AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances D; Vol. 22; N°1; pp 122-136
Submitted Oct. 2017; Revised Dec. 20, 2017, Accepted Dec. 31, 2017

A New Crypto-based Processor System Design Using FPGA

*Suravi Lahiri, **Kaustav M. Chatterjee, ***Swarup K. Mitra

* Department of Computer Science & Engineering, MAKAUT, India (slahiri.iahiri06@gmail.com)
**Technical Head, Ardent Computech Pvt. Limited, India (Onlykaustav007@gmail.com)
***Associate Professor of M.C.K.V., India (swarup.subho@gmail.com)

Abstract

Cryptography is the art of keeping messages secretly and it plays an important role in securing the data from unauthorized access and use. In this paper we have developed a crypto based processor using RSA algorithm. RSA algorithm is a highly securable, public key algorithm. It is used to encrypt and decrypt the messages to send it over the transmission channel securely. The main purpose of our work in this paper is to develop a system to transmit the message to the intended recipient in such a way so that an unauthorized person cannot be able read or alter it.

This crypto based processor has been implemented on FPGA- which is a reconfigurable platform and designed by Very High speed integrated circuit Hardware Description Language (VHDL) and simulated by using Xilinx ISE 8.2i. This paper describes the detailed implementation of RSA encryption/decryption algorithm on FPGA using 32 bits key size.

Key words

Cryptography, FPGA, RSA, VHDL.

1. Introduction

Nowadays with the development of telecommunications especially the internet and mobile network have expanded the domain of information transmission in such a way which in turn presents new challenges for securing the data from unauthorized access and use. Cryptography is the science of information security. The word is derived from the Greek word *crypto's,* meaning hidden. Cryptography is mostly associated with scrambling plain text into cipher text (this process is called as encryption), and then back again to plain text (known as decryption) [1].

There are generally two types of cryptographic based encryption algorithm- one is symmetric key algorithm which is also called as private key algorithm and another one is asymmetric key algorithm which is also called as public key algorithm [2]. RSA is a public key algorithm. In this paper a new crypto based processor has been designed and implemented using RSA algorithm. The whole system has been implemented on FPGA and simulated using Xilinx ISE 8.2i. We have used RSA algorithm because it is highly securable public key algorithm. In this paper we have provided detailed implementation of RSA encryption/decryption algorithm on FPGA using 32 bits key size.

2. CRYPTOGRAPHIC ALGORITHM

2.1 Data Encryption Standard (DES)

DES is a symmetric key algorithm in which two types of keys are used- public and private key. Here same key is used for public and private key for encryption and decryption purpose, that's why it is called symmetric algorithm. This is a cipher that operates on 64-bit blocks of data, using a 56-bit key. It is a 'private key' system. DES is the archetypal block cipher — an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. In the case of DES, the block size is 64 bits [3].

2.2 RSA

RSA is a public-key system designed by Rivest, Shamir, and Adleman. RSA is one of the first practicable public-key cryptosystems and is widely used for secure data transmission. In this algorithm, there are two types of key are used. Public key and private key. Here different key is used for encryption and decryption. Public key is announced for the public and private key is used for receiver. That's why it is called Asymmetric algorithm. In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician.

2.3 Comparision of DES and RSA

From the above discussions we can say that asymmetric algorithm is much better for encryption and decryption system. Assymetric algorithm is much more securable than the

symmetric algorithm [4]. So, we have developed a new crypto based processor system using RSA – a assymetric algorithm.

Table 1. Comparison of DES and RSA

Feature	DES	RSA
Speed	High	Low
Deposit of keys	Needed	Needed
Data block size	64 bits	Min 512 bits
Key length	64 bits	Min 512 bits
Use of data space	Full, 64 bits	Variable
Ciphering & deciphering key	Same	Different
Ciphering & deciphering algorithm	Different	Same

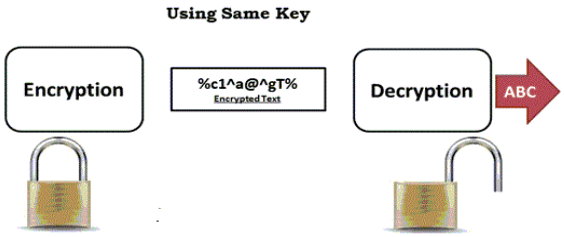

Figure 1. Symmetric Algorithm Concept

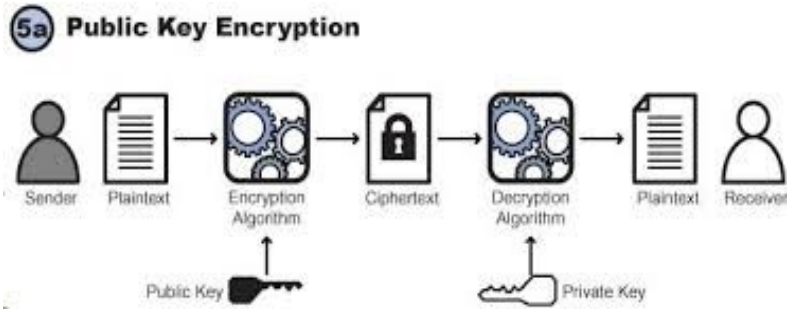

Figure 2. Assymetric Algorithm Concept

3. RSA ALGORITHM

RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it was classified until 1997. A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large

enough, only someone with knowledge of the prime factors can feasibly decode the message [5], [6], [7].

