Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 34 (2014) 647 – 654

Procedia Computer Science

International Symposium on Emerging Inter-networks, Communication and Mobility (EICM 2014)

# Efficient FPGA Implementation of RSA Coprocessor Using Scalable Modules

Qasem Abu Al-Haija*, Mahmoud Smadi, Monther Al-Ja'fari and Abdullah Al-Shua'ibi

*King Faisal University, Department of Electrical Engineering, Al-Ahsa 31982, P.O. Box 380*

**Abstract**

RSA Cryptosystem is considered the first practicable secure algorithm that can be used to protect information during the communication. The significance of high security and efficient implementations of RSA have formed the base of many cryptographic engineering researches. In fact, the implementation of RSA Cryptosystem is heavily based on modular arithmetic and exponentiation involving large prime numbers. In this paper, we propose an efficient FPGA design and architecture for RSA cryptosystem using ALTERA FPGA Hardware Kit. The proposed design comprises six levels: random two prime numbers, parallel multiplication of the prime numbers and their decremented values, get encryption key, get decryption key, encryption and decryption levels. As the modular multiplication is considered as the heart of RSA computations, Interleaved Algorithm was particularly chosen as an efficient solution to speed up the modular multiplication. The experimental part of this work has been synthesized for both ALTERA Cyclone IV EP4CE115F29C7 and VERTIX VII VC707 FPGA kits and resulted in a maximum frequencies of 15.725 MHz, 17.629 MHz respectively. These findings make our design comparable and a good choice for efficient RSA Cryptoprocessor design. The results for the FPGA implementation for EC design using these curves is also proposed in this paper.


Selection and peer-review under responsibility of Conference Program Chairs

*Keywords:* Public Key Cryptography; RSA Algorithm; FPGA; ALTERA; Modular Arithmetic.

## 1. Introduction

Due to the rapid evolution of technology and its growth at a daily rate, the communication and information transfer has become so easy between sides. On the other hand, the world is in need of protecting information from intruders. As Internet expands exponentially, the need for secure communication and electronic commerce increase very quickly [1]. Therefore, to increase the privacy and information security from strangers, some techniques would

* Corresponding author. Tel.: +966-13-5895400; fax: +966-13-5871068.
*E-mail address: qalhaija@kfu.edu.sa;*


Selection and peer-review under responsibility of Conference Program Chairs
doi:10.1016/j.procs.2014.07.092

be used to insure security. Such technique is called cryptography. Accordingly, it provides some services such as confidentiality, integrity, and authentication.

However, cryptography plays a big rule in electronic security communications which nowadays become one of the noticeable contemporary issues. For instance, the banks, governments, organizations, and companies are extremely in needs of securing their database. Nevertheless, RSA cryptography is a public key cryptosystem that is one of the most effective and functional schemes in information security [2]. Generally, RSA is a type of public key cryptography. Both its encryption and decryption rely on public algorithms. Since everyone can encrypt a message, only authorized users have the ability to decipher the message.

This paper proposes an efficient architecture and design for RSA Cryptoprocessor by using different scalable modules such: the random number generator and the primality tester as well as modular multiplication and inversion.

### *1.1. RSA Algorithm - Revisited*

RSA Cryptography is a well-known example of public key cryptographic algorithms with robust encryption/decryption processes. RSA algorithm can be defined as follows:

Pick two large distinct prime numbers (p) and (q).

$$n = (p \times q)$$

$$\varphi(n) = (p-1) \times (q-1)$$

Where $\varphi(n) < n$

After that, pick up a number for the encryption key (e) which also called the public key and the decryption key (d) which also called the private key, where e is computed by generating random positive integers which must be co-prime and d is the inverse of e mod m as below:

$$1 < e < \varphi(n);$$

Where $\gcd(\varphi(n), e) = 1$

$$d = e^{-1} \bmod \varphi(n);$$

Where $(e \times d) \bmod \varphi(n) = 1$

$$E = D^e \bmod n$$

$$D = E^d \bmod n$$

### *1.2. Research Methodology*

The proposed methodology throughout this work consists of the following steps:

