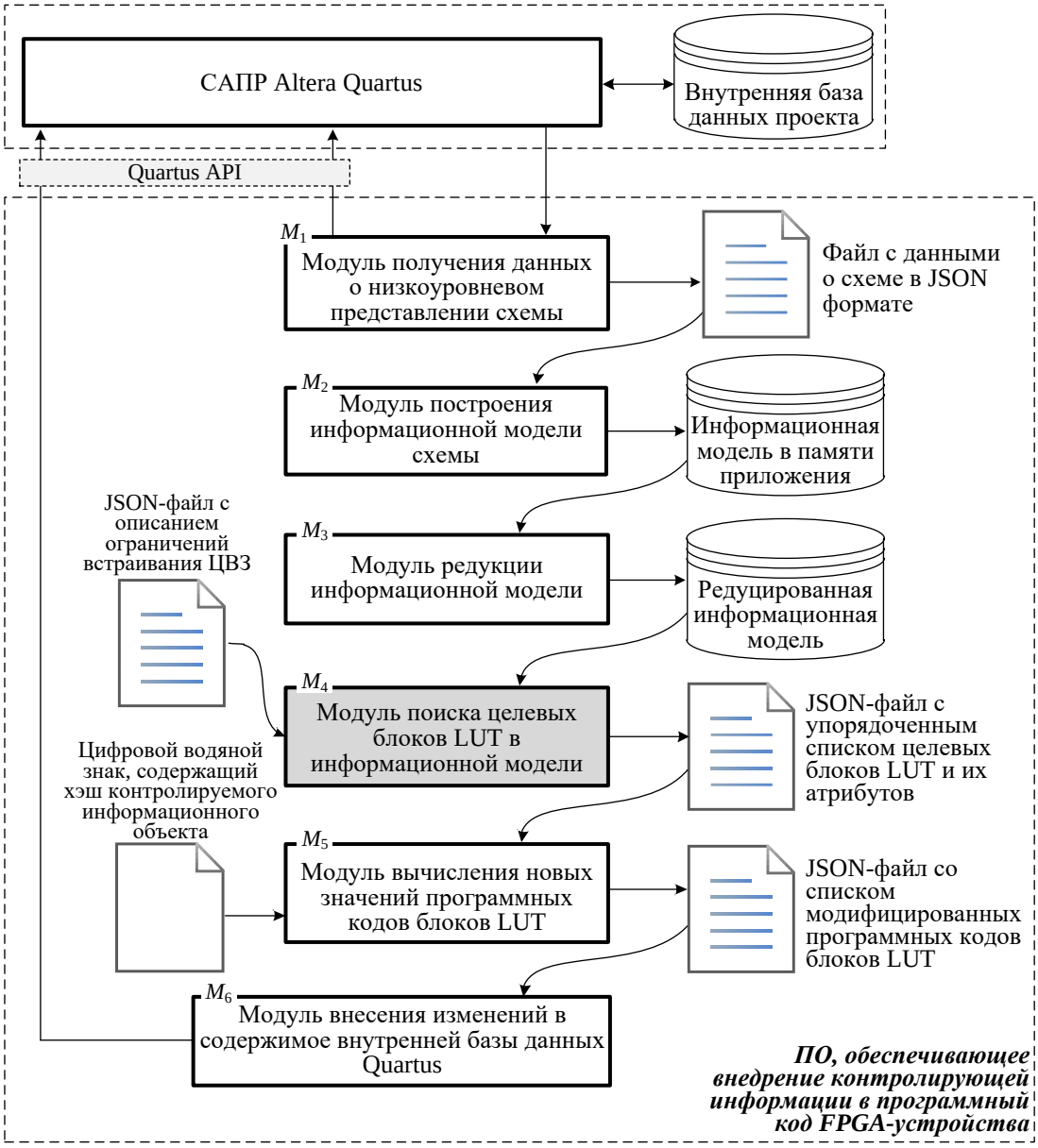

Рис. 1. Граф потока данных, описывающий структуру программного обеспечения, выполняющего внедрение в программный код ЦВЗ с контрольной информацией

Для поиска целевых блоков LUT необходима информация о связях на полном множестве блоков LUT в схеме устройства. В представленной информационной модели и в базе данных САПР Altera Quartus для каждого узла имеется информация об источнике сигнала для каждого из входных портов этого узла. Эта информация полностью описывает всю совокупность связей между узлами. Однако выполнение поиска целевых блоков по информации, представленной таким образом, характеризуется крайне высокой вычислительной сложностью. Это связано с тем, что при выполнении поиска необходимо выделить пары последовательно соединенных блоков LUT и выбрать пары, удовлетворяющие условиям ограничений встраивания ЦВЗ.