3.1 Algorithm

1. Select random prime numbers *p* and *q*, and check that *p != q.*
2. Compute modulus *n = pq.*
3. Compute phi, = *(p - 1)(q - 1)*
4. Select public exponent *e*, 1<e<phi, such that *gcd (e, phi) = 1*
5. Public key is *{n, e}*, private key is *d.*
6. Encryption: c= m^e mod n,
7. Decryption: m= c^d mod n.

3.2 Example of RSA algorithm

a. Choose p = 3 and q = 11
b. Compute n = p * q = 3 * 11 = 33
c. Compute φ(n) = (p - 1) * (q - 1) = 2 * 10 = 20
d. Choose e such that 1 < e < φ(n) and e and n are co prime. Let e = 7
e. Compute a value for d such that (d * e) % φ(n) = 1. One solution is d = 3 [(3 * 7) % 20 = 1]
f. Public key is (e, n) => (7, 33)
g. Private key is (d, n) => (3, 33)
h. The encryption $c = 2^7 \% 33 = 29$
i. The decryption of $c = 29$ is $m = 29^3 \% 33 = 2$

4. HARDWARE AND SOFTWARE SPECIFICATION

4.1 Field Programmable Gate Array (FPGA)

Field programmable gate arrays (FPGA's) are semiconductor devices. It can be reprogrammed to desired application after manufacturing. Field programmable gate array contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allows the block to be wired together –somewhat like many logic gates that can be inter-wired in many configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory [9].

4.2 Xilinx ISE 8.2i

It is announced as a **I**ntegrated **S**oftware **E**nvironment tool.The ISE 8.2i designed environment enables 30 percent faster performance than previous generation FPGAs.The ability to meet timing budget is the number one productivity issue fasing design today.The advanced

features and performance of the Vertex family can be quickly and efficiently utilized with ISE 8.2i.

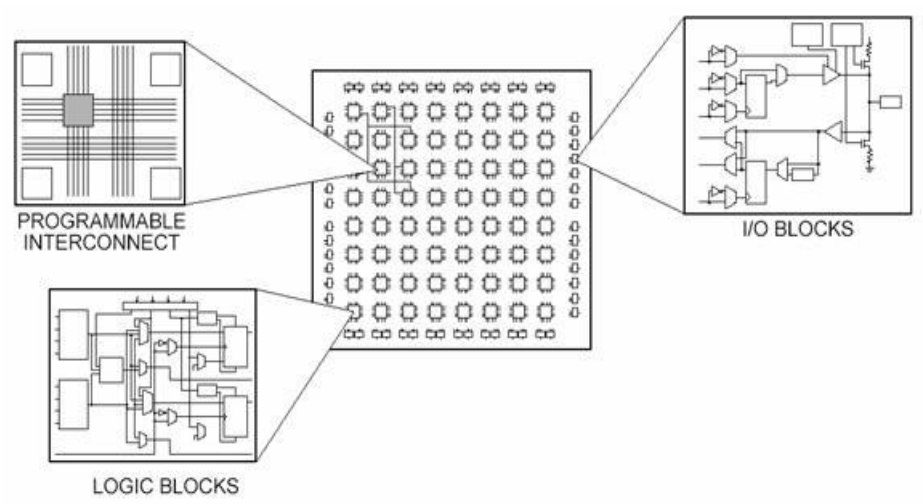

Figure 3. Basic parts in FPGA

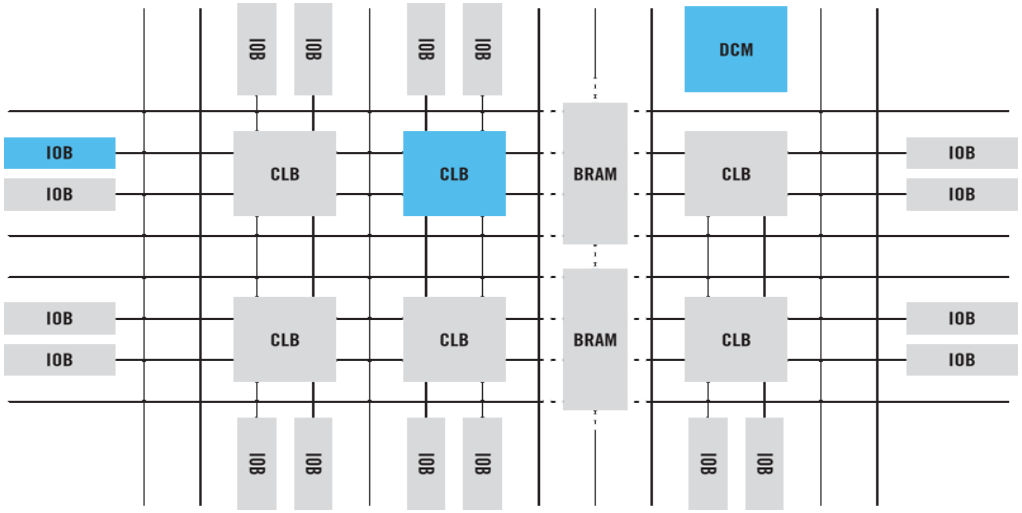

Figure 4. Building block of FPGA

5. IMPEMENTATION OF RSA ALGORITHM ON FPGA

The whole implementation of the program is divided into three categories. The categories are modular multiplication [9][13], key generation [10] which is the main program and encryption and decryption part [11], [12]. These three categories are discussed below:

5.1 Modular Multiplication

The first step of the algorithm involves multiplying two large prime numbers ‘p’ and ‘q’ (a prime number is a number which is divisible only by that number) and through additional operations deriving a set of two numbers that constitutes the public key and another set that is the private key. The prime numbers ‘p’ and ‘q’ should be large such that it will be very difficult to derive. From two selected prime number the computer will generate the public and the private key. From the two prime numbers the public key modulus(n) will be generated. n=p*q, where n is the modulus and p & q are the prime numbers

5.2 Key Generation

In the second step of the algorithm the value for public and the private key has to be determined in order to hide the secret message in such a way that it will be difficult to find the private key by only knowing the public key [15].At first we have to generate the value of Euler’s toitent function by multiplying (p-1) and (q-1) and then we have to choose a number ‘e’ such that ‘e’ is co-prime to φ(n), where φ(n) is the Euler’s totient function that counts the number of positive integers less than or equal to ‘*n’* that are relatively prime to ‘*n’*.

Compute φ(n) = *(p - 1)(q - 1)*
Select public exponent *e*, 1<e< φ(n), such that *gcd (e,* φ(n)*) = 1*
In order to determine the value of private key ‘*d’* in such a way so that ‘d’ is the multiplicative inverse of the public key ‘e’.

(d * e) mod φ(n) = 1.

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

Here the public key is (n, e) and the private key is d.

Only the public keys are announced publicly. One cannot easily compute the value of private key d from the numbers ‘e’ and ‘n’ because the computation is time consuming since the larger numbers are involved. Moreover, it is impossible to compute d without knowing the value of φ.

5.3 Encryption and Decryption

In the encryption stage, the user data is encrypted first by using the public key. The encryption is carried out by modular and exponential operation [14].

$c = m^e \bmod n$, where c is the encrypted message, (n, e) is the public key and m is the user message.

In the decryption stage, the encrypted message ‘c’ is sent to receiver and the receiver decrypts the encrypted message by using his private key ‘d’.

$q = c^d \bmod n$, where q is the decrypted message, c is the encrypted message and d is the private key.

6. RESULTS

6.1 Output in Xilinx 8.2i

6.1.1 Case 1

Table 2. Input and Output Parameters

Parameter	Description	Value
m	User message	2
p, q	Prime numbers	3, 11
e	Public key exponential	7
c	Encrypted output	29
q	Decrypted output	2

Notes: here public key is (n, e) => (33, 7) and private key is d => 3.

