A HARDWARE IMPLEMENTATION OF RIVEST-SHAMIR-ADLEMAN CO-PROCESSOR FOR RESOURCE CONSTRAINED EMBEDDED SYSTEMS

ARUL A/L PANIANDI

UNIVERSITI TEKNOLOGI MALAYSIA

A HARDWARE IMPLEMENTATION OF RIVEST-SHAMIR-ADLEMAN CO-PROCESSOR FOR RESOURCE CONSTRAINED EMBEDDED SYSTEMS

ARUL A/L PANIANDI

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MAY 2006

Specially dedicated to
my beloved family

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my deepest gratitude is to Professor Dr. Mohamed Khalil bin Haji Mohd Hani for giving me the opportunity to explore new grounds in the computer-aided design of electronic systems without getting lost in the process. His constant encouragement, support and guidance were key to bringing this project to a fruitful completion. I have learnt and gained much in my two years with him, not only in the field of research, but also in the lessons of life.

My sincerest appreciation goes out to all those who have contributed directly and indirectly to the completion of this research and thesis. Of particular mention are lecturer Encik Nasir Shaikh Husin for for his sincere guidance and the VLSI-ECAD lab technicians, En. Zulkffli bin Che Embong and En. Khomarudden bin Mohd Khair Juhari, in creating a conducive learning and research environment in the lab.

Many thanks are due to past and present members of our research group at VLSI-ECAD lab. I am especially thankful to my colleagues Avinash, Kie Woon, Hau Yuan Wen, Kwee Siong, Izzeldin, Illiasaak, Heng San, Shikin and Chew for providing a supportive and productive environment during the course of my stay at UTM. At the same time, the constant encouragement and camaraderie shared between all my friends in campus made life in UTM an enriching experience.

Finally, I would like to express my love and appreciation to my family who have shown unrelenting care and support throughout this challenging endevour.

ABSTRACT

The concern with security problems has been rapidly increasing as computers and Internet services become a more pervasive part of our daily life. This need is further fueled by the advent of mobile electronic devices like smart cards, mobile phones and hardware tokens. Public key cryptographic systems such as RSA (Rivest-Shamir-Adleman) are vital in providing this security in terms of authentication, private key exchange, and digital signatures. Unfortunately, current RSA implementations are either resource exhaustive or too slow. In this thesis, a fast and configurable hardware implementation of the RSA algorithm for public key cryptography is presented that addresses the issues above. The designed RSA co-processor core is actually a modular exponentiation hardware engine, which is the basic arithmetic operation in implementing a RSA public key encryption and decryption algorithm. The computation intensive modular multiplication operation is based on the Montgomery's algorithm and implemented using systolic array architecture. The modules in the RSA co-processor are modeled using VHDL hardware description language before being integrated with Altera's softcore general-purpose processor, Nios II, and standard peripherals to form a complete cryptosystem in SoPC environment. Embedded C language codes are then written to test the functionality of the RSA co-processor on hardware. Upon verification, a demonstration application prototype that performs RSA encryption and decryption is developed using Visual Basic 6.0. This RSA co-processor core is able to encrypt and decrypt data with variable key lengths up to 4096 bits. The 1024 bit implementation uses 7000 Logic Elements (LE) on the Altera Stratix EP1S40-F780C5 FPGA development board which roughly translates to 49,000 gates. Encryption takes 2 ms while decryption takes 79 ms with the clock frequency of 40MHz. The speed and area constraint achieved is comparable and even better than several other research and commercial implementations.

