HARDWARE ACCELERATION OF SECURE HASH ALGORITHM 3

NG LAI BOON

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer and Microelectronic Systems)

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

JANUARY 2020

DEDICATION

This project report is dedicated to my parents, who taught me never to give up and always strive to be the best. It is also dedicated to my siblings, who provided me with moral support throughout the entire length of the project.

ACKNOWLEDGEMENT

First of all, I would like to take this opportunity to express my deepest gratitude to Associate Professor Ir. Dr. Muhammad Nadzir bin Marsono, my master project supervisor who has very supportive in giving constructive advice and encouragement to enable me to complete the project. With his guidance, I have successfully solved many problems encountered in this project.

I would also like to thank my beloved parents and other family members for their continuous support and encouragement especially when I faced problems in this project. Without their support and encouragement, I would not have gone this far, especially juggling between work and academic workload simultaneously.

ABSTRACT

Secure Hash Algorithm-3 (SHA-3) is the most recent and efficient cryptography hash functions widely used in most information security applications. Contemporary, SHA-3 were built as software where the performance of the cryptographic function is based on the performance of the general-purpose CPU. Since SHA-3 is frequently used in requires multiple operations per data input and is generally inefficient running on a general-purpose CPU. To improve the performance of the SHA-3 function on data encryption progress, an alternative solution is to implement the SHA-3 algorithm as a hardware accelerator. The proposed accelerator is targeted at the most computation-intensive function in the C-based SHA-3 algorithm which is Keccak-f that is executed 97.56% in an SHA-3 operation and synthesized using Xilinx's Vivado High-Level Synthesis (HLS) to hardware implementations targeted for FPGAs. Besides, the proposed SHA-3 accelerator is employed in the architectural optimization approaches based on the concepts of loop pipelining, loop unrolling, and memory array mapping. Considering the trade-offs between the performance and hardware cost, the SHA-3 architecture in terms of the high throughput and less resource is identified. Incorporated with four-stage sub-pipelining and fully loop unrolling on five permutation steps, followed by memory array partitioning by factor of 25, new SHA-3 hardware architecture is proposed. The proposed SHA-3 accelerator is able to achieve up to 47.7Gbps throughput and operate up to 722.5 MHz. As compared to other existing works, the proposed SHA-3 implementation achieves high throughput performance and operation frequency at a reasonable cost.

ABSTRAK

Algorithma selamat cincangan-3 (SHA-3) adalah fungsi cincangan kriptografi terkini dan cekap yang digunakan secara meluas dalam kebanyakan aplikasi keselamatan maklumat. Kontemporari, SHA-3 dibinakan sebagai perisian di mana prestasi fungsi kriptografi didasarkan pada prestasi CPU tujuan umum. Oleh sebab SHA-3 sering digunakan dalam memerlukan banyak operasi setiap input data dan secara amnya tidak berkesan berjalan pada CPU tujuan umum. Untuk meningkatkan prestasi fungsi SHA-3 pada kemajuan penyulitan data, penyelesaian alternatif adalah untuk melaksanakan algoritma SHA-3 sebagai pemecut perkakasan. Pemecut yang dicadangkan ini disasarkan kepada fungsi intensif pengkomputeran dalam algoritma SHA-3 yang berasaskan C iaitu Keccak-f yang dijalankan 97.56 % dalam operasi SHA-3 dan disintesis menggunakan Xilinx's Vivado High-Level Synthesis (HLS) kepada pelaksanaan perkakasan yang disasarkan untuk FPGA. Selain itu, pemecut SHA-3 ini digunakan dalam pendekatan pengoptimuman seni bina berdasarkan konsep talian paip gelung, gelung yang tidak bergulung, dan pemetaan pelbagai memori. Memandangkan pertukaran antara prestasi dan kos perkakasan, seni bina SHA-3 dari segi daya tampung tinggi dan kurang sumber dikenalpasti. Diperbadankan dengan sub-talian paip empat peringkat dan gelung penuh yang tidak bergulung pada lima langkah permutasi, diikuti dengan pemisahan tatasusunan memori dengan faktor 25, seni bina perkakasan SHA-3 baru dicadangkan. Pemecut SHA-3 yang dicadangkan mampu mencapai sehingga 47.7Gbps penghantaran dan beroperasi sehingga 722.5 MHz. Berbanding dengan kerja-kerja yang sedia ada, pelaksanaan SHA-3 yang dicadangkan mencapai prestasi tinggi dan kekerapan operasi yang tinggi pada kos yang munasabah.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AES	-	Advanced Encryption Standard
ASIC	-	Application-Specific Integrated Circuit
ECC	-	Elliptic Curve Cryptography
FPGA	-	Field-Programmable Gate Array
GPP	-	General Purpose Processor
HLS	-	High Level Synthesis
IoT	-	Internet-of-Things
ISE	-	Integrated Synthesis Environment
LUT	-	Lookup Table
MACs	-	Message Authentication Codes
MD5	-	Message-Digest Algorithm
NIST	-	National Instituteof Standards and Technology
RAM	-	Random Access Memory
RSA	-	Rivest-Sharmir-Adleman
RTL	-	Register Transfer Level
SHA	-	Secure Hash Algorithm
SoC	-	System-on-Chip
TPA	-	Throughput to Area Ratio
VHDL	-	Very High Speed Integrated Circuit Hardware Description Language

