УДК 004.056.55:004.274

Е. В. БРОШЕВАН, А. Е. ПЕРЕПЕЛИЦЫН, В. С. ХАРЧЕНКО

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков, Украина

МАСШТАБИРУЕМАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ ШИФРОВАНИЯ НА ПЛИС: АЛГОРИТМ IDEA

Приводятся результаты обзора реализации на ПЛИС алгоритмов шифрования данных. Предлагается общий подход к реализации масштабируемых проектов ПЛИС (FPGA). Анализируются особенности алгоритма IDEA в контексте задачи масштабирования. Предлагается вариант его масштабируемой реализации на ПЛИС с помощью статической параметризации количества параллельных и последовательных конструкций на всех уровнях декомпозиции проекта. Рассматривается пример использования IP-ядра в условиях различных требований к производительности и величине аппаратных ресурсов. Описывается целесообразность последовательной генерации подключей для дешифрования.

Ключевые слова: *алгоритм IDEA, ПЛИС, FPGA, масштабируемая архитектура, параметризация.*

Введение

Постоянное увеличение объема конфиденциальной информации, появление новых методов и средств несанкционированного доступа к данным ведет к развитию отрасли защиты информации. К средствам криптографической защиты относятся аппаратные, программно-аппаратные и программные реализации криптографических алгоритмов преобразования данных, однако, именно аппаратная реализация является более надежной и быстрой.

Нестандартные аппаратные решения реализации криптографических алгоритмов могут быть получены с использованием программируемых логических интегральных схем (ПЛИС) [1], позволяющих проектировать цифровые устройства с использованием высокоуровневых языков описания аппаратуры, что снижает трудоёмкость разработки и допускает повторное использование кода за счёт применения IP-ядер[2].

Кроме того, одним из важных преимуществ реализации криптографических алгоритмов на ПЛИС является возможность построения параллельной и асинхронной архитектуры, превосходящей по производительности решения, основанные на микропроцессорах, GPU или CPU [3].

В то же время применение ПЛИС, включая их наиболее гибкий тип FPGA, для реализации таких алгоритмов ограничивается фиксированными параметрами имплементируемого проекта. Подобное ограничение является традиционным при поиске компромисса между универсальной и специализированной реализациями. Поэтому важно иметь средства, позволяющие расширить границы специализированных решений на ПЛИС (далее подразумевается FPGA) до унифицированных или универсальных вариантов.

Целью данной работы является исследование возможности построения масштабируемого проекта для криптографического алгоритма на ПЛИС, позволяющего учесть доступные аппаратные ресурсы. Для достижения цели в первой части статьи проводится обзор существующих криптографических решений на ПЛИС, а затем описывается подход к масштабированию. В качестве проекта, иллюстрирующего этот подход, выбран блочный алгоритм шифрования IDEA. В качестве инструмента для масштабирования используется механизм статической параметризации (СП). Параметризация – это процесс выбора параметров проекта, которые позволяют изменять его характеристики без перепроектирования [2]. СП – это параметризация без динамического изменения параметров.

1. Обзор существующих реализаций алгоритмов шифрования на ПЛИС

Наиболее часто реализуемыми алгоритмами шифрования на ПЛИС являются алгоритмы RSA, AES, DES, ГОСТ 28147-89. Основной целью большинства работ по исследованию реализации указанных алгоритмов является достижение эффективного использования ресурсов кристалла и максимально возможной скорости шифрования данных. В таблице 1 приведены результаты сравнения особенностей алгоритмов: длины входных данных, размера ключей, используемых математических операций. Асимметричный алгоритм шифрования RSA используется как для шифрования данных, так и для создания электронной цифровой подписи [4]. Безопасность этого алгоритма основана на трудности разложения на множители больших целых чисел. Он используется во многих криптографических приложениях, включая систему шифрования PGP, в кото-

рой также применяется рассматриваемый алгоритм IDEA [5]. Алгоритм RSA реализуется на ПЛИС чаще других ассиметричных алгоритмов шифрования в разных интерпретациях для различных целей [6]. Основной особенностью этих реализаций является использование алгоритма Монтгомери для снижения затрат аппаратных ресурсов и достижения требуемой производительности в FPGA реализации [7].

