Encryption Design Based on FPGA using VHDL

Murtada M. Abdulwahab[1], Abdul Rasoul J. Alzubaidi[2]

[1]Electronic Dept -Faculty of Engineering and Technology- Gezira University-Sudan
[2]Electronic Dept-Engineering College-Sudan University for science and Technology

ABSTRACT
There is a quiet, international battle underway, a battle that impacts every data consumer and producer. The important part of this battle are the cryptographers who work to protect our national security and the privacy of our personal information through increasingly strong methods of encryption, This work, developed a double encryption design using Very high speed integrated circuits Hardware Description Language (VHDL) and a Field Programmable Gate Arrays technology
(FPGA) aimed to ensure privacy to the transmitted data over open networks. The presented work is a type of modern encryption technique. It used a combination of transposition and substitution encryption techniques that help to generate complex cipher. The implementation design was developed and tested with the aid of Xilinx ISE.9.2. The results obtained prove the reliability and applicability of the system. The paper
discussed the provided performance of several FPGA devices.

Keywords
VHDL,ISE9.2i,FPGA, Encryption.

1. INTRODUCTION

Encryption is a process to transform a piece of information into an incomprehensible form. The input to the transformation is called plaintext (or cleartext) and the output from it is called ciphertext (or cryptogram). The reverse process of transforming ciphertext into plaintext is called decryption (or decipherment). Notice that plaintext and ciphertext are a pair of respective notions: the former refers to messages input to, and the latter, output from, an encryption algorithm. Plaintext needn't be in a comprehensible form; for example, in the case of double encryption, a ciphertext can be in the position of a plaintext for re-encryption. Usually, cleartext means messages in a small subset of all possible messages which have certain recognizable distributions. In general ciphers can be distinguished into two types classical or modern according to the type of input data [7]. It should be useful to point out that the two basic working principles of the classical ciphers: substitution and transposition are still the most important kernel techniques in the construction of modern symmetric encryption algorithms. A combinations of substitution and transposition ciphers founded in two important modern symmetric encryption algorithms: Data Encryption Standard (DES) and Advance Encryption Standard ,AES, [1].

A transposition cipher (also called permutation cipher) transforms a message by rearranging the positions of the elements of the message without changing the identities of the elements. Transposition ciphers are an important family of classical ciphers, in additional substitution ciphers, which are widely used in the constructions of modern block ciphers Modern encryption methods can be divided according to two criteria [7]: the type of key used, and the type of input data. By type of key used ciphers are divided into:

- Symmetric key algorithms (Private-key cryptography), where the same key is used for encryption and decryption.
- Asymmetric key algorithms (Public-key cryptography), where two different keys are used for encryption and decryption.

The objectives of this work was to develop a modern technique using a combination of transposition and substitution techniques in order to present an efficient encryption design that help to produce secure cipher and can be more reliable . The paper also aimed to demonstrate how FPGAs can address the need for faster recovery. This encryption algorithm targeted for small embedded applications. It was initially designed for software implementations in controllers, smart cards or processors.

2. MATERIALS AND METHODS

Materials:
The model design synthesizing and implementing (i.e Translate Map & Place and Route) VHDL code with FPGA device are completely done on Xilinx –project navigator ,ISE 9.2i .
Methods:

In a substitution cipher, the encryption algorithm $\epsilon k(m)$ is a substitution function which replaces each $m \epsilon M$ (input data) with a corresponding $c \epsilon C$ (cipher) The substitution function is parameterized by a secret key k. The decryption algorithm Dk(c) is merely the reverse substitution. In general, the substitution can be given by a mapping [1]:

$$\pi : M \rightarrow C \qquad (1)$$

And the reverse substitution is just the corresponding inverse mapping

Manuscript received December 5, 2012
Manuscript revised December 20, 2012

π-1 : C $\rightarrow$ M (2)

The previous substitution function mapping explains the essence of this work. The base of encryption is to use unpredictable methods.