ABSTRAK

Penggunaan computer dan Internet yang semakin meluas menyebabkan perhatian yang diberikan terhadap isu keselamatannya kian meningkat. Keperluaan ini semakin terasa dengan penggunaan alatan elektronik mudah alih seperti kad pintar, telefon bimbit dan token perkakasan. Sistem kriptografi kunci-awam seperti RSA (Rivest-Shamir-Adleman) amat penting dalam menyediakan sekuriti ini daripada aspek autentikasi, pertukaran kunci-persendirian dan tandatangan digital. Malangnya, implementasi RSA yang sedia ada sama ada terlalu besar atau terlalu perlahan. Dalam tesis ini, suatu implementasi teras kripto berdasarkan algoritma RSA, yang laju dan mudah diubah-suai, dicadangkan bagi menangani isu di atas untuk sistem kriptografi kunci-awam. Teras RSA yang direka ini sebenarnya adalah perkakasan ekponensasi modular, yang merupakan operasi aritmetik asas dalam enkripsi dan dekripsi yang dinyatakan oleh algoritma RSA. Operasi pendaraban modular yang intensif komputasi adalah berdasarkan algoritma Montgomery dan dilaksanakan dengan senibina tatasusunan sistolik. Modul-modul dalam teras RSA ini dimodelkan menggunakan bahasa deskripsi perkakasan, VHDL, sebelum digabungkan dengan mikropemproses buatan Altera, Nios II dengan persistan langsung, untuk membentuk sistem kripto menerusi SoPC. Kod C terbenam kemudian ditulis untuk menguji kesahihan implementas iteras RSA atas perkakasan. Setelah terbukti sahih, sebuah prototaip aplikasi demonstrasi yang melaksanakan enkripsi dan dekripsi RSA dibangunkan menggunakan Visual Basic 6.0. Teras RSA ini boleh enkrip dan dekrip data dengan variasi panjang kunci sehingga 4096 bit. Implementasi 1024 bit menggunakan 7000 Elemen Logic (LE) perkakasan FPGA Altera Stratix EP1S40-F780C5, iaitu lebih kurang 49,000 get logik. Enkripsi mengambil masa lebih kurang 2 ms manakala dekripsi 79 ms dengan frekuensi 40 MHz. Kelajuan dan limitasi saiz yang dicapai adalah sama ada standing atau lebih baik berbanding implementasi penyelidikan atau komersial yang lain.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

AES	-	Advanced Encryption Standard
API	-	Application Programming Interface
ASIC	-	Application Sppecific Integrated Circuit
CAD	-	Computer Aided Design
CECG		Communications-Electronics Security Group
CPLD		Complex Programmable Logic Device
CPU	-	Central Processing Unit
CRT		Chinese Remainder Theorem
DES	-	Data Encryption Standard
DMA	-	Direct Memory Access
ECC	-	Elliptic Curve Cryptography
FPGA	-	Field Programmable Gate Array
GB		Gigabyte
GCD		Greatest Common Divisor
GCHQ		Government Communications Headquarters
GUI	-	Graphical User Interface
HDL		Hardware Development Language
IDE		Integrated Development Environment
I/O	-	Input/Output
IP	-	Intellectual Property
LE	-	Logic Element
LSB	-	Least Significant Bit
MHz	-	Megahertz
MIT		Massachusetts Institute of Technology
MSB	-	Most Significant Bit
NIST		National Institute of Standards and Technology

PC	-	Personal Computer
PDA		Personal Digital Assistant
PIO	-	Parallel Input Output
PKI	-	Public Key Infrastructure
PLD	-	Programmable Logic Device
PRNG	-	Pseudo Random Number Generator
RAM	-	Random Access Memory
RISC	-	Reduced Instruction Set Computer
RSA	-	Rivest-Shamir-Adleman
SDK	-	System Development Kit
SHA-1	-	Secure Hash Algorithm
SoC	-	System-on-Chip
SOPC	-	System-on-Programmable-Chip
UART	-	Universal Asynchronous Receiver Transmitter
USB	-	Universal Serial Bus
UTM		Universiti Teknologi Malaysia
VHDL	-	Very High Speed Integrated Circuit Hardware Description Language
VB	-	Visual Basic
VLSI	-	Very Large Scale Integration

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This thesis proposes the design and implementation of a RSA cryptographic co-processor on FPGA. The design applies the System-on-Chip (SoC) technology to produce a RSA cryptosystem that performs operations such as encryption, decryption and key generation. The aim is to produce a RSA co-processor that strikes a balance between speed and area so that it is both compact and fast enough for commercial implementation. This first chapter covers background of research, problem statement, research objectives, scope of work, significance and contribution of the research, and finally thesis organization.

1.9 Background

The use of mobile electronic devices like smart cards, wireless handsets, PDAs, PCs, and network equipment, are becoming more prevalent since the turn of the new millennium. Their various applications cover almost every aspect of human life, including some very important fields like commerce and person identification. These embedded systems are ubiquitously used to capture, store, manipulate, and exchange sensitive information over insecure mediums, and consequently, they are subject to increasing security concerns.

