Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Efficient Implementation for PRINCE Algorithm in FPGA Based on the BB84 Protocol

To cite this article: Alharith A. Abdullah and Noor R. Obeid 2021 *J. Phys.: Conf. Ser.* **1818** 012216

View the article online for updates and enhancements.

You may also like

- Quantum cryptography and combined schemes of quantum cryptography communication networks
  A.Yu. Bykovsky and I.N. Kompanets

- Thwarting the photon-number-splitting attack with entanglement-enhanced BB84 quantum key distribution
  Carl F Sabottke, Chris D Richardson, Petr M Anisimov et al.

- A monogamy-of-entanglement game with applications to device-independent quantum cryptography
  Marco Tomamichel, Serge Fehr, Jdrzej Kaniewski et al.

PRIME
PACIFIC RIM MEETING
ON ELECTROCHEMICAL AND SOLID STATE SCIENCE
HONOLULU, HI
Oct 6–11, 2024
Abstract submission deadline:
April 12, 2024
Learn more and submit!
Joint Meeting of
The Electrochemical Society
•
The Electrochemical Society of Japan
•
Korea Electrochemical Society

This content was downloaded from IP address 37.238.40.40 on 29/03/2024 at 07:27

Efficient Implementation for PRINCE Algorithm in FPGA Based on the BB84 Protocol

Alharith A. Abdullah*[1], Noor R. Obeid*[2]
College of Information Technology, University of Babylon, Babil, Iraq
Emails: alharith@itnet.uobabylon.edu.iq

Abstract. The current study introduces a hardware-implemented PRINCE block cipher within Field Programmable Gate Array (FPGA) determined by the quantum cryptography protocol (BB84). Most security-related software applications of cryptographic algorithms tend to be rather slow and of no efficiency. So as to present a solution to this issue, a new hardware architecture is suggested for speeding up the execution of the PRINCE algorithm and increasing its flexibility, yet with more security. Concurrent computing designs allow an encryption block data of 64 bits during a single clock cycle, resulting in the reduction of hardware area and the production of a higher throughput and relatively lower latency. Higher speed processing and lower power consumption are other features that have been observed. This could be achieved by means of implementing the encrypting, decrypting and quantum key schedule using little hardware sources, followed by the development of a sufficient hardware architecture model for the PRINCE algorithm through very high speed integrated circuit hardware description language (VHDL). The synthetization of this VHDL design is eventually performed in FPGA boards. As for the present study, two FPGA boards have been employed, namely Virtex-4 and Kintex-7. The resulting data indicates the throughput and efficiency values to be (2.029 Gbps) and (1.9 Mbps/slice) for Virtex-4, and (3.931 Gbps) and (7.290 Mbps/slice) for Kintex-7, all respectively.

Keywords:
Lightweight, Prince, FPGA, VHDL, BB84.

I. Introduction

Quantum cryptography is considered to be among the most useful solutions to ensure information security systems nowadays, describing the ultimate method for key distribution. It is different from the various of classical cryptosystems used today, as these security systems regularly face issues due to computational power and the difficulty of implementing mathematical issues, meanwhile the quantum cryptography security is based on physics laws. [1]

PRINCE algorithms is an important algorithm within the field of Quantum cryptography. It depends on a group of logic design operations to encrypt the data, as well as a BB84 protocol which can be defined as a quantum key distribution scheme often clarified as being a method of secure communication through a private key in one-time pad encryption between parties. Moreover, the Virtex-4 presents high throughput about 2032 Mbps, with a total equivalent of 956 slices and an efficiency of 2.126 Mbps/Slice.

Quantum cryptography and especially Quantum Key Distribution (QKD) is used as the most research direction in the past twenty years and now proceeds to a majority field of the quantum field, where the QKD enables the Secret Key Establishment mechanism within two users' connection,

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

using a combination of classical and quantum channels. The essential benefit of QKD deals with the “quantumness” of the signals transferred within the quantum channel, which leads to detect any eavesdropping threats affecting the communication line. This characteristic of QKD leads to a particular property that is not achieved by means of classical cryptography techniques. It provides Key Establishment beside the high-security standard as unconditional or information-theoretic security. Quantum lightweight cryptography methods are recommended to deal with the several problems of conventional cryptography, for instance, limitations related to physical size, the requirements of processing, the limitation of memory and energy consumption [2].

