19. *Kneser R., Ney H.* Improved backing-off for m-gram language modeling, *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. I, Detroit, Michigan: 1995, pp. 181-184.
20. *Chen S.F., Goodman J.* An empirical study of smoothing techniques for language modeling, *Computer Science Group, Harvard University, Cambridge, Massachusetts, TR-8-98, August, 1998*.

Статью рекомендовал к опубликованию д.т.н., профессор И.И. Левин.

**Белозеров Андрей Александрович** – ФГУП «НТЦ «Орион»; e-mail: melnikov@linfotech.ru; 127018, г. Москва, ул. Образцова, д. 38, стр. 1; сотрудник.

**Вахлаков Дмитрий Владимирович** – сотрудник.

**Пересыпкин Владимир Анатольевич** – научный консультант; к.т.н.

**Мельников Сергей Юрьевич** – ООО «Лингвистические и информационные технологии»; зам. директора; к.ф.-м.н.

**Сидоров Евгений Сергеевич** – сотрудник.

**Belozerov Andrey Alexandrovich** – FGUP "NTC "Orion""; e-mail: melnikov@linfotech.ru; 38, Obraztsova street, build. 1, Moscow, 127018, Russia; worker.

**Vakhlakov Dmitriy Vladimirovich** – worker.

**Peresypkin Vladimir Anatol'evich** – research consultant; cand. of eng. sc.

**Melnikov Sergey Yur'evich** – OOO "Lingvisticheskie I informatsionye tehnologii" (Limited Liability Company); deputy Director; cand. of phys.-math. sc.

**Sidorov Evgeniy Sergeevich** – worker.

УДК 004.056.55 DOI 10.18522/2311-3103-2016-12-4254

**Л.К. Бабенко, Д.В. Голотин, О.Б. Макаревич**

## СОЗДАНИЕ И ИССЛЕДОВАНИЕ МАЛОРЕСУРСНОЙ РЕАЛИЗАЦИИ ПОТОЧНОГО ШИФРА TRIVIUM*

*В современном мире Интернет вещей особое место занимает легковесная криптография, наиболее эффективно обеспечивающая безопасность сетевых структур различного типа. Одной из проблем легковесной криптографии является оптимизация аппаратных решений с целью повышения эффективности их использования. Легковесная криптография – новое малоисследованное направление. Предметом данной статьи является поточный шифр Trivium. Данный шифр является финалистом проекта eSTREAM, по профилю 2 (поточные шифры для аппаратной реализации). Целью работы является реализация и исследование аппаратной модели легковесного шифра Trivium, в сравнении с другими его реализациями. Результаты работы могут быть использованы для решения задач по защите информации в условиях ограниченных ресурсов. Шифр Trivium является одним из самых эффективных в плане соотношения быстродействия, надёжности и легковесности. Для определения характеристик представленной аппаратной реализации шифра, на разработанной модели проведен анализ экспериментальных данных. Устройство шифрования реализовано на плате Марсоход2bis, на основе ПЛИС cyclone EP4CE6E22C8 с 6272 элементами. Частота внутреннего генератора микросхемы 100 МГц. Шифрующее устройство обрабатывает данные с компьютера, посылаемые по виртуальному com-порту, реализованному через USB. Передача данных из компьютера на ПЛИС и обратно осуществляет программа-клиент Serial Com port v1.2. В статье приводится схема устройства и его бло-*

* Работа поддержана грантом РФФИ № 15-07-00595.

*ков. Приведены результаты экспериментальных исследований созданного устройства, и оценка его эффективности, сравнение с другими существующими реализациями. В заключении подведён итог проделанной работы, представлены теоретические и практические результаты, представлен прогноз возможного повышения эффективности.*

*FPGA; ПЛИС; Verilog; интернет вещей; Quartus II; Trivium; аппаратная реализация; легковесная криптография; com-порт.*

**L.K. Babenko, D.V. Golotin, O.B. Makarevich**

## DEVELOPMENT AND STUDY OF SHORT-LIFE REALIZATION OF THE TRIVIUM STREAM CIPHER

*In today's world of the Internet of Things a special place takes the lightweight cryptography, which provides the most effective security networks of various types. One of the challenges is to optimize the lightweight cryptography hardware solutions to improve the efficiency of their use. Lightweight cryptography is a new unexplored direction. The subject of this article is a stream cipher Trivium. This cipher is a finalist of the project eSTREAM, the profile 2 (stream ciphers for hardware implementation). The target is a study and implementation of the hardware model lightweight cipher Trivium, in comparison with the other implementations. The results can be used in solving the problems of data protection in resource-limited settings. Cipher Trivium is one of the most efficient in terms of performance ratio, reliability and lightness. To determine the characteristics of the represented hardware implementation of the cipher carried out is the analysis of experimental data on the developed model. Device encryption has been implemented on board Marsohod2bis, FPGA cyclone EP4CE6E22C8 c 6272 elements. The frequency of the internal oscillator circuit is 100 MHz. Encrypting device processes the data from the computer, sent over the virtual com-port, realized via USB. The transfer of data from the computer to the FPGA and back performs client program Serial Com port v1.2. The article provides a diagram of the device and its units. Given are the results of experimental studies of the created device and its effectiveness assessment compared with other existing implementations. In conclusion, the conducted work is summed up, theoretical and practical results, a forecast of possible efficiency gains are presented.*

*The FPGA; PLD; the Verilog; Internet of things; Quartus II; Trivium; hardware implementation; lightweight cryptography; com-port.*

**Введение.** В настоящее время бурно развивается направление интегрирования различных устройств через сеть. Не имеет особого значения, какой канал связи используют данные устройства, так как любой информационный обмен по этим каналам приводит к возможности перехвата данных [1, 2]. Живым примером необходимости пересмотра защитных мер может служить новость о взломе огромного числа принтеров всего одним злоумышленником [3].