Algorithm:

The described design algorithm illustrated in Fig (1) accepts 128 bit input data that is divided into 16(8-bit) sub-blocks. Two stages of encryption were required to process the accepted data. The design used different procedure for each one. The uses of double encryption stage makes it complex, this provides more authentication on the output cipher. The sequence events of algorithm are applied to each input byte.

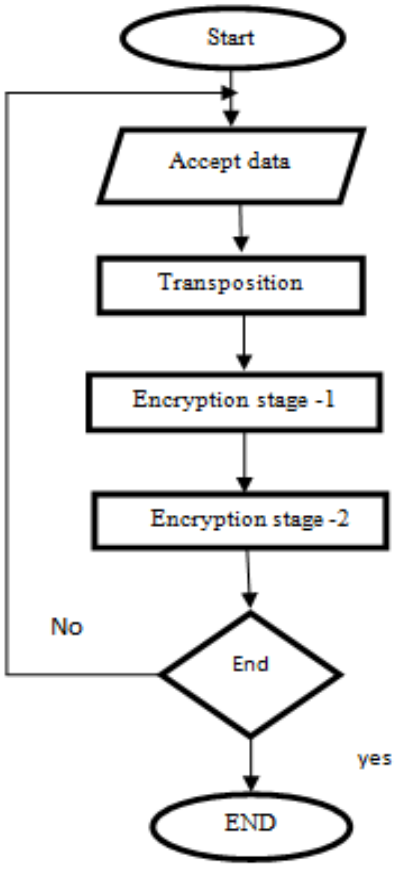

Figure 1: Algorithm Steps

Transposition stage used to improve the design reliability. The used transposition's mechanism is described in fig (2). Each byte changes position following the illustrated path in the chart.

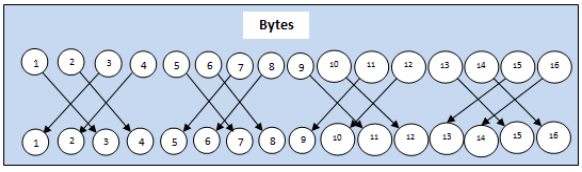

Figure 2 : Transposition chart

Design Overview:

The synthesis process used to transform the VHDL code into model design [4] illustrated in fig (3). It divided into blocks each one used to perform a specific function according to theirs internal logic composition. The design used the resources of the FPGA device. The complete design required about 128 of the 4 input LUTs in addition to 273 IOBs of the available device resources. However the top model design view consists of four inputs and a cipher output. A clock unit links all blocks and used to control the input\output process.

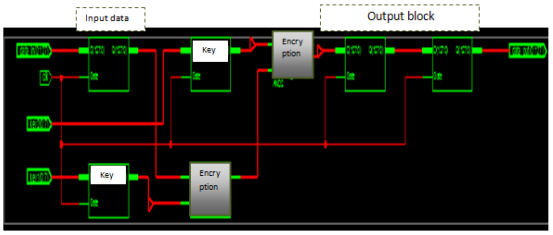

Figure (3): RTL design blocks

In general description the input block is responsible of dividing data into bytes also it performs the transpositions operation, and then data pass through two encryption blocks which consist of a logic combination ,each encryption block has its own key . Finally the output block, this is used to form the output cipher stream.

RESULTS

The simulation process applied using ISE VHDL simulator in order to ensure that the design is well performed. The test bench process results presented in fig (4) show the inputs and outputs parameters, the simulation used an FPGA device of Spartan3 type. The results accuracy approved the ability of using this design.

Device Family: Spatran3

Tools used:

- Xilinx ISE 9.2i
- ISE VHDL simulator.
- Device: xc3s1000-5fg675

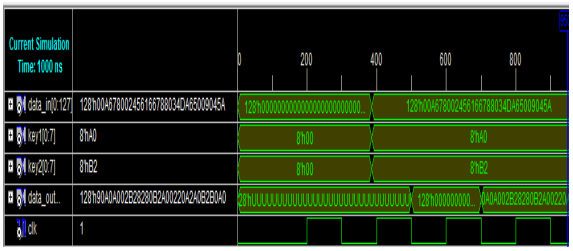

Figure 4: Behavioral simulation

By checking the results obtained from fig(4) it was clear obviously that the design works correctly similar to the previous algorithm illustrated in fig(1) . The test accomplished successfully without errors.