- The approach for proceeding in the proposed solution started by studying the several cryptographic algorithms and number theories that can be applied for RSA and understanding the parameters in each individual algorithm.
- A mathematical model has been developed to compute RSA operations separately in which they will be connected through the sophisticated system bus based on the modular reduction system.
- The mathematical solution of the proposed system has been parallelized efficiently using the well-known methods of parallelism and pipelined operations such as parallel multipliers.
- The solution was converted to hardware modules and described using the hardware description language – VHDL [3]. This phase is considered as the core phase of this work where the VHDL code will be applied to the FPGA ALTERA Kit [4] as a next stage of this step.
- The mathematical solution was verified using the mathematical software application- Maple Worksheets for Cryptography (Maple soft) [5] and the hardware implementation were verified by using the Synthesizer - Xilinx ISE Foundation [6].

## 2. Proposed RSA Coprocessor Design

The complete system design is shown in figure 1, Proposed Hardware Design Schematic Diagram of RSA. Also, Top Level Block Diagrams of RSA (Key Generator, Encryption, and Decryption) are shown in figure 2 respectively. The main modules are as follows:

- Three modules of the Random Number Generator (Random-p, Random-q, RNG-e).
- Two modules of the Primality Testing (P.T.).
- Two decrementer units (-1).
- Two Sequential Multipliers (Multiplier) Based Cary Save Adder (CSA) [2].
- Greatest Common Divider module using Euclidean algorithm (GCD).
- Inversion Unit using Extended Euclidean Algorithm (Modular Inverse).

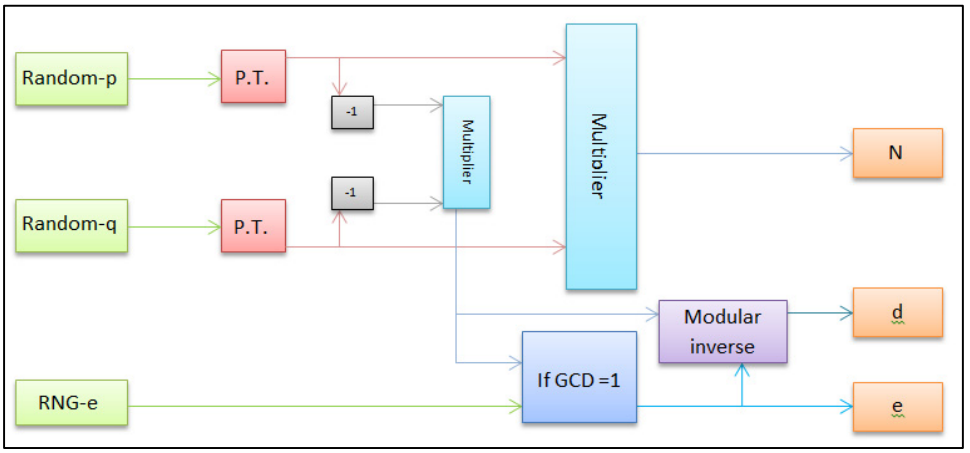


Figure 1: Proposed Hardware Design Schematic Diagram of RSA

Also, the proposed design was divided into five main levels of execution.

- Firstly, RSA Algorithm starts with finding two large prime numbers P and Q. This was accomplished in one level called level1 where we started with the random number generators. These two random numbers are assigned as the input of two components of primality testing working in parallel. That is, each random number will carry out the process separately.
- Now, we have got P and Q as random numbers. Then from the RSA Algorithm, we should calculate the modulus 'N' as the multiplication of P by Q, and 'phi' as the decremented values of P by Q. Therefore, we considered these operations as they work in parallel as one level called level 2. 'N' and phi are now ready and were treated independently.
- After that, we should find the encryption key 'e' such that the GCD between it and phi is equal to one which means that phi and e are co-prime. We have done this in one level as it cannot be treated in parallel with any other operations and is named as level 3.
- Now, we have the encryption key, and we can find the cipher text as the modular exponentiation of the message to the power the encryption key module N, defined as the encryption level. In order to decrypt the message, we need the decryption key which is defined as the modular inverse of the encryption key module N, defined as level 4.
- Finally, we have the last level (level 5) as the decryption level which is defined as the modular exponentiation of the cipher text to the power the decryption key module N, defined as the decryption level.


