УДК 004.0315 DOI: 10.18698/2541-8009-2023-8-924

# ВЫЧИСЛИТЕЛЬ ХЕШ-ФУНКЦИИ SHA-256

**С.В. Астахов** fzastahov@gmail.com
**Д.И. Вариханов** denis.varihanov@ya.ru

**МГТУ им. Н.Э. Баумана, Москва, Российская Федерация**

**Аннотация**

*Представлен проект устройства, выполняющего расчет внутреннего цикла алгоритма хеширования SHA-256. Устройство позволяет рассчитывать внутренний цикл алгоритма SHA-256 в соответствии со стандартом Secure Hash Standard. Хеш-функции, в том числе SHA-256, применяются главным образом для вычисления контрольных сумм, работы с электронной подписью и построения уникальных идентификаторов для наборов данных. Широкое применение хеш-функций в современных информационных системах обусловливает актуальность работы. При проектировании проведен анализ объекта разработки на функциональном уровне, разработана функциональная схема устройства, подготовлено описание устройства на языке Verilog, выполнен синтез RTL-схемы устройства.*

**Ключевые слова**

*SHA-256, хеш-функция, ПЛИС, FPGA, вычислитель, Verilog, Xilinx, цифровая схемотехника*

Поступила в редакцию 22.06.2023


**Введение.** Несмотря на то что хеш-функции легко могут быть реализованы программным путем, часто возникают ситуации, когда их требуется применять для большого количества сообщений (например, при работе с блокчейном). В этих условиях разница в производительности программной и аппаратной реализации становится принципиальной (в пользу последней) [1].

В рамках данной статьи рассмотрена аппаратная реализация вычислителя, осуществляющего вычисление одной из самых широко применяемых хеш-функций — SHA-256. Эта хеш-функция входит в состав семейства хеш-функций SHA-2, разработанных Агентством национальной безопасности США и опубликованных Национальным институтом стандартов и технологий в федеральном стандарте обработки информации FIPS PUB 180-2 [2].

**Алгоритм расчета SHA-256.** Все хеш-функции семейства SHA-2 построены на основе структуры Меркла — Дамгора, предусматривающей разбиение входных сообщений произвольной длины на блоки фиксированной длины и работающей с ними по очереди с помощью функции сжатия, каждый раз принимая входной блок с выходным от предыдущего прохода [3].

В случае SHA-256 каждое сообщение разбивается на блоки по 16 32-битных слов, алгоритм пропускает каждый блок сообщения через цикл с 64 итерациями.

В разработанном устройстве осуществляется преобразование сообщения в рамках этих 64 циклов, разбиение исходного сообщения на блоки возлагается на другое (внешнее) цифровое устройство или стороннюю программу. Схема одной итерации алгоритма представлена на рис. 1.

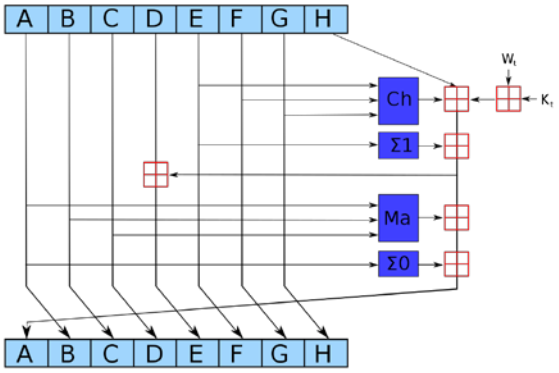


**Рис. 1.** Схема одной итерации алгоритма SHA-256:
A, B, C, D, E, F, G, H — служебные переменные; Ch, Σ1, Ma, Σ0 — математические функции, описанные в FIPS PUB 180-2 и RFC 4634 [4]; t — номер итерации; $K_t$ — служебная константа; $W_t$ — слово из блока сообщения

**Разработка функциональной схемы устройства.** По результатам анализа предметной области было заключено, что устройство должно состоять из блока управления, блока памяти переменных, блока памяти констант, выходного буфера, мультиплексирующего блока и вычислительного блока (рис. 2).

Каждые 64 цикла исполнения алгоритма блок памяти переменных инициализируется извне, в остальных циклах он сохраняет значения, полученные в предыдущем цикле. Поэтому данный блок должен иметь два информационных входа для соответствующих целей.

