REVIEW ON AREA OPTIMIZATION AND SIMULATION OF SHA-3 (KECCAK-512)

Priya Kotewar[#1], Prof. R.N.Mandavgane[#2]
Prof. D.M.Khatri[#3]
[1]*Student Mtech(VLSI) BDCE Sevagram, Wardha, India,*
[2] *Associate ProfessorBDCE, Sevagram, India,*[3]*Assistant ProfessorBDCE,Sevagram, India,*

***Abstract*—: The new SHA-3 hash algorithm wasannounced on October 2, 2012 by the NIST and will be used to provide security to any application which requires hashing, pseudo-random number generation, and integrity checking. Keccak is the new Standard Hash Algorithm (SHA-3) to complement the hash standard SHA-2. After five years of tough competition hash function has been selected.SHA-3, a subset of the cryptographic primitive family Keccak .It is a cryptographic hash function designed by Guido Bertoni.SHA core is composed of the Data path and the Controller, the Controller is implemented using Finite State Machines. The finite state machines responsible for input and output are almost identical for all hash function candidates.Keccak is based on four basic logic operations: xor, and, not and rotate. We are improving the performance of SHA-3 by reducing the area as compare to previously design SHA-3 algorithm.**

***IndexTerms*—SHA-3, Keccak-512, Data path,Controllers.**

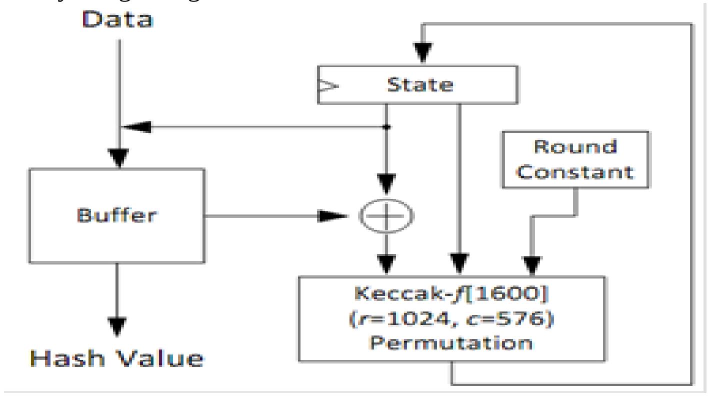
1. Introduction

Security has become a crucial aspect in the design and use of computer systems and networks. Whether one is designing a wireless communication system, web application, or network protocol, addressing security is an essential engineering criterion. Though a well-designed system is built from a multitude of components, the use of cryptography as a building block is almost unanimous. Cryptography is used to address many security issues, the most pertinent of which are confidentiality, integrity, and authentication. Cryptography encompasses the design of cryptographic primitives, basic building blocks, and protocols/schemes that use these building blocks to construct complex security systems.

SHA-3, a subset of the cryptographic primitive family Keccak is a cryptographic hash function designed by Guido Bertoni, Joan Daemen, Michel, Peeters, and Gilles Van Assche, building upon RadioGatun. In October 2, 2012, Keccak was selected as the winner of the NIST hash function competition.SHA-3 uses the sponge construction in which message blocks are XORed into a subset of the state, which is then transformed as a whole. In the version used in SHA-3, the state consists of a 5×5 array of 64-bit words, 1600 bits total.Hash algorithms are used widely for cryptographic applications that ensure the authenticity ofdigital documents, such as digital signatures and message authentication codes. These algorithms take an electronic file and generate a short "digest," a sort of digital fingerprint of the content. A good hash algorithm has a few vital characteristics. Any change in the original message, however small, must cause a change in the digest, and for any given file and digest, it must be infeasible for a forger to create a different file with the same digest.