Figure 2: Top Level Block Diagrams of RSA (Key Generator, Encryption, Decryption)

## 3. Implementation Environment

The proposed work is to design and implement an efficient RSA cryptosystem that could be used to assure security in many communication network applications. Our proposed solution is applied using FPGA technology along with the VHDL as a hardware description language. The implementation phase of the proposed architecture included hardware and software equipments. Accordingly, following are the main component that has been used to accomplish the implementation and coding phase:

- ALTERA FPGA DE2-115 Hardware Kit: we have targeted the device of Cyclone II EP2C35F672C6 with EPCS16/ 16-Mbit serial configuration device. Such device is shown in figure 3, Altera DE2-115 Development and Education Board.

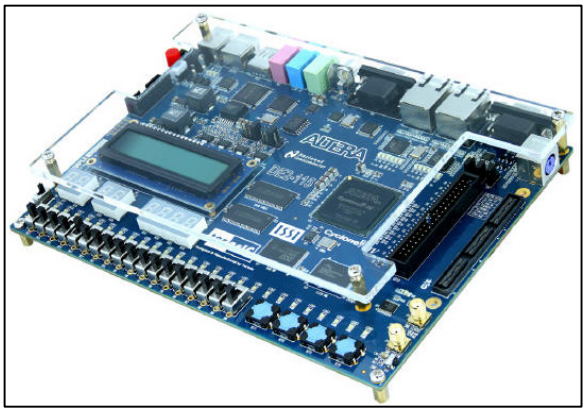

Figure 3: Altera DE2-115 Development and Education Board

- High Performance Computer Specifications: the purpose of such high characteristic is to enhance the simulation and synthesizing times. Table 1 shows the full description of this computer.

Table 1. High Performance Computer Specifications

| Components | Specifications |
|---|---|
| Processor | 4th generation Intel® Core™ i7-4770 processor quad-core [3.4GHz,8MB Shared Cache] |
| Operating system | Windows 8.1 64 bit |
| Memory | 16GB DDR3 - 1600MHz |
| Hard drive | 2TB 7200 RPM SATA |
| Graphics | 2GB AMD Radeon R7 240 [DVI, HDMI, & DVI to VGA] |
| Screen | 23" LED Display |

- Quartus II for Altera FPGA: This software is compatible most Altera FPGA families [9] as it is considered as our main tool for the design coding and synthesizing. We have used Quartus II version 13.0/64 bit edition to target our FPGA board (Altera DE2-115 Cyclone IV EP4CE115F29C7). However, the Quartus II software is a user friendly and considered as a typical environment to describe the hardware using VHDL. The main menu of Quartus II shown in figure 4. Also, Quartus II software supports specific versions of the EDA simulators for RTL and gate-level simulation. Therefore, we have used ModelSim-Altera Edition version 10.1d as a simulation tool for proposed design. A sample of ModelSim-Altera environment is shown in Figure 5.