DEVICE PERFORMANCE:
The following discussion focuses on the performance of various FPGA devices. Synthesis is carried out using different family's types of Vertex and Spartan devices. Tables (1.upto 6) show the execution results of each device. The evaluating gives the opportunity to recognize the available resources of each device. It determines utilization ratio and time summary for each one. It is acknowledged that synthesis results are with some (approx.8) warnings which are allowable [6].
Spartan Devices Performance:

Table1: program synthesis using the device xc3s1000-5-fg676

Spartan3(target device xc3s1000 package fg675 speed-5)			
Logic utilization	**used**	**available**	**utilization**
Number of slices	230	7680	2%
Number of 4input LUTs	128	15360	0%
Number of bonded IOBs	273	391	69%
Number of CLOCKs	1	8	12%

Timing Summary:
Minimum period: 4.855ns (Maximum Frequency: 205.977MHz)
Minimum input arrival time before clock: 1.572ns
Maximum output required time after clock: 6.141ns
Total delay of the output data path : 6.141ns (5.460ns logic, 0.681ns route).
The number of signals not completely routed for this design is: 0
The average connection delay for this design is: 1.078
The maximum PIN delay is:
3.584
The average connection delay on the 10 worst nets is: 3.239

Table 2 : program synthesis using the device Xc2s200-6fg456

Spartan2(target device xc2s200 package fg456 speed-6)			
Logic utilization	**used**	**available**	**utilization**
Number of slices	230	2352	9%
Number of 4input LUTs	128	4704	2%
Number of bonded IOBs	273	284	96%
Number of CLOCKs	1	4	25%

Timing Summary:
Minimum period: 9.943ns (Maximum Frequency: 100.573MHz)
Minimum input arrival time before clock: 2.520ns
Maximum output required time after clock: 6.897ns.
Total delay of the output data path: 6.897ns (5.862ns logic, 1.035ns route).
The number of signals not completely routed for this design is: 0
The average connection delay for this design is: 1.296
The maximum PIN delay is:
5.640
The average connection delay on the 10 worst nets is: 4.615

Table 3: program synthesis using the device xc2s50eft256-7.

Spartan2E(target device xc2s50E package ft256 speed-7)			
Logic utilization	**used**	**available**	**utilization**
Number of slices	230	768	29%
Number of 4input LUTs	128	1536	8%
Number of bonded IOBs	273	178	153%
Number of CLOCKs	1	4	25%

More than 100% of Device resources are used (OVERMAPPED).
Virtex Devices Performance:

Table 4 : program synthesis using the device xcv400- 5- bg432 .

Virtex(target device xcv400 package bg432 speed-5)			
Logic utilization	**used**	**available**	**utilization**
Number of	230	4800	4%

slices			
Number of 4input LUTs	128	9600	1%
Number of bonded IOBs	273	316	86%
Number of CLOCKs	1	4	25%

Timing Summary:
Minimum period: 10.841ns (Maximum Frequency: 92.246MHz).
Minimum input arrival time before clock: 2.676ns
Maximum output required time after clock: 7.630ns
Total delay of the output data path: 7.630ns (6.480ns logic, 1.150ns route).
The number of signals not completely routed for this design is: 0
The average connection delay for this design is: 1.710
The maximum PIN delay is :
6.176
The average connection delay on the 10 worst nets is: 5.631

Table 5: program synthesis using the device xc4LX15 -12-FF668 .

Virtex4(target device xc4VLX15 package FF668 speed-12)			
Logic utilization	**used**	**available**	**utilization**
Number of slices	230	6144	3%
Number of slice LUTs	128	12288	1%
Number of bonded IOBs	273	320	85%
Number of CLOCKs	1	32	3%

Timing Summary:
Minimum period: 2.261ns (Maximum Frequency: 442.243MHz)
Minimum input arrival time before clock: 1.037ns
Maximum output required time after clock: 3.839ns
Total delay of the output data path: 3.839ns (3.573ns logic, 0.266ns route).
The number of signals not completely routed for this design is: 0
The average connection delay for this design is: 0.980
The maximum PIN delay is:
2.034
The average connection delay on the 10 worst nets is: 1.581

Table 6: program synthesis using the device xc5vlx85-3-ff676.