In this paper, the parallel hardware design within the Xilinx environment has been used to apply the PRINCE algorithms on the Field-programmable gate array (FPGA) board as based on the BB84 Protocol. The used FPGA board, Virtex-4, examines the suggested hardware model.

There are many related research directions presented in this work, as in [3] who suggested an effective architecture within hardware based on the hummingbird algorithm. They applied a specific partial loop unrolling which involves both implementation performance and the area of hardware. The used system based on the Verilog HDL and simulated in ModelSim with a particular compiler which is known as a cadence RTL, thereby implementing the low-cost Spartan 3 FPGA board. Furthermore, a comparison with the existing implementations is drawn. System results provided a 6% reduction of the area around, whereas the throughput got roughly doubled.

Due to the large number of data generated in an environment of Smart Grid, a security system is required to maintain the power data being confidential and integral. The lightweight transmission scheme has been suggested in [4]. They merge the "one-time pad" mechanism with a quantum cryptography scheme, using a quantum random number features with quantum key distribution protocol for ensuring that the key distribution remains secure. Furthermore, they produced a new lightweight stream cipher encryption algorithm. The result analyses of the used system showed how the used scheme could guarantee the power data to be transmitted securely.

In [5], the general quantitative key distribution (QKD), a secure key exchange, has been proposed depending on the laws of quantum physics instead of arithmetic complexity. They presented a comprehensive scope of the most commonly followed protocols for goods, including IPsec and TLS. By following a key exchange model, they suggested the way in which QKD could be integrated into such security applications. They proposed a support layer for providing a range of QKD services shared among the QKD protocol and any security application.

The authors in [6] illustrated a method based on a quantum key distribution called BB84. It provides a description of governing quantum physics to work properly within any system inside quantum cryptography. They displayed the specific analysis details of the BB48 system, followed by the discussion of its operating procedures and security characteristics. In addition, different quantum system implementation challenges are described [6].

Additionally, [7] introduced a framework for simulation and modeling quantum key distribution protocols and analyzing the impact of the components of the experimental photonic on the QKD process by employing commercial photonic simulation tools called "OptiSystem". It is useful for the implementation of the quantum key distribution components. The used system simulates BB84 operation having various security threats and noise scheme of key distribution.

As for [8] the authors performed an IP-Core scheme of PRINCE lightweight algorithm based on the main features of execution-speed and low-resource hardware applied on Field Programmable Gate Arrays (FPGA). The new FPGA IP-core is to speed-up the performance of PRINCE, superseding software implementation that is typically slow and inefficient**.**

In [9] [14], the authors show a PRINCE block cipher algorithm applied in hardware, taking into consideration the feature of latency which is required for the security of real-time needs. The simulation results show the encryption and decryption with the minimum costs added.

In [10], the authors implement the lightweight scheme within specific cryptographic devices to analyze the efficiency and security operations of PRINCE and RECTANGLE within an attack environment implementation described as a Differential Power Analysis (DPA) attack. The used attack reduces key search space from $(2)^{128}$ to 33008 for PRINCE and $(2)^{80}$ to 288 for RECTANGLE.In [11], the authors proposed a secure and effective lightweight cryptographic algorithm for implementation within smaller computing devices, based on the Linux benchmark tool as Fair Evaluation of Lightweight Cryptographic Systems (FELICS) for testing the quality of encryption state. The suggested algorithm manages to reduce the method of using processing cycles but simultaneously, it produces appropriate security.

The sections of the current study are presented in the following way: Section II represents and explanation of the used PRINCE algorithm and the main algorithm components. Section III describes the BB84 Quantum Key Distribution Protocol. Section IV deals with the (VHDL) implementation of PRINCE algorithm, whereas Section V sheds light on the simulation and results. Eventual conclusions upon this study are presented in Section VI.