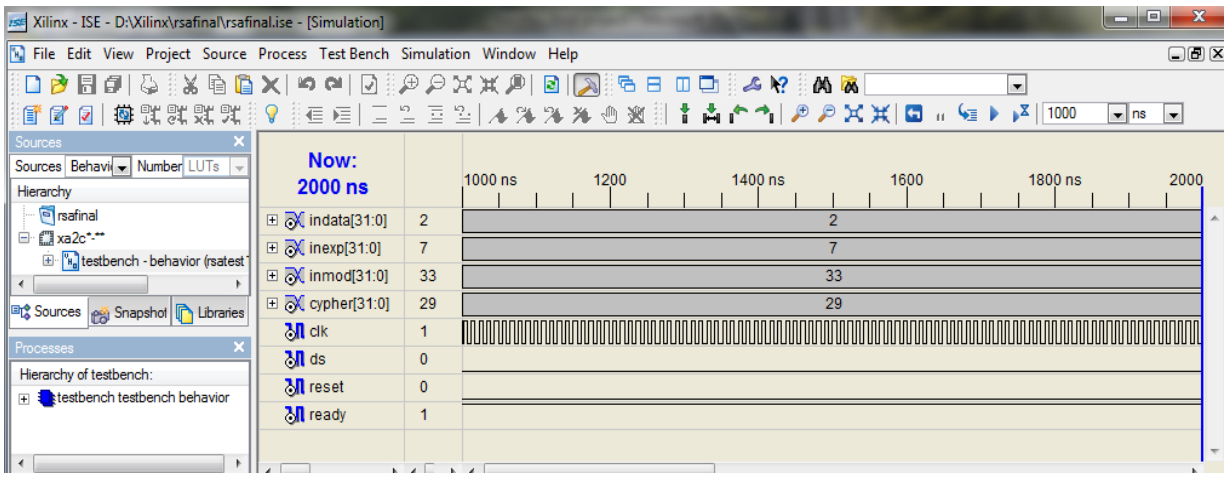

Figure 5. Encrypted output

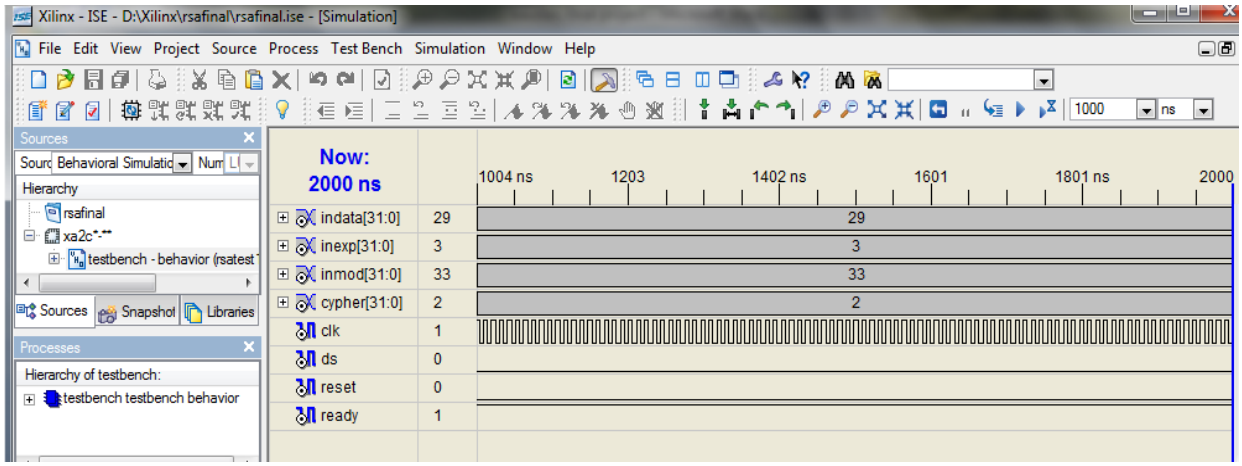

Figure 6. Decrypted output

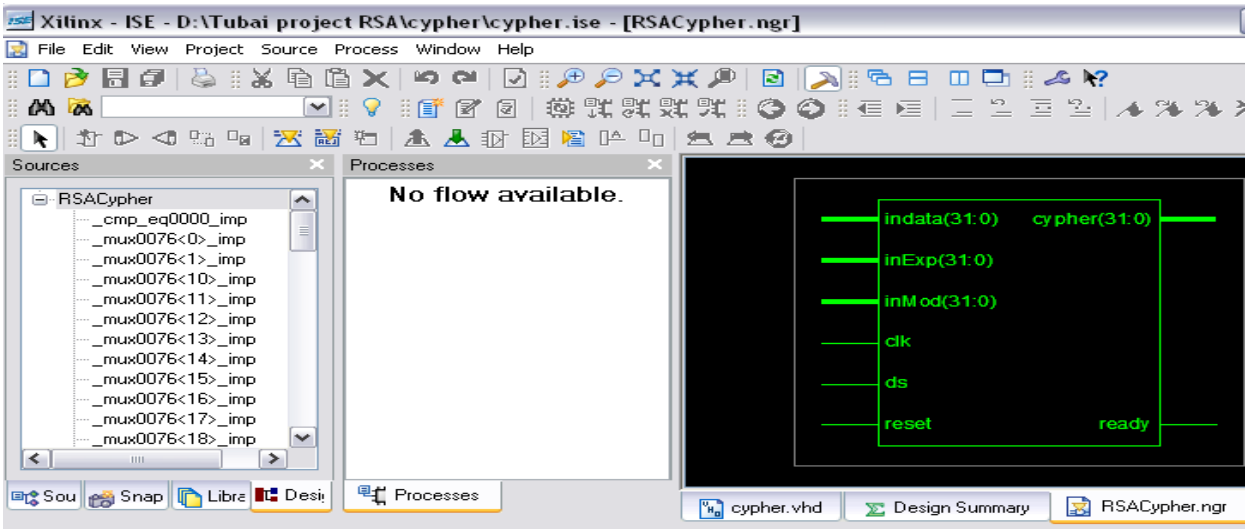

Figure 7. RTL schematic

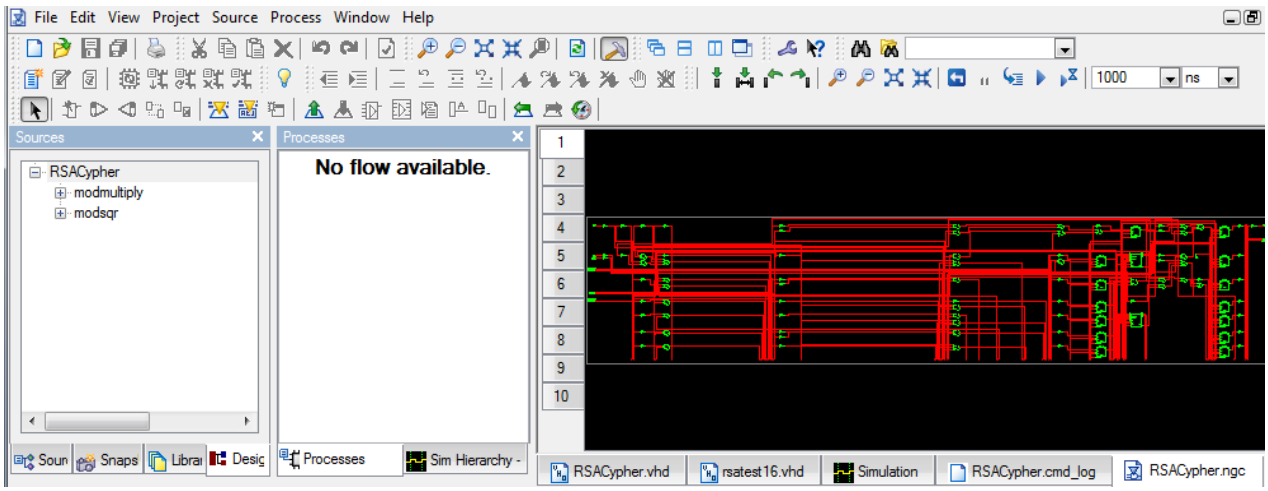

Figure 8. Look up table

6.1.2 Case 2

Table 3. Input and Output Parameters

Parameter	Description	Value
m	User message	5
p, q	Prime numbers	3, 11
e	Public key exponential	7
c	Encrypted output	24
q	Decrypted output	5

Notes: here public key is (n, e) => (33, 7) and private key is d => 3.

