НАУЧНОЕ ИЗДАНИЕ МГТУ ИМ. Н. Э. БАУМАНА

НАУКА и ОБРАЗОВАНИЕ

Эл № ФС77 - 48211. Государственная регистрация №0421200025. ISSN 1994-0408

электронный научно-технический журнал

Реализация хэш-функций, основанных на обобщенных клеточных автоматах, на базе ПЛИС: производительность и эффективность

 # 01, январь 2014

DOI: 10.7463/0114.0675812

Ключарёв П. Г.

УДК 004.056.55

Россия, МГТУ им. Н.Э. Баумана
pk.iu8@yandex.ru

Введение

В условиях повышения требований к защищенности информации, передаваемой по различным каналам связи и повышения скоростей передачи информации, большую важность приобретают высокоскоростные криптографические хэш-функции. В работе [2] было предложено семейство криптографических хэш-функций, основанных на обобщенных клеточных автоматах. Мы будем называть это семейство GRACE–H. В той же работе была высказана гипотеза о том, что такие хэш-функции могут быть весьма эффективно реализованы аппаратно, например, на программируемых логических интегральных схемах (ПЛИС). Целью настоящей статьи является подтверждение этой гипотезы путем тестирования производительности реализации хэш-функций семейства GRACE–H на ПЛИС.

1. Хэш-функции

Автором [1] разработано семейство псевдослучайных функций, вида

$$S_c\colon B^k \times B^n \to B^m.$$

Эти функции основываются на обобщенных клеточных автоматах и могут быть заданы формулой

$$S_c^A(key, x) = pr_m(F_A(x \parallel key \parallel c, r)), \tag{1}$$

где $x \parallel y$ — конкатенация x и y; $F_A(x, r)$ — заполнение на шаге r клеточного автомата A с начальным заполнением x; r — число шагов клеточного автомата; $pr_m\colon B^* \to B^m$ —

функция, возвращающая некоторые m элементов аргумента, индексы которых заданы наперед; A — обобщенный клеточный автомат; $c \in B^t$ — некоторая константа, вес которой близок к значению $\left[\frac{t}{2}\right]$; $t = N - (n+k)$; N — число вершин графа обобщенного клеточного автомата.

Как указано в работе [1], при правильном выборе параметров такие функции нельзя отличить от случайных с помощью статистических тестов.

Теперь опишем работу хэш-функции из семейства GRACE–H. Пусть сообщение X разбито на блоки длины n: $X = (x_1, x_2, \ldots, x_q)$ (если длина сообщения не кратна длине блока, дополним его до длины блока, например, нулями). Хэш, имеющий длину m, вычисляется по формуле

$$H(X) = pr_m\left(S^A_{c_2}\left(c_3, \bigoplus_{i=1}^{q} S^A_{c_1}(i, x_i)\right)\right), \tag{2}$$

где c_1, $c_2 \in B^t$ — некоторые различные константы, вес которых близок к значению $[t/2]$; $c_3 \in B^k$ — константа, вес которой близок к значению $[k/2]$.

Важным для обеспечения криптографических свойств является правильный выбор графов клеточных автоматов и локальных функций связи. В качестве семейства графов выбирается Y-семейство графов Любоцкого — Филипса — Сарнака. Подробные сведения о построении рассматриваемого семейства хэш-функций можно найти в работе [2].

2. Реализация на ПЛИС и производительность

В качестве платформы для реализации выбраны программируемые логические интегральные схемы. Сравнение производительности осуществлялось с хэш-функцией Keccak [8, 7], являющейся стандартом США (SHA-3), хэш-функциями BLAKE [9], Groestl [5], JH [12], Skein [10], которые являются финалистами конкурса NIST hash function competition, а также хэш-функцией SHA-256 [3], тоже являющейся стандартом США.

