# ASIC IMPLEMENTATION OF LOW LATENCY MONTGOMERY MODULAR EXPONENTIATION

LIEW PUI YEN

UNIVERSITI TEKNOLOGI MALAYSIA

# ASIC IMPLEMENTATION OF LOW LATENCY MONTGOMERY MODULAR EXPONENTIATION

LIEW PUI YEN

A project report submitted in 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

JULY 2022

## DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

## ACKNOWLEDGEMENT

The completion of a project usually depends on cooperation, coordination, and combined efforts from different people. While I am performing my project assignment, I had to take some trusted people's support and encouragement, who deserve my greatest gratitude. First, I would like to express my sincere appreciation to my Final Year Project course's lecturer, Dr. Shahidatul Sadiah Binti Abdul Manan for her guidance in accomplishing the task she has given to me. Her guideline helped a lot when I am doing my design project. Besides, I am also grateful and appreciate the effort of other lecturers and staffs for providing cooperation, valuable information, suggestions, and guidance in the preparation of this final year project.

Deepest thanks and appreciations to my parents, family, special mates, and others for their cooperation, encouragement, constructive suggestion and full of support for the project completion, from the beginning until the end. Moreover, I would like to express gratitude to my family and friends for the moral and financial support. I would also like to thank for my course mates for giving a big cooperation to my project. They often provide me with valuable suggestion to improve my project. Finally, I would also like to thank to all the people who have supported me to complete the project directly or indirectly.

## ABSTRACT

Nowadays, electronic communication devices tend to design smaller in size, lighter in weight, lower in cost and higher performance. Individual may tend to use electronic communication devices when exchanging sensitive matters, such as personal details, contract documents, company secrets and specific passwords are sent to other parties. Since internet is one of the important key contacts and electronically communicates with billions of people, protection for the transmission of important messages over the internet is vital. Encryption plays a vital role for every user in ensuring security of communication within the organization. Hence the algorithms needed for safe communication. The motivation of this project is to protect digital data in computer confidentiality, as it is often stored on computer systems and distributed through the internet or other computer networks. Rivest-Shamir-Adleman algorithm is first introduced by Ron Rivest, Adi Shamir and Leonard Adelman in 1977, and it is known as one of the famous public key cryptography algorithms since it is an asymmetric cryptography. Besides, the theory behind RSA is relatively simple and easy for modification purpose as it relies on algorithm such as factorization and modular exponentiation. In this paper, the whole process and algorithm has been described for 256-bit key size. Due to the bit length of modulus, the work included different but suitable implementation, which is the basic, radix-4 and radix-16 implementations to reduce the speed of cipher-decipher process. Implementation on Verilog HDL using Vivado Design Suite software has been done. Enhancement on speed and delay is the main constraint of this project. According to the synthesis results, the radix-16 Montgomery Multiplier implemented in RSA cipher can be implemented with a nearly 60% reduction in encryption latency. However, radix implementation will involve more loop unrolling steps that resulted in a higher gate count. It is conceivable to absorb the increase in the gate count in the RSA cipher in return for performance as chip technology improves.

# ABSTRAK

Pada masa kini, teknologi peranti komunikasi elektronik cenderung kepada bentuk saiz yang lebih kecil, lebih ringan, lebih rendah dalam kos dan prestasi yang lebih tinggi. Individu mungkin cenderung menggunakan peranti komunikasi elektronik apabila bertukar maklumat yang sensitif, seperti butiran peribadi, dokumen kontrak, rahsia syarikat dan kata laluan, khusus dihantar kepada pihak lain. Memandangkan internet adalah salah satu hubungan utama penting dan berkomunikasi secara elektronik dengan berbilion orang, perlindungan untuk penghantaran mesej penting melalui internet adalah penting. Penyulitan memainkan peranan penting bagi setiap pengguna dalam memastikan keselamatan komunikasi dalam organisasi. Oleh itu, algoritma diperlukan untuk komunikasi perlu selamat. Motivasi projek ini adalah untuk melindungi data digital dalam kerahsiaan komputer, kerana ia sering disimpan pada sistem komputer dan diedarkan melalui internet atau rangkaian komputer lain. Algoritma Rivest-Shamir-Adleman diperkenalkan oleh Ron Rivest, Adi Shamir dan Leonard Adelman pada tahun 1977, dan ia dikenali sebagai salah satu algoritma kriptografi kunci awam yang terkenal kerana ia adalah kriptografi asimetrik. Selain itu, teori di sebalik RSA adalah agak mudah dan mudah untuk tujuan pengubahsuaian kerana ia bergantung pada algoritma seperti pemfaktoran dan eksponensial modular. Dalam makalah ini, semua modul dalam algoritma RSA telah diterangkan untuk saiz utama 256 bit. Disebabkan kepanjangan bit modulus, kerja itu termasuk pelaksanaan yang berbeza tetapi sesuai, iaitu pelaksanaan asas, radix-4 dan radix-16 untuk mengurangkan kelajuan proses cipher-decipher.Pelaksanaan Verilog HDL menggunakan perisian Quartus II telah dilakukan. Peningkatan kelajuan adalah kekangan tujuan utama untuk projek ini. Mengikut keputusan sintesis, radix-16 Montgomery Multiplier yang dilaksanakan dalam cipher RSA boleh dilaksanakan dengan pengurangan hampir 60% dalam kependaman penyulitan. Ia boleh difikirkan untuk menyerap peningkatan dalam kiraan pintu dalam cipher RSA sebagai balasan untuk prestasi apabila teknologi cip bertambah baik.