На рис. 2 показан фрагмент схемы, образованный десятью блоками: LUT_1–LUT_{10}. Схема получена по редуцированной модели схемы FPGA-базированного устройства, в которой удалены все блоки, отличные от блоков LUT. На представленном фрагменте блоки LUT образуют семь пар: LUT_1–LUT_8, LUT_1–LUT_4, LUT_2–LUT_4, LUT_3–LUT_4, LUT_3–LUT_9, LUT_4–LUT_{10}, LUT_5–LUT_6. Процесс поиска таких пар по имеющейся в информационной модели информации вида «порту блока LUT_i соответствует источник сигнала для этого порта» вычислительно сложен и плохо формализуем. Предлагается выполнить преобразование информации о связях блоков LUT путем приведения ее к виду «блок LUT_i соответствует блоку LUT_j, который является приемником выходного сигнала блока LUT_i».

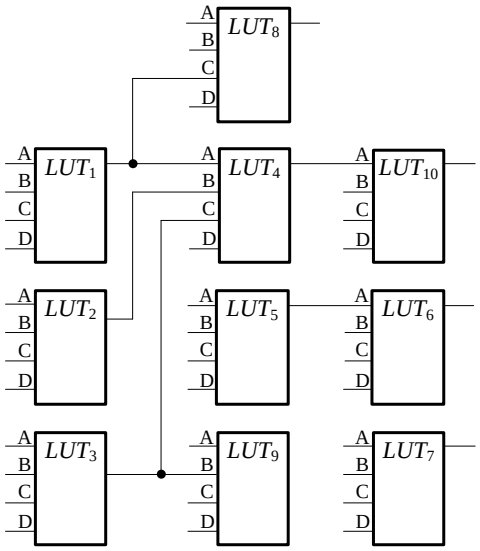

Рис. 2. Пример: фрагмент схемы, полученной из информационной модели FPGA-базированного устройства

На рис. 3 представлена блок-схема предлагаемой процедуры поиска целевых блоков LUT. Процедура базируется на последовательном переборе всех узлов (блоков LUT), содержащихся в информационной модели схемы. Основой процедуры является получение списка всех узлов в модели, для которых текущий узел является источником сигнала. Локальный результат, полученный для узла, являющегося текущим на данной итерации цикла, оценивается на предмет соответствия ограничениям встраивания ЦВЗ. На рис. 3 эта оценка состоит в сравнении количества узлов, для которых текущий узел является источником сигнала. При превышении этим количеством заданного ограничениями значения, текущий узел и образованные им пары не включаются в результаты глобального поиска целевых блоков. Эта оценка соответствия ограничениям может быть дополнена. В частности в такую оценку могут быть включены требования правил формирования стего-пути (пути встраивания ЦВЗ), которые являются компонентами секретного ключа встраивания и извлечения ЦВЗ. Результатом выполнения предложенной процедуры является список, включающий в себя локальные списки связей блоков LUT, образующих пары целевых блоков для процесса встраивания ЦВЗ.