ПЛИС представляет собой микросхему, реализующую набор однотипных элементов, которые могут быть реконфигурированы различным образом. Путем программирования ПЛИС определяется конфигурация этих ячеек и связи между ними, что позволяет реализовывать произвольную цифровую схему. Описание таких схем осуществляется на специализированном языке. Наиболее часто применяются такие языки, как Verilog и VHDL. В этой статье реализация проводится на языке VHDL. Для компиляции VHDL-файла и моделирования использовались САПР Altera Quartus II 12.1 и Altera ModelSim Starter Edition 10.0d.

Наиболее важной хэш-функцией сравнения является, несомненно, Keccak (SHA-3), поскольку она является новым стандартом США. Среди данных, имеющихся в открытой печати (в частности, см. [6]), наилучшая производительность этой функции была достигнута для

ПЛИС Stratix III фирмы Altera. Кроме того, для этой ПЛИС известны результаты и других хэш-функций сравнения. В связи с этим в целях сравнения мы будем использовать именно эту ПЛИС. В то же время максимальная производительность может быть достигнута на более новой ПЛИС того же семейства — Stratix V. Поэтому мы приведем данные по семейству GRACE-H и для этой ПЛИС (результаты тестирования производительности на ПЛИС Stratix V для хэш-функций сравнения в литературе не найдены).

В качестве критериев сравнения были выбраны производительность и эффективность аппаратной реализации. Под эффективностью аппаратной реализации мы будем понимать отношение производительности к ресурсоемкости. Под ресурсоемкостью будем понимать количество блоков адаптивной логики (ALUT), требуемых для реализации.

Хэш-функцию из семейства GRACE-H с размером обобщенного клеточного автомата N, реализованную в s потоков, будем обозначать GRACE-H-N-s. В эксперименте рассматривались обобщенные клеточные автоматы размеров 390, 522 и 1010. Длина блока хэш-функции для этих клеточных автоматов составляла, соответственно, 256, 384 и 832. Число шагов клеточного автомата было выбрано равным 10.

В табл. 1 приведены результаты моделирования работы хэш-функций из семейства GRACE-H на ПЛИС Altera Stratix V. Приведены данные о тактовой частоте (МГц), ресурсоемкости (в блоках адаптивной логики ALUT), производительности (Мбит/с) и эффективности реализации (Мбит/(ALUT·с)). Эти же данные отражены на рис. 1 и 2. Видно, что максимальная производительность превышает 190 Гбит/с.

Таблица 1

Быстродействие и эффективность аппаратной реализации хэш-функций семейства GRACE-H на ПЛИС Altera Stratix V

Алгоритм и реализация	Частота, МГц	Ресурсоемкость, ALUT	Производительность, Мбит/с	Эффективность, Мбит/(ALUT · с)
GRACE-H-390-1	550	1068	14080	13,18
GRACE-H-390-2	535	1633	27392	16,77
GRACE-H-390-4	478	2851	48947	17,17
GRACE-H-390-8	383	5212	78438	15,05
GRACE-H-522-1	519	1454	19930	13,71
GRACE-H-522-2	521	2238	40013	17,88
GRACE-H-522-4	409	3962	62822	15,86
GRACE-H-522-8	357	7339	109670	14,94
GRACE-H-1010-1	479	2998	39853	13,29
GRACE-H-1010-2	441	4550	73382	16,13
GRACE-H-1010-4	357	8067	118810	14,73
GRACE-H-1010-8	298	15187	198349	13,06