**TABLE OF CONTENTS**

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| RSA | - | Rivest, Shamir, and Adleman |
| HDL | - | Hardware Description Language |
| CPU | - | Central Processing Unit |
| GPU | - | Graphic Processing Unit |
| MM | - | Montgomery Modular Multiplication |
| DC | - | Design Compiler |
| FPGA | - | Field-Programmable Gate Array |
| CPLD | - | Complex Programmable Logic Device |
| LVT | - | Low V Threshold |
| HVT | - | High V Threshold |

## LIST OF SYMBOLS

| | | |
|---|---|---|
| mod | - | Modulus |
| gcd | - | Greatest Common Divider |

# CHAPTER 1

# INTRODUCTION

## 1.1 Problem Background

Today, internet is one of the most important outlets of contact and information for human beings. Thousands of people communicate electronically with each other [1]. Besides, information technology is also increasing lively throughout these days. Unfortunately, internet can no longer guarantee the provision of secure information. Although the internet was built to be durable and effective, it was also not to be inherently safe. There are different types of search engines continue to grow along with mushroomed viruses, bugs, spam, and hackers which can easily steal confidential data [2]. The important aspects for delivery and storage of data and information are security and confidentiality problem. To ensure information is safe and complete, human need to think of ways to provide strong protection for information in the virtual world [3].

To deliver secure services, many technologies depend on public cryptography. The essential operations in processing many types of public-key cryptosystems are modular multiplication and modular division with a lengthy modulus. One of the most extensively used public key algorithms today is Rivest, Shamir, and Adleman (RSA). The computation of modular exponentiation in RSA needs repeated modular multiplications. Thus, computation time for this algorithm is extremely large and not practical to be used since data transmission often need a high speed. Besides, when future technology system requirements and real-time computing speed are considered in wireless communications and personal communications systems, speed improvement will be getting more critical.

Since the division process is time-consuming in modular reduction, Montgomery devised a novel approach that avoids division. The Montgomery multiplication

algorithm is a well-known method of implementing a modular multiplication architecture (MM). It's a time-saving approach for modular multiplication with any modulus. Instead of divisions, which are utilised in a traditional modular operation, the method employs simple multiplications by a power of two. However, because these Montgomery designs are frequently sophisticated, it's not always clear whether they offer the required speed. Thus, it is worth to research and implement more on this new modular multiplication architecture.

On the other hand, the apparent challenge in factoring huge semi-primes is the foundation for RSA cryptography. Factoring two primes is used as the reverse of multiplication, and it becomes complicated when the values of the two prime numbers grow larger. Consider the RSA Factoring Challenge, which was established by RSA Laboratories in 1991. There are still many moduli that need to be factored. On December 12, 2009, a total of 13 researchers calculated a 768 bits RSA modulus (232 decimal digit number) over the course of two years, by utilising hundreds of simultaneous computers, a work equivalent to about 2000 years of computation on a single-core 2.2 GHz AMD CPU. It is proved that the longer the modulus' key length, the longer it takes to factorise. Since the algorithm strength is depends on the key length, it seems to be important to discover a more efficient factoring algorithms and advances in cryptanalysis techniques.