II. PRINCE Algorithm

The PRINCE algorithm is a (64-bit) substitution-permutation network (SPN) of the lightweight block cipher holding a 128-bit key. It consists of the cipher core with 12 rounds, and each core function round adds a round-dependent constant and a "fixed-key, sixteen 4×4 parallel s-boxes and a linear diffusion". The initial half is described as a secret key of 128-bit, employed as the "pre- and post-whitening keys", meanwhile its remaining part is applied in a direct manner within the round functions. Therefore, the implementation of the decrypting function required the key first to be (XOR-ed) with a constant value, so as to reuse the corresponding circuit for decryption state. The overhead of producing a decrypting state should thereby be consequently reduced. PRINCE is described as the first lightweight block cipher to take into consideration the latency feature. The prior system designs cover the essentials on a small footprint work within the hardware. The encryption scheme in the PRINCE algorithm can be achieved in a perfect one clock cycle whenever an unrolled hardware architecture is implemented. Such an aim could be achieved by keeping down the requirement of the design area, where designers choose the ideal s-box for keeping the number of actions in the linear diffusion section minimal. Figure 1 explains the building and the components of the PRINCE cipher.

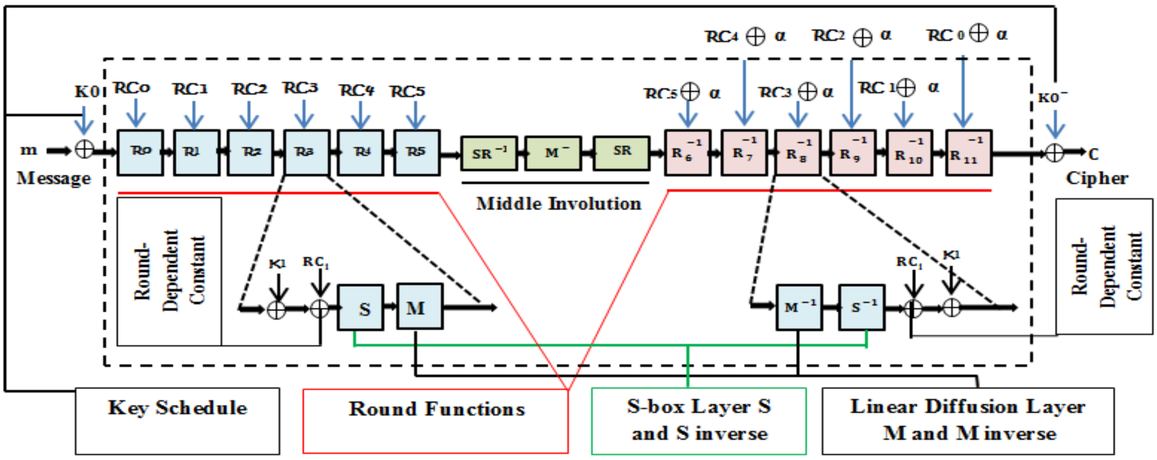


Figure (1): The PRINCE Core Cipher components

The main components of the PRINCE algorithm are sorted as:

Key Schedule: It described as the 128-bit secret key, which is divided into two key schemes: the first one is k0 and the second is k_1. The key (k_0) is applied in a direct manner as the pre-whitening key, whereas the other one (k0) is used toward post-whitening:

$$K_0' = (k0 >>> 1) \oplus (k_0 >> 63)$$

It seems to be a modest alteration of k0. Pre-and post-whitening mention adding key elements both before and after the operation of the PRINCE core cipher. As mentioned, PRINCE core processing is described as the PRINCE core. The key k_1 is applied in a direct way within the process of the key addition phase of the round functions R and R^{-1}. The XOR operation is illustrated in Figure (2).

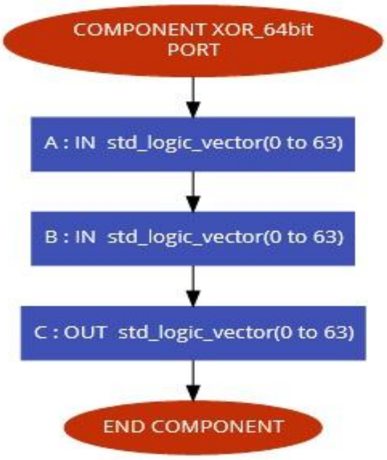

COMPONENT XOR_64bit PORT
A : IN std_logic_vector(0 to 63)
B : IN std_logic_vector(0 to 63)
C : OUT std_logic_vector(0 to 63)
END COMPONENT

Figure (2): The used XOR operation

Round Functions R and R^{-1}: This component contains an XOR scheme with a constant key k1, with an XOR beside a round-dependent constant RC, a S-box layer S and the inverse of S as S^{-1} and a linear diffusion of M and the inverse of M as M^{-1}.