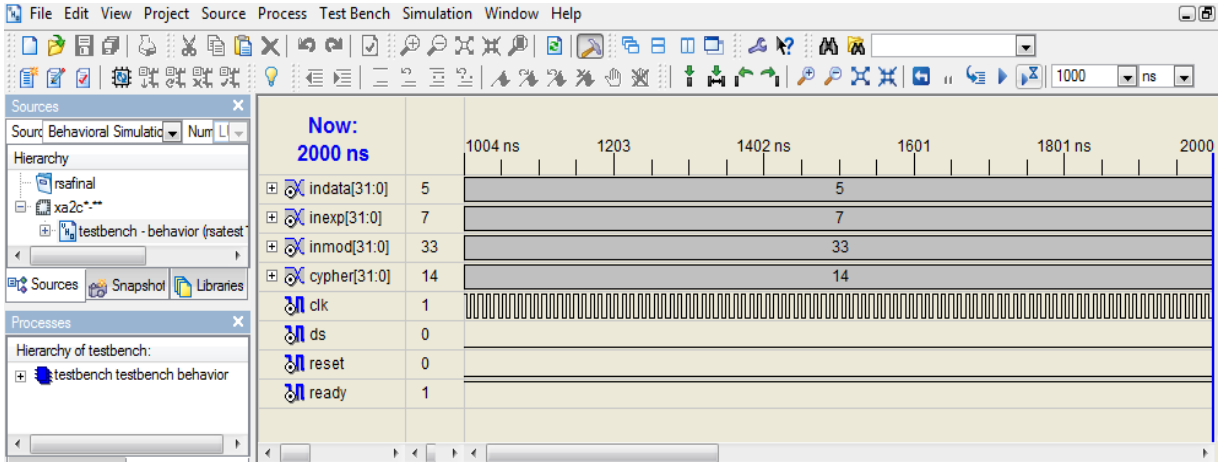

Figure 9. Encrypted output

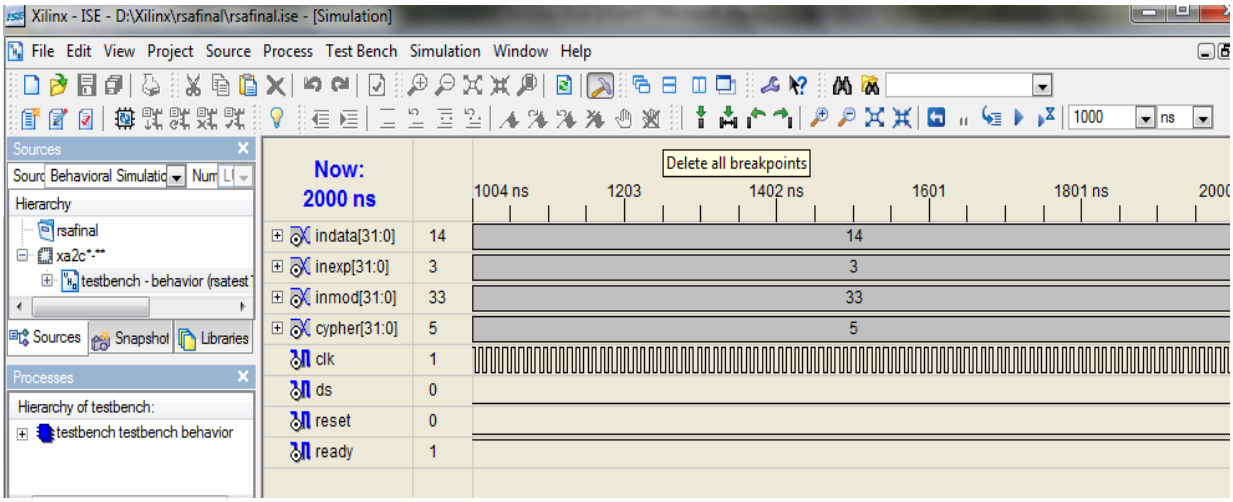

Figure 10. Decrypted output

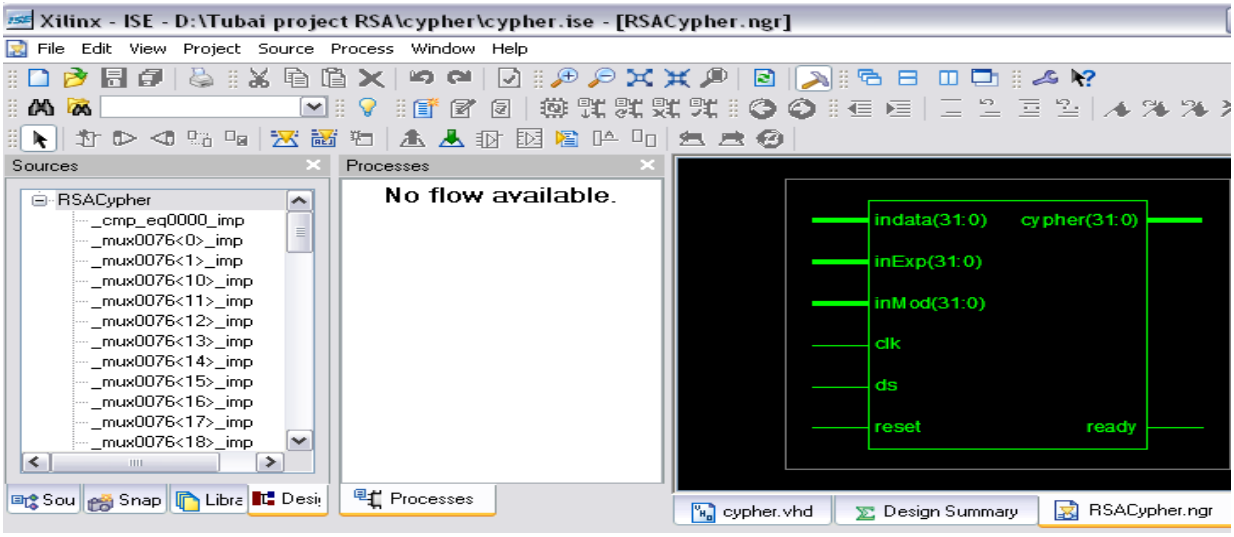

Figure 11. RTL schematic

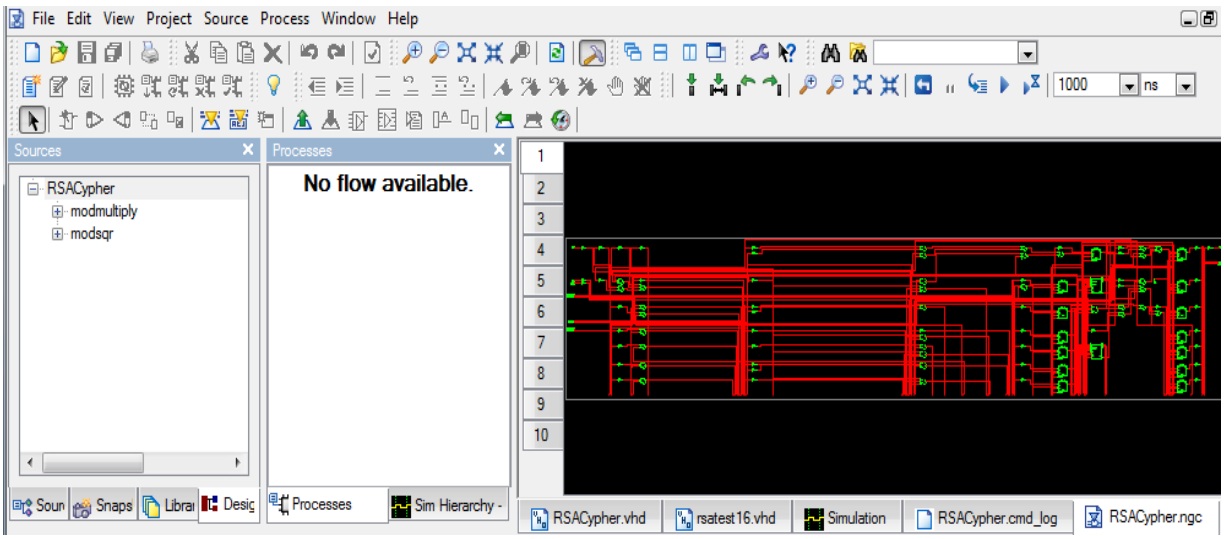

Figure 12. Look up table

6.1.4 Case 3

Table 4. Input and Output Parameters

Parameter	Description	Value
m	User message	30
p, q	Prime numbers	3, 11
e	Public key exponential	7
c	Encrypted output	24
q	Decrypted output	30

Notes: here public key is (n, e) => (33, 7) and private key is d => 3.