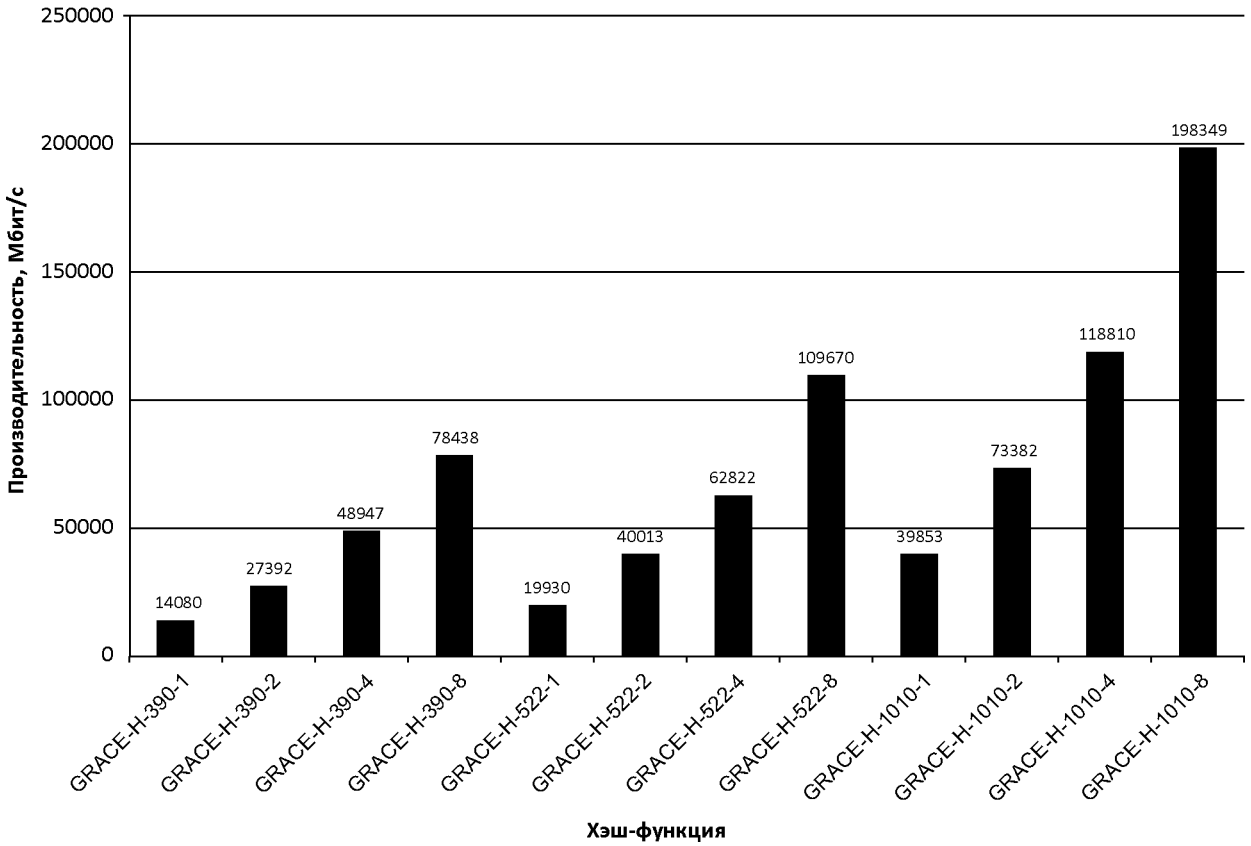

Рис. 1. Производительность при реализации на ПЛИС Altera Stratix V

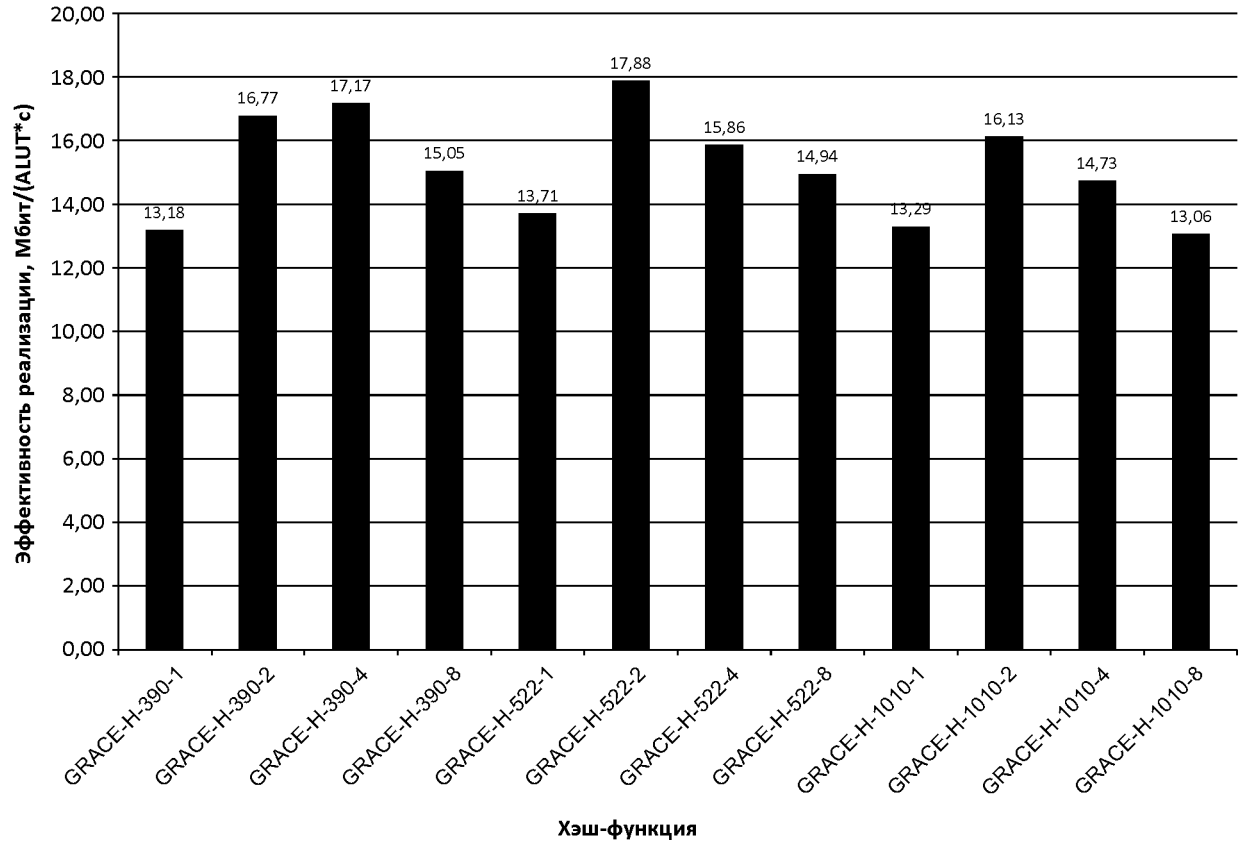

Рис. 2. Эффективность реализации на ПЛИС Altera Stratix V

Сравнение производительности и эффективности для ПЛИС Altera Stratix III, результаты которого представлены в табл. 2, осуществлялось, прежде всего, с различными реализациями Keccak, представленными в работах [4], [6] (работа авторов алгоритма Keccak, представлены две реализации, различающиеся ресурсоемкостью) и [11] (также представлены две реализации, отличающиеся ресурсоемкостью). Кроме того, как было указано выше, в сравнении участвовали хэш-функции BLAKE, Groestl, JH, Skein и SHA-256, данные по которым взяты из работы [4]. Данные по производительности и эффективности реализации также приведены на рис. 3, 4.

Таблица 2

Быстродействие и эффективность аппаратной реализации хэш-функций семейства GRACE-H, а также хэш-функций Keccak, Blake, Groestl, JH, Skein и SHA-256 на ПЛИС Altera Stratix III