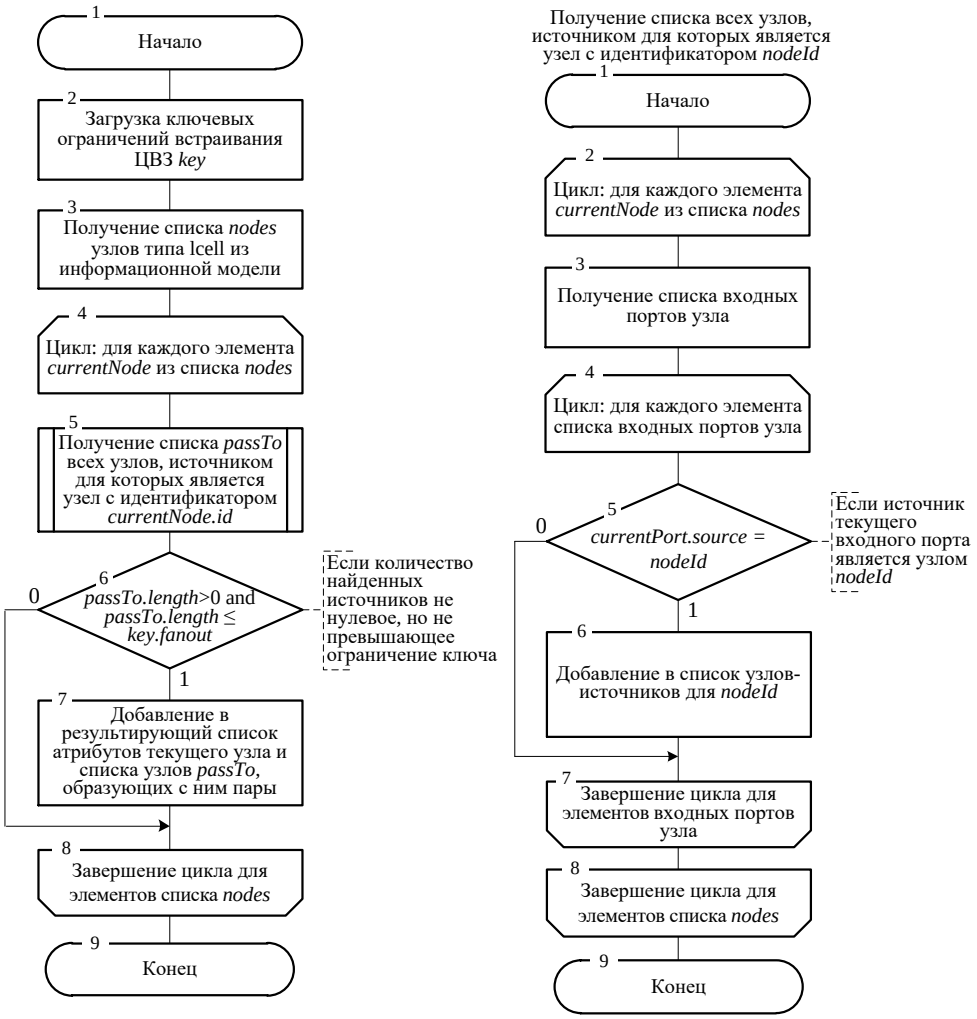

Рис. 3. Блок схема предлагаемой процедуры поиска целевых блоков LUT

Выводы

Предложенная в работе формализованная процедура позволяет найти в информационной модели схемы FPGA-базированного устройства совокупность целевых блоков LUT для встраивания разрядов ЦВЗ, который содержит информацию для контроля целостности программного кода устройства.

В работе обоснована возможность практического применения предложенной методики посредством программного приложения, взаимодействующего с САПР FPGA через соответствующий интерфейс API.

Методика может найти применение для организации подсистемы подготовки данных в рамках системы контроля целостности программного кода микросхем FPGA. Дальнейших исследований требует вопрос оптимизации поиска в информационной модели, представленной как совокупность объектов вида «ключ–значение».

Список использованной литературы

1. Vanderbauwhede, W. High-performance computing using FPGAs [Text] / W. Vanderbauwhede, K. BenkridSpringer. – New-York: CRC Press. – 2016. – 525 p.

2. Green Experiments with FPGA [Text] / A. Drozd, J. Drozd, S. Antoshchuk, V. Antonyuk, K. Zashcholkin, M. Drozd, O. Titomir // Green IT Engineering: Components, Networks and Systems Implementation / eds. V. Kharchenko, Y. Kondratenko, J. Kacprzyk. – Berlin: Springer International Publishing. – 2017. – Vol. 105, P. 219–239. DOI: 10.1007/978-3-319-55595-9_11

3. Clock Networks and PLLs in Cyclone IV Devices [Electronic resource] – Режим доступу: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-iv/cyiv-51005.pdf (дата звернення 02.04.2018).