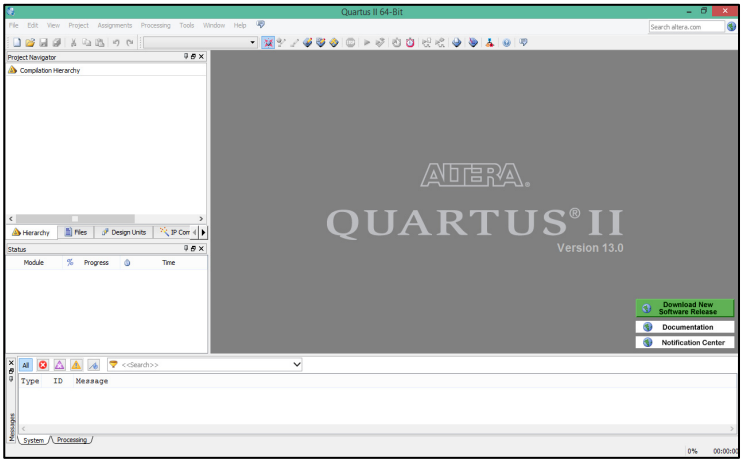


Figure 4: Quartus II main menu

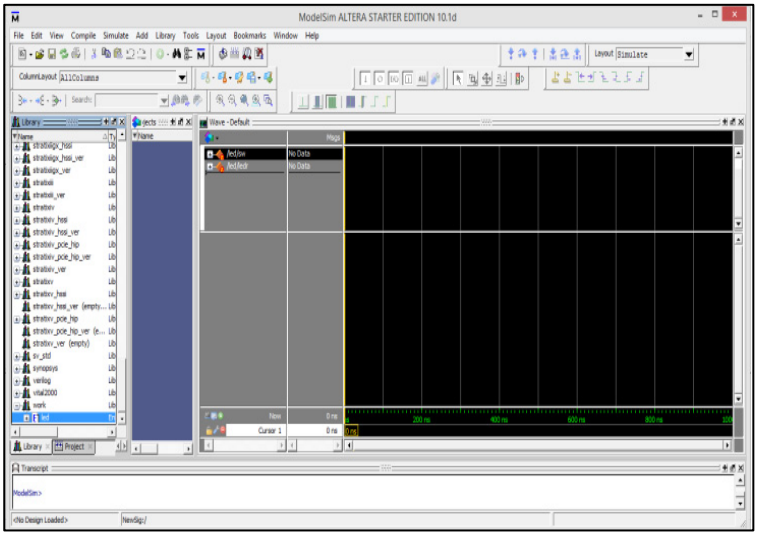


Figure 5: Modelsim Altera Edition 10.1d

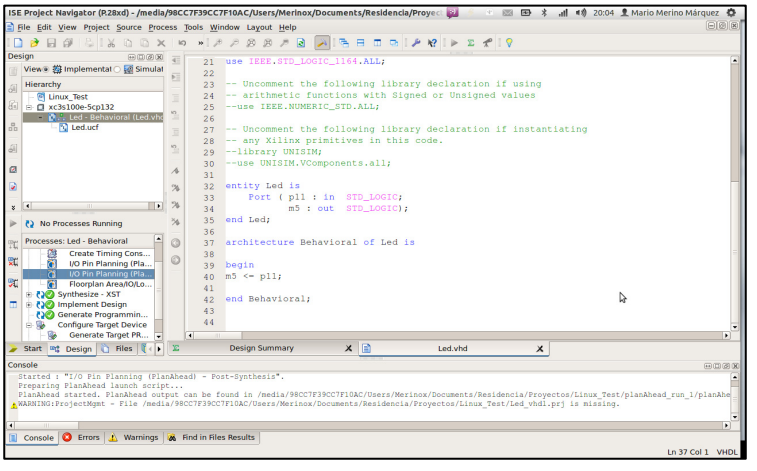


Figure 6: Xilinx ISE Design Suite version 6

- ISE Synthesizer Design Suite: we used the Xilinx ISE Design Suite version 14.2 for synthesizing purpose targeting another FPGA board family which is VERTIX VII VC707 FPGA where Xilinx FPGAs are more expensive and powerful [6]. The main menu of the ISE Synthesizer Design is shown in figure 6.

## 4. Experimental Results and Analysis

The proposed design was described using VHDL, Simulated using Modelsim, and synthesized using both Quartus II and Xilinx ISE synthesis tools for two different FPGA boards [2]. In this section, we will examine the Area-Time factor [7] to assess the cost of the proposed design. Table 2 shows the critical path delay and maximum frequency as well the area the design that synthesized in Altera DE2-155 and Virtex-7 VC707 FPGA boards. The proposed parallel RSA Cryptoprocessor design using sequential multiplier with redundant CSA has enhanced the area-time factor of hardware implementation. As seen in the table 2, the critical delay is less and better for Vertex-7 FPGA boards due to their higher speed grades [10]. Altera's and Xilinx's basic building block are a 4-input look-up table (LUT), a flip-flop and some additional circuitry that Altera calls a logic element (LE) and Xilinx calls a logic cell (LC). However, the area almost the same because of the ratio between Logic Elements and the Logic Cells is 1.125:1 [11].