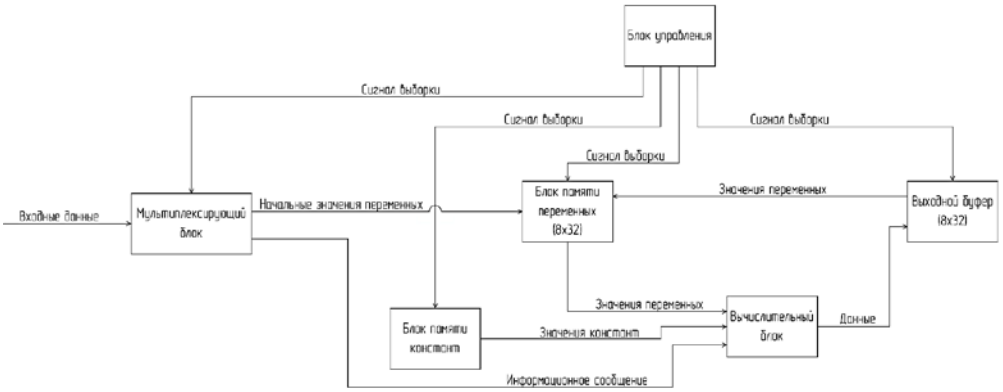

**Рис. 2.** Функциональная схема вычислителя

Мультиплексирующий блок в начале каждого раунда шифрования перенаправляет значения служебных переменных с информационного входа схемы в память. После этого данных блок перенаправляет значения информационных сообщений на вход вычислительного блока.

Вычислительный блок принимает на вход значения переменных из блока памяти переменных, значение очередной служебной константы и фрагмент информационного сообщения. Данные блок отвечает непосредственно за расчеты, описанные в стандарте SHA-256.

Выходной буфер принимает значения, полученные в ходе вычислений, и передает их на выход устройства, а также в блок памяти переменных [5].

Блок памяти констант представляет собой постоянную память, хранящую 64 служебных константы, которые необходимы для расчетов.

Блок управления на основе тактирующего сигнала генерирует необходимые сигналы выборки для блоков памяти, поскольку из-за большой разрядности все они имеют внутри себя схему выборки и для операций ввода-вывода используют мультиплексирование с разделением по времени.

**Разработка блока вычислений.** Блок вычислений представляет собой набор модулей, вычисляющих значения функций Ch, Σ1, Ma, Σ0, и является комбинационной схемой [6]. В качестве примера рассмотрим модуль, вычисляющий функцию Σ0. Функция задается формулой

$$\Sigma 0 = \left(\mathrm{a\,rotr\,2}\right)\mathrm{xor}\left(\mathrm{a\,rotr\,13}\right)\mathrm{xor}\left(\mathrm{a\,rotr\,22}\right),$$

где a — служебная переменная алгоритма вычисления хеш-функции SHA-256, является первой частью результата вычислений (см. рис. 1); rotr — операция побитового циклического сдвига вправо; xor — побитовая операция «исключающее или».

Код на языке Verilog, реализующий данные вычисления, приведен в листинге ниже.

***Исходный код модуля сигма-0***

```
// Подключение описания функции циклического сдвига вправо (rotr)
`include "right_cyclic_shift.v"
module func_sigma0(in_A, func);
    input wire[31:0] in_A;              // Список входов
    output wire[31:0] func;             // Список выходов
    wire[31:0] A2, A13, A22;            // Список соединений
    right_cyclic_shift #(2)
    A2_node( .out (A2), .num (in_A));   // A2 := A rotr 2
    right_cyclic_shift #(13)
    A13_node( .out (A13), .num (in_A)); // A13 := A rotr 13
    right_cyclic_shift #(22)
    A22_node( .out (A22), .num (in_A)); // A22 := A rotr 22
// Выход := A2 xor A13 xor A22
```

```
    assign func = A2 ^ A13 ^ A22;
endmodule
```

RTL-схема данного модуля, сгенерированная Xilinx ISE на основе приведенного кода, показана на рис. 3 [7].

Аналогично тому как к модулю сигма-0 был подключен модуль циклического сдвига вправо, модули вычисления функций Ch, Σ1, Ma, Σ0 были подключены к главному модулю блока вычислений.