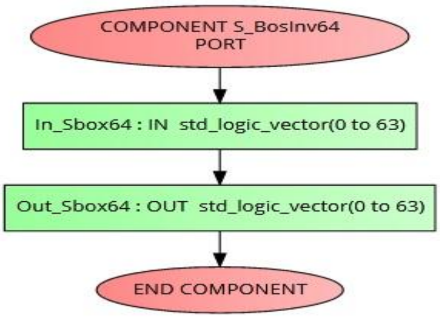
The Round-Dependent Constant Component: It is applied as the $RC_i \oplus RC11_{-i} = \alpha$ for $0 \leq i \leq 1$, with α in hexadecimal as α=coac29b7c97c50dd.

S-box Layer S and the inverse of S as S^{-1}: This layer maps the 4-bit to 4-bit, illustrated in Table 1.

Table I. The used S-box Layer of Prince Algorithm.

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S (X)	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4
X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S^{-1} (X)	B	7	3	2	F	D	8	9	A	6	4	0	5	E	C	1

The S-box operation within the used system described as in Figure (3):


COMPONENT S_BosInv64
PORT
In_Sbox64 : IN std_logic_vector(0 to 63)
Out_Sbox64 : OUT std_logic_vector(0 to 63)
END COMPONENT

Figure(3) The used S-box scheme operation.

Linear Diffusion Layer M and the inverse of M as M^{-1}: It is implemented with XORs three input bits so as to generate one output bit. Respective output bits make use of differing input bits. This layer is planned as an approach to maximize the diffusion features. The used M^{-1} within the proposed system is presented in Figure (4):

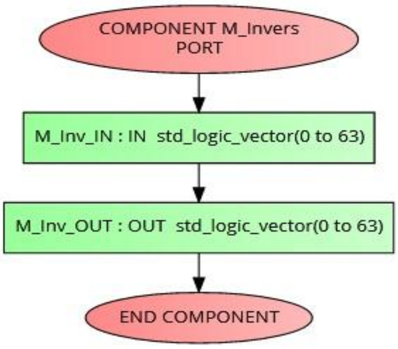

COMPONENT M_Invers
PORT
M_Inv_IN : IN std_logic_vector(0 to 63)
M_Inv_OUT : OUT std_logic_vector(0 to 63)
END COMPONENT

Figure (4): the used M^{-1} operations.

The Middle Involution: This part includes the functions of SR^{-1}, SR, and M^{-} schemes. The former two share some form of similarity with the AES ShiftRows procedure, whereas the latter one is merely a linear diffusion. The middle involution work illustrates a connector for the round function of the forward and inverse operations. Such a case produces the consequence of the encrypting operation with key k being the same as the decrypting process with key ($k \oplus \alpha$). The main operation done by SR is described in Figure (5).

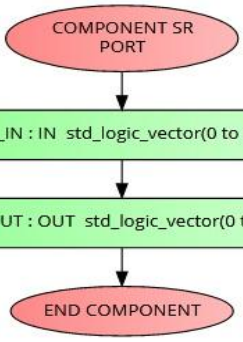

COMPONENT SR PORT
SR_IN : IN std_logic_vector(0 to 63)
SR_OUT : OUT std_logic_vector(0 to 63)
END COMPONENT

Figure (5): the used SR operations.

III. BB84

Bennett and Brassard managed to present BB84 in 1984 as being the first QKD method. In brief, within the BB84 the partners (Alice and Bob) want to agree on a secret key concept regarding no eavesdropper (Eve) to gain important information, and the operations are summarized as follows: Alice sends any bit of the secret key in one of a collection of conjugate bases in another side (not recognized by Eve), and the key used remains protected through the impossibility of holding the state of a quantum system bases simultaneously [12]. Figure (6) depicts the main operation steps within the BB84 protocol.