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Research Background

Cryptographic algorithm is a process of converting plaintext into ciphertext. It has become an extremely important feature in modern cloud computing systems and has widely used in protecting information within many kinds of network and cloud applications such as encryption and decryption. With the widely use of these systems, information security attacks are also evolving, therefore security has become one of the most important factors in the new generation networking applications [1]. Various cryptographic algorithms have been developed for secure communication and data encryption, they can be divided into three different categories such as symmetric encryption, asymmetric encryption, and hash functions. Symmetric encryption is public-key encryption such as Advanced Encryption Standard (AES) and BlowFish that use the same key for encryption and decryption. Asymmetric encryption is private key encryption such as Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC) that use different keys for encryption and decryption. The third cryptographic algorithm - hash function is a one-way hash algorithm used in data integrity and its example is the message-digest algorithm (MD5) and Secure Hash Algorithm (SHA).

Data integrity is important for security applications and it is used to confirm system data has not been distorted during transmission, storage and restore. Cryptography hash functions are commonly used in integrity verification and it is an efficient hashing technique [2]. SHA-3 is the most recent and efficient cryptographic hash functions widely used in digital signature schemes, message authentication codes (MACs) and several other information security applications. [3]. SHA-3 is the new Secure Hash Algorithm released by the National Institute of Standards and Technology (NIST) in 2015 based on the Keccak function developed by Bertoni et al. [2]. This algorithm is proposed to replace the existing common hash functions like SHA-1, SHA-2, and MD-5 for better security.

Conventional cryptography algorithms are generally computationally intensive because their complexity and requires high number rounds to encrypt/decrypt, therefore consume a high rate of power and time when working on Internet of Thing (IoT) devices [4]. State-of-the-art SHA-3 were built as software where the performance of the cryptography function is based on the performance of the general-purposed central processing unit (CPU). Since the hash function will be frequently used, it is requiring many operations per input and is generally inefficient on a general-purpose CPU. With the increase of the message data size, the difficulty and process time are also increasing. Therefore, the acceleration of the SHA-3 algorithm is needed for an incoming modern computing system.

To improve the performance of the SHA-3 function on data hashing progress, an alternative solution is implementing the SHA-3 algorithm as hardware accelerator. In recent years, cryptographic hash functions are widely implemented in hardware due to they are proven to be more secure than in software realizations [5]. Field-programmable gate array (FPGA) has been widely used to improve the performance of processors for specific applications due to its unique properties such as re-configurable and support parallelization and pipelining that not exist in the general-purpose processor (GPP). This specialization of FPGA provides a compromise between the flexibility of a GPP and the performance of an Application-Specific Integrated Circuit (ASIC). The weakness and strength of GPP, FPGA and ASIC are shown in Table 1.1. Due to comparable performance in FPGA, this means that for application implementation, FPGA is better than the general-purpose system due to portability, low cost and low power consumption [6]. Moreover, FPGA owns a property of parallelism and make it have real-time computation. Hence, implementing an SHA-3 algorithm in FPGA makes the system to be more viable and able to provide better performance compared to software.