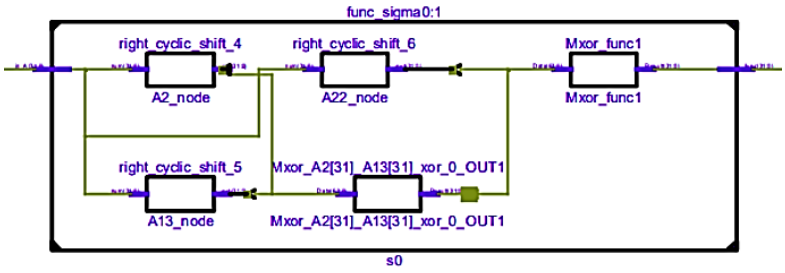


**Рис. 3.** RTL-схема модуля Сигма-0

**Тестирование блока вычислений.** В результате тестирования блока вычислений была получена временна́я диаграмма, представленная на рис. 4 [8]. В нижней половине диаграммы представлены входные значения, в верхней — выходные (время указано в пикосекундах — ps).

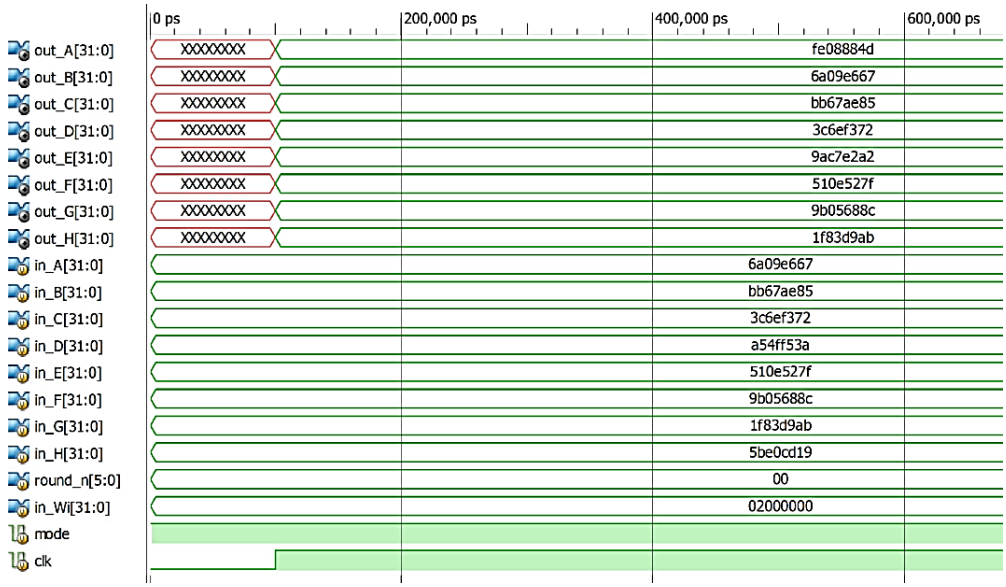


**Рис. 4.** Временная диаграмма работы блока вычислений

С целью проверки полученных результатов вычисление результатов выполнения одного раунда SHA-256 было повторно проведено вручную с теми же

начальными значениями [9]. Полученные двумя способами результаты совпали, следовательно, тестирование прошло успешно.

**Соединение основных блоков.** После разработки блока вычислений были спроектированы блок памяти переменных, блок памяти констант, выходной буфер, мультиплексирующий блок. Их внутреннее устройство довольно тривиально, а общие принципы работы очевидны из функциональной схемы. Блок памяти переменных и выходной буфер являются последовательными схемами на базе 32-битных регистров, они мультиплексируют ввод-вывод во времени, что позволяет взаимодействовать с устройством по 32-битной шине данных [10].

RTL-схема выходного буфера показана на рис. 5, в ее правой части находятся регистры, в левой — схема управления записью.