Table 2: Comparisons of different 32 bit RSA designs

| # | Reference | size (bit) | Area | Fmax (MHz) | Time (ns) |
|---|---|---|---|---|---|
| 1 | H. Thapliyal | 32 | 15235 | 0.663 | 1507 |
| 2 | V. Garg | 32 | 28045 | 8.947 | 111.76 |
| 3 | Proposed design using Altera | 32 | 10427 | 15.725 | 63.590 |
| 4 | Proposed design using Vertix | 32 | 9414 | 17.629 | 56.724 |

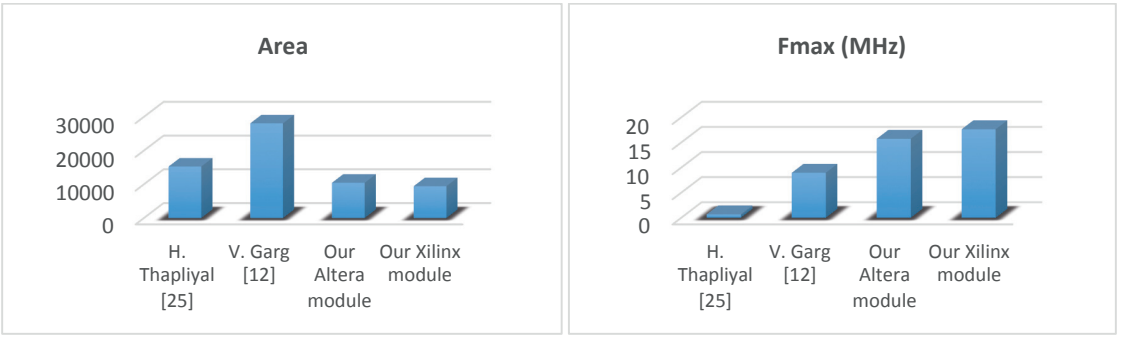


Figure 7: Comparison Charts for Area and Maximum Frequency

Also, Table 2 along with figure 7 shows the comparison between our proposed design and other two designs: H. Thapliyal Design [12] and V. Garg Design [8]. H. Thapliyal used Xilinx Spartan family which resulted in a gate delay 1.507 µs with area of 15235 (14942 FMAP and 293 HMAP) for RSA circuitry using 8x8 overlay multiplier architecture with 16 bit by 16 bit Vedic division. Alternatively, V. Garg used Virtex-5 - xc5vlx50t-2ff1136 Xilinx board along with the algorithm for encryption and decryption is "Square and multiply". Therefore, V. Garg got more efficient design in the critical time delay, but not in the area. However, our proposed design is considered as an improvement for both the critical time delay and the area due to the use of the scalable and efficient design modules and units as well as the parallel RSA Cryptosystem design based on a CSA sequential multiplier

## 5. Conclusions and Recommendations

An efficient FPGA design and implementation of RSA Cryptoprocessor using scalable modules has been discussed and proposed in this paper. The design has been executed for 32-bit encryption and decryption keys using

a new parallelized architecture involving several hardware modules and modular arithmetic units. The final coding of the coprocessor were synthesized using both ALTERA Cyclone IV EP4CE115F29C7 and VERTIX VII VC707 FPGA kits and resulted in a maximum frequencies of 15.725 MHz, 17.629 MHz respectively as well as analyzed and compared with several RSA designs. The synthesizing reports showed attractive and comparable results in terms of area and time (performance).

**Acknowledgements**

Authors appreciate the publication support of College of Engineering at King Faisal University (KFU). Special Thanks due to Prof. Amjad Omar (Chairman of EE Department) and Mr. Ibrahim Al-Daej (Administrator of Financial Affairs) for their support and help.

