# Design and Verification of New Area Efficient RSA Algorithm: Review

**1Ritu, 2Shivam Solnki**
*2Asst. Prof., TITECH, Jabalpur*

***Abstract:*** **This paper, present a new structure to develop 64-bit RSA encryption engine on FPGA that can be used as a standard device in the secured communication system. The RSA algorithm has three parts i.e. key generation, encryption and decryption. The algorithm also requires random prime numbers so a primarily tester is also design to meet the needs of the algorithm. We use right-to-left-binary method for the exponent calculation. This reduces the number of cycles enhancing the performance of the system and reducing the area usage of the FPGA. These blocks are coded in Verilog and are synthesized and simulated in Xilinx 13.2 design suit.**

***Keywords:*** **RSA, Verilog, Cryptosystem, Decryption, Encryption, Implementation, Key Generation, Modular Exponentiation**

## I.INTRODUCTION

There has been a lot of work going on in the field of cryptography and in the recent years it has increased exponentially. As the usage of communication system increases so does the need for securing data over those channels. Many algorithms are designed to meet these needs. Cryptographic algorithms have two major types: symmetric and Asymmetric [1]. Symmetric cryptography requires sharing of a single key at both ends. The problem is the selection of the key privately. In asymmetric (Public key) cryptography this problem is overcome by using an algorithm that deals with two keys. One key is for encryption and the other one is to decrypt the same message. The idea of publishing one key (the public key) and keeping the other one secrete (the private key) can surely make the whole procedure more secure and protected. Only those will be able to read the massage who may also have the private key as well, it is necessary to have both keys if someone encrypt the massage [2]. RSA algorithm belongs to this type of cryptography. This problem is discussed in many ways [12] has provided the high speed RSA implementation of FPGA platforms, [13] showed the high speed RSA implementation of a public key block cipher-MQQ for FPGA platforms. [14] also provided the implementation of RSA algorithm on FPGA. In this paper much work has done by the [14] but here we are modifying the our proposed engine for 64 bits RSA encryption. Here we are extended the work given by [14], also other algorithms like LFSR, Miller Rabin, Extended Euclidean and Modular exponentiation have been successfully implemented by using the proposed technique of XILINX ISE 13.2.

As far as the significance of the RSA is concern it can be used as a tool for exchanging the secrete information such as massages and conversation by generating the keys and producing digital signatures. However, the complexity comes from calculating the prime factors of large numbers. This work implements the modular exponentiation operation by simple right-to-left-binary method, which helps to reduce the processing time.

## II.LITERATURE WORK

Rupali Verma et al [1] Modular multiplication determines the efficiency of RSA cryptosystem as modular multiplication is core operation in RSA. The efficiency of modular multiplication can be improved by algorithmic improvement. The long operands in Montgomery modular multiplication can be added with carry save adders. Implementations of carry save adders on FPGAs require more area. This paper presents the implementation results of RSA on FPGAs based on carry save Montgomery. Parallel computations decrease the path length and hence reduce the critical path delay of Montgomery modular multiplication. Carry save adders and compressors have large area requirement. This is due to the mapping of carry and sum logic to different Look up tables (LUTs) in a slice. Carry save based architectures are suitable for high throughput applications but may not be suitable for applications run on constrained devices. The Place and Route (Metric) results show that each CLB does not have 100% utilization of its resources. Floorplanning may improve the use of resources in CLBs and hence reduce the area results of the design. The Look up tables within a slice in FPGAs can be used for varied functionality like logic functions, shift registers etc. The synthesis software maps the code to LUTs. Existing FPGAs resources like BRAM (Block RAM), IP cores and embedded multipliers further reduce the area requirements of design. The carry save Montgomery architectures focus on reducing time at

 

the cost of area requirement. Efficiency of these architectures can be improved further by reducing the area. This can be done by utilizing the underlying FPGA architecture for design implementation. Hence the available FPGA resources can be harnessed to improve the efficiency. Efficient mapping techniques can be applied for mapping carry save architectures so that less LUTs are occupied and results in efficient designs with less area requirements.

Michael Bourg et al[2] The biometric encryption system is a significant addition in the areas of privacy, security and convenience among its users. The intent of this research is to propose an RSA based biometric encryption system which can be realized on field programmable gate arrays (FPGAs) using hardware software co-design methods. Due to the high number of hackers that stand to profit from sub-par security methods, the proposed design will serve as a high level of security. This implementation can be applied in many areas of life including but not limited to password replacement, building and equipment access, and payroll and timekeeping procedures.
The implementation of biometric encryption on FPGAs will help to revolutionize the security and privacy industry. This study proposes a biometric encryption system, wherein the RSA algorithm is implemented in Java. This Java method can be called by C program to allow this implementation to be used by FPGA for high speed processing. Furthermore, the ability to calculate very large numbers by breaking them into an RNS will make the process suitable for real-time requirements. The possibility of hardware-software co-design of the biometric encryption system enables the proposed method to provide an efficient solution for the security applications.

Ari Shawkat Tahir et al [3] RSA cryptographic algorithm used to encrypt and decrypt the messages to send it over the secure transmission channel like internet. The RSA algorithm is a secure, high quality, public key algorithm. In this paper, a new architecture and modeling has been proposed for RSA public key algorithm, the suggested system uses 1024-bit RSA encryption/decryption for restricted system. The system uses the multiply and square algorithm to perform modular operation. The design has been described by VHDL and simulated by using Xilinx ISE 12.2 tool. The architectures have been implemented on reconfigurable platforms FPGAs. Accomplishment when implemented on Xilinx_Spartan3 (device XC3S50, package PG208, speed -4) which confirms that the proposed architectures have minimum hardware resource, where only 29% of the chip resources are used for RSA algorithm design with realizable operating clock frequency of 68.573 MHz. In this paper, new an efficient architecture has been proposed to implement an optimized 1024-bit RSA encryption/decryption algorithm for restricted system using multiply and square algorithm to process the Modular exponential for encryption and decryption. The whole system is implemented using VHDL code targeting Spartan3 (device XC3S50, package PG208, speed -4) from Xilinx. The whole design is tested using Xilinx ISE Design Suite 12.2 tool. The system speed achieved is 68.573 MHz.