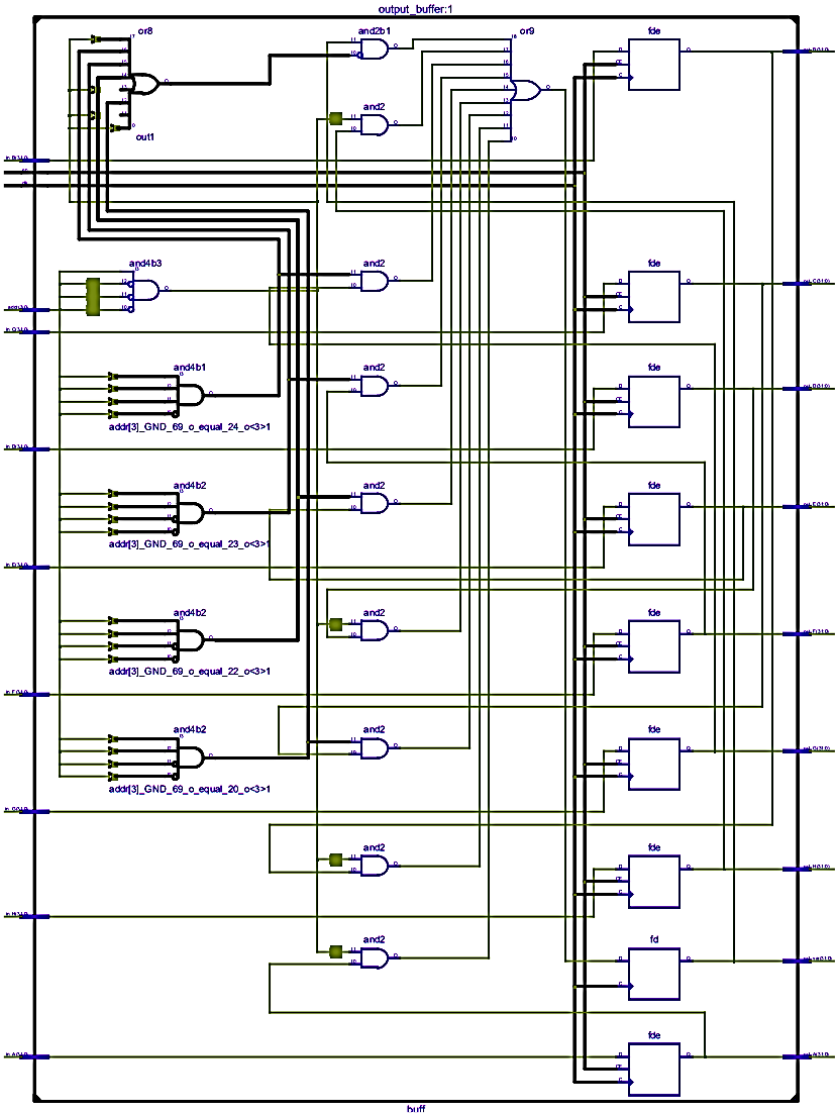

**Рис. 5.** RTL-схема выходного буфера

Временная диаграмма работы основных блоков устройства представлена на рис. 6 (время указано в миллисекундах — us).

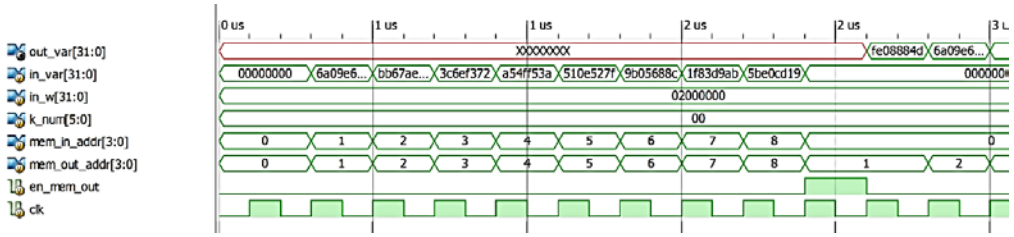


**Рис. 6.** Временная диаграмма работы основных блоков устройства

Для проверки основных блоков устройства использованы те же значения, что и для проверки блока вычислений, однако теперь значения подаются на вход (вторая строка временной диаграммы) и выход (первая строка временной диаграммы) с мультиплексированием по времени.

**Блок управления.** Поскольку сочетания сигналов выборки полностью определяются последовательным номером исполняемого цикла алгоритма SHA-256, для обеспечения корректной работы блоков памяти и упрощения внешнего интерфейса устройства был разработан блок управления. Временна́я диаграмма работы блока управления приведена на рис. 7. Этот блок на основе тактового сигнала изменяет значение внутреннего счетчика.