## 1.2 Problem Statement

To maintain the confidentiality of digital data, an encryption key is used, since it is often stored on computer systems and distributed via the internet or other computer networks. Asymmetric cryptography allows the use of public key during encryption and private key during decryption. Security is one of the main concerns to protect data in a complicated internet system. Due to limited battery power, security requirements are becoming increasingly crucial for private data transfer through mobile devices with internet access. It is vital to create efficient hardware architectures for applications that required energy-efficient cryptosystem, to perform quick modular multiplications but consume low energy.

Besides, RSA algorithm is a high secure cryptosystem used for data transmission. But, for real-time applications such as video processing, using RSA algorithm to perform encryption or decryption include a lot of calculation and the speed is far too slow [4]. As a result of the developed wide range of decomposition techniques, key length will increase to assure safety, resulting in increased computation. In addition, it also requires processing a big number of modular multiplications repeatedly. Therefore, optimized hardware implementation is required to provide low delay RSA performance by using fast modular multiplications [5].

## 1.3 Hypothesis

Montgomery algorithm is a feasible option for modular multiplication and exponentiation to attain speed improvement or area reduction in an asymmetric cryptosystem. We believe that if we focus on improving the Montgomery algorithm through the hardware architecture, a high throughput and low latency RSA encryption/decryption architecture can be achieved after some practical implementations.

## 1.4 Objectives

The objectives of the research are:

(a) To design a fast and area efficient architecture for Montgomery Modular Exponentiation and integrate the design in RSA algorithm. Besides, it able to validate that the algorithm used in cipher is correct and encrypted data is match with expected result.

(b) To verify the functionality and synthesis the RTL design of the RSA algorithm using proper synthesis tools.

(c) To analyse the performance matric in terms of area, latency, power and throughput for Radix n-th implementation regard to normal implementation.

## 1.5 Scope

The aim of the project is to design and improve the critical constraint of an encryption cipher unit using Hardware Description Language. Meanwhile, functionality of encryption and decryption is still the most essential part for an encryption cipher. The effect of performance due to different techniques of design in algorithms is vary. Therefore, a suitable modification technique should be chosen to use as to either improve on its power, performance, and area. Detailed scopes are elaborated to obtain accurate result and compare with the existing results.

To perform this project, the first step is to implement a basic RSA algorithm using RTL design for baseline comparison purpose. Then, design a High-Radix Montgomery Modular Multiplier to perform the modular exponentiation in this algorithm. The design is required to be compiled successful and waveform is simulated to verify the algorithm's functionality. By using Synopsys VCS tool, SAIF file is also generated to perform power analysis. The design then can be used to perform synthesis by using Synopsys 32nm Generic Library in Synopsys Design Compiler. The original basic algorithm's performance should be compared with the basic design after implementation, in term of latency, throughput, power consumption, and area. Since performance of area and power are not in the scope of work, only latency and throughput will be optimized for this algorithm.

Besides, the design will first be simulated using three numbers of bit length as input, which are 256 bits, 1024 bits and 2048 bits. This is due to the security claim and security level are typically expressed in bits. The lengthy the key, the complicated the process of cracking. Due to the number of bits used as key, the radix value that is suitable to be implemented in Montgomery Multiplier will need to be dividable with the key length with no remainder. Thus, a basic, radix-4 and radix-16 will be chosen to be implemented, since the higher the radix value, the bigger the lookup table used.

## 1.6 Report Structure

The structure of this project report is organized as follows: Chapter 1 will cover the introduction to the problem, a brief background on the main motivation of tackling this topic, the problem that has been captured, the hypothesis that has been concluded and what's planned to be achieved in this study. In Chapter 2, the background of an encryption system, details of algorithm in RSA encryption/decryption system and a comprehensive critical review on the popular techniques have been stated in detail. Chapter 3 presents the methodology of the project is presented in this chapter which including the general methodology, design overview as well as the software tool that applied to the project. Chapter 4 provides the overall result of the whole project work. It concentrates the analysis of the results obtained for the project using Vivado Design Suite and Design Compiler. It also interprets the findings along with discussion related to critical path delay, latency, area, and power. Lastly, Chapter 5 summarizes the project findings by fulfilling all the requirements based on the project objectives. In addition, it also includes some useful recommendations to make further improvement to the design architecture.

## REFERENCES


[1] Misha Kay, J. Santos, and M. Takane, "Global Observatory for EHealth: Safety and security on the Internet Challenges and advances in Member States," *Glob. Obs. eHealth Ser.*, vol. 4, p. 92, 2011.