This concern can be addressed effectively by the application of crypto algorithms in these devices. Security mechanisms utilize crypto algorithms (public-key ciphers, symmetric encryption, hashing functions, etc.) as building blocks in a suitable scheme to achieve the desired security services. The fundamental security requirements include confidentiality, authentication, data integrity, and non-repudiation. To provide such security services, normally systems use public key cryptography. Among the various public key cryptography algorithms, the RSA cryptosystem [Rivest *et al*, 1978] is the best known and widely used public key crypto algorithm today. It is named after Ron Rivest, Adi Shamir and Len Adleman, who invented it in 1977.

Since RSA is the current de-facto public key crypto algorithm, numerous implementations of RSA have been done throughout the world. Two main approaches are pursued, which are software implementations and hardware implementations. Software solutions are slower in performance compared to hardware implementations since they are not dedicated to the RSA operation. To achieve optimal system performance while maintaining physical security, it is desirable to implement the RSA algorithm in hardware. Hardware implementations also can be made tamper-resistant and clone-free.

1.2 Problem Statement

Public key cryptosystems have proved to be essential in the security of electronic transactions especially with the sudden boom in electronic commerce and transmissions of secure personal data. Since their invention in 1976 by Whitfield Diffie and Martin Hellman [1976] to solve the key management problem in symmetric key cryptography, various public key cryptosystems such as RSA, El-Gamal and ECC, have been proposed. Public key cryptography can be used not only for privacy (encryption), but for authentication as well. Unfortunately, its drawback is that it performs much slower compared to symmetric key cryptography.

As the RSA algorithm provides high security and easy to implement, it quickly became the most widely used public key cryptosystem. Its advantage is that it is able to provide privacy, confidentiality and digital signatures using the same key pair, and based on the same mathematical operation. However, due to its underlying complex wide-operand modular arithmetic, the RSA operation requires a long computation time. Software implementations of RSA are about 100 times slower than DES while hardware implementations of RSA are about 1000 times slower than DES. (Schneier, 1996)

Due to increasing data rates and complexity of security protocols, software solutions are not sufficient to keep up with the computational demands of crypto processing. Thus, hardware implementation presents a viable solution to implement a RSA cryptosystem. Unfortunately, due to its underlying complex wide-operand modular arithmetic, the implementation of RSA in hardware poses a design challenge in itself. Coupled with the very fast speed requirement, the design challenge increases dramatically when we further add in the resource constraint issue of mobile electronic devices.

Although a plethora of RSA cryptosystems in hardware exists, most of them are tailored to high-speed applications thus do not display a suitable compromise between speed and utilized hardware resources. As hardware resources are cost critical factors in devices like smart cards and hardware tokens, current implementations of RSA cores are unsuitable for them.

Therefore, a compact yet reasonably fast RSA co-processor core is much needed to facilitate the upcoming of cryptographic functions in mobile devices. The RSA co-processor core design should be able to strike good a balance between speed and resource utilization. The design should also be parameterized so that it can be scaled up or down from the 1024 bits for either a more compact implementation with some compromise to the level of security, or a larger design with higher security.

This flexibility in design could not be provided by full custom and semi custom ASIC solutions. However, reconfigurable logic like FPGA and CPLD can provide this flexibility. In hardware implementation, the FPGA has become the chosen platform for any proof-of-concept design, before being committed to an ASIC (Application-Specific Integrated Circuit) or VLSI implementation. Other than that, FPGA also allows for rapid prototyping which makes them suitable for implementations of crypto hardware on embedded systems.

1.3 Objectives

From the discussion in the previous sections, the objectives of the work presented in this thesis are as follows:

1) To design and implement a 1024 bit RSA core which is able to perform RSA encryption and decryption within stipulated area and speed constraints. The design also has to be parameterizable so that it can be reconfigured for different key lengths.

2) To design an embedded RSA cryptosystem that integrates the RSA core with an embedded processor on a System-on-Programmable Chip (SoPC) platform.

3) To develop a prototype for demonstration of real-world RSA cryptography as a verification system in PC environment through the use of Graphical User Interface (GUI). A simple file encryption system is developed as the demonstration application prototype.

1.4 Scope of Work

Based on the outlined objectives above, available hardware and software resources, and the time frame allocated, this research project is narrowed down to the following scope of work.