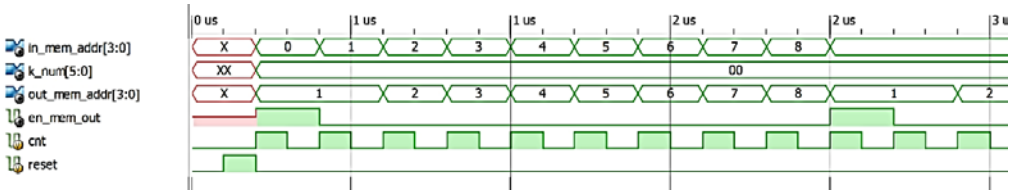


**Рис. 7.** Временная диаграмма работы блока управления

На основе значения внутреннего счетчика генерируются управляющие сигналы, соответствующие входным сигналам на рис. 6 (за исключением начального адреса выходного буфера, который не оказывает влияния на корректность работы устройства).

**Заключение.** В рамках представленной статьи рассмотрен процесс проектирования вычислителя хеш-функции SHA-256. Составлена функциональная схема устройства, в соответствии с которой вычислитель был представлен в виде совокупности блоков: блока управления, блока памяти переменных, блока памяти констант, выходного буфера, мультиплексирующего и вычислительного блоков. Приведены примеры исходного кода описания некоторых блоков вычислителя

и их RTL-схемы. Выполнена проверка работоспособности ключевых блоков устройства посредством моделирования временных диаграмм их работы.

**Литература**


[1] Hashimoto Y., Noda S. *Pricing of Mining ASIC and Its Implication to the High Volatility of Cryptocurrency Prices. Social Science Research Network*. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3368286 (дата обращения 08.04.2019).

[2] Dang Q.H. *Secure Hash Standard (SHS)*. Gaithersburg, National Institute of Standards and Technology, 2015, 36 p. https://doi.org/10.6028/NIST.FIPS.180-4

[3] Семашко А.В., Кулаков А.В. Криптографическая хеш-функция. *Информационные системы и технологии. Матер. докл. XXIV Междунар. науч.-технич. конф., посв. 100-летию Нижегородской радиолаборатории: сб. тр*. Нижний Новгород, НГТУ, 2018, с. 534–538.

[4] Eastlake D., Hansen T. *RFC 4634, US Secure Hash Algorithms*. New Jersey, AT&T Labs, 2006, 108 p.

[5] Попов А.Ю. *Проектирование цифровых устройств с использованием ПЛИС*. Москва, МГТУ им. Н.Э. Баумана, 2009, 79 с.

[6] Уилкинсон Б. *Основы проектирования цифровых схем*. Москва, Вильямс, 2004, 320 с.

[7] Cong J., Liu B., Neuendorffer S., Noguera J., Vissers K., Zhang Z. High-level synthesis for FPGAs: from prototyping to deployment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011, vol. 30 (4), pp. 473–491. https://doi.org/10.1109/TCAD.2011.2110592

[8] Spear C. *System Verilog for verification*. Marlboro, Springer Science, 2008, 425 p.

[9] *Разбираем каждый шаг хэш-алгоритма SHA-256*. URL: https://habr.com/ru/companies/selectel/articles/530262/ (дата обращения 16.06.2023).

[10] Forster K., Mull A., Doehla S., Gerhaeuzer K., Heuberger A. *Vorrichtung und Verfahren zur Übertragung einer Mehrzahl von Informationssignalen in einem flexiblen Zeitmultiplex*. Patent no. EP2230784A1, Deutschland, H04J 3/16, 2009, 24 p.



**Астахов Сергей Викторович** — бакалавр кафедры «Компьютерные системы и сети», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

**Вариханов Денис Игоревич** — бакалавр кафедры «Компьютерные системы и сети», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

**Научный руководитель** — Ким Тамара Александровна, ассистент кафедры «Компьютерные системы и сети», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.