Алгоритм и реализация	Частота, МГц	Ресурсоемкость, ALUT	Производительность, Мбит/с	Эффективность, Мбит/(ALUT · с)
GRACE-H-390-1	385	1088	9399	8,64
GRACE-H-390-2	377	1488	18408	12,37
GRACE-H-390-4	358	2540	34961	13,76
GRACE-H-390-8	323	4388	63086	14,38
GRACE-H-522-1	372	1481	13623	9,20
GRACE-H-522-2	374	2012	27393	13,61
GRACE-H-522-4	348	3457	50977	14,75
GRACE-H-522-8	309	5964	90527	15,18
GRACE-H-1010-1	362	2904	28723	9,89
GRACE-H-1010-2	329	3926	52209	13,30
GRACE-H-1010-4	323	6794	102515	15,09
GRACE-H-1010-8	260	11698	165039	14,11
Keccak (Gai и др. [4])	296	4458	8200	1,44
Keccak (Strömbergson 1 [11])	176	2670	10240	3,84
Keccak (Keccak team 1 [6])	206	4684	8700	1,86
Keccak (Strömbergson 2 [11])	133	242	35	0,14
Keccak (Keccak team 2 [6])	359	855	70	0,08
BLAKE	109	1965	1710	0,87
Groestl	270	3101	4024	1,30
JH	364	3120	3164	1,01
Skein	52	3602	911	0,25
SHA-256	212	963	1676	1,74