4. Logic Elements and Logic Array Blocks in Cyclone IV Devices [Electronic resource] – Режим доступу: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-iv/cyiv-51002.pdf (дата звернення 02.04.2018).

5. Drozd, M. Safety-Related Instrumentation and Control Systems and a Problem of the Hidden Faults [Text] / M. Drozd, A. Drozd // 10th International Conference on Digital Technologies, Zhilina, Slovak Republic. – 2014 – P. 137–140. DOI: 10.1109/ DT.2014.6868692.

6. Vacca, J. Computer and information security, 2nd edition [Text] / J. Vacca. – USA, Waltham: Morgan Kaufmann Publishers, 2013. – 1280 p.

7. Ferguson, N. Cryptography engineering [Text] / N. Ferguson, B. Schneier, T. Kohno. – Hoboken: Wiley, 2013. – 354 p.

8. Vasu, S. An Integrity Verification System for Images Using Hashing and Watermarking [Text]/ S. Vasu, S. George, P. Deepthi // Proceedings of the International Conference on Communication Systems and Network Technologies. – 2012. – P. 85–89.

9. Andress, J. Building a Practical Information Security Program [Text] / J. Andress, M. Leary. – Cambdidge: Syngress. – 2016. – 202 p.

10. Andina, J. FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics [Text] / J. Andina. – CRC Press. – 2017.

11. Shih, F. Multimedia Security: Watermarking, Steganography, and Forensics [Text] / F. Shih. – Boston: CRC Press. – 2013. – 424 p.

12. Zashcholkin, K. The Control Technology of Integrity and Legitimacy of LUT-Oriented Information Object Usage by Self-Recovering Digital Watermark [Text] / K. Zashcholkin, O. Ivanova // CEUR Workshop Proceedings. – 2015. – Vol. 1356. – P. 486–497.

13. Защелкин, К.В. Информационная технология внедрения самовосстанавливающих цифровых водяных знаков в LUT-ориентированные контейнеры [Текст] / К.В. Защелкин, Е.Н. Иванова // Электротехнические и компьютерные системы. – 2014. – №16 (92). – С. 78–84.

14. Drozd, A. Use of natural LUT redundancy to improve trustworthiness of FPGA design [Text] / A. Drozd, M. Drozd, M. Kuznietsov // CEUR Workshop Proceedings. – 2016. – Vol. 1614. – P. 322–331.

15. Drozd, O. Improving of a circuit checkability and trustworthiness of data processing results in LUT-based FPGA components of safety-related systems [Text] / O. Drozd, M. Drozd, O. Martynyuk, M. Kuznietsov // CEUR Workshop Proceedings. – 2017. –Vol. 1844. – P. 654–661.

16. Защелкин, К.В. Методика и программная реализация построения информационной модели LUT-схемы, размещенной в среде FPGA [Текст] / К.В. Защелкин // Вісник Кременчуцького національного університету імені М. Остроградського. – №1 (108). – 2018. – С. 46–51.

References

1. Vanderbauwhede, W. and Benkrid, K. (2016). *High-performance computing using FPGAs*. Springer, New-York.

2. Drozd, A., Drozd, J., Antoshchuk, S., Antonyuk, V., Zashcholkin, K., Drozd, M. and Titomir, O. (2017). Green Experiments with FPGA, In: V. Kharchenko, Y. Kondratenko and J. Kacprzyk, ed. *Green IT Engineering: Components, Networks*

and Systems Implementation, vol. 105. Berlin, Springer International Publishing, pp. 219-239. DOI: 10.1007/978-3-319-55595-9_11

3. *Clock Networks and PLLs in Cyclone IV Devices* [online] Available at: www.altera.com/ content/dam/altera-www/global/en_US/pdfs/literature/hb/ cyclone-iv/cyiv-51005.pdf [Accessed 02.04.2018].