Одним из наиболее эффективных методов защиты информации является криптография. Благодаря ей можно защитить информацию, передавая по каналам связи в зашифрованном виде, а расшифровывать её уже непосредственно при получении. Но тогда появляется необходимость в том, чтобы каждый узел сети обладал криптографическими средствами, необходимыми для обмена. Данное требование сильно отражается на расходах компании, а также сказывается на быстродействии обмена информацией. Поэтому необходимо вводить требования к криптографическим устройствам, связанные со стоимостью и быстродействием. Появляется необходимость оптимизации этих параметров при построении криптографических систем информационной безопасности.

Особое влияние на обеспечение информационной безопасности оказывает переход к интернету вещей, когда каждое устройство будет интегрироваться в глобальную сеть, что многократно увеличит объём информации, подлежащей защите. В статье рассмотрена аппаратная реализация алгоритма легковесного шифрования [4–6] Trivium [7–11] и сравнение эффективности полученного устройства с другими существующими его реализациями [12–14].

**Постановка задачи.** Целью работы является создание и исследование аппаратной модели шифра Trivium. Данная статья рассматривает процесс создания модели шифра Trivium, и даёт сравнительный анализ полученной аппаратной модели с другими реализациями шифра Trivium, результаты которых опубликованы в статьях “A low-cost implementation of Trivium”[12] и “ Lightweight Cryptography From An Engineers Perspective ”[13].

**Описание алгоритма**. Шифр Trivium является проектом eSTREAM. Шифр реализован на трёх сдвиговых регистрах с комбинацией линейных и не линейных обратных связей. Длина регистров первого – 93 бита, второго – 84 бита и третьего – 111 бит. Работа алгоритма разбивается на три этапа:

1. Загрузка ключа и инициализирующего вектора в регистры.
2. Инициализация начального состояния регистров.
3. Генерация выходного потока и шифрования на нём отрытого теста.

Все этапы выполняются последовательно, и переход на следующий осуществляется только после завершения предыдущего.

Объём выходного потока, полученного при работе данного шифра, может достигать $2^{64}$ бит [7].

Шифр Trivium является аппаратно-ориентированным (он стал финалистом отбора eSTREAM [7], по профилю 2 – аппаратно-ориентированные шифры), тем не менее имеет достаточно хорошие результаты при программной реализации.

**Загрузка инициализирующей информации**. Для работы алгоритма используются два источника ключевой информации – секретный ключ и инициализирующий вектор [7-10]. Длина каждого составляет 80 бит. Ключ загружается в первый регистр, длина которого составляет 93 бита. Так как длина ключа составляет 80 бит, остальное 13 бит заполняется нулями. Инициализирующий вектор загружается во второй регистр, длина которого - 84 бита и аналогично первому регистру неиспользуемые биты заполняются нулями. Третий регистр, длиной 111 бит, заполняется нулями, кроме последних трёх бит, которые заполняются единицами. Данные единицы оказывают существенное влияние на генерируемую псевдослучайную последовательность.

**Инициализация.** После загрузки ключевой информации Trivium происходит инициализация. На данном этапе алгоритм выполняется 288*4 раз, без генерации выходного потока. На рис. 1 приведено в виде псевдокода описание работы алгоритма

for $i = 1$ to $4 \cdot 288$ do

$t_1 \leftarrow s_{66} + s_{91} \cdot s_{92} + s_{93} + s_{171}$

$t_2 \leftarrow s_{162} + s_{175} \cdot s_{176} + s_{177} + s_{264}$

$t_3 \leftarrow s_{243} + s_{286} \cdot s_{287} + s_{288} + s_{69}$

$(s_1, s_2, \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$

$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$

$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$

end for

*Рис. 1. Инициализация работы алгоритма*

На данном рисунке $t_1$, $t_2$, $t_3$ – биты, формирующие сдвиг внутри битов регистра. Для формирования этих битов используются биты всех трёх регистров. Само формирование происходит при помощи операций «·» и «+» (операции «И» и исключающего «ИЛИ»).

$S_i$ – биты регистров. На рис. 1 биты регистров объединены в один регистр. Можно выделить их по отдельности – первый регистр ($S_1$, $S_2$, $S_3$ … $S_{93}$), второй регистр ($S_{94}$…$S_{177}$) и третий регистр ($S_{178}$… $S_{288}$).

Инициализация является важным этапом шифрования, так как от неё зависит стойкость к различного рода атакам полученного далее выходного потока. Без инициализации значительно проще получить ключевую информацию, а сам алгоритм становится уязвимым [7].

**Генерация выходного потока**. Генерация выходного потока в Trivium происходит побитно, но так как сдвиговые биты, полученные при помощи комбинации операции XOR и логического умножения, то за один машинный такт можно вычислить более одного бита.