The NIST team praised the Keccak algorithm for its many admirable qualities, including its elegant design and its ability to run well on many different computing devices. The clarity of Keccak's construction lends itself to easy analysis and it has higher performance in hardware implementations than SHA-2 or any of the other finalists. The Keccak sponge function family is characterized by three parameters: the bitrate r, the capacity c and the diversifier d. The sponge construction uses r +c bits of state, of which r are updated with message bits between each application of Keccak-f during the absorbing phase and output during the squeezing phase. The remaining c bits are not directly affected by message bits, nor are they taken as output.[1]

Keccak inherits the flexibility of the sponge and duplex constructions. As a sponge function, Keccak has arbitrary output length. This allows to simplify modes of use where dedicated constructions would be needed for fixed-output-length hash functions. It can be natively used for, e.g., hashing, full domain hashing, randomized hashing, stream encryption, and MAC computation. In addition, the arbitrary output length makes it suitable for tree hashing. The duplex function it can be used in clean and efficient modes as a reseedable pseudo-random bit generator and for authenticated encryption. Efficiency of duplexing comes from the absence of output transformation.Keccak has a thick safety margin. Figure shows the Keccak-256 core.



Keccak core

2. LITERATURE REVIEW

A design strategy and evaluation criteria for a fair candidates is proposed. A SASEBO-GII field-programmable gate array (FPGA) board as a common platform is used in combination with well-defined hardware and software interfaces. All 256-bit version candidates are compared with respect to area, throughput, latency, power, and energy consumption. The second contribution is that we provide both FPGA and 90-nm CMOS application-specific integrated circuit (ASIC) synthesis results and thereby are able to compare the results. Our third contribution is that they release the source code of all the candidates and by using a common, fixed, publicly available platform, our claimed results become reproducible and open for a public verification. They conclude that the obtained FPGA results represent rather reliable way of estimating the ASIC performance, especially with respect to speed and area. [2]

A two-staged pipelined architecture of the new SHA-3 algorithm is presented. The core can operate on both one-block and multi-block messages. In this paper a two-staged pipelined architecture of the new SHA-3 (Keccak) algorithm is presented. The core can operate on both one-block and multi-block messages. Special effort has been paid and different design alternatives have been studied to derive efficient FPGA implementations in terms of throughput and throughput/area metrics. The proposed Xilinx Virtex-5, Virtex-6, and Virtex-7 FPGA technologies and achieves significant improvements compared to existing FPGA implementations. Future work include optimized FPGA implementations of the finalized SHA-3 standard. [3]

The software implementation of Keccak - 512 on two platforms - Intel core-is and Cavium Networks Octeon embedded platform is done in this paper. Along with this benchmarking of our code on the former platform and its comparison with benchmarking results of other Keccak implementations. The perfonnance of Keccak-512 is evaluated on a resource-efficient platform and a resource-constrained platform for short input messages. The performance result of Intel core-i5 2450M is magnificent as compare to Octeon CN5860 is an embedded system in terms of speed in MB/S. [4]

Propose an efficient concurrent error detection scheme is used in order to provide reliable architectures for this algorithm. The secure hash algorithm has been selected in 2012 and will be used to provide security to any application which requires hashing, pseudo-random number generation, and integrity checking. This algorithm has been selected based on various benchmarks as security, performance, and complexity. Subpipelining is to overcome the inherent throughput degradation of this time-redundancy approach. A time-redundancy scheme is presented for error detection of the recently-standardized secure cryptographic SHA-3 algorithm, i.e., Keccak.[5]

All the SHA-3 candidates are compared in this paper The comparative study involves different hash functions criteria such as; security, structure, and performance and cost, to measure the robustness of the algorithms through the Fundamentals Security Measurement Factors of Hash Function (FSMFHF) of Secure Hash Algorithm (SHA). [6]

Focus on two of these candidate algorithms, namely BLAKE and Shabal by presenting the common structure for all the SHA3 candidates. After the designing of VLSI circuit the hardware evaluations is done on FPGA and ASIC.SHA-256 algorithms and SHA-3 are compared in the same technology, after comparison found that the SHA3 candidates provide higher throughput but cost more area. Because of raising the complexity of circuit, the efficiency of SHA3 is lower than SHA-256.The common architecture and FSM are presented for all the SHA3 candidates, and design VLSI circuit for two SHA-3 candidates namely BLAKE and shabal and implemented by ASIC and FPGA. SHA-3 algorithm has larger throughput and higher circuit complexity, and is more suitable for high speed and security hardware implementation.[7]