4. *Logic Elements and Logic Array Blocks in Cyclone IV Devices* [online] Available at: https://www.altera.com/content/dam/altera-www/ global/en_US/pdfs/literature/hb/cyclone-iv/cyiv-51002.pdf [Accessed 02.04.2018].

5. Drozd, M. and Drozd, A. (2014). Safety-Related Instrumentation and Control Systems and a Problem of the Hidden Faults. In: *10th International Conference on Digital Technologies*, Zhilina, Slovak Republic, pp. 137–140. DOI: 10.1109/DT.2014.6868692

6. Vacca, J. (2013). *Computer and information security*, 2nd edition. USA, Waltham: Morgan Kaufmann Publishers.

7. Ferguson, N., Schneier, B. and Kohno, T. (2013). *Cryptography engineering*. Hoboken: Wiley.

8. Vasu, S., George, S. and Deepthi, P. (2012). An Integrity Verification System for Images Using Hashing and Watermarking. In: *Proceedings of the International Conference on Communication Systems and Network Technologies*, pp. 85–89.

9. Andress, J. and Leary, M. (2016). *Building a Practical Information Security Program*. Cambridge: Syngress.

10. Andina, J. (2017). *FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics*. CRC Press.

11. Shih, F. (2013). *Multimedia Security: Watermarking, Steganography, and Forensics*. Boston: CRC Press.

12. Zashcholkin, K. and Ivanova, O. (2015). The Control Technology of Integrity and Legitimacy of LUT-Oriented Information Object Usage by Self-Recovering Digital Watermark. *CEUR Workshop Proceedings*, vol. 1356, pp. 486–497.

13. Zashcholkin, K. and Ivanova, O. (2014). Information technology of embedding self-recovery digital watermark in LUT-oriented containers [Informacionnaja tehnologija vnedrenija samovosstanavlivajushhih cifrovyh vodjanyh znakov v LUT-orientirovannye kontejnery] *Electrotrotechnic and computer systems*, No16 (92), pp. 78–84. DOI: 10.15276/etks.16.92.2014.11

14. Drozd, A., Drozd, M. and Kuznietsov, M. (2016). Use of natural LUT redundancy to improve trustworthiness of FPGA design. *CEUR Workshop Proceedings*, vol. 1614, pp. 322–331.

15. Drozd, O., Drozd, M., Martynyuk, O. and Kuznietsov, M. (2017). Improving of a circuit checkability and trustworthiness of data processing results in LUT-based FPGA components of safety-related systems, *CEUR Workshop Proceedings*, vol. 1844, pp. 654–661.

16. Zashcholkin, K. (2018). The technique and software implementation of creation of the LUT-circuit information model placed in FPGA environment [Metodika i programmnaja realizacija postroenija informacionnoj modeli LUT-shemy, razmeshhennoj v srede FPGA], *Transactions of Kremenchuk Mykhailo Ostrohradskyi National University*, No 1 (108), pp. 46–51.

THE SEARCH OF LUT UNITS IN INFORMATION FPGA-BASED DEVICE MODEL WITHIN THE FRAMEWORK OF PROGRAM CODE INTEGRITY MONITORING

K. V. Zashcholkin, O. M. Ivanova

Odessa National Polytechnic University

Abstract. *The problem of provision of the program code integrity of the computer system FPGA-based components is analyzed. It is noted in the article that the perspective direction of integrity monitoring of such kind of components is the embedding of monitoring hash immediately in a program code in the form of digital watermark. It is also noted that one of the important stages of preparation for embedding the digital watermark in FPGA program code is the LUT unit selection from the information FPGA-based device model. The mentioned units are the place of immediate location of the digital watermark. The unit should be selected with considering the natural restrictions and secret key ones used for embedding the digital watermark. A formalized procedure of the target LUT unit search in the information model of FPGA-device circuit was proposed. This unit program code is the place of immediate imbedding the digital watermark. The approaches to software implementation of the offered procedure are considered. The analysis of CAD Altera Quartus structure, in the environment of which the target procedure is to be implemented, was made. As a result of analysis the possibility of interaction of software realizing the proposed procedure with CAD Altera Quartus through the corresponding software interface API Quartus was found out. The possibility to obtain the information necessary for the creation of LUT-circuit information model through API Quartus was researched. The approaches to the automated analysis of program code and structure of FPGA-projects with*

the view of their integrity monitoring were further developed. The procedure offered in the work and the software, which implements it, can be applied in organizing the data preparation subsystem within the framework of the system of FPGA chip program code integrity monitoring.