Из графиков видно, что производительность и эффективность аппаратной реализации хэш-функций семейства GRACE-H существенно (до 16 и до 4 раз соответственно) превышает производительность и эффективность лучших аппаратных реализаций хэш-функции Keccak, которые, в свою очередь, имеют более высокие показатели по сравнению с хэш-функциями Blake, Groestl, JH, Skein и SHA-256.

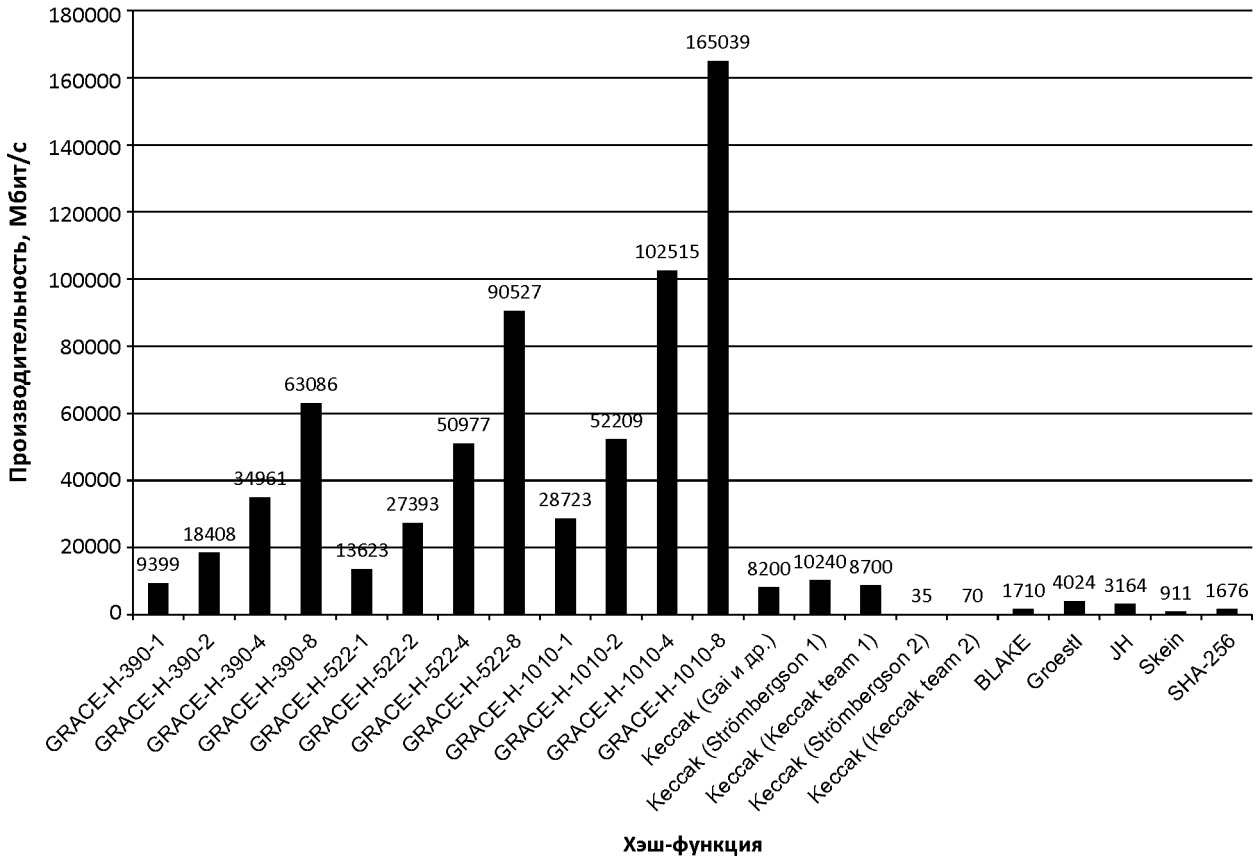