A new Standard Hash Algorithm (SHA-3) supplies more security, a public competition was organized by NIST in 2007. The paper focus on two of these candidate algorithms, namely BLAKE and Shabal and present the common structure for all the SHA3 candidates. They also design the VLSI circuit and give the hardware evaluations on FPGA and ASIC. Compared with SHA-256 algorithms in the same technology, they found that the SHA3 candidates provide higher throughput but cost more area. Because of raising the complexity of circuit, the efficiency (TP/area) of SHA3 is lower than SHA-256. They present the common architecture and FSM for all the SHA3 candidates. The VLSI circuits of two SHA-3 candidate algorithms: BLAKE, and Shabal are designed, and implemented by ASIC and FPGA as well. Compared with SHA-256, SHA-3 algorithm has larger throughput and higher circuit complexity, and is more suitable for high speed and security hardware implementation. [8]

3. EXPERIMENTAL STUDY

Each SHA core is composed of the Data path and the Controller. The Controller is implemented using three main Finite State Machines, working in parallel, and responsible for the Input, Main Processing, and the Output, respectively. As a result, each circuit can simultaneously perform the following three tasks: output hash value for the previous message, process a current block of data, and read the next block of data. The parameters of the interface are selected in such a way that the time necessary to process one block of data is always larger or equal to the time necessary to read the next block of data. This way, the processing of long streams of data can happen at full speed, without any visible input interface overhead. The finite state machines responsible for input and output are almost identical for all hash function candidates; the third state machine, responsible for main data processing, is based on a similar template. The similarity of all designs and reuse of common building blocks assures a high fairness of the comparison.

Keccak is based on four basic logic operations: xor, and, not and rotate. For every message block, an input is zero-extended to produce a 160**0**-bit state. This state can be viewed as a 5x5 array of 64-bit words.

4. FUTURE WORK AND CONCLUSION

In this propose work we are using the data path and controller .The implementation of controller is done by using Finite State Machine. The finite state machines responsible for input and output are almost identical for all hash function candidates the third state machine, responsible for main data processing, is based on a similar template. We are improving the performance of SHA-3 by reducing the area as compare with previously design SHA-3.

In future work, we will try to develop SHA-4 to improve security.

REFRENCES:

[1] Keccak Hash Function, NIST (National Institute of Standards and Technology),(2014, Mar.)[Online].Available:http://csrc.nist.gov/groups/ST/hash/sha-3

[2] Miroslav Knezevic, Kazuyuki Kobayashi, Jun Ikegami, Shin ichiro Matsuo, Akashi Satoh,Unal Kocabas¸ Junfeng Fan, “Fair and Consistent Hardware Evaluation of Fourteen Round Two SHA-3 Candidates”.IEEE-2012

[3] George S. Athanasiou, George-Paris Makkas, Georgios Theodoridis “High Throughput Pipelined FPGA Implementation of the New Sha-3 Cryptographic Hash Algorithm”. IEEE-2014

[4] Aisha Malikl, Arshad Aziz, Dur- e-Shahwar Kunde ,Moiz Akhter,”Software Implementation of Standard Hash (SHA-3) Keccak on Intel Core-i5 and Cavium Networks Octeon Plus embedded platform”.IEEE-2013

[5] Siavash Bayat-Sarmadi, Mehran Mozaffari-Kermani, and Arash Reyhani-Masoleh.”Efficient and Concurrent Reliable Realization of the Secure Cryptographic SHA-3 Algorithm”.IEEE-201

[6] Imad Fakhri Alshaikhli, Mohammad A. Alahmad, Khanssaa Munthir”.Comparison And Analysis Study of Sha-3 Finalists”.IEEE-2013

[7] Liang Han, Bai Guoqiang. “Hardware implementation analysis of SHA-3 candidates algorithms”IEEE-2010.

[8] K. Latif, M. Rao, A. Aziz, and A. Mahboob, “Efficient hardware implementations and hardware performance evaluation of SHA-3”IEEE-.2014

[9] S. Tillich, M. Feldhofer, M. Kirschbaum, T. Plos, J. Schmidt, and A.Szekely,“Uniform evaluation of hardware implementations of the round-two SHA-3 candidates,” presented at the 2nd SHA-3 Candidate “.Conf., Santa Barbara, CA, 201