На рис. 2 представлен псевдокод функции генерации выходного потока. Битами, участвующими в работе алгоритма являются 66, 69, 91, 92, 93, 162, 171, 175, 176, 177, 243, 286, 287, 288. Сдвиговыми битами будут 1, 94, 178. Несложно вычислить расстояние между наиболее близкими битами, сгенерированными на данном цикле, и битами, которые будут участвовать в работе данного алгоритма на следующем цикле. Для первого регистра это расстояние между битом 1 и битом 66, что составляет 65 бит, для второго регистра это расстояние между 94 и 162 битом, что составляет 68 бит, и для третьего это расстояние между 178 и 243 битом, что составляет 65 бит. Отсюда следует что максимальное количество генерируемых бит за один такт составляет 64 бита.

for $i$ = 1 to N do

$t_1 \leftarrow s_{66} + s_{93}$
$t_2 \leftarrow s_{162} + s_{177}$
$t_3 \leftarrow s_{243} + s_{288}$

$z_i \leftarrow t_1 + t_2 + t_3$

$t_1 \leftarrow s_{66} + s_{91} \cdot s_{92} + s_{93} + s_{171}$
$t_2 \leftarrow s_{162} + s_{175} \cdot s_{176} + s_{177} + s_{264}$
$t_3 \leftarrow s_{243} + s_{286} \cdot s_{287} + s_{288} + s_{69}$

$(s_1, s_2, \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$
$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$
$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$

end for

*Рис. 2. Генерация выходного потока*

Стоит отметить то, что генерация и инициализация имеют почти одинаковый набор операций, и отличаются лишь тем, что при инициализации не генерируется выходной поток. Данное сходство можно использовать для уменьшения количества используемых элементов при реализации на ПЛИС.

**РЕАЛИЗАЦИЯ АППАРАТНОЙ МОДЕЛИ TRIVIUM.**

**1. Выбор средств.**

Средствами для создания аппаратной модели были выбраны программируемая логическая интегральная схема (ПЛИС), использована плата Марсоход2bis [15], реализованная на основе ПЛИС фирмы Altera Ccyclone 4 EP4CE6E22C8, с 6272 логическими элементами.

Реализация алгоритма Trivium на ПЛИС выполнена на программном комплексе Quartus II с использованием языка описания аппаратуры Verilog.

**2. Описание устройства.**

Данное устройство должно принимать данные с компьютера, шифровать их и полученный результат возвращать обратно на компьютер.

Структурная схема устройства приведена на рис. 3.

Наличие блоков приёма и передачи обусловлено необходимостью синхронизации шифрующего устройства и компьютера, а также приведения выходных данных в вид воспринимаемый компьютером.

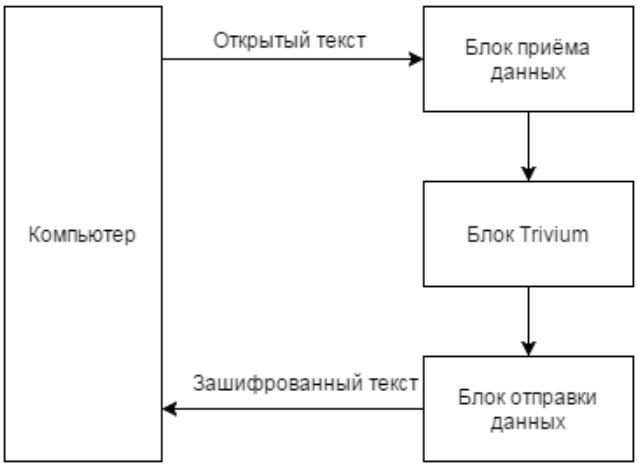


*Рис. 3. Структурная схема аппаратной модели шифра*

**3. Приём данных.**

Для обмена данными с компьютером был выбран виртуальный com-порт, реализованный через USB, потому что на плате Марсоход2bis располагается микросхема ft2232h, реализующая com-порт. Работа данного интерфейса заключается в следующем. При передаче по com-порту сначала посылается логический 0, затем байт информации, и затем com порт равный 1. Если передача данных не осуществляется линия находится в 1. Длина стоп бита может быть 1, 1.5, 2 бита, дополнительно может присутствовать бит чётности [16, 17]. Блок приёма разрабатываемой модели ориентирован на 1 стоп бит и с отсутствующим битом чётности. На рис. 4 приведена функциональная схема блока.

На рис. 4 есть параметр k. Этот параметр является делителем частоты. Так как частота генератора ПЛИС намного выше частоты com-порта возникает необходимость в делители частоты.

**4. Шифрование данных.**

Шифрование происходит по алгоритму Trivium [7, 11, 18]. Особенностью в его реализации является то, что он должен быть синхронизирован с блоком передачи информации и её приёма. Для связи с приёмником существует сигнал st_rc, который присутствует на рисунке 4, сигнализирующий о приёме одного байта. Помимо этого, в этом блоке, также как и в приёмнике, существует делитель частоты.

Работу алгоритма можно разбить на три части: загрузка ключевой информации, инициализация, генерация последовательности, используемой для шифрования. Блок-схема этапа инициализации представлена на рис. 5.

Загрузка ключевой информации происходит побайтно, с синхронизацией с блоком приёмника. Когда загрузка ключевой информации закончена, происходит процесс инициализации, с частотой кварцевого генератора (установленного на плате Марсоход2bis) 100 МГц.