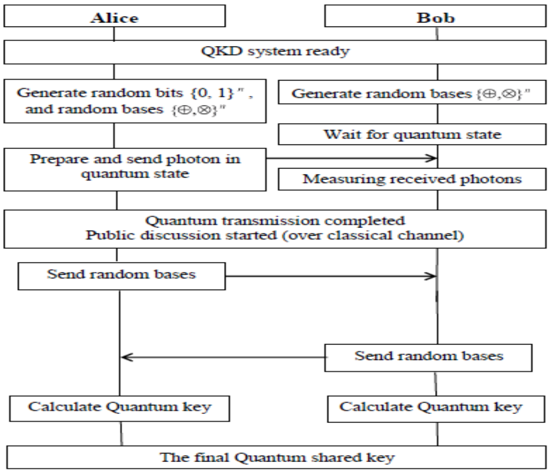

Alice
Bob
QKD system ready
Generate random bits {0, 1}" , and random bases {⊕,⊗}"
Generate random bases {⊕,⊗}"
Wait for quantum state
Prepare and send photon in quantum state
Measuring received photons
Quantum transmission completed
Public discussion started (over classical channel)
Send random bases
Send random bases
Calculate Quantum key
Calculate Quantum key
The final Quantum shared key

Figure (6): The flowchart of BB84 Quantum Key Distribution Protocol steps.

IV. VHDL Implementation of prince algorithm

As far as we have experienced, there is no prior case of the PRINCE algorithm being implemented within the hardware component and based on quantum cryptography. The used system implemented the PRINCE algorithm on Field Programmable Gate Arrays (FPGAs) because of the adaptability given by the FPGA technology. The authors have designed a hardware using VHDL which has been tested using simulation and test vector. The main processes of the used system are described in Figure (7). Beside the common sender and receiver, the used system consists of the PRINCE algorithm, BB84 QKD Quantum key distribution system and the main steps of the PRINCE algorithm, and the system schemes are described as follows: The PRINCE design includes three ports, two of them for input sorting, as the 64-bits for plaintext and the 128-bits for key, and the third port is describing the cipher-text as 64-bits. The whole number of pins used for the input/output sections are 256-pins. The used scheme for the FPGA board implemented is Virtex-4 for its larger number of logic resources and input/output pins. The BB84 protocol within the used system acts as a quantum key distribution scheme, which consists of a sender (Alice), and a receiver (Bob) connected using a specific channel called a classical public channel. The used object of BB84 is described as a way for the secure communication of private keys between two communication sides in a one-time pad encryption. So, BB84 produces a perfectly shared random key between the sender and receiver to be used in the algorithm. The PRINCE decryption design scheme is similar to the design of the encryption scheme. The decryption scheme needs the previous round key in the encrypting scheme to work as an input key for the initial round of decrypting. This procedure is achieved by employing the operation of XOR for key (K_1) with α or RC11 to make the new key (K_1) intrude towards the decrypting scheme.