Симметричный алгоритм шифрования DES в США являлся национальным стандартом в период с 1977 по 1980 года. На данный момент он считается устаревшим, и чаще всего используется его более криптоустойчивый вид (3DES, DESX) [12]. DES может быть построен на ПЛИС в виде комбинационной логической схемы. К достоинствам такой реализации отнесем отсутствие дополнительного управляющего автомата, простоту интерфейса, безопасность ключа, хранимого в ПЛИС. К недостаткам – большую площадь на кристалле и отсутствие возможной конвейеризации [9].

AES является одним из самых известных симметричных блочных алгоритмов шифрования, принятых в качестве стандарта шифрования США [8]. Алгоритм прост для понимания и популярен благодаря компактности записи на языках программирования и высокой скорости шифрования [13].

Таблица 1

Анализ особенностей существующих реализаций алгоритмов шифрования на ПЛИС

Характеристики алгоритма	**Алгоритм**				
	AES (подстановочно-перестановочная сеть)	**DES (сеть Фейстеля)**	**ГОСТ 28147-89 (сеть Фейстеля)**	**IDEA (модификация сети Фейстеля)**	**RSA**
Тип шифрования	Симметричный, блочный				Асимметричный
Длина блока, бит	128	64	64	64	
Число раундов	10	16	32	8	
Размер ключа, бит	256	56	256	128	1024
Основные операции	Табличная замена байтов, циклический сдвиг, перемешивание столбцов, поразрядное сложение по модулю 2	Подстановка, перестановка, побитовое исключающее ИЛИ	Сложение по модулю 2^{32}, подстановка, побитовое исключающее ИЛИ, циклический сдвиг	Умножение по модулю $2^{16}+1$, сложение по модулю 2^{16}, побитовое исключающее ИЛИ	Операция возведения в степень по модулю большого числа
Анализ содержания публикаций, рассматривающих реализацию на ПЛИС	Обзор эффективной реализации алгоритма при параллельном расположении раундов и расположении шифрования и дешифрования на одном модуле. Анализ полученной частоты, скорости шифрования и количества занимаемых логических элементов [8].	Обзор аппаратной реализации алгоритма шифрования DES, целью которой было получение скорости шифрования не меньшей, чем у современных криптопроцессоров. Анализ преимуществ и недостатков данной реализации на ПЛИС [9].	Оценка производительности реализации алгоритма на GPU и FPGA. Сравнение предварительного прогнозирования производительности и экспериментально полученных результатов [10].	Исследование особенностей реализации шифрования для мобильной связи средствами DSP, FPGA и ASIC на примере IDEA. Сравнение полученных данных о скоростях шифрования и производительности. Описание преимуществ FPGA для таких реализаций [11].	Анализ работы алгоритма с различными длинами ключей на разных модулях с разным направлением вычислений и количеством сумматоров. Сравнение полученных скоростей шифрования, частот и количества логических элементов [6].
Кристалл ПЛИС	Xilinx Virtex 5	Xilinx Virtex – E	Altera Arria II GX EP2AGX125	Xilinx XC4000	Altera Cyclone IV EP4CE115F29C7
Частота	347,6 МГц	19,4 МГц	125 МГц	33 МГц	13,31 МГц
Скорость	44,5 Гб/с	4,25 Гб/с	748 Мб/с	528 Мб/с	12,81 Кб/с

Поскольку ПЛИС представляет собой гибкую и легко настраиваемую платформу для реализации встраиваемых защитных механизмов, алгоритм AES используется в системах на кристалле (SoC) для обеспечения безопасного функционирования. Ввиду того, что это алгоритм высокой вычислительной сложности, в работах, посвященных его реализации на ПЛИС, описываются различные программно-аппаратные решения для проектирования SoC [14].

Отечественный стандарт шифрования ГОСТ-28147-89 формировался с учетом мирового опыта, недостатков и нереализованных возможностей криптоалгоритма DES, развитием которого он является. Этот стандарт рекомендован к использованию для защиты любых данных, представленных в виде двоичных последовательностей [15].

В исследованиях по реализации алгоритма ГОСТ 28147-89 производилось сравнение производительности при использовании сопроцессоров GPU и сопроцессоров на базе ПЛИС. Преимуществом реализации на ПЛИС является возможность задания собственной аппаратной логики, что позволяет снизить количество операций за счёт параллельной табличной подстановки и отсутствия обращений к памяти во время обработки блока [10].