Keywords: *FPGA, integrity monitoring, LUT-oriented architecture, program code, digital watermark, programmable logic integrated circuits.*

ПОШУК ЦІЛЬОВИХ БЛОКІВ LUT В ІНФОРМАЦІЙНІЙ МОДЕЛІ FPGA-ПРИСТРОЮ В МЕЖАХ ЗАДАЧІ КОНТРОЛЮ ЦІЛІСНОСТІ ПРОГРАМНОГО КОДУ

К. В. Защолкін, О. М. Іванова

Одеський національний політехнічний університет

Анотація. *Розглянуто проблему забезпечення цілісності програмного коду FPGA-базованих компонентів комп'ютерних систем. Відзначено, що перспективним напрямком контролю цілісності компонентів такого роду є вбудовування контрольного хеша безпосередньо в програмний код у вигляді цифрового водяного знаку. Також відзначено, що одним з важливих етапів підготовки до вбудовування цифрового водяного знаку в програмний код FPGA є вибір цільових блоків LUT з інформаційної моделі FPGA-базованого пристрою. Зазначені блоки є місцем безпосереднього розміщення цифрового водяного знаку. Вибір блоків повинен провадитися з урахуванням природних обмежень і обмежень секретного ключа, використовуваного для вбудовування цифрового водяного знаку. Запропоновано формалізовану процедуру пошуку в інформаційній моделі схеми FPGA-пристрою цільових блоків LUT, призначених для безпосереднього вбудовування цифрового водяного знаку. Розглянуто підходи до програмної реалізації запропонованої процедури. Виконаний аналіз структур САПР Altera Quartus, в середовищі якого пропонується реалізувати зазначену формалізовану процедуру. Виявлено можливість взаємодії програмного забезпечення, що реалізує запропоновану процедуру, з САПР Altera Quartus через відповідний API інтерфейс Quartus. Досліджена можливість отримання через API Quartus інформацію, необхідну для побудови інформаційної моделі LUT-схем. Дістали подальшого розвитку підходи до автоматизованого аналізу структури та програмного коду FPGA-проектів з метою контролю їх цілісності. Пропоновані в роботі формалізована процедура та програмне забезпечення, яке її реалізує, можуть знайти застосування для організації підсистеми підготовки даних в рамках системи контролю цілісності програмного коду мікросхем FPGA.*

Ключові слова: *FPGA, контроль цілісності, LUT-орієнтована архітектура, програмний код, цифровий водяний знак, програмовані логічні інтегральні схеми.*

Получено 25.03.2018

Защелкин Константин Вячеславович, кандидат технических наук, доцент кафедры Компьютерных интеллектуальных систем и сетей Одесского национального политехнического университета. Просп. Шевченко, 1, Одесса, Украина, E-mail: const-z@te.net.ua, тел.: (048) 734-83-22

Zashcholkin Kostiantyn, PhD, Associate Professor, Department of Computer Intellectual Systems and Networks, Odessa National Polytechnic University, Shevchenko ave., 1, Odessa, Ukraine, E-mail: const-z@te.net.ua, tel.: (048) 734-83-22

ORCID ID: 0000-0003-0427-9005

Иванова Елена Николаевна, старший преподаватель кафедры Компьютерных систем Одесского национального политехнического университета. Просп. Шевченко, 1, Одесса, Украина, E-mail: enivanova@ukr.net, тел.: (048) 734-83-91

Ivanova Olena, Senior Lecturer, Department of Computer Systems, Odessa National Polytechnic University, Shevchenko ave., 1, Odessa, Ukraine, E-mail: enivanova@ukr.net, tel.: (048) 734-83-91

ORCID ID: 0000-0002-4743-6931