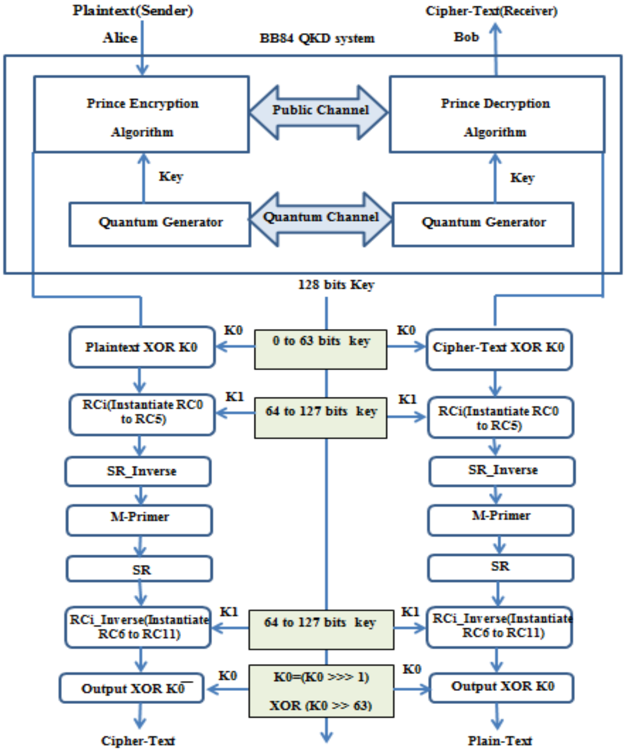


Figure (7): The proposed data flow of encryption/decryption operations

V. Simulation and Result

The used system implemented the Xilinx ISE V14.5 WebPACK and (ModelSimXE P.58f) for design synthesis and simulation tool, which is based on the (VHDL) for each of Virtex-4 and Kintex-7. The main objective of the used design is providing lower latency and economic hardware implementation, as well as improving the security by adding BB84 protocol. Through the used model design, the minimal potential gate is examined for producing a lower latency hardware PRINCE algorithm component. The results that have been obtained for each FPGA board indicate a relatively high throughput and efficiency, as they display throughput and efficiency rates of (2.029 Gbps) and (1.9 Mbps/slice) for Virtex-4, respectively. The parameter of energy consumed in Virtex-4 is 0.165W. Figure (8:A) shows the implementation of BB64 in Xilinx simulation tool (ModelSimXE P.58f).

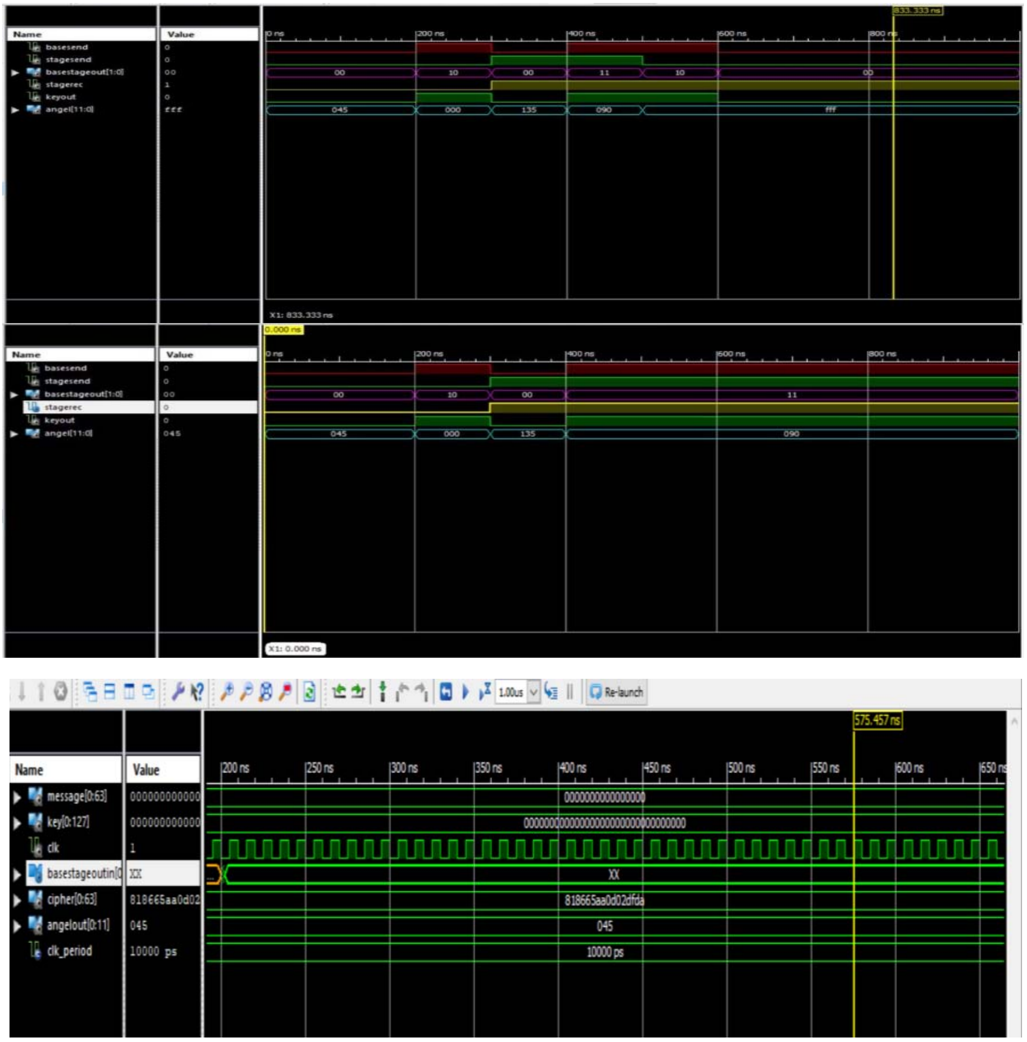


Figure (8): (8:A, 8:B): Simulation BB64 core in Xilinx.