[2] I. T. U. Apnic and M. Ipv, "Internet Security Introduction," no. May, 2016.

[3] T. Tietoturvallisuuden, "Introduction to Information Security Broadcast encryption," *Network*, pp. 1–26, 2010.

[4] N. Nedjah and L. de M. Mourelle, "Parallel computation of modular exponentiation for fast cryptography," *Int. J. High Perform. Syst. Archit.*, vol. 1, no. 1, pp. 44–49, 2007, doi: 10.1504/IJHPSA.2007.013290.

[5] E. Öksüzoğlu and E. Savaş, "Parametric, secure and compact implementation of RSA on FPGA," *Proc. - 2008 Int. Conf. Reconfigurable Comput. FPGAs, ReConFig 2008*, pp. 391–396, 2008, doi: 10.1109/RECONFIG.2008.13.

[6] "4. Cryptography and the Web - Web Security, Privacy & Commerce, 2nd Edition [Book]." https://www.oreilly.com/library/view/web-security-privacy/0596000456/ch04.html (accessed Dec. 06, 2019).

[7] R. B. Lee, A. M. Fiskiran, and R. B. Lee, "Performance Scaling of Cryptography Operations in Servers and Mobile Clients," no. May, 2014.

[8] C. F. Kerry and P. D. Gallagher, "Digital Signature Standard (DSS)," Jul. 2013, doi: 10.6028/NIST.FIPS.186-4.

[9] H. P. Ashok and G. U. Kharat, "Parallel Artificial Bee Colony Optimisation for Solving Curricula Time-Tabling Problem," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2016, no. 1, pp. 1–8, 2016, doi: 10.15680/IJIRCCE.2016.

[10] S. Process *et al.*, "International Journal of Scientific Research," vol. I, no. 2, pp. 96–106, 2016.

[11] B. Rothke, "A look at the Advanced Encryption Standard (AES)," *Inf. Secur. Manag. Handbook, Sixth Ed.*, pp. 1151–1158, 2007, doi: 10.1201/9781439833032.ch89.

[12] "A Brief History of Encryption | Security | TechNewsWorld." https://www.technewsworld.com/story/70437.html (accessed Nov. 29, 2019).

[13] K. Cramer, W. Winfree, and K. Hodges, “Proceedings of SPIE ‘Thermosense-XXVIII,’” vol. 6205, no. November, p. 62051B1, 2006, doi: 10.1117/12.2538414.

[14] E. Barker, A. Roginsky, G. Locke, and P. Gallagher, “Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths,” *NIST Spec. Publ.*, no. January, pp. 800–131, 2011.

[15] J. Nechvatal *et al.*, “Report on the development of the Advanced Encryption Standard (AES),” *J. Res. Natl. Inst. Stand. Technol.*, vol. 106, no. 3, pp. 511–577, 2001, doi: 10.6028/jres.106.023.

[16] J. Nechvatal, E. Barker, D. Dodson, M. Dworkin, J. Foti, and E. Roback, “Status report on the first round of the development of the advanced encryption standard,” *J. Res. Natl. Inst. Stand. Technol.*, vol. 104, no. 5, pp. 435–459, 1999, doi: 10.6028/jres.104.027.

[17] A. Shamir, “New directions in croptography,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2162, p. 159, 2001, doi: 10.1007/3-540-44709-1_14.

[18] Shieny, “Known Plaintext Attack History of the Known Plaintext Attack Breaking the Enigma Code How Good are Classic Ciphers ?,” 2008.

[19] “RSA Algorithm | Learn List of Possible Attacks on RSA Algorithm.” https://www.educba.com/rsa-algorithm/ (accessed Feb. 06, 2022).

[20] I. K. Salah, A. Darwish, and S. Oqeili, “Mathematical Attacks on RSA Cryptosystem Royal Scientific Society , Jordan Amman Arab University for Graduate Studies , ( sabbatical leave from Al Balqa Applied University ),” *J. Comput. Sci.*, vol. 2, no. 8, pp. 665–671, 2006.

[21] “Security of RSA - GeeksforGeeks.” https://www.geeksforgeeks.org/security-of-rsa/ (accessed Feb. 06, 2022).

[22] M. Domb, “Distributed Modular Multiplication to Be Processed by a Network of Limited Resources Devices,” *Advances in Intelligent Systems and Computing*, vol. 1183. pp. 104–109, 2021, doi: 10.1007/978-981-15-5856-6_9.