Рис. 3. Производительность при реализации на ПЛИС Altera Stratix III

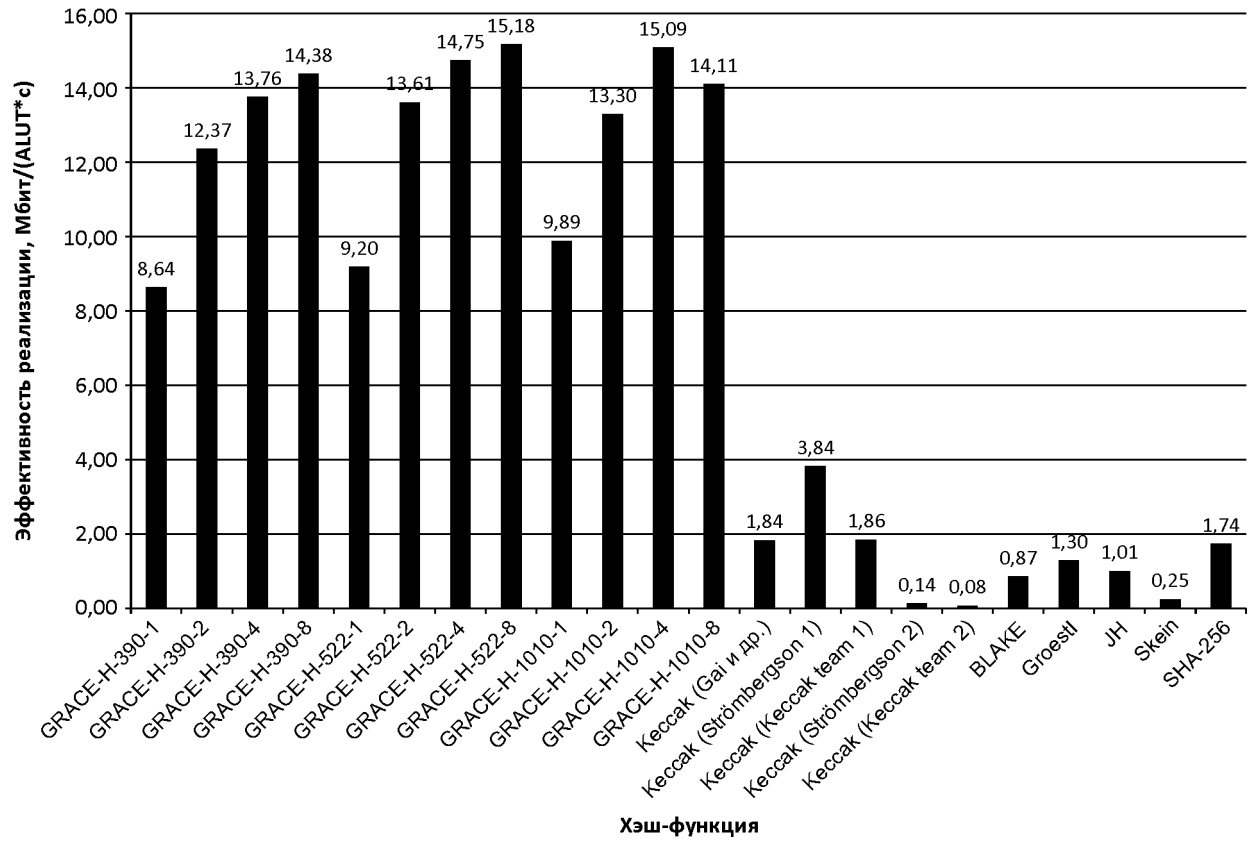

Рис. 4. Эффективность реализации на ПЛИС Altera Stratix III

Заключение

Итак, в статье продемонстрировано, что при реализации на ПЛИС, семейство хэш-функций GRACE-H обладает значительно более высокими показателями производительности и эффективности аппаратной реализации, по сравнению с хэш-функциями Keccak, Blake, Groestl, JH, Skein и SHA-256.

