INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATION ENGINEERING & TECHNOLOGY (IJECET)

ISSN 0976 – 6464(Print)
ISSN 0976 – 6472(Online)
Volume 5, Issue 10, October (2014), pp. 34-44

Journal Impact Factor (2014): 7.2836 (Calculated by GISI)
www.jifactor.com

IJECET

© I A E M E

HARDWARE IMPLEMENTATION OF THE SERPENT BLOCK CIPHER USING FPGA TECHNOLOGY

Mai Hossam Taher[1]

Electronics and Communications Dept.Mansoura University, Egypt

Ali E.Taki El_Deen[2]

IEEE senior member, Alexandria University, Egypt

Mohy E.Abo-Elsoud[3]

IEEE senior member, Electronics and Communications Dept. Mansoura University, Egypt

ABSTRACT

Governments, military, corporations, financial institutions, hospitals and private businesses amass a great deal of confidential information about their employees, customers, products, research and financial status. Most of this information is now collected, processed and stored on electronic computers and transmitted across networks to other computers. This leads to the need of securing data from any unauthorized access. This paper presents the Serpent encryption and the proposed decryption algorithms. The results are simulated and implemented design using ModelSim 6.5 and Xilinx ISE 14.2 platform. The new Xilinx Spartan-6 is speed up the Serpent algorithm and reduce the power consumption. Comparisons with other Symmetric Block Algorithms are discussed.

Keywords: Cryptography, Serpent, AES, Mars, Twofish, RC6, FPGA, Hardware Implementation.

1. INTRODUCTION

Cryptography is referred to as study of secret. This is a process where a readable message is converted into a form which is unreadable to others except for the one it is intended to. Whenever confidential information is sent, there is possibility of an unauthorized third party attack in order to learn the confidential information. Cryptanalysis is the process by which an unintended receiver discovers the decryption process, and thereby the plaintext[1].

Modern encryption techniques start with the Data Encryption Standard (DES) which was developed in the early 1970s at IBM and based on an earlier design by Horst Feistel. But it is expired

now because DES has a relatively small 56-bit key which was becoming vulnerable to brute force attacks. In addition, the DES was designed primarily for hardware and is relatively slow when implemented in software[2].

While Triple-DES avoids the problem of a small key size, it is very slow even in hardware; it is unsuitable for limited-resource platforms; and it may be affected by potential security issues connected with the (today comparatively small) block size of 64 bits.

For these reasons, the US National Institute of Standards and Technology has issued a call for a successor algorithm, to be called the Advanced Encryption Standard or AES. The essential requirement is that AES should be both faster than triple DES and at least as secure: it should have a 128 bit block length and a 256 bit key length (though keys of 128 and 192 bits must also be supported).Fifteen algorithms were submitted to the First AES Candidate Conference in August 1998[3].One year later, NIST announced the five finalists: MARS, RC6™, Rijndael, Serpent and Twofish.

In 2006, Rijndael has also received some criticism suggesting that its mathematical structure might lead to attacks in the future.

Because of above problems that lead to using **Serpent** block cipher which has the highest security factor and simplest design equation.

Serpent designed by Ross Anderson, Eli Biham and Lars Knudsen as a candidate for the Advanced Encryption Standard. It was a finalist in the AES competition. Serpent and Rijndael are somewhat similar[4].Although it was not finally selected, Serpent was considered very secure and with a high potential in hardware implementations[5].

A field-programmable gate array (FPGA) is a large-scale integrated circuit that can be programmed after it is manufactured rather than being limited to a predetermined, unchangeable hardware function. The term "field-programmable" refers to the ability to change the operation of the device "in the field," while "gate array" is a somewhat dated reference to the basic internal architecture that makes this after-the-fact reprogramming possible.

FPGA chip adoption across all industries is driven by the fact that FPGAs combine the best parts of application-specific integrated circuits (ASICs) and processor-based systems. FPGAs provide hardware-timed speed and reliability.

The paper is organized as: Section (2) introduces Serpent Encryption Algorithm. Section (3) presents the proposed Serpent Decryption Algorithm. Section (4) discusses Specifics of FPGA Implementation. Section (5) Comparisons with other Symmetric Block Cipher are given. Section (6) discusses Experimental Results. Section (7). The advantages of Spartan-6 are represented. Finally the paper is concluded in section (8).

2. SERPENT ENCRYPTION ALGORITHM

Serpent is a 32-round substitution permutation network (SPN) operating on four 32-bit words, thus having a block size of 128 bits[3].In fig.1, Serpent encrypts a 128-bit plaintext to a 128-bit ciphertext in 32 rounds with 33 sub keys. The user key length is assumed to be variable but in the proposal, it is fixed to be 128, 192 or 256 bits. It should be mentioned that the short keys with less than 256 bits are mapped to 256 bits keys by appending one ’1’ bit to the MSB end followed by as many ’0’ bits as required to produce 256 bits. The cipher consists of an initial permutation IP, 32 rounds, and a final permutation FP. Each round involves a key mixing operation, a pass through S-boxes, and a linear transformation. In the last round, the linear transformation is replaced by an additional key mixing operation. The whole data path from the plaintext *P* to the cipher text *C* can be formally described by a sequence of the following equations[6]:

$$B0 := IP(P) \quad (1)$$

$$Bi+1 := LT(SBoxi \bmod 8(B \oplus Ki)), i = 0 \ldots 30 \quad (2)$$

$$B32 := SBox7(B31 \oplus K31) \oplus K32 \quad (3)$$

$$C := FP(B32) \quad (4)$$

2.1. Key Mixing

At each round, a 128-bit sub key Ki is XORed with the current intermediate data Bi Fig.2.