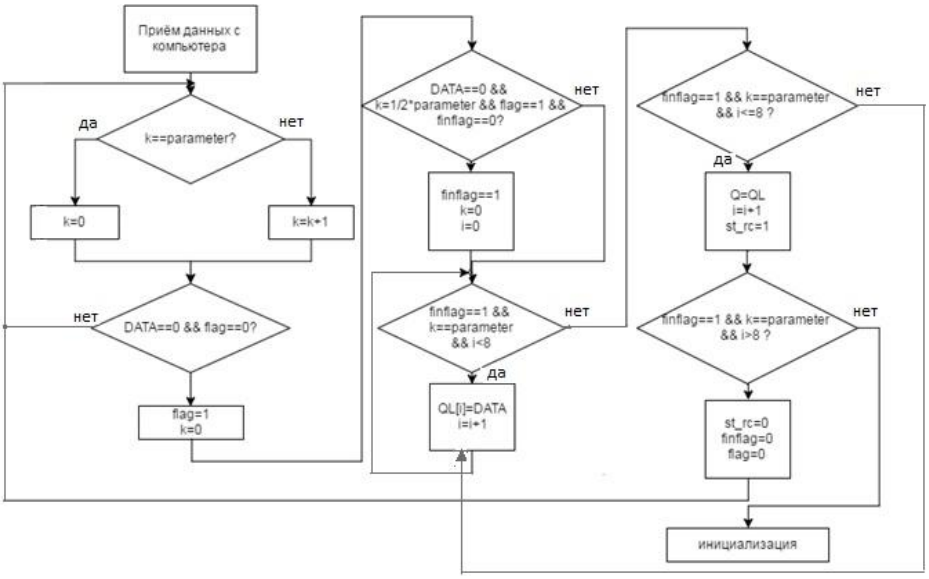


*Рис. 4. Блок-схема приёма информации*

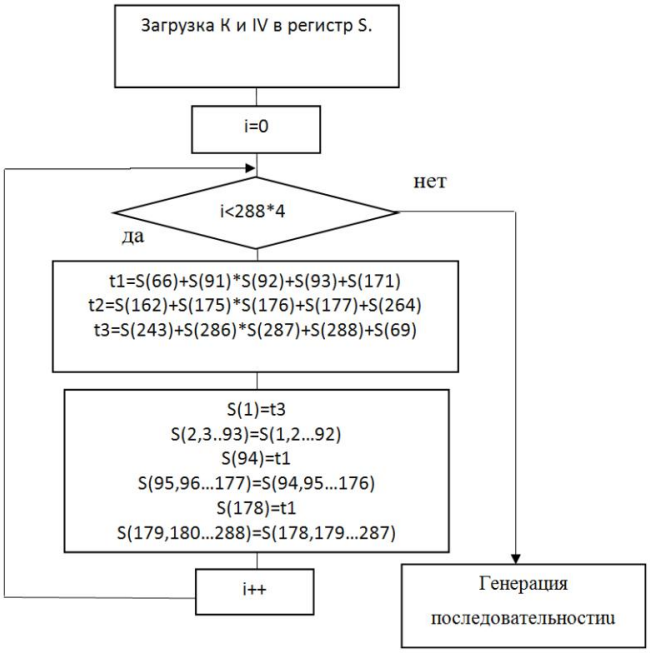


*Рис. 5. Блок-схема инициализации*

После окончания процесса инициализации происходит процесс шифрования полученных данных.

**5. Передача данных компьютеру.**

Данный блок отвечает за синхронизацию с компьютером при отправке данных. Как уже было сказано, приём и передача осуществляется с подачи старт бита равного 0, а оканчивается подачей стоп бита равного 1 [17]. А также передача ведётся с фиксированной скоростью, в данном случае 9600 бит/с. Для достижения данной скорости используется всё тот же делитель частоты, реализованный на счётчике.

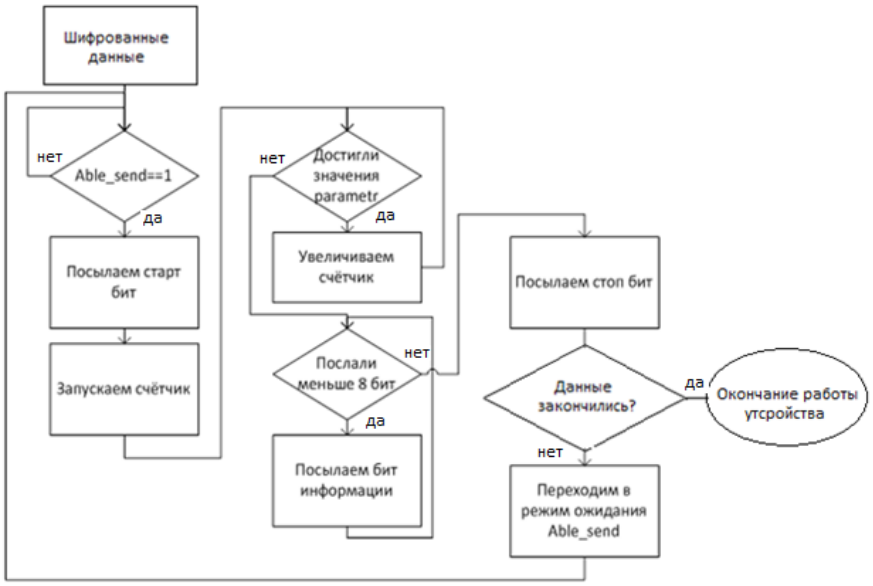


*Рис. 6. Блок-схема модуля, отвечающего за отправление зашифрованных данных*

Данный блок синхронизирован с шифрующим блоком при помощи специализированного сигнала able_send. На рис. 6 приведена блок-схема данного блока.

**6. Реализация аппаратной модели.**

При реализации данных блоков на плис был использован программный комплекс Quartus II. Данный комплекс реализован для работы с ПЛИС фирмы Altera. Комплекс был дополнен инструментами моделирования MedelSim [19] для выявления различных ошибок в процессе проектирования. На рис. 7 приведён скриншот отладки проекта блока шифрования в среде ModelSim.