1) As specified by the research objectives, a hardware implementation of 1024 bit RSA must consist of approximately 50,000 gates and must be able to perform the RSA encryption and decryption operation in less than 100 ms. Similarly, a 2048 bit RSA implementation must consist of approximately 100,000 gates and must be able to perform the RSA encryption and decryption operation in less than 400 ms. (MyMS, 2004)

2) The RSA co-processor, henceforth known as UTM-RSA_CoProcessor, is designed using VHDL. The design must be parameterizable so that the co-processor can be reconfigured to other key sizes, based on the security level and the hardware resources required by targeted applications.

3) The UTM-RSA_CoProcessor is integrated with the Nios II embedded processor to form the RSA Processor. The proposed RSA Processor is to fit into an Altera Stratix EP1S40F780C5 FPGA chip (which contains 41250 LEs (Logic Elements) or an equivalent of 14×10^6 system gates). The running frequency of the proposed cryptosystem with the RSA Processor is limited to 40 MHz.

4) The proposed RSA cryptosystem must be able to generate the RSA key pairs on chip, which means the RSA keys does not need to leave the embedded system. However, the issue of secure storage of the keys generated or used in the cryptosystem will not be addressed. (In actual applications like the Public Key Infrastructure, the public key is generated by a Certification Authority)

5) The test and validation methodologies are carried out to verify the functional operations of the RSA Processor. Cryptanalysis techniques to measure the security level of the embedded system will not be covered in this work.

6) A simple file encryption system is developed to validate the RSA cryptosystem. The current version is able to encrypt /decrypt a file limited to size of not more 4 GB. For a file larger than this size, the file needs to be chopped into multiple smaller files.

1.5 Research Strategies

The following research strategies have been applied during the course of research to ensure a complete and quality research is carried out.

1. The speed and area constraints are set based on the problems and stringent requirements demanded by industries in the commercial environment, which in turn increases the design challenge many times.

2. The established RSA algorithms are studied and the necessary algorithmic modifications (without changing the actual algorithm itself) are determined for efficient mapping of the algorithm onto hardware.

3. The designed RSA co-processor (UTM-RSA_CoProcessor) is integrated with a general-purpose embedded processor to obtain a complete RSA Processor on a System-on-Programmable Chip (SoPC) platform.

4. An application demonstration prototype is developed as the means to perform the RSA cryptosystem's verification on real-world test patterns.

1.6 Research Contribution and Project Delivery

1) A comprehensive design technique for design of an RSA core limited by computation speed and design area constraints for application in resource constrained embedded systems.

2) Design of a complete embedded RSA cryptosystem that incorporates a 32-bit RISC embedded general-purpose Nios II processor. Besides performing encryption and decryption, it also is able to perform on-chip RSA key generation.

3) An application demonstration prototype performing a real-world application that incorporates the UTM-RSA_CoProcessor and the Nios II processor to form the RSA Processor, and communicating with the standard PC to form the RSA Cryptosystem. Figure 1.1 below shows the system architecture of the proposed RSA cryptosystem.