Автор благодарит А.М. Карондеева за помощь в подготовке материалов для этой статьи.

Работа выполнена при финансовой поддержке РФФИ (грант № 12-07-31012).

Список литературы

1. Ключарёв П.Г. Построение псевдослучайных функций на основе обобщенных клеточных автоматов // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2012. № 10. Режим доступа: http://technomag.bmstu.ru/doc/496381.html (дата обращения 01.12.2013).
2. Ключарёв П.Г. Криптографические хэш-функции, основанные на обобщенных клеточных автоматах // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 1. Режим доступа: http://technomag.bmstu.ru/doc/534640.html (дата обращения 01.12.2013).
3. FIPS PUB 180-4. Secure Hash Standard (SHS). U.S. Department of Commerce, National Institute of Standards and Technology (NIST), 2012. 30 p.
4. Gaj K., Homsirikamol E., Rogawski M. Fair and comprehensive methodology for comparing hardware performance of fourteen round two SHA-3 candidates using FPGAs // Cryptographic Hardware and Embedded Systems, CHES 2010. Springer, 2010. P. 264–278. (Ser. Lecture Notes in Computer Science; vol. 6225). DOI: 10.1007/978-3-642-15031-9_18.
5. Gauravaram P., Knudsen L.R., Matusiewicz K., Mendel F., Rechberger C., Schlaffer M., Thomsen S.S. Grøstl — a SHA-3 candidate. Submission to NIST. 2011. Available at: http://www.groestl.info/Groestl.pdf, accessed 01.12.2013.
6. Bertoni G., Daemen J., Peeters M., Van Assche G., Van Keer R. Keccak implementation overview. 2012. Available at: http://keccak.noekeon.org/Keccak-implementation-3.2.pdf, accessed 01.12.2013.
7. Bertoni G., Daemen J., Peeters M., Van Assche G. Keccak specifications. Submission to NIST (Round 2). 2009. Available at: http://keccak.noekeon.org/Keccak-specifications-2.pdf, accessed 01.12.2013.
8. Bertoni G., Daemen J., Peeters M., Van Assche G. Keccak sponge function family main document. Submission to NIST (Round 2). 2010. Available at: http://keccak.noekeon.org/Keccak-main-2.1.pdf, accessed 01.12.2013.

9. Aumasson J.-P., Henzen L., Meier W., Phan R.C.-W. SHA-3 proposal BLAKE. Submission to NIST. 2010. Available at: https://131002.net/blake/blake.pdf, accessed 01.12.2013.
10. Ferguson N., Lucks S., Schneier B., Whiting D., Bellare M., Kohno T., Callas J., Walker J. The Skein Hash Function Family. Submisson to NIST. 2008. Available at: http://www.skein-hash.info/sites/default/files/skein1.1.pdf, accessed 01.12.2013.
11. Strömbergson J. Implementation of the Keccak Hash Function in FPGA Devices. 2008. Available at: http://www.strombergson.com/files/Keccak_in_FPGAs.pdf, accessed 01.12.2013.
12. Wu H. The Hash Function JH. Submission to NIST (round 3). 2011. Available at: http://www3.ntu.edu.sg/home/wuhj/research/jh/jh_round3.pdf, accessed 01.12.2013.

SCIENTIFIC PERIODICAL OF THE BAUMAN MSTU

SCIENCE and EDUCATION

EL № FS77 - 48211. №0421200025. ISSN 1994-0408

electronic scientific and technical journal

The FPGA realization of the general cellular automata based cryptographic hash functions: Performance and effectiveness

01, January 2014
DOI: 10.7463/0114.0675812

Klyucharev P. G.