На рис. 7 показаны сигналы

1. Clk – сигнал, генератора, с частотой 100 МГц. Играет роль синхронизатора.
2. DATA – сигнал эмулирующий данные, посылаемые с блока-приёмника.
3. Q – сигнал, эмулирующий данные, посылаемые на блок передатчика.
4. QTO – переменная, хранящая значение сигнала Q, до пересылки.
5. T1, t2, t3, s1, s2, s3 – переменные, хранящие значения сдвиговых битов. T1, t2, t3 используются при инициализации, а s1, s2, s3 при генерации выходного потока. Различие состоит в том, что при инициализации для достижения максимальной скорости используются 64 потока, а для генерации 8, так как для шифрования передаётся за один такт только 8 бит.
6. Headreg – регистр, который представляет собой три регистра в Trivium.
7. Control – переменная, используемая для определения в каком режиме работает алгоритм.

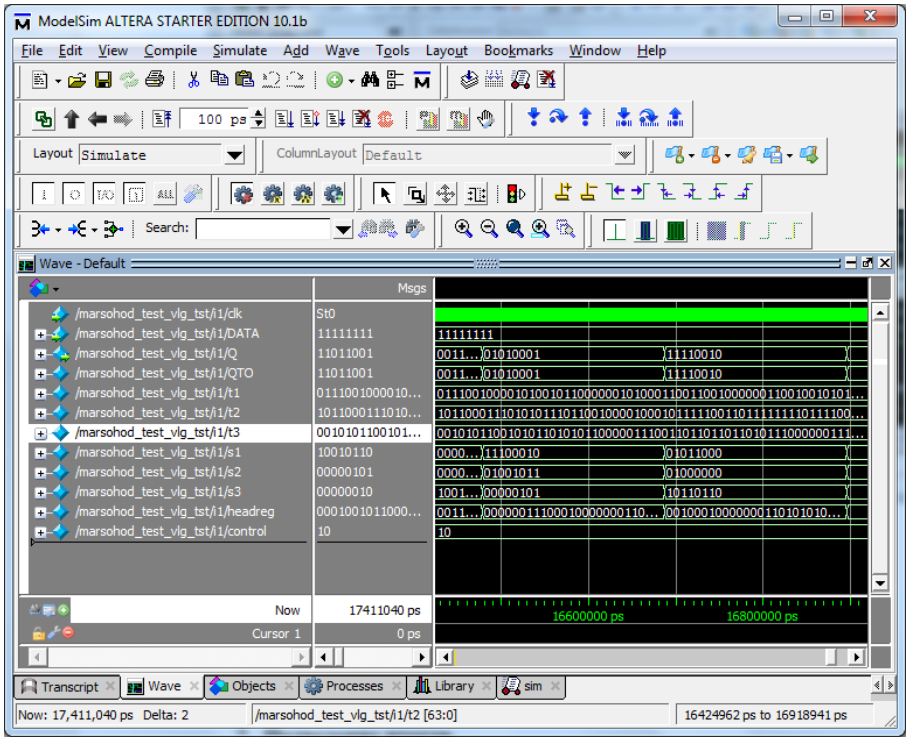


*Рис. 7. Моделирование блока Trivium в среде ModelSim*

После успешного моделирования в среде ModelSim, проект был перенесён на ПЛИС. Для обмена данными [16] с прошитой микросхемой использовалась программа-терминал Serial Port Terminal v1.2. Обмен данными был представлен в виде обмена шестнадцатеричными числами, пара таких чисел эмитирует 1 байт. Для проверки устройства данные были зашифрованы и расшифрованы, что можно увидеть из сравнения рис. 8 и 9.