[23] A. Razaque, W. Jinrui, W. Zancheng, Q. B. Hani, M. A. Khaskheli, and W. A. Bhutto, “Integration of CPU and GPU to accelerate RSA modular exponentiation operation,” *2018 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2018*, pp. 1–6, 2018, doi: 10.1109/LISAT.2018.8378036.

[24] E. Ochoa-Jimenez, L. Rivera-Zamarripa, N. Cruz-Cortes, and F. Rodriguez-

Henriquez, “Implementation of RSA Signatures on GPU and CPU Architectures,” *IEEE Access*, vol. 8, pp. 9928–9941, 2020, doi: 10.1109/ACCESS.2019.2963826.

[25] Q. Hu, M. Duan, Z. Yang, S. Yu, and B. Xiao, “Efficient Parallel Secure Outsourcing of Modular Exponentiation to Cloud for IoT Applications,” *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12782–12791, 2021, doi: 10.1109/JIOT.2020.3029030.

[26] P. Lara, F. Borges, R. Portugal, and N. Nedjah, “Parallel modular exponentiation using load balancing without precomputation,” *J. Comput. Syst. Sci.*, vol. 78, no. 2, pp. 575–582, 2012, doi: 10.1016/j.jcss.2011.07.002.

[27] A. Bhattacharjya, X. Zhong, and X. Li, “A lightweight and efficient secure hybrid rsa (shrsa) messaging scheme with four-layered authentication stack,” *IEEE Access*, vol. 7, no. 5, pp. 30487–30506, 2019, doi: 10.1109/ACCESS.2019.2900300.

[28] R. Abid *et al.*, “An optimised homomorphic CRT-RSA algorithm for secure and efficient communication,” *Pers. Ubiquitous Comput.*, 2021, doi: 10.1007/s00779-021-01607-3.

[29] K. Venkata Reddy, C. Simranjeet Singh, V. Desalphine, and D. Selvakumar, “A Low Latency Montgomery Modular Exponentiation,” *Procedia Comput. Sci.*, vol. 171, pp. 800–809, 2020, doi: 10.1016/j.procs.2020.04.087.

[30] G. Sassaw, C. J. Jiménez, and M. Valencia, “High radix implementation of Montgomery multipliers with CSA,” *Proc. Int. Conf. Microelectron. ICM*, pp. 315–318, 2010, doi: 10.1109/ICM.2010.5696148.

[31] S. D. Thabah, M. Sonowal, R. U. Ahmed, and P. Saha, “Fast and Area Efficient Implementation of RSA Algorithm,” *Procedia Comput. Sci.*, vol. 165, no. 2019, pp. 525–531, 2019, doi: 10.1016/j.procs.2020.01.024.

[32] I. Of, “روطت مُ ةيسمش ة فحال مُ ماظن ذيفنتو ميمصت Design and Implementation of,” vol. 67, no. 6, pp. 0–70, 2020.

[33] A. F. Tenca and Ç. K. Koç, “A scalable architecture for modular multiplication based on Montgomery’s algorithm,” *IEEE Trans. Comput.*, vol. 52, no. 9, pp. 1216–1222, 2003, doi: 10.1109/TC.2003.1228516.

[34] M. Knežević, F. Vercauteren, and I. Verbauwhede, “Faster interleaved modular multiplication based on Barrett and Montgomery reduction methods,” *IEEE Trans. Comput.*, vol. 59, no. 12, pp. 1715–1721, 2010, doi:

10.1109/TC.2010.93.

[35] Synopsys, “Design Compiler User Guide,” no. December, pp. 1–465, 2010, [Online]. Available: http://acms.ucsd.edu/_files/dcug.pdf%5Cnpapers2://publication/uuid/479A4C81-9E9B-4EDE-8CD5-AB9F12130421.

[36] P. L. Montgomery, “Modular multiplication without trial division,” *undefined*, vol. 44, no. 170, pp. 519–521, 1985, doi: 10.1090/S0025-5718-1985-0777282-X.

[37] A. Nadjia, A. Mohamed, and I. Mohamed, “Montgomery modular exponentiation on FPGA,” *Proc. Int. Conf. Microelectron. ICM*, 2012, doi: 10.1109/ICM.2012.6471439.

[38] S. Yeşil, A. N. Ismailoğlu, Y. C. Tekmen, and M. Aşkar, “Two fast RSA implementations using high-radix montgomery algorithm,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2, 2004, doi: 10.1109/ISCAS.2004.1329332.