Алгоритм IDEA является международным алгоритмом шифрования данных, запатентованный швейцарской фирмой Ascom. Он представляет собой симметричный блочный алгоритм и известен тем, что применяется в пакете программ шифрования корпорации PGP [5].

Алгоритм IDEA выделен в отдельную группу, поскольку представляет наибольший интерес с точки зрения иллюстрации принципа масштабирования, ввиду наличия однотипных раундов, которые могут быть организованы последовательно или параллельно. Это позволяет управлять потребляемыми ресурсами за счёт изменения времени обработки одного блока.

2. Анализ алгоритма IDEA

Входными данными для алгоритма IDEA являются 128-битный ключ и 64-битные блоки, на которые разбивается весь исходный текст. Каждый блок делится на 4 подблока по 16 бит, т.к. все операции выполняются с 16-битными числами. Для шифрования и дешифрования используется один и тот же алгоритм, как и предполагает симметричное шифрование, но с разными подключами. Для шифрования данных исходный 128-битный ключ разбивается на 6 подключей по 16 бит для каждого раунда и 4 подключа для выходного преобразования.

Шифрование состоит из 8 раундов и выходного преобразования, представляющего собой укороченный раунд. В каждом раунде над 16-битными блоками и подключами осуществляются следующие операции: сложение по модулю 2^{16}, умножение по модулю $2^{16}+1$, побитовое исключающее ИЛИ. Особенностью алгоритма является несовместимость этих операций, т.е. никакие две из них не удовлетворяют как дистрибутивному закону, так и ассоциативному, что существенно затрудняет криптоанализ. Выбор этого алгоритма обусловлен наличием в нем простых алгебраических операций, легкостью для понимания, относительной простотой операций в виду итеративного повторения раундов и высокой оценкой одного из самых известных криптологов Брюса Шнайера, который характеризовал его как самый лучший и надежный блочный алгоритм [16].

3. Реализация масштабируемой архитектуры криптомодуля на ПЛИС

Специфика ПЛИС реализации обуславливает возможность решения большинства задач как последовательным, так и параллельным способом, что существенно влияет на требуемое количество ресурсов для реализации проекта. В общем случае, решение на ПЛИС предполагает реализацию многократно повторяющихся операций, зачастую в конвейерном исполнении. При этом некоторые вычисления нужны лишь для однократного расчета констант, что актуализирует их компактную реализацию. В других проектах эти же вычисления могут использоваться в качестве основных, что требует от них высокой производительности.

3.1. Принцип масштабирования

Учёт возможности масштабирования вычислительного модуля в зависимости от требований проекта, в котором он используется, позволяет создать универсальное IP-ядро. Масштабирование может быть реализовано за счёт предоставления разработчику возможности выбора параллельной или последовательной реализации отдельных операций в составе компонента. Использование параметризации может позволить настроить модуль в соответствии с требованиями проекта по производительности или по количеству ресурсов. Определение требуемых значений параметров может быть выполнено как экспериментально, так и на основании методики оценки ресурсов, учитывающей параметризацию.

3.2. Особенности параметризируемой реализации алгоритма IDEA

Алгоритм IDEA основан на полностью фиксированных величинах. К ним можно отнести разрядности обрабатываемого блока данных, принимаемого ключа и формируемых подключей. Кроме того, количество раундов всегда равно восьми, а число

16-битных подключей, получаемых сдвигом ровно на 25 разрядов исходного ключа, всегда равно 52. Полная детерминированность задачи обуславливает отсутствие необходимости параметризации вышеописанных величин, т.к. они фиксированы в рамках самого алгоритма. Однако, кроме функциональной параметризации, проект на ПЛИС допускает параметризацию способа реализации архитектуры. Наличие восьми однотипных раундов позволяет применить параметризацию структурного описания проекта на ПЛИС. Разработанная реализация алгоритма IDEA предполагает наивысшую скорость работы для случая асинхронной архитектуры. Ее отличительной чертой является возможность получения всего набора из 52 подключей без использования сдвигающего регистра и регистров данных, что достигается использованием прямой коммутации.

3.3. Разработка масштабируемой реализации алгоритма IDEA

Алгоритм шифрования IDEA не содержит зависимости по данным между обрабатываемыми блоками, а следовательно, допускает распараллеливание за счёт увеличения числа одновременно кодируемых или декодируемых блоков. Этот результат может быть достигнут за счёт параметризации числа параллельных каналов, являющихся отдельными криптомодулями. В таком случае производительность определяется исключительно количеством доступных аппаратных ресурсов выбранного кристалла или ограничениями проекта верхнего уровня, в состав которого входит IP-ядро криптомодуля.