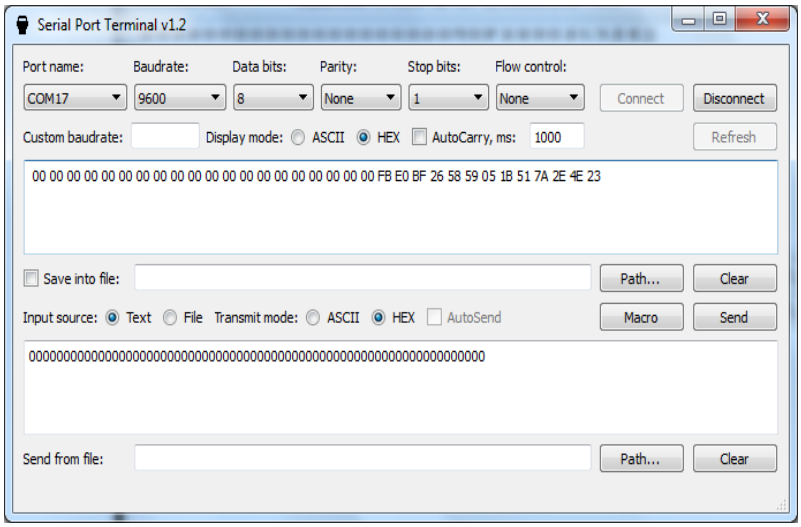


*Рис. 8. Шифрование данных*

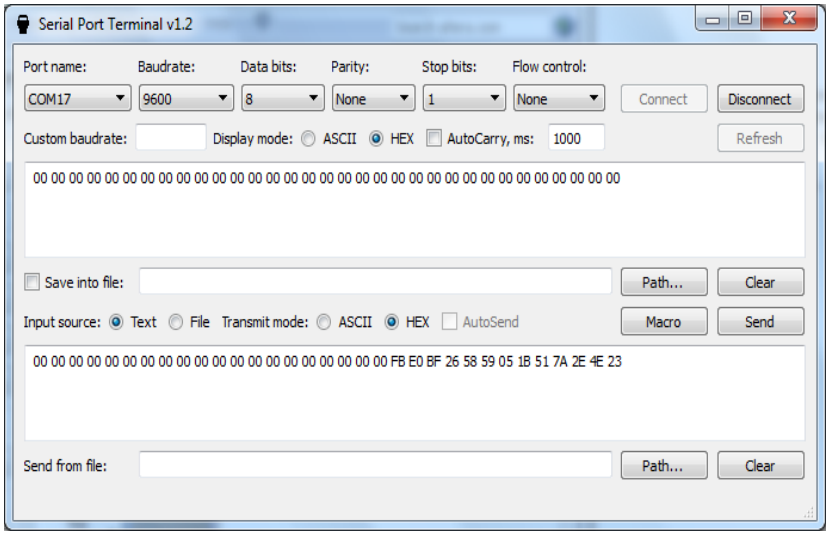


*Рис. 9. Дешифрование данных*

Как видно из рис. 8 и 9 первые 20 байт выводятся без шифрования. Это связано с тем, что вначале схема принимает данные, которые будут исполнять роль управляющего вектора и ключа, каждый из которых по 80 бит. 80 бит = 10 байт, и того 20 байт. Эти данные не шифруются, а выводятся на экран для удобства (легче понять, когда начинается шифрование, легче расшифровывать зашифрованные данные). В реальном устройстве такие данные выводить не стоит, так как в шифровании теряется смысл, если злоумышленник будет знать ключ, но данное устройство реализовано как модель шифра, а не для коммерческого использования.

Для установления того, что данный шифр работает в соответствии с эталоном, были высчитаны в Excel биты выходного потока и сравнены с битами, полученными при работе шифра, реализованного на ПЛИС. Была проведена проверка на 8 байт информации, и данные совпали при сравнении (особенностью является то, что при вычислении в Excel первым шёл младший бит байта, а последним шёл старший бит байта, что связано с тем, что вычисления были побитовыми, а в устройстве ПЛИС всё было наоборот, так как там вычисления были байтовыми, и первым шёл старший бит). На рис. 10 приведён скриншот Excel, на котором видно два первых байта.

Данные байты равны 11111011 и 1110000, если их представить в правильном виде. Данные байты равны FB и E0, полученные при работе шифрующего устройства.

**Сравнение аппаратной модели с существующими аналогами.** Разработанная аппаратная модель является эффективной по причине малого использования ресурсов, так как при дополнительных затратах элементов для организации последовательного порта, размер всего проекта составляет 1712 элементов и работает на 8 потоках, при условии того, что создателями данного шифра было подсчитано, что для реализации данного шифра с 1 выходным потоком потребуется 3488 элементов, а для 64 потоков потребуется 5504 логических элемента [9, 20].

Помимо этого, основываясь на информации о результатах реализации Trivium из статьи Good T., Benaissa M. [12], где перечисляются результаты по нескольким шифрам, можно так же сделать вывод об эффективности шифра, так как количество элементов для реализации данного шифра на 8-ми потоках 2801 элемент.

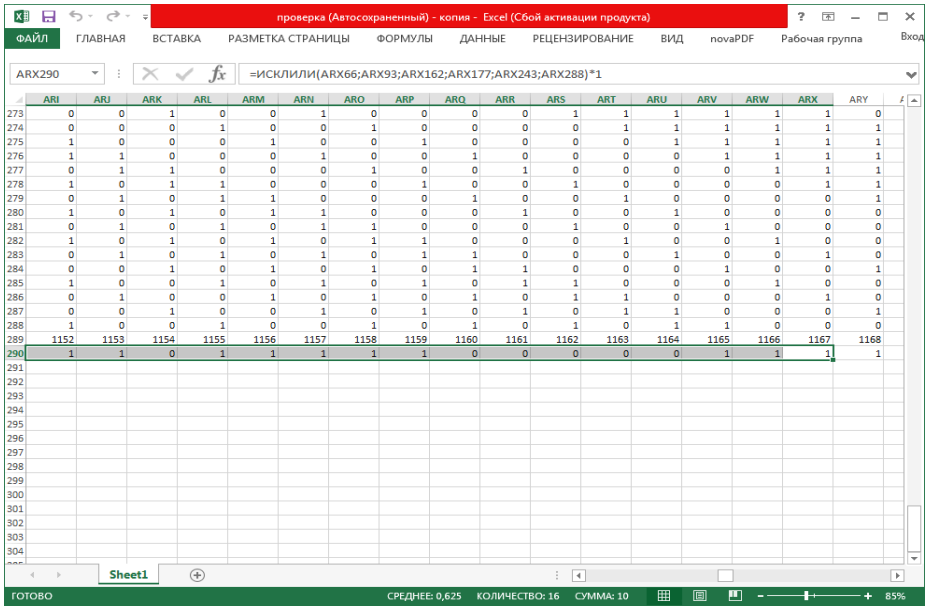


*Рис. 10. Вычисление битов выходного потока в Excel*

Так же проведено сравнение со стандартной реализацией данного шифра, результаты о которой взяты из статьи “A low-cost implementation of Trivium” [13] под авторством Mentens N., Genoe J., Preneel B., Verbauwhede. По результатам этого сравнения можно отметить высокую эффективность данного устройства, так как количество элементов, используемых в аналоге составляет 2017 элементов.

**Заключение.** Данная аппаратная реализация заняла на чипе 1712 элементов, что является, по результатам сравнения с аналогом, хорошим результатом в плане экономии места на чипе.