**Ссылку на эту статью просим оформлять следующим образом:**

Астахов С.В., Вариханов Д.И. Вычислитель хеш-функции SHA-256. *Политехнический молодежный журнал*, 2023, № 08 (85). http://dx.doi.org/10.18698/2541-8009-2023-8-924

# SHA-256 HASH FUNCTION CALCULATOR

**S.V. Astahov** fzastahov@gmail.com
**D.I. Varihanov** denis.varihanov@ya.ru

**Bauman Moscow State Technical University, Moscow, Russian Federation**

## Abstract

*The paper presents design of a device that calculates inner cycle of the SHA-256 hashing algorithm. The device makes it possible to calculate the SHA-256 algorithm inner cycle in accordance with the Secure Hash Standard. Hash functions, including the SHA-256, are mainly used to calculate checksums, work with the electronic signature and construct unique identifiers for the data sets. Widespread use of the hash functions in modern information systems determines relevance of this work. During the design, the development object at the functional level was analyzed, the device functional diagram was elaborated, the device description was prepared in the Verilog language, and the device RTL diagram synthesis was performed.*

## Keywords

*SHA-256, hash function, FPGA, calculator, Verilog, Xilinx, digital circuitry*

Received 22.06.2023


## References

[1] Hashimoto Y., Noda S. *Pricing of Mining ASIC and Its Implication to the High Volatility of Cryptocurrency Prices. Social Science Research Network*. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3368286 (accessed April 08, 2019).

[2] Dang Q.H. *Secure Hash Standard (SHS)*. Gaithersburg, National Institute of Standards and Technology, 2015, 36 p. https://doi.org/10.6028/NIST.FIPS.180-4

[3] Semashko A.V., Kulakov A.V. Aspects of ensurance the integrity of information using block chain transactions research. *Informatsionnye sistemy i tekhnologii. Mater. dokl. XXIV Mezhdunar. nauch.-tekhnich. konf., posv. 100-letiyu Nizhegorodskoy radiolaboratorii: sb. tr*. [Information systems and technologies. Proceedings of the XXIV International Scientific and Technical Conference Dedicated to the 100th Anniversary of the Nizhny Novgorod Radio Laboratory]. Nizhniy Novgorod, NNSTU Publ., 2018, pp. 534–538. (In Russ.).

[4] Eastlake D., Hansen T. *RFC 4634, US Secure Hash Algorithms*. New Jersey, AT&T Labs, 2006, 108 p.

[5] Popov A.Yu. *Proektirovanie tsifrovykh ustroystv s ispol'zovaniem PLIS* [Designing digital devices using FPGAs]. Moscow, BMSTU Press, 2009, 79 p. (In Russ.).

[6] Uilkinson B. *Osnovy proektirovaniya tsifrovykh skhem* [Fundamentals of digital circuit design]. Moscow, Vil'yams Publ., 2004, 320 p.

[7] Cong J., Liu B., Neuendorffer S., Noguera J., Vissers K., Zhang Z. High-level synthesis for FPGAs: from prototyping to deployment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011, vol. 30 (4), pp. 473–491.

https://doi.org/10.1109/TCAD.2011.2110592
[8] Spear C. *System Verilog for verification*. Marlboro, Springer Science, 2008, 425 p.
[9] *Razbiraem kazhdyy shag khesh-algoritma SHA-256* [Parsing each step of the SHA-256 hash algorithm]. URL: https://habr.com/ru/companies/selectel/articles/530262/ (accessed June 16, 2023).
[10] Forster C., Mull A., Doehla S., Gerhaeuzer K., Heuberger A. *Apparatus and method for transmitting a plurality of information signals in flexible time-division multiplexing*. Patent no. US8804768B2, USA, H04J 3/16, 2014, 21 p.

**Astahov S.V.** — Bachelor, Department of Computer Systems and Networks, Bauman Moscow State Technical University, Moscow, Russian Federation.
**Varihanov D.I.** — Bachelor, Department of Computer Systems and Networks, Bauman Moscow State Technical University, Moscow, Russian Federation.

**Scientific advisor** — Kim T.A., Assistant, Department of Computer Systems and Networks, Bauman Moscow State Technical University, Moscow, Russian Federation.

**Please cite this article in English as:**
Astakhov S.V., Varikhanov D.I. SHA-256 hash function calculator. *Politekhnicheskiy molodezhnyy zhurnal*, 2023, no. 08 (85). (In Russ.). http://dx.doi.org/10.18698/2541-8009-2023-8-924