**Appendix A. Block Circuit Diagram Simulated from Quartus II**

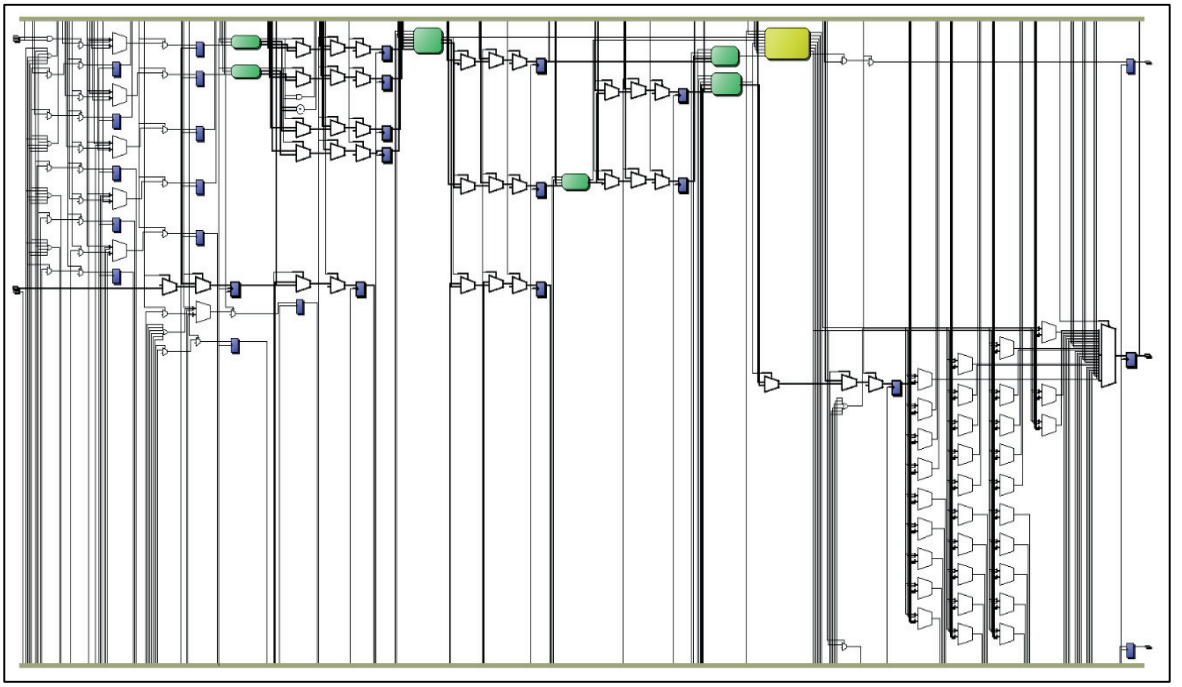

Figure 8: RSA Cryptoprocessor

**References**

1. L. Z. a. L. B. Haiyong Xie. Architectural Analysis of Cryptographic Applications for Network Processors. *IEEE First Workshop on Network Processors*, with HPCA-8, 2002.
2. Q. Abu Al-Haija and L. Tawalbeh. Efficient Algorithms & Architectures for Elliptic Curve Crypto-Processor Over GF (P) Using New Projective Coordinates Systems. *Journal of Information Assurance and Security*, vol. 1, no. 6, 2011.
3. D. L. Perry.VHDL: Programming by Example. Fourth ed., *The McGraw-Hill Companies*, 2002.
4. A. Corporation. Introduction to the Quartus® II Software. *Quartus*, 2013.
5. W. M. Incorporation. Maple User Manual, Copyright Maplesoft, a division of Waterloo Maple Inc., 2007.
6. X. Corporation. Xilinx ISE 10 Tutorial: A Tutorial on Using the Xilinx ISE Software to Create FPGA Designs for the XESS XSA Boards. *XESS Corp,* 2008.
7. Q. Abu Al-Haija and A. Al Badawi. Cost-effective design for binary Edwards elliptic curves crypto-processor over GF (2N) using parallel multipliers and architectures. *Int. J. of Information and Computer Security*, 2013 Vol.5, No.3, pp.236 – 250.

8. V. Garg and V. Arunachalam. Architectural Analysis of RSA Cryptosystem on FPGA. *International Journal of Computer Applications (0975 – 8887)*, vol. 26, no. 8, July 2011.
9. Altera. Quartus II Web Edition Software. *Altera*, Nov. 2013. [Online]. Available: http://www.altera.com. [Accessed 11 May. 2014].
10. Xilinx. Command Line Tools User. 19 April 2010. [Online]. Available: *http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/devref.pdf*. [Accessed 5 May 2014].
11. "Device Comparison," Altera, [Online]. Available: *http://www.altera.com/cgi-bin/device_compare.pl.* [Accessed 5 May 2014].
12. H. Thapliyal and M. Srinivas. VLSI Implementation of RSA Encryption System Using AncientIndian Vedic Mathematics. *Center for VLSI and Embedded System Technologies, International Institute of Information Technology*, Hyderabad-500019, 2005.