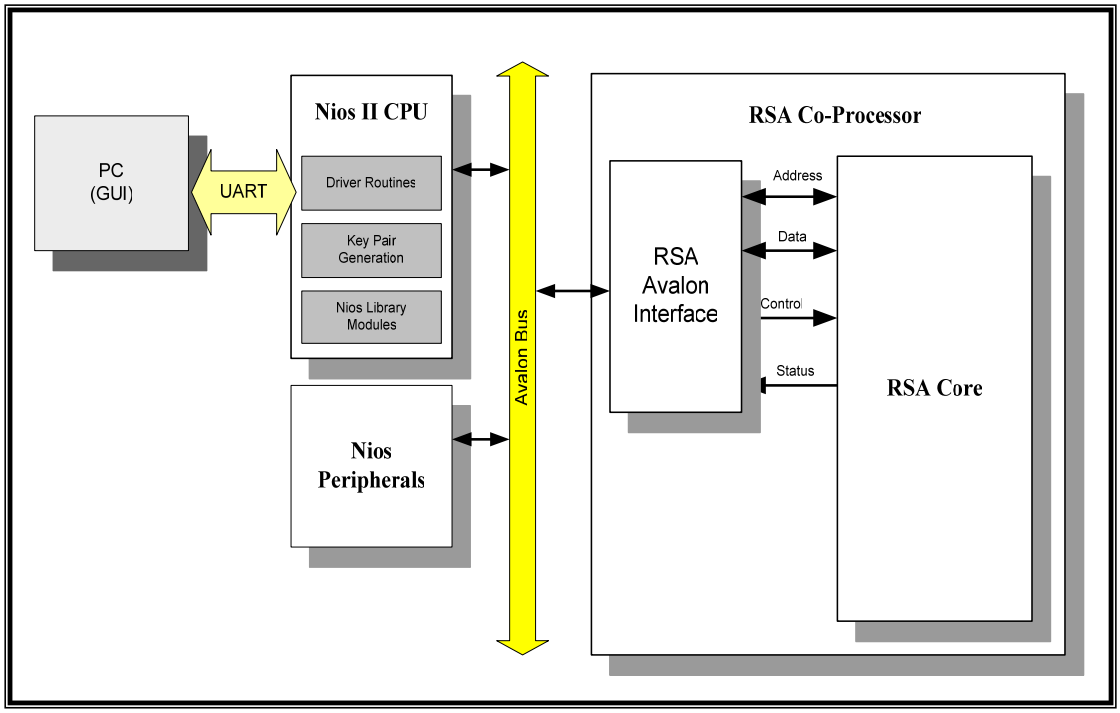

Figure 1.1 : System Architecture of Proposed RSA Cryptosystem

1.7 Thesis Organization

The work in this thesis is conveniently organized into eight chapters. The first chapter presents the motivation and research objectives and follows through with research scope and research contribution before concluding with thesis organization.

The second chapter provides brief summaries of the literature reviewed prior to engaging the mentioned scope of work. Several topics related to this research are reviewed to give an overall picture of the background knowledge involved. Summary of the literature review is given to clarify the research rationale.

Chapter three presents the design methodologies that are employed.

Chapter four focuses on the discussion of the implemented RSA algorithm, specifically the modular exponentiation and modular multiplication algorithms. This is followed by outlines of the necessary algorithmic modifications for better hardware implementation

Chapter five delivers the detailed description of the design of the RSA core based on the modified algorithms. First, a top-level view of the RSA cryptosystem is given before the design of each module is presented in both the top-down and bottom-up approach.

Chapter six explains the design of the RSA cryptosystem. First the design of the interface module for the RSA core is presented, followed by the development of the device drivers and embedded subroutines, the APIs and finally the RSA File Encryption Cryptosystem.

Chapter seven presents the tests that are carried out to verify the RSA cryptosystem. First, the hardware simulations of individual modules are presented. Then, this is followed by tests on the cryptosystem by using embedded software.

In the final chapter of the thesis, the research work is summarized and deliverables of the research are stated. Suggestion for potential extensions and improvements to the design is also given.

1.8 Summary

In this chapter, an introduction was given on the background and motivation of the project. The need for a compact yet fast, hardware implementation of RSA algorithm is pointed out. Based on those, several objectives were identified and scope of project was set to achieve the desired implementation. The UTM-RSA_CoProcessor was proposed to perform RSA computations on resource constrained embedded systems. The following chapter will discuss the literature relevant to the research and look into some previous work accomplished on the design of RSA hardware.

REFERENCES

Altera Corporation. (2003a). "*Nios Hardware Development Tutorial*". Altera Corporation.

Altera Corporation. (2003b). "*Introduction to Quartus II*". Altera Corporation.

ALTERA CORPORATION. (2003C). "*SOPC BUILDER DATA SHEET*". ALTERA CORPORATION.

Altera Corporation. (2004a). "*Nios II Hardware Development Tutorial*". Altera Corporation.

Altera Corporation. (2004b). "*Nios II Processor Reference Handbook*". Altera Corporation.

Altera Corporation. (2004c). "*Upgrading Nios Processor Systems to Nios II Processor"*. Altera Corporation.

Bajard J, Didier L, and Kornerup P, (1998). An RNS Montgomery Modular Multiplication Algorithm. *IEEE Transactions on Computers*, 47(7):766–76.

Bajard J and Imbert L, (2004). A Full RNS Implementation of RSA. *IEEE Transactions On Computers*. Vol. 53, No. 6.

Blakley. G.R, (1983). A Computer Algorithm for the Product AB Modulo M. *IEEE Transactions on Computer*. 32(5) : 407-500

Blum.T & Paar. C, (1999). Montgomery Modular Exponentiation on Reconfigurable Hardware. *14th IEEE Symposium on Computer Arithmetic*. 70-77.

Blum.T & Paar. C, (2001). High-Radix Montgomery Modular Exponentiation on Reconfigurable Hardware. *IEEE Transactions on Computers*. Vol. 50(7). Pg: 759-764.