Скорость данной реализации ограничена скоростью com порта 9600бит/с (1200бай/с). Если учесть, то что инициализация работала на частоте кварцевого генератора с использованием 64 потоков, можно сделать вывод что при другом методе передачи информации на шифрование по алгоритму Trivium, скорость данного блока может достигать 763 Мбайт/с (данная скорость равна скоростям других реализаций, использующих 64 потока [13, 18, 20]), из чего можно сделать вывод, о том, что быстродействие данной реализации ограничено скоростью обмена информацией с компьютером через com-порт и для повышения быстродействия следует использовать более быстрые интерфейсы передачи информации (передача по HDMI, USB 3.0, Gigabit Ethernet).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Жуков А.Е.* Легковесная криптография. Ч. 1 // Вопросы кибербезопасности. – 2015. – № 1 (9). – С. 18.
2. *Жуков А.Е.* Легковесная криптография. Ч. 2 // Вопросы кибербезопасности. – 2015. – № 2 (10). – С. 10.
3. История о том, как один злоумышленник заставил тысячи принтеров по всему миру печатать листовки со свастикой // GeekTiemes коллективный блог. – URL: https://geektimes.ru/post/273536/ (дата обращения: 4.02.2016).
4. *Preneel B.* Perspectives on Lightweight Cryptography // Bart Pernel pesonal site. – URL: http://homes.esat.kuleuven.be/~preneel/preneel_lightweight_shanghaiv1.pdf (дата обращения 20.10.2015).

5. *Aoki K., Ichikawa T., Kanda M., Matsui M., Moriai S., Nakajima J., Tokita T.* Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms – Design and Analysis // Selected Areas in Cryptography (SAC), LNCS. – 2001. – Vol. 2012. – P. 39-56.
6. *Poschmann A., Leander G., Schramm K., Paar C.* New lighweight crypto algorithms for RFID // In Proceedings of The IEEE International Symposium on Circuits and Systems 2007 – ISCAS 2007. – 2007. – P. 1843-1846.
7. *De Canniere C. and Preneel B.* Trivium Specifications // eSTREAM: ECRYPT Stream Cipher Project Report 2005/030. – URL: http://www.ecrypt.eu.org/stream/. (дата обращения: 10.11.2015).
8. *De Canniere C. and Preneel B.* Trivium A Stream Cipher Construction Inspired by Block Cipher Design Principles // eSTREAM: ECRYPT Stream Cipher Project Report 2006/021 (2015). – URL: http://www.ecrypt.eu.org/stream/ (дата обращения: 13.12.2015).
9. eSTREAM portfolio // eSTREAM, the ECRYPT Stream Cipher Project. – 2015. – URL: http://www.ecrypt.eu.org/stream/ (дата обращения: 15.01.2016).
10. *Khazaei S., Hasanzadeh M.M., and Kiaei M.S.* Linear Sequential Circuit Ap-proximation of Grain and Trivium Stream Ciphers // eSTREAM, ECRYPT StreamCipher Project, Report 2005/063. – URL: http://www.ecrypt.eu.org/stream/papersdir/2007/008.pdf (дата обращения: 13.12.2015).
11. *Бабенко Л.К., Голотин Д.В.* Об особенностях функционирования и реализации поточного шифра Trivium // Известия ЮФУ. Технические науки. – 2015. – № 5 (166). – P. 103-111.
12. *Mora Gutiérrez, Jiménez Fernández, Valencia Barrero.* Low power implementation of Trivium stream cipher. Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation // 22nd International Workshop. – 2012. – P. 113-120.
13. *Good T., Benaissa* M. Hardware Results for selected Stream Cipher Candidates. State of the Art of Stream Ciphers 2007 // SASC: Workshop Record. – February 2007. – P. 120-128.
14. *Бабенко Л.К., Беспалов Д.А., Макаревич О.Б., Чесноков Р.А., Трубников Я.А.* Разработка и исследование программно-аппаратного комплекса шифрования по алгоритму Present для решения задач малоресурсной криптографии // Известия ЮФУ. Технические науки. – 2014. – № 2 (151). – P. 174-180.
15. Спецификация платы Marsohod2bis // Marsohod: open source hardware project. FPGA и CPLD блог. – 2016. Режим доступа: URL: https://marsohod.org/11-blog/289-marsohod2bis (дата обращения: 20.01.2016).
16. Интерфейсный модуль на FT2232D // Easy Electronics: блог посвященный цифровой электронике. Режим доступа: URL: http://easyelectronics.ru/interfejsnyj-modul-na-ft2232d.html (дата обращения: 11.10.2015).
17. UART и с чем его едят // Geektimes: коллективный блог. – 2016. Режим доступа: https://geektimes.ru/post/253786/.
18. Axel Poschmann: Lightweight Cryptography From An Engineers Perspective // Horst-Görtz-Institut für Sicherheit. – 2016. – P. 28-32.
19. Симулятор ModelSim // Marsohod: open source hardware project. Режим доступа: URL: http://marsohod.org/11-blog/118-modelsim (дата обращения: 11.12.2015).
20. *Nele Mentens, Jan Genoe, Bart Preneel, Ingrid Verbauwhede*. A low-cost implementation of Trivium // Preproceeding of SACS. – 2008. – P. 197-204.

REFERENCES

1. *Zhukov A.E.* Legkovesnaya kriptografiya [Lightweight cryptography]. Part 1, *Voprosy kiberbezopasnosti* [Cybersecurity], 2015, No. 1 (9), pp. 18.
2. *Zhukov A.E.* Legkovesnaya kriptografiya [Lightweight cryptography]. Part 2, *Voprosy kiberbezopasnosti* [Cybersecurity], 2015, No. 2 (10), pp. 10.
3. Istoriya o tom, kak odin zloumyshlennik zastavil tysyachi printerov po vsemu miru pechatat' listovki so svastikoy [The story of how one man made thousands of printers worldwide to print leaflets with a swastika], *GeekTiemes kollektivnyy blog* [GeekTiemes collective blog]. Available at: https://geektimes.ru/post/273536/ (accessed 4 February 2016).