1.2 Problem Statement

Various cryptography algorithms have been developed for security applications. SHA-3 is the most recent and efficient cryptographic hash functions widely used in most security applications. Conventional SHA-3 were built as software where the

Table 1.1: Weakness and strength of GPP, FPGA and ASIC

	GPP	**FPGA**	**ASIC**
Execution	Sequential	Parallel	Parallel
Performance	Low	High	Very High
Flexibility	Excellent	Good	Poor
Power efficient	Low	Moderate	High
Design cost	Small	Large	Very Large
Time to market	Excellent	Good	Poor

performance of the cryptographic function is based on the performance of the general-purpose CPU. Conventional cryptography algorithms are generally computation-intensive algorithms because of their complexity and require a high number of rounds to encrypt. Since the hash function will be frequently used, it is requiring many operations per input and is generally inefficient on a general-purpose CPU.

To improve the performance of the SHA-3 hash algorithm to overcome the bottleneck of software progressing in GPP, several hardware solutions such as co-processors utilize specialized hardware to do parallel acceleration in algorithm computation [7]. They can significantly improve the performance of the SHA-3 algorithm and speed up computation. However, this solution also brings additional resource overhead to the silicon design. Several research studies on different kinds of hardware implementations of the SHA-3 algorithm have been proposed. The objectives of hardware implementation are either towards the architecture with the high-speed performance or lightweight design [5].

In order to optimize the hardware implementation to improve latency, throughput and overcome performance bottlenecks, several optimization concepts that usually used in the designs are loop rolling and unrolling, loop and data-flow pipelining and array mapping. However, the performance of the SHA-3 algorithm can be further improved with other techniques or mixed techniques above. Therefore, a further improvement of acceleration for the SHA-3 algorithm is needed to speed up the hashing process.

1.3 Project Objective

This project aims to design and develop hardware acceleration of the SHA-3 algorithm. To achieve the aim, the objectives of this project are:

1. To perform software profiling to investigate the time-consuming SHA-3 algorithm.
2. To propose hardware accelerator architecture for improved SHA-3 performance.
3. To evaluate the performance-area trade-off of the proposed SHA-3 hardware accelerator against the conventional SHA-3 implementation.

1.4 Project Scope

The scope of this project is focusing on the architecture design of the hardware accelerator. The hardware accelerator is implemented using Vivado High-Level Synthesis (HLS) for the FPGA platform. The proposed work will focus on developing the hardware accelerator for the most intensive computational function of the SHA-3 algorithm, Keccak-f function. Besides that, the performance of the proposed SHA-3 accelerator is simulated and evaluated in terms of the maximum frequency, latency, throughput and resource needed to process the SHA-3 algorithm.

1.5 Chapter Organization

The report for project is organized with 5 chapters. Chapter 1 provide the introduction of the project including research background, problem statement, objective, and scope. Chapter 2 reviews the SHA-3 operation and the related literature review on the state-of-the-art for hardware acceleration of the hash algorithm. Chapter 3 describes the methodology for the overview of proposed approaches and hardware accelerator architecture. Chapter 4 elaborates on the execution work, experiment setup, results for software profiling, hardware implementation, C/RTL cosimulation

and also the comparison with the existing work. Lastly, Chapter 5 concludes the project accomplishment and discuss on the future works.

REFERENCES

1. Regazzoni, F. and Ienne, P. Instruction Set Extensions for Secure Applications. *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2016. 1529–1534.

2. Chandran, N. R. and Manuel, E. M. Performance Analysis of Modified SHA-3. *Procedia Technology*, 2016. 24: 904 – 910. International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015).

3. Eissa, A. S., Elmohr, M. A., Saleh, M. A., Ahmed, K. E. and Farag, M. M. SHA-3 Instruction Set Extension for a 32-bit RISC processor architecture. *2016 IEEE 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. 2016. 233–234.