Bauman Moscow State Technical University
105005, Moscow, Russian Federation
pk.iu8@yandex.ru

In the paper the author considers hardware implementation of the GRACE-H family general cellular automata based cryptographic hash functions. VHDL is used as a language and Altera FPGA as a platform for hardware implementation. Performance and effectiveness of the FPGA implementations of GRACE-H hash functions were compared with Keccak (SHA-3), SHA-256, BLAKE, Groestl, JH, Skein hash functions. According to the performed tests, performance of the hardware implementation of GRACE-H family hash functions significantly (up to 12 times) exceeded performance of the hardware implementation of previously known hash functions, and effectiveness of that hardware implementation was also better (up to 4 times).

Publications with keywords: cryptography, FPGA, cellular automata, hash-function, hardware
Publications with words: cryptography, FPGA, cellular automata, hash-function, hardware

References

1. Klyucharev P.G. Postroenie psevdosluchaynykh funktsiy na osnove obobshchennykh kletochnykh avtomatov [Construction of pseudorandom functions based on generalized cellular automata]. *Nauka i obrazovanie MGTU im. N.E. Baumana* [Science and Education of the Bauman MSTU], 2012, no. 10. DOI: 10.7463/1112.0496381.
2. Klyucharev P.G. Kriptograficheskie khesh-funktsii, osnovannye na obobshchennykh kletochnykh avtomatakh [Cryptographic hash functions based on generalized cellular automata]. *Nauka i obrazovanie MGTU im. N.E. Baumana* [Science and Education of the Bauman MSTU], 2013, no. 1. DOI: 10.7463/0113.0534640.

3. *FIPS PUB 180-4. Secure Hash Standard (SHS)*. U.S. Department of Commerce, National Institute of Standards and Technology (NIST), 2012. 30 p.
4. Gaj K., Homsirikamol E., Rogawski M. Fair and comprehensive methodology for comparing hardware performance of fourteen round two SHA-3 candidates using FPGAs. *Cryptographic Hardware and Embedded Systems, CHES 2010*. Springer, 2010, pp. 264–278. (Ser. Lecture Notes in Computer Science; vol. 6225). DOI: 10.1007/978-3-642-15031-9_18.
5. Gauravaram P., Knudsen L.R., Matusiewicz K., Mendel F., Rechberger C., Schlaffer M., Thomsen S.S. *Grøstl — a SHA-3 candidate*. Submission to NIST. 2011. Available at: http://www.groestl.info/Groestl.pdf, accessed 01.12.2013.
6. Bertoni G., Daemen J., Peeters M., Van Assche G., Van Keer R. *Keccak implementation overview*. 2012. Available at: http://keccak.noekeon.org/Keccak-implementation-3.2.pdf, accessed 01.12.2013.
7. Bertoni G., Daemen J., Peeters M., Van Assche G. *Keccak specifications*. Submission to NIST (Round 2). 2009. Available at: http://keccak.noekeon.org/Keccak-specifications-2.pdf, accessed 01.12.2013.
8. Bertoni G., Daemen J., Peeters M., Van Assche G. *Keccak sponge function family main document*. Submission to NIST (Round 2). 2010. Available at: http://keccak.noekeon.org/Keccak-main-2.1.pdf, accessed 01.12.2013.
9. Aumasson J.-P., Henzen L., Meier W., Phan R.C.-W. *SHA-3 proposal BLAKE*. Submission to NIST. 2010. Available at: https://131002.net/blake/blake.pdf, accessed 01.12.2013.
10. Ferguson N., Lucks S., Schneier B., Whiting D., Bellare M., Kohno T., Callas J., Walker J. *The Skein Hash Function Family*. Submisson to NIST. 2008. Available at: http://www.skein-hash.info/sites/default/files/skein1.1.pdf, accessed 01.12.2013.
11. Strömbergson J. *Implementation of the Keccak Hash Function in FPGA Devices*. 2008. Available at: http://www.strombergson.com/files/Keccak_in_FPGAs.pdf, accessed 01.12.2013.
12. Wu H. *The Hash Function JH*. Submission to NIST (round 3). 2011. Available at: http://www3.ntu.edu.sg/home/wuhj/research/jh/jh_round3.pdf, accessed 01.12.2013.