4. *Preneel B.* Perspectives on Lightweight Cryptography, *Bart Pernel pesonal site*. Available at: http://homes.esat.kuleuven.be/~preneel/preneel_lightweight_shanghaiv1.pdf (accessed 20 October 2015).
5. *Aoki K., Ichikawa T., Kanda M., Matsui M., Moriai S., Nakajima J., Tokita T.* Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms – Design and Analysis, *Selected Areas in Cryptography (SAC), LNCS*, 2001, Vol. 2012, pp. 39-56.
6. *Poschmann A., Leander G., Schramm K., Paar C.* New lighweight crypto algorithms for RFID, *In Proceedings of The IEEE International Symposium on Circuits and Systems 2007 – ISCAS 2007*, 2007, pp. 1843-1846.
7. *De Canniere C. and Preneel B.* Trivium Specifications, *eSTREAM: ECRYPT Stream Cipher Project Report 2005/030*. Available at: http://www.ecrypt.eu.org/stream/ (accessed 10 November 2015).
8. *De Canniere C. and Preneel B.* Trivium A Stream Cipher Construction Inspired by Block Cipher Design Principles, *eSTREAM: ECRYPT Stream Cipher Project Report 2006/021 (2015).* Available at: http://www.ecrypt.eu.org/stream/ (accessed 13 December 2015).
9. eSTREAM portfolio, *eSTREAM, the ECRYPT Stream Cipher Project*, 2015. Available at: http://www.ecrypt.eu.org/stream/ (accessed 15 January 2016).
10. *Khazaei S., Hasanzadeh M.M., and Kiaei M.S.* Linear Sequential Circuit Ap-proximation of Grain and Trivium Stream Ciphers, *eSTREAM, ECRYPT StreamCipher Project, Report 2005/063*. Available at: http://www.ecrypt.eu.org/stream/papersdir/2007/008.pdf (accessed 13 December 2015).
11. *Babenko L.K., Golotin D.V.* Ob osobennostyakh funktsionirovaniya i realizatsii potochnogo shifra Trivium [The main features functioning and implementation stream cipher Trivium], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 5 (166), pp. 103-111.
12. *Mora Gutiérrez, Jiménez Fernández, Valencia Barrero.* Low power implementation of Trivium stream cipher. Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation, *22nd International Workshop*, 2012, pp. 113-120.
13. *Good T., Benaissa* M. Hardware Results for selected Stream Cipher Candidates. State of the Art of Stream Ciphers 2007, *SASC: Workshop Record*, February 2007, pp. 120-128.
14. *Babenko L.K., Bespalov D.A., Makarevich O.B., Chesnokov R.A., Trubnikov Ya.A.* Razrabotka i issledovanie programmno-apparatnogo kompleksa shifrovaniya po algoritmu Present dlya resheniya zadach maloresursnoy kriptografii [Software and hardware development and research of encryption algorithm present for solving problems of the lightweight cryptography], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 2 (151), pp. 174-180.
15. Spetsifikatsiya platy Marsohod2bis [Specification Board Marsohod2bis], *Marsohod: open source hardware project. FPGA i CPLD blog*, 2016. Available at: https://marsohod.org/11-blog/289-marsohod2bis (accessed 20 January 2016).
16. Interfeysnyy modul' na FT2232D [Interface module for FT2232D], *Easy Electronics: blog posvyashchennyy tsifrovoy elektronike*. Available at: http://easyelectronics.ru/interfejsnyj-modul-na-ft2232d.html (accessed 11 October 2015).
17. UART i s chem ego edyat [UART and with what it eat], *Geektimes: kollektivnyy blog* [Geektimes: a collective blog], 2016. Available at: https://geektimes.ru/post/253786/.
18. Axel Poschmann: Lightweight Cryptography From An Engineers Perspective, *Horst-Görtz-Institut für Sicherheit,* 2016, pp. 28-32.
19. Simulyator ModelSim [Simulator ModelSim], *Marsohod: open source hardware project*. Available at: http://marsohod.org/11-blog/118-modelsim (accessed 11 December 2015).
20. *Nele Mentens, Jan Genoe, Bart Preneel, Ingrid Verbauwhede*. A low-cost implementation of Trivium, *Preproceeding of SACS*, 2008, pp. 197-204.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

**Бабенко Людмила Климентьевна** – Южный федеральный университет; e-mail: blk@tsure.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634361518; кафедра безопасности информационных технологий; профессор.

**Макаревич Олег Борисович** – e-mail: mak@tsure.ru; кафедра безопасности информационных технологий; профессор.

**Голотин Денис Владимирович** – e-mail: ww.golotin@mail.ru; тел.: +79514944226; кафедра безопасности информационных технологий; студент.

**Babenko Lyudmila Klimentevna** – Southern Federal University; e-mail: blk@tsure.ru; 2, Chehov street, Taganrog, 347928, Russia; phone: +78634361518; the department of information technologies security; professor.

**Makarevich Oleg Borisovich** – e-mail: mak@tsure.ru; the department of information technologies security; professor.

**Golotin Denis Vladimirovich** – e-mail: ww.golotin@mail.ru; phone: +79514944226; the department of information technologies security; student.