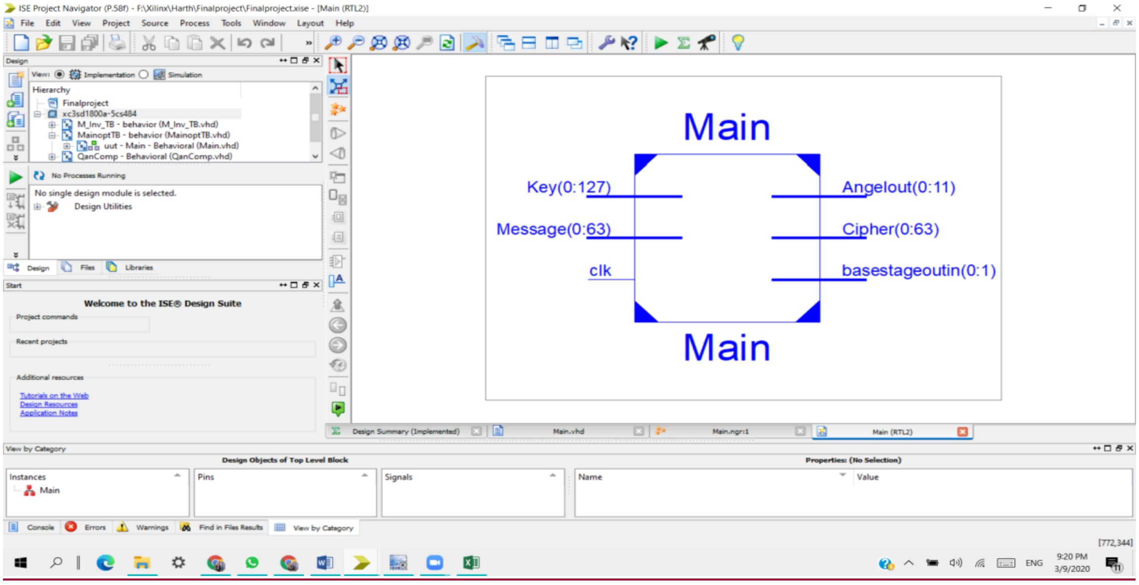

ISE Project Navigator (P.58f) - F:\Xilinx\Harth\Finalproject\Finalproject.xise - [Main (RTL2)]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
Finalproject
xc3sd1800a-5cs484
M_Inv_TB - behavior (M_Inv_TB.vhd)
MainoptTB - behavior (MainoptTB.vhd)
uut - Main - Behavioral (Main.vhd)
QanComp - Behavioral (QanComp.vhd)
No Processes Running
No single design module is selected.
Design Utilities
Design Files Libraries
Start
Welcome to the ISE® Design Suite
Project commands
Recent projects
Additional resources
Tutorials on the Web
Design Resources
Application Notes
Main
Key(0:127)
Message(0:63)
clk
Angelout(0:11)
Cipher(0:63)
basestageoutin(0:1)
Main
Design Summary (Implemented)
Main.vhd
Main.ngr1
Main (RTL2)
View by Category
Design Objects of Top Level Block
Properties: (No Selection)
Instances
Main
Pins
Signals
Name
Value
Console Errors Warnings Find in Files Results View by Category
[772,344]
ENG
9:20 PM
3/9/2020

Figure (9): RTL fig

Table (2) describes the FPGA implementations synthesis for PRINCE algorithm and proposed design PRINCE-BB64.