## III.PROBLEM STATEMENT

Problem with Michael Bourg et al[1] is that the Currently, the majority of biometric encryption methods used are either on a very small scale, or involve a third party for authorization and personnel confirmation. Our study shows that biometrics is the safest form of security and privacy available.

Problem with Rupali Verma et al [2] is that the Parallel computations decrease the path length and hence reduce the critical path delay of Montgomery modular multiplication. Carry save adders and compressors have large area requirement

Problem with Ari Shawkat Tahir et al [3] is that the restricted system using multiply and square algorithm to process the Modular exponential for encryption and decryption

## IV.METHODOLOGY

The design of the architectures was done using Very High Speed Integrated Circuit Hardware Description Language (VHDL) and the complete source codes for 32 to 1024 bit implementations of Fast Montgomery, Faster Montgomery and Optimized Interleaved multipliers are available in electronic form. For the implementation of the multipliers, a very structured approach was used which shows the hierarchical decomposition of the multipliers into sub modules The basic units of the architectures which comprises carry save adders, shift registers and registers were modeled as components which are independently functional.

 

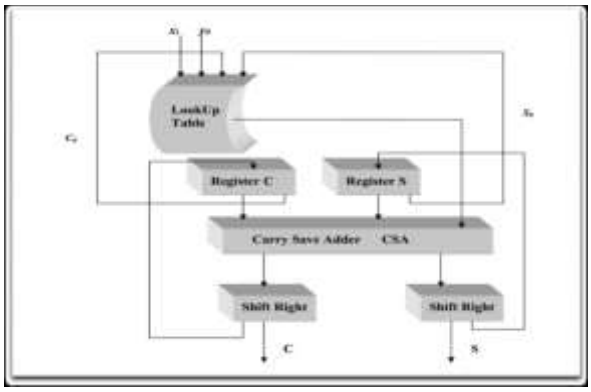


Fig. 1: Block diagram showing components that were implemented for the Faster Montgomery Architecture

## V.CONCLUSION

In this paper, study and problems of three different approaches for three different applications have been done and a new and efficient architecture is been planned for future work it will be an design of optimized 1024-bit RSA encryption/decryption algorithm for restricted system using multiply and square algorithm to process the Modular exponential for encryption and decryption, new The carry save Montgomery architectures focus on reducing time at the cost of area requirement. Efficiency of these architectures can be improved further by reducing the area. This can be done by utilizing the underlying FPGA architecture for design implementation. Hence the available FPGA resources can be harnessed to improve the efficiency.

## REFERENCES

[1] Rupali Verma, Maitreyee Dutta, Renu Vig, FPGA Implementation of RSA based on Carry Save Montgomery Modular Multiplication, 2016 International Conference on Computational Techniques in Information and Communication
Technologies(ICCTICT), DOI: 10.1109/ICCTICT.2016.7514561, ISBN: 978-1-5090-0082-1/16©2016 IEEE

[2] Michael Bourg, Pramod Govindan, RSA Based Biometric Encryption System Using FPGA for Increased Security, IEEE International Carnahan Conference on Security Technology (ICCST), 2016, DOI: 10.1109/CCST.2016.7815699, ISBN: 978-1-5090-1072-1/16©2016 IEEE, Orlando, FL, USA

[3] Ari Shawkat Tahir, Design and Implementation of RSA Algorithm using FPGA, INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol. 14, No. 12,

[4] Sushanta Kumar Sahu, Manoranjan Pradhan, FPGA Implementation of RSA Encryption System International Journal of Computer Applications (0975 – 8887) Volume 19– No.9, April 2011

[5] A.R.Landge1, A.H. Ansari, RSA algorithm realization on FPGA, ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 7, July 2013

[6] John Fry, Altera Corporation – Europe, Martin Langhammer, RSA & Public Key Cryptography in FPGAs, CF-032305-1.0, Copyrights atlera corporations

[7] Rehan Shams, Fozia Hanif Khan and Mohammad Umair, Cryptosystem an Implementation of RSA Using Verilog, International Journal of Computer Networks and Communications Security, VOL. 1, NO. 3, AUGUST 2013, 102–109 Available online at: www.ijcncs.org ISSN 2308-9830

[8] Vibhor G. , Aruna c. (2011).”Architectural analysis of RSA crypto system on FPGA “, International Journal of Computer Applications , Volume 26-No8.

[9] Chiranth E, Chakravarthy H.V.A, Nagamohanareddy P, Umesh T.H, Chethan Kumar M., (2011 ),“Implementation of RSA Cryptosystem Using Verilog”, International Journal of Scientific & Engineering Research, Volume 2, Issue 5, pp.1-7.

[10] Muhammad I. I., Mamun B.I. R., handaker A. and Sazzad H., (2007),“FPGA Implementation of RSA Encryption Engine with Flexible Key Size”, International journal of communication, Issue 3, Volume 1, pp. 107-113.

[11] Rokon I.R., Rahman M.,(2009),“Efficient hardware implementation of RSA cryptography”, 3rd International Conference on Anticounterfeiting, Security, and Ident

 