Ограничение ресурсов выбранного корпуса ПЛИС или проекта, использующего IP-ядро, могут быть таковы, что даже одна асинхронная реализация не сможет быть выполнена. В этом случае возникает необходимость в перестройке проекта из асинхронного в синхронный с разным количеством асинхронных компонентов, что также может быть реализовано посредством параметризации архитектуры. Например, количество последовательно реализованных раундов может быть сокращено в 2 раза за счёт двукратного увеличения времени вычисления и введения тактирования. В таком случае результат формируется за два такта. Предельным случаем такой параметризации является реализация лишь одного компонента с функциональностью раунда и вычисление одного результата за восемь тактов.

Также возможна параметризация арифметических операций умножения и сложения по модулю для экономии ресурсов за счёт использования последовательных вычислений вместо параллельных. Это важно, если параметризируемое IP-ядро входит в состав проекта, основная функциональность которого не связана с шифрованием и существует необходимость реализации алгоритма без требований к производительности в условиях ограниченных аппаратных ресурсов. Например, расчёт подключей для дешифрования достаточно выполнить единожды для заданного ключа. Для их нахождения используются операции мультипликативной и аддитивной инверсии, которые в параллельном исполнении требуют колоссальных аппаратных ресурсов. В связи с этим возникает необходимость создания последовательной синхронной реализации указанных операций, что многократно снизит аппаратные затраты за счёт увеличения длительности этих вычислений.

Выводы

Преимущества ПЛИС и развитие гибридных технологий на их основе позволяет расширять спектр возможных решений для реализации криптоалгоритмов. Дополнительные возможности предоставляются при использовании масштабируемых IP-ядер, которые позволяют варьировать параметры алгоритмов. Это подтверждается разработанным вариантом масштабируемого решения алгоритма шифрования IDEA. Масштабирование предлагаемого варианта реализации алгоритма на ПЛИС достигается с помощью статической параметризации количества параллельных и последовательных конструкций на всех уровнях декомпозиции проекта.

Литература

1. Куликова, А. С. Реализация многоверсионного поточного криптопреобразования данных с использованием бесключевых хеш-функций на программируемой логике [Текст] / А. С. Куликова, И. В. Лысенко // Системи обробки інформації. –2012. – № 7 (105). – С. 22 - 26.

2. Kulanov, V. Parameterized IP Infrastructures for Fault-Tolerant FPGA-Based Systems: Development, Assessment, Case-Study [Text] / V. Kulanov, V. Kharchenko, A. Perepelitsyn // Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2019), 2009. – P. 322–325.

3. Perepelitsyn, A. FPGA Technologies in Medical Equipment: Electrical Impedance Tomography [Text] / A. Perepelitsyn, D. Shulga // Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2012), 2012. – P. 437-440.

4. Качанов, А. П. Анализ алгоритмов шифрования данных с открытым ключом [Текст] / А. П. Качанов, А. С. Дацько // Автоматика та приладобудування. Вестник НТУ "ХПИ". – 2011. – № 11. – С. 56-61.

5. Макаренко, С. Алгоритмы шифрования для передачи данных в открытых сетях [Текст] / С. Макаренко, А. Брусникин // Прав., нормат. та метрол. забезп. системи захисту інформації в Україні. – 2001. – Вип. 2. – С. 223 – 229.

6. Škobić, V. Hardware Modules of the RSA Algorithm [Text] / V. Škobić, B. Dokić, Ž. Ivanović // Serbian Journal of Electrical Engineering. – February 2014. – Vol. 11, No. 1. – P. 121 – 131.

7. FPGA Implementation of RSA [Text] / R. Ghayoula, E. Hajlaoui, T. Korkobi, M. Traii, H. Trabelsi / World Academy of Science, Engineering and Technology. – 2008. – Vol. 2, No. 8. – P. 848 – 852.

8. Gurmail, S. High Throughput AES Encryption Algorithm Implementation on FPGA [Text] / S. Gurmail, M. Rajesh // International Journal of Computer Technology and Applications. – 2011. – P. 1993-1996.