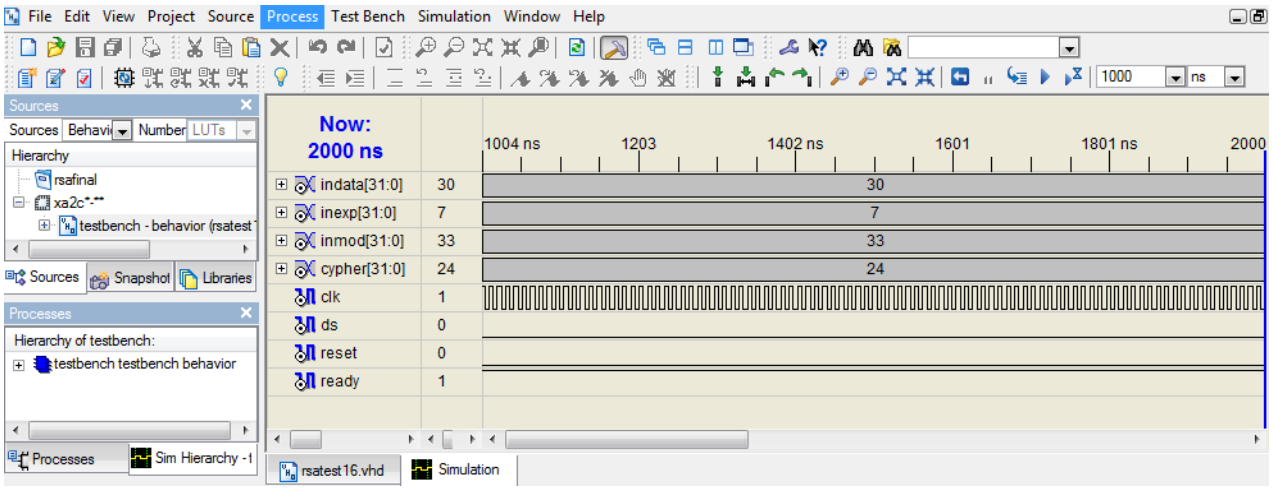

Figure 13. Encrypted output

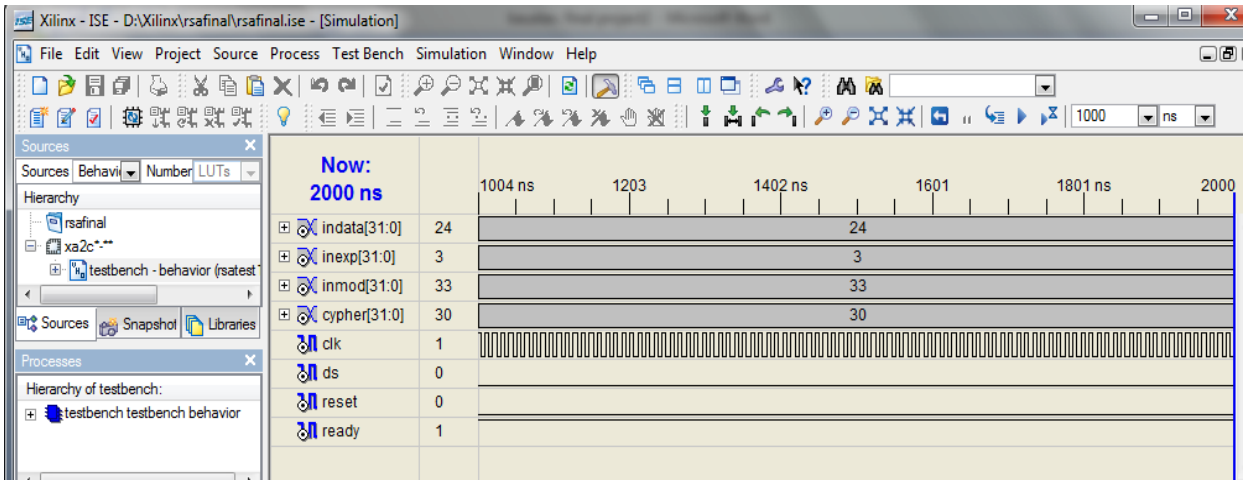

Figure 14. Decrypted output

6.2 Output in FPGA Spartan-3E Kit

The Spartan-3E High Volume Starter Kit gives designers instant access to the complete platform capabilities of the Spartan-3E family. The Spartan®-3E FPGA Starter Kit is a complete development board solution giving designers instant access to the capabilities of the Spartan-3E family. Complete kit includes board, power supply, evaluation software, resource CD (application notes, white papers, data sheets, etc.), and USB cable. The Spartan 3E Starter Board provides a powerful and highly advanced self-contained development platform for designs targeting the Spartan 3E FPGA from Xilinx.

6.2.1 Parts included in this FPGA

- Development board
- Universal power supply 100-240V, 50/60 Hz
- ISE® WebPACK™ software, ISE Foundation™ software evaluation, and the Embedded Development Kit (EDK)
- Starter Kit resource CD
- USB cable