Brickell. E. F, (1983). A Fast Modular Multiplication Algorithm With Application To Two Key Cryptography. *Advances in Cryptology, Proceedings CRYPTO'82*. Plenum, pp. 51-60.

Certicom Corporation. (1998). "*The Elliptic Curve Cryptosystem for Smart Cards*." Certicom Research.

Certicom Corporation. (2000a). *The Elliptic Curve Cryptosystem: Current Public-Key Cryptographic Systems*. Certicom Research.

Ciet M, Neve M, Peeters E and Quisquater J, (2003). Parallel FPGA Implementation of RSA with Residue Number Systems. *46th IEEE Midwest Symposium on Circuits and Systems*. Egypt.

Cocks. C, (1973). A Note on Non-Secret Encryption. *CESG Research Report*

Daly A and Marnane W, (2002). Efficient Architectures for implementing Montgomery Modular Multiplication and RSA Modular Exponentiation on Reconfigurable Logic. *FPGA'02,* February 24-26. Monterey, California, USA,.

Diffie, W., and Hellman, M.(1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, November.

DigitalCoreDesign, (2004). *DFPADD, DFP2INT, DMAC, DFPSQRT, DR8051*. Bytorn, Poland. Note: http://www.digitalcoredesign.com.

Eldridge S.E and Walter C.D, (1993). Hardware implementation of Montgomery's modular multiplication algorithm. *IEEE Transactions on Computers*. Vol .42(6). Pg :693–699.

Fang Yingli and Gao Zhiqiang, (2001). A new RSA cryptosystem hardware implementation based on high-radix Montgomery's algorithm. *Proceedings. 4th Int. Conf. on*. Page(s):348 – 351

Franklin. M, Levine.L, Anthony. D and Brentrup. R, (2004). PKI: *A Technology Whose Time Has Come in Higher Education*. In: EDUCAUSE Review,

Gai. W and Chen. H, (1996). A Systolic Linear Array For Modular Multiplication. *2nd International Conference on ASIC*. Pages 171–174.

Grobschsdl J, (2000). The Chinese Remainder Theorem and its Application in a High-speed RSA Crypto Chip. *IEEE*.

Hau Y. W. (2005). "*An Embedded Cryptosystem Implementing Symmetric Cipher and Public-Key Crypto Algorithms in Hardware.*" Universiti Teknologi Malaysia: M.Sc.

Ireland D, (2001). *BigDigits Multiple-Precision Arithmetic Library Version 1.0*. D.I. Management Services Pty Limited.

Ishii. S, Ohyama. K and Yamanaka. K, (1994).A single-chip RSA processor implemented in a 0.5 μm rule gate array. *Proceedings of Seventh Annual IEEE Int. ASIC Conf. and Exhibit.* Page(s): 433 – 436.

Iwamura. K, Matsumoto. T, and Imai H, (1994). Montgomery Modular-Multiplication Method And Systolic Arrays Suitable For Modular Exponentiation. *Electronics and Communications*. March. Japan. Part 3, 77(3):40–51.

Keller. S.S, (2004). The RSA Validation System (RSAVS). *National Institute of Standards and Technology (NIST)*. USA.

Khalil. M and Tan S.L, (2000). FPGA Implementation of RSA Public-Key Cryptographic Coprocessor. *IEEE TENCON Proceedings*. Volume: 3.

Khalil M, Koay K H, (1999). VHDL Module Generator: A Rapid-prototyping Design Entry Tool for Digital ASICs. *Jurnal Teknologi UTM*, December. 31:45-61.

Knuth D.E, (1981). *The Art of Computer Programming*, Volume 2: Seminumerical Algorithms. 2nd edition, Reading, MA: Addison-Wesley.

Koblitz. N, (1987). Elliptic Curve Cryptosystems. *Mathematics of Computation*, Vol. 48, n.177, pp. 203-209

Koc.C.K and Hung. C.Y, (1991). Bit Level Systolic Arrays For Modular Multiplication. *Journal of VLSI Signal Processing*, Vol 3, pp. 215-223.

Koren. I, (1993). *Computer Arithmetic Algorithms*. Englewood Cliffs, New Jersey. Prentice-Hall Inc.

Kung. H.T, (1980). The Structure of Parallel Algorithm. *Advances in Computers.* Vol.19. Pg:65-112. Academic Press, Inc.