Virtex5(target device xc5vlx85 package FF676 speed-3)			
Logic utilization	**used**	**available**	**utilization**
Number of slices Registers	400	51840	0%
Number of slice LUTs	128	51840	0%
Number of bonded IOBs	273	440	62%
Number of CLOCKs	1	32	3%

Timing Summary:
Minimum period: 2.228ns (Maximum Frequency: 448.752MHz)
Minimum input arrival time before clock: 0.846ns
Maximum output required time after clock: 2.702ns
Total delay of output data path: 2.702ns (2.467ns logic, 0.235ns route).
The number of signals not completely routed for this design is: 0
The average connection delay for this design is: 1.320
The maximum PIN delay is:
4.320
The average connection delay on the 10 worst nets is: 3.451

DISCUSSIONS

The discussions aimed to point out the required performance and composition of the FPGA device. The results From Tables (1 upto 5) gives that the design needed to use 230, 128, 273, and 1, respectively to the available slices, 4input LUTs, bonded IOBs and clocks. The results found that the device in table (3) is not recommended to use because the design is too large for the given device and package. Fast speed device is required in such application to obtain fast data recovery. The time reports of the devices in table(1), table(2), table(4) , table(5) and table(6) gives that the provided performance of each device respectively to 205MHZ ,100MHZ, 92MHz , 442MHZ and 448MHz. which demonstrate that vertex5 works faster than other

devices . The device uses 273 IBOs and provides performance upwards of 448 MHz and it is also provides the less output time of the other testing devices. To be more reasonable in our evaluating it should be known that other factors such as power consumption and device cost should be account in particular when the evaluating expands to cover other known designs that work in the same field [2]. The implementation results lead to observe that in terms of area requirements It was no doubt that this design generally exhibits the smallest hardware consumption than the implementing design in [5] .Also it should be noted that the implementing design is almost faster and has less delay time compared to [5]. Finally it was known according to [3] that Spartans devices have low cost than Virtex.

CONCLUSION &RECOMMENDATIONS:

Today's connected society requires secure data encryption devices to preserve data privacy and authentication in critical application, this helps for developing more researches works to improve the ordinary using of encryption techniques . This work develops a secure encryption system that used FPGA technology in order to provide fast recovery .The paper studied most of the known encryption methods. The design performance was examined to different families of FPGA devices. Future work can be performed to allow more complexity by using additional techniques such as using one of the keys as a public key and the other can be used as private key. Also expanding keys will be useful.

REFERENCES

[1] Douglas. S, 'Cryptography: Theory and Practice', CRC Press LLC,(1995).
[2] Dimitrios. M and Ioannis. P, Power consumption estimations vs measurements for FPGA-based security cores.In: proceeding of International Conference on Reconfigurable Computing and FPGAs, Chania, Greece, (2008).
[3] Deming.C, Jason.C, and Peichan. P, " FPGA Design Automation: A Survey", now Publishers In, (2006).
[4] PONG P. C, 'RTL HARDWARE DESIGN USING VHDL Coding for Efficiency, Portability, and Scalability' , Cleveland State University, A John Wiley & Sons, INC , (2006).
[5] Rabie . A.M , Magdi . S, " Hardware Implementation of the Stone Matamorphic cipher " , International Journal of Computer Science & Network Security vol 10 (8):54-60, (2010).
[6] Sounak Samanta B.E, 'FPGA Implementation of AES Encryption and Decryption,Electronics & Communication Engg, ,Sardar Vallabhbhai National Institute of Technology,surat,(2002).
[7] Wenbo. M.H, 'Modern Cryptography Theory and Practice', Packard Company, Prentice Hall PTR ,(2003).