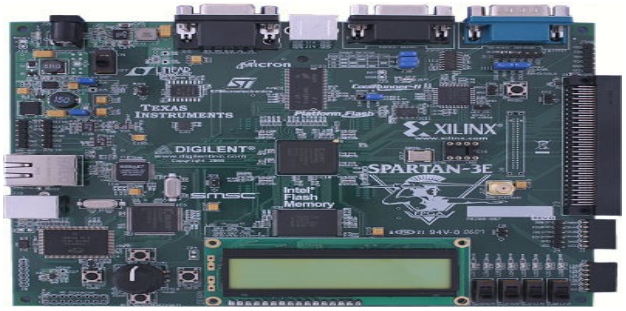

Figure 15. Spartan-3E Kit

6.2.2 Case 1

When data input is 2

Figure 16. Output in FPGA

6.2.3 Case 2

When data input is 5

Figure 17. Output in FPGA

6.2.3 Case 2.

When data input is 30

Figure 18. Output in FPGA

7. CONCLUSIONS

In cryptographic applications, FPGA's can be used as dedicated computers in order to perform some computations at very high frequencies. It allows us to reach high encryption rates for different Block ciphers. A consequence is that these implementations can speed up well-known attacks and improve their effectiveness, with some additional constraints that can be solved either by modifying algorithms or by hardware/ software co-designs. In this paper we have tried to make an overview of the features that make reconfigurable hardware a viable tool in the design as well as in the implementation by using cryptographic applications and with the help of FPGA we have studied the performance of various output of the RSA algorithm according to the various input. In this paper we have explained the details implementation of the RSA encryption/decryption algorithm in FPGA kit named as Spartan3E and we have successfully developed a crypto based processor system. This crypto based processor is applicable in different privacy system.

ACKNOWLEDGMENT

I would like to express my special thanks and appreciation to my guides for their support and encouragement throughout this work.

REFERENCES

[1] S.N. Kumar, Review on network security and cryptography, 2015, International transaction of electrical and computer engineers system, vol. 3, no. 1, pp. 1-11. Available: http://pubs.sciepub.com/iteces/3/1/1

[2] S. Kashyap, E.N. Madan. (2015, Apr). A review on network security and cryptographic algorithm, 2015, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 4, pp. 1414-1418. Available:https://www.ijarcsse.com/docs/papers/Volume_5/4_April2015/V5I4-0249.pdf

[3] S. Karthik, A. Muruganandam, Data encryption and decryption by using triple DES and performance analysis of crypto system, 2014, International Journal of Scientific Engineering and Research (IJSER), vol. 2, no. 11, pp. 2347-3878. Available: http://www.ijser.in/archives/v2i11/SjIwMTM0MDM=.pdf

[4] D. Pointcheval, Assymetric cryptography and practical security, 2002, Journal of Telecommunications and Information Technology, vol. 4, pp. 2-26. Available: http://www.di.ens.fr/users/pointche/Documents/Papers/2002_jtit.pdf

[5] Saranya, Vinothini, Vasumathi, A study on RSA algorithm for cryptography, 2004, International Journal of Computer Science and Information Technologies, vol. 5, no. 4, pp. 5708709. Available:http://ijcsit.com/docs/Volume%205/vol5issue04/ijcsit20140504192.pdf

[6] Y.G. Nentawe, Data encryption and decryption using RSA algorithm in a network environment, 2013, *International Journal of Computer Science and Network Security*, vol. 13, no. 7, pp. 9-13. Available: http://paper.ijcsns.org/07_book/201307/20130702.pdf

[7] I. Jahan, M. Asif, L.J. Rozario, Improved RSA cryptosystem based on the study of number theory and public key cryptosystem, 2015, American Journal of Engineering Research (AJER), vol. 4, no. 1, pp. 143-149. Available: http://www.ajer.org/papers/v4(01)/S040101430149.pdf

[8] S. Kilts, Architecting speed, 2007, Advanced FPGA design, Architecture, Implementation, and Optimization, Hoboken, New Jersey: John Wiley & Sons Inc., ch. 1, pp. 2-16.

[9] A.R. Landge, A.H. Ansari, RSA algorithm realization on FPGA, 2013, International Journal of Advanced Research in Computer Engineering & Technology(IJARCET), vol. 2, no. 7, pp. 2323-2327. Available: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-2-ISSUE-7-2323-2327.pdf

[10] R. Behera, P. Abhisek, FPGA implantation of RSA algorithm and to develop a crypto based security system, M.S. thesis, Dept. Electronics & Communication Engg, National Institute of Technology, Rourkela, India, 2012-2013.

[11] A. Thobbi, S. Dhage, P. Jadhav, A. Chandrachood, Implantation of RSA encryption algorithm in FPGA, 2015, American Journal of Engineering Research, vol. 4, no. 6, pp. 144-151. Available: http://www.ajer.org/papers/v4(06)/Q04601440151.pdf

[12] S.K. Sahu, M. Pradhan, FPGA implementation of RSA encryption system, 2011, International Journal of Computer Application, vol. 19, no. 9, pp. 10-12. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.3453&rep=rep1&type=pdf

[13] A.S. Tahir, Design and implementation of RSA algorithm using FPGA, 2015, International Journal of Computer & Technology, vol. 14, no. 12, pp. 6361-6366. Available: https://www.researchgate.net/publication/282249995_Design_and_Implementation_of_RSA_Algorithm_using_FPGA

[14] K. Shehata, H. Hussien, S. Yehia, FPGA implementation of RSA encryption algorithm for e-passport application, 2014, International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 8, no. 1, pp. 82-85. Available: http://waset.org/publications/9997304/fpga-implementation-of-rsa-encryption-algorithm-for-e-passport-application

[15] V. Garg, V.A. Chalams. Architectural analysis of RSA crypto system on FPGA, 2011, International Journal of Computer Applications, vol. 26, no. 8, pp. 32-34. Available: https://pdfs.semanticscholar.org/b8d8/16642aa0d09730994236a31d50f834bd4598.pdf