Algorithm	Block Size	Device	Clock Cycle	Max Freq (MHz)	Throughput (Mbps)	Total Equiv Slices	Efficiency (Mbps/Slice)	Power (Watt)	Delay (ns)
PRINCE[9]	64	Virtex-4FF668	1	31.765	2032	956	2.126	0.165	31.48
PRINCE-BB84	64	Virtex-4FF668	1	31.715	2029	1026	1.978	0.165	31.53
PRINCE-BB84	64	Kintex-7	1	61.42	3931	539	7.290	0.045	16.28

VI. Conclusion

The current study introduced a hardware design of FPGA for the implementation of PRINCE algorithm by using quantum cryptography protocol (BB84). The hardware architecture has been created for encrypting the input data during a single clock cycle with the maintenance of a higher throughput and efficiency, and lower power consumption, ensuring high security depending on the concepts of quantum. Such an economic design is significant in giving a more comprehensive view of the implementation costs, aiming to implement the algorithm in a smart card or other portable devices. Both FPGA board results indicated an increase in throughput and efficiency with a lower power consumption than other studied pointed out, as the results present a throughput and efficiency value of (2.029 Gbps) and (1.9 Mbps/slice) for Virtex-4, and (3.391 Gbps) and (7.29 Mbps/slice) for Kintex-7, all respectively. The rate of power consumption each of Virtex-4 and Kintex-7 were (0.165W) and (0.045 W), respectively.

References

[1] S. K. Lenka, V. Ojha, A. Sharma, "Security of Entanglement Based Version of BB84 protocol for Quantum Cryptography", IEEE, 2010.

[2] R. Alléaume, J. Bouda, C. Branciard, T. Debuisschert, M. Dianati, "Quantum Key Distribution and Cryptography", SECOQC, 2007.

[3] M. Vanitha and Subha, "High Throughput Area Efficient Architecture for Light Weight Cryptography", Advances in Systems Science and Application Vol.15 No.4, 2015.

[4] Y. Li, P. ZHANG, R. HUANG, "Lightweight Quantum Encryption for Secure Transmission of Power Data in Smart Grid", IEEE,2019.

[5] J. Borghoff, A. Canteaut, T. G¨uneysu, E. Bilge Kavun, "PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications", LNCS 7658, 2012.

[6] A. Mink, S. Frankel and Ray Perlner, "Quantum Key Distribution (QKD) and Commodity Security Protocols: Introduction and Integration ", International Journal of Network Security & Its Applications (IJNSA), Vol 1, No 2, July 2009.

[7] P. Winiarczyk, W. Zabierowski, "BB84 analysis of operation and practical considerations and implementations of quantum key distribution systems", CADSM, 2011.

[8] A. Buhari, Z. Ahmad, H. Zainuddin, S. Saharudin, "An Efficient Modeling and Simulation of Quantum Key Distribution Protocols Using OptiSystem", IEEE Symposium on Industrial Electronics and Applications (ISIEA),2012.

[9] Y. Amer Abbas, R. Jidin, N. Jamil, M. Reza Z'aba and M. Ezanee Rusli, "PRINCE IP-core on Field Programmable Gate Arrays (FPGA)", Research Journal of Applied Sciences, Engineering and Technology, 2015.

[10] J. Borghoff, A. Canteaut, T. G¨uneysu, E. Bilge Kavun, "PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications", LNCS 7658, 2012.

[11] R. Selvam, D. Shanmugam, and S. Annadurai, "Side Channel Attacks: Vulnerability Analysis of PRINCE and RECTANGLE using DPA", Hardware Security Research Group,2014.

[12] S. Rana, S. Hossain, H. Imam Shoun, Mohammod Abul Kashem, "An Effective Lightweight Cryptographic Algorithm to Secure Resource-Constrained Devices", (IJACSA) International Journal of Advanced Computer Science and Applications, 2018.

[13] G. Cordone, C. Fabbri, "Quantum Key Distribution Protocol Literature Review and BB84 Simulation ",CS456, 2014.

[14] [1] Abbas, Y. A., Jidin, R., Jamil, N., Z'aba, M. R., Rusli, M. E., & Tariq, B. (2014, November). Implem1entation of PRINCE algorithm in FPGA. In Proceedings of the 6th International Conference on Information Technology and Multimedia (pp. 1-4). IEEE.