9. Уваров, Н. С. Аппаратная реализация алгоритма шифрования DES на базе FPGA [Текст] / Н. С. Уваров // Моделирование, компьютерное проектирование и технологии производства электронных средств. – Минск : БГУИР, 2013. – С. 18.

10. Андреев, А. Е. Прогнозирование производительности при реализации алгоритмов на гибридных архитектурах с сопроцессорами [Электронный ресурс] / А. Е. Андреев, И. М. Силкин, Ю. В. Шафран // Режим доступа: http://www.science-education.ru. – 17.03.2014.

11. Mencer, O. Hardware Software Tri-Design of Encryption for Mobile Communication Units [Text] / O. Mencer, M. Morf, M. Flynn // Proc. Int'l Conference on Acoustics, Speech, and Signal Processing (ICASSP), IEEE '98. – May 1998. – P. 3045-3048.

12. Баркалов, А. А. Реализация алгоритма шифрования DES на базе FPGA [Текст] / А. А. Баркалов, А. А. Красичков, В. О. Кузьменко // Наукові праці ДонНТУ. – 2009. – Вип. 147. – С. 116-120.

13. Смит, Р. Разгадка тайны AES [Текст] / Р. Смит // LAN magazine / журнал сетевых решений. – 2001. – Том. 7, № 5. – С. 110-115.

14. Comparison of FPGA-based Implementation Alternatives for Complex Algorithms in Networked Embedded Systems - the Encryption Example [Text] / E. Heinrich, S. Staamann, R. Joost, R. Salomon // Emerging Technologies and Factory Automation (ETFA 2008). – 2008. – P. 1449-1456.

15. Сударев, И. В. Криптографическая защита телефонных сообщений [Текст] / И. В. Сударев // Специальная техника. – 1998. – № 2. – С. 47 – 55.

16. Шнайер, Б. Прикладная криптология [Текст] / Б. Шнайер. – М. : Триумф, 2002. – 374 с.

Поступила в редакцию 17.03.2014, рассмотрена на редколлегии 25.03.2014

Рецензент: д-р техн. наук, проф. В. И. Хаханов, Харьковский национальный университет радиоэлектроники, Харьков, Украина.

МАСШТАБОВАНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ШИФРУВАННЯ НА ПЛІС: АЛГОРИТМ IDEA

Є. В. Брошеван, А. Є. Перепелицин, В. С. Харченко

Наводяться результати огляду ПЛІС реалізацій алгоритмів шифрування даних. Пропонується загальний підхід до реалізації масштабованих проектів ПЛІС (FPGA). Аналізуються особливості алгоритму IDEA в контексті завдання масштабування. Пропонується варіант його масштабованої реалізації на ПЛІС з використанням статичної параметризації кількості паралельних і послідовних конструкцій на всіх рівнях декомпозиції проекту. Розглядається приклад використання IP-ядра в умовах різних вимог до продуктивності і розміру апаратних ресурсів. Описується доцільність послідовної генерації підключів для дешифрування.

Ключові слова: алгоритм IDEA, ПЛІС, FPGA, масштабована архітектура, параметризація.

FPGA-BASED SCALABLE IMPLEMENTATION OF ENCRYPTION ALGORITHMS: IDEA ALGORITHM

E. V. Broshevan, A. E. Perepelitsyn, V. S. Kharchenko

The overview results of the FPGA implementation of encryption algorithms are presented. A general approach to the FPGA-based scalable implementation is offered. The features of the IDEA algorithm in the context of scalable realizations are analyzed. The method of its FPGA-based scalable implementation using static parameterization of the number of parallel and sequential structures at all levels of decomposition of the project is suggested. An example of the IP-core usage in terms of different performance requirements and the amount of hardware resources is described. The practicability of subkeys generation for decoding in sequential manner is described.

Key words: IDEA algorithm, PLD, FPGA, scalable architecture, parameterization.

Брошеван Евгения Викторовна – студент кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ», Харьков, Украина, e-mail: evgeniya.broshevan@live.ru.

Перепелицын Артём Евгеньевич – аспирант, ассистент кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ», Харьков, Украина, e-mail: a.perepelitsyn@csn.khai.edu.

Харченко Вячеслав Сергеевич – д-р техн. наук, профессор, зав. каф. компьютерных систем и сетей Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ», Харьков, Украина, e-mail: v_s_kharchenko@ukr.net.