McIvor C, McLoone M and McCanny J.V, (2004). Modified Montgomery modular multiplication and RSA exponentiation techniques. *Computers and Digital Techniques, IEE Proc*. Volume 151. Issue 6, Page(s):402 – 408.

Menezes A.J, Oorschot P.C and Vanstone S.A (2001). "*Handbook of Applied Cryptography*", CRC Press

Miller V.S, (1986), "*Use of Elliptic Curves in Cryptography*", Advances in Cryptology-CRYPTO 85, Springer-Verlag, pp 417-428

Montgomery P.L, (1985). Modular Multiplication Without Trial Division. *Mathematics of Computation*. 44:512-521.

MyMS, (2004). A discussion held on the current demands of the commercial industry on the hardware implementation of the RSA Algorithm in resource constrained embedded systems. Date of discussion : 4 October 2004. Time of discussion : 9.00am to 12.00p.m. Involved parties : Prof. Dr. Mohamed Khalil (UTM), Arul Paniandi (UTM), Hau Yuan Wen (UTM), Lau Boon Leong (MyMS), Tee Kok Kim (MyMS), Mohammad Khir (MyMS). Place of discussion : Malaysia Mictrotronic Solutions (M) Sdn. Bhd, 2300 Century Square, 63000 Cyberjaya, Selangor.

Orup, H, (1995). Simplifying Quotient Determination In High-Radix Modular Multiplication. *Proceedings 12th Symposium on Computer Arithmetic*. Pages : 193–199.

Rivest R.L, Shamir.A, and Adleman.L. (1978). A Method for Obtaining Digital Signature and Public-Key Cryptosystems, *Comm. in ACM*, 21:120-126.

Schmeh, K. (2003). "*Cryptography and Public Key Infrastructure on the Internet*" John Wiley & Sons, Inc

Schneier, B. (1996). "*Applied Cryptography : Protocols, Algorithm and Source Code in C*." 2nd Edition. N.Y : John Wiley & Sons, Inc.

Shand. M and Vuillemin J, (1993). Fast implementations of RSA cryptography. *11th IEEE Symposium on Computer Arithmetic*, pages 252–259.

Stalling, W. (1999). "*Cryptography and Network Security: Principles and Practice"*. 2nd Ed. Upper Saddle River, New Jersey: Prentice Hall.

Steffen. A, (2000). Secure Communications in Distributed Embedded Systems. *Security Group of the Zurich Univ.of Applied Sciences*. Winterthur.

Takagi.N and Tagima.S, (1992). Modular Multiplication Hardware Algorithms With A Redundant Representation And Their Application To RSA Cryptosystem. *IEEE Transaction on Computers*. Vol 41, no 7, pp. 887-891.

Tsai. W.C, Shung. C.B, Wang. S.J, (2000). Two Systolic Architecture for Modular Multiplication. *IEEE Transactions on VLSI Systems*. Vol. 8(1). Pg : 102-107

Vuillemin. J.E, Bertin.P, Roncin. D, Shand.M, Touati H.H and Boucard. P, (1996). Programmable Active Memories : Reconfigurable Systems Come of Age. *IEEE Transactions on VLSI Systems*. Vol. 4(1). Pg : 56-59.

Walter C.D, (1993). Systolic Modular Multiplication. *IEEE Transactions on Computers*. 42(3):376–8.

Walter C.D, (1999). Montgomery Exponentiation Needs no Final Subtractions. *Electronics Letters*. 35(21):1831-1832.

Wang P.A, (1997). New VLSI Architectures Of RSA Public Key Cryptosystems. *Proceedings of 1997 IEEE International Symposium on Circuits and Systems*. Volume 3. Pages 2040–2043.

Win. E.D, Mister. S, Preneel. B, and Wiener. M, (1998). On the Performance of Signature Schemes Based on Elliptic Curves. *Algorithm Number Theory Symposium III*. Springer-Verlag. 252-266.

Wu C.H, Hong J.H, Wu.C.W, (2001). VLSI Design of RSA Cryptosystem Based on the Chinese Remainder Theorem. *Journal Of Information Science And Engineering.* Vol.17. Pg : 967-980.

Zhu Keija; Xu Ke; Wang Yang; Min Hao, (2003). A Novel ASIC Implementation of RSA Algorithm. *Proceedings. 5th Int. Conf. on*. 21-24 Oct. 2003. Volume 2. Page(s):1300 – 1303.