4. Usman, M. and Khan, S. SIT: A Lightweight Encryption Algorithm for Secure Internet of Things. *International Journal of Advanced Computer Science and Applications*, 2017. 8.

5. Wong, M. M., Haj-Yahya, J., Sau, S. and Chattopadhyay, A. A New High Throughput and Area Efficient SHA-3 Implementation. 2018. 1–5.

6. Hamada, T., Benkrid, K., Nitadori, K. and Taiji, M. A Comparative Study on ASIC, FPGAs, GPUs and General Purpose Processors in the O(N-2) Gravitational N-body Simulation. *Proceedings - 2009 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2009*, 2009.

7. Wu, X. and Li, S. High Throughput Design and Implementation of SHA-3 Hash Algorithm. *2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC)*. 2017. 1–2.

8. Rao, J., Ao, T., Xu, S., Dai, K. and Zou, X. Design Exploration of SHA-3 ASIP for IoT on a 32-bit RISC-V Processor. *IEICE Transactions on Information and Systems*, 2018. E101.D(11): 2698–2705.

9. Bertoni, G., Daemen, J., Peeters, M. and Assche, G. V. The Keccak reference. *STMicroelectronics: NXP Semiconductors*, 2011: 1 – 69.

10. Bertoni, G., Daemen, J., Hoffert, S., Peeters, M., Assche, G. V. and Keer, R. V. Keccak Specifications Summary, 2011. URL https://keccak.team/keccak_specs_summary.html.

11. Sideris, A., Sanida, T. and Dasygenis, M. Hardware Acceleration of SHA-256 Algorithm Using NIOS-II Processor. *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. 2019. 1–4.

12. K.N., D. and Bhakthavatchalu, R. Parameterizable FPGA Implementation of SHA-256 using Blockchain Concept. *2019 International Conference on Communication and Signal Processing (ICCSP)*. 2019. 0370–0374.

13. Ahmed, K. E. and Farag, M. M. Hardware/Software Co-design of a Dynamically Configurable SHA-3 System-on-Chip (SoC). *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*. 2015. 617–620.

14. Ioannou, L., Michail, H. E. and Voyiatzis, A. G. High Performance Pipelined FPGA Implementation of the SHA-3 Hash Algorithm. *2015 4th Mediterranean Conference on Embedded Computing (MECO)*. 2015. 68–71.

15. Adnan, S. M. and Aziz, A. Resource Efficient and Area Optimized Grøstl Implementation on FPGA. *2012 International Conference on Open Source Systems and Technologies*. 2012. 1–4.

16. Sugier, J. Simplifying FPGA Implementations of BLAKE Hash Algorithm with Block Memory Resources. *Procedia Engineering*, 2017. 178: 33–41. URL http://www.sciencedirect.com/science/article/pii/S1877705817300577.

17. Arora, H. GPROF Tutorial: How to use Linux GNU GCC Profiling Tool, 2012. URL https://www.thegeekstuff.com/2012/08/gprof-tutorial.

18. Feist, T. White Paper: Vivado Design Suite, 2012. URL https://www.xilinx.com/support/documentation/white_papers/wp416-Vivado-Design-Suite.pdf.

19. Vivado Design Suite User Guide: High-Level Synthesis, 2019. URL https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug902-vivado-high-level-synthesis.pdf.

20. Baldwin, B. and Marnane, W. P. An FPGA Technologies Area Examination of the SHA-3 Hash Candidate Implementations. *Cryptology ePrint Archive, Report 2009/603*. 2009.

21. 7 Series FPGAs Configurable Logic Block, 2016. URL `https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf`.

22. Stallings, W. *Computer Organization and Architecture: Designing for Performance*. 10th ed. Pearson. 2015.

23. Jivsov, A. C Implementation of the SHA-3 and Keccak with Init/Update/Finalize Hashing API (NIST FIPS 202/Etherium), 2015. URL `https://github.com/brainhub/SHA3IUF`.