2.2. The S-Boxes

Serpent has 8 individual 1×16 S-Boxes which repeat every 8 round. Every 128-bit input of the S-Box will be divided to 32 blocks of 4-bits and every block will be applied to a 4×4 S-Box. The outputs of these 32 S-Boxes will be concatenated again together to perform a 128-bit block.

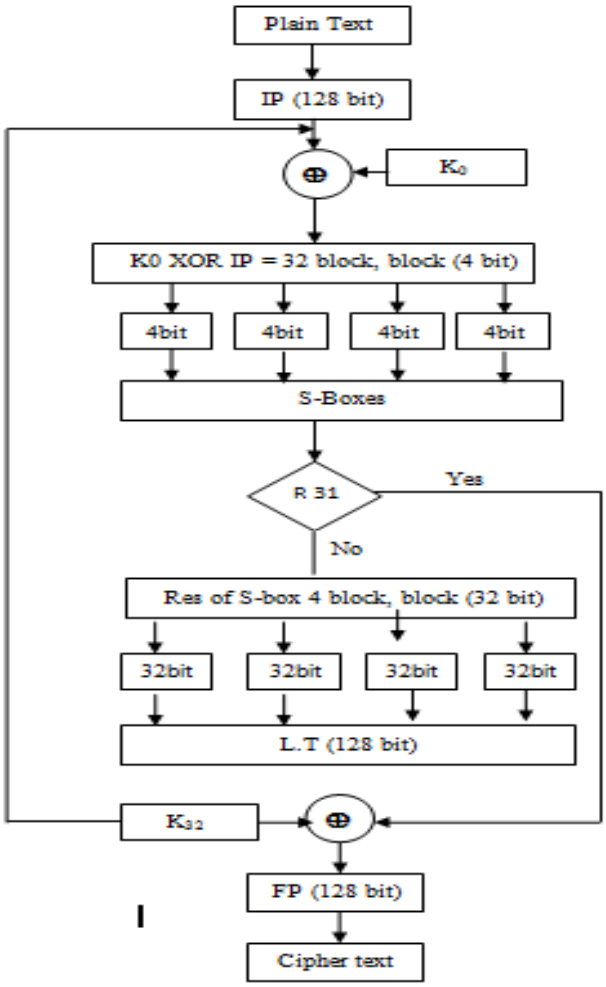


Fig 1: Serpent Encryption

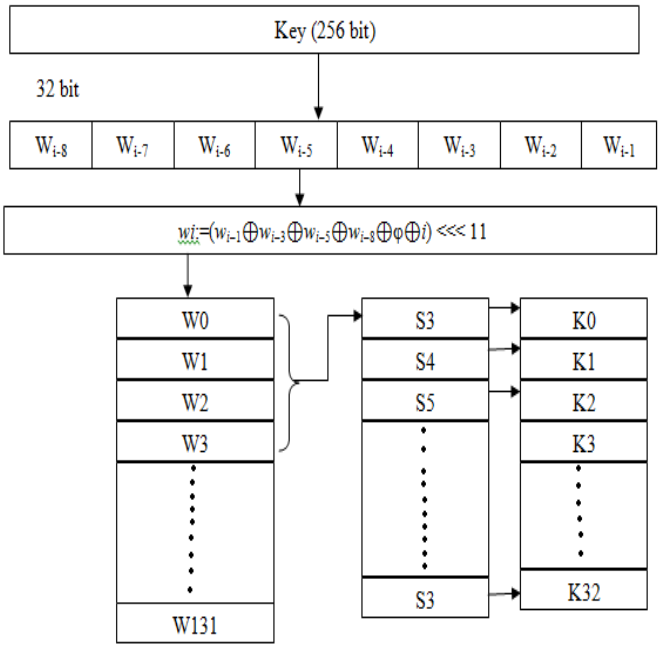


Fig 2: Key Mixing

2.3. Linear Transform

The 128-bit output of S-Box will be divided to four 32-bits and these words are linearly mixed by some shift, rotate and XOR operations. A complete round of Serpent is described as [6]:

$$X0, X1, X2, X3 := (Bi \oplus Ki)$$
$$X0 := X0 <<< 13$$
$$X2 := X2 <<< 3$$
$$X1 := X1 \oplus X0 \oplus X2$$
$$X3 := X3 \oplus X2 \oplus (X0 << 3)$$
$$X1 := X1 <<< 1$$
$$X3 := X3 <<< 7$$
$$X0 := X0 \oplus X1 \oplus X3$$
$$X2 := X2 \oplus X3 \oplus (X1 << 7)$$
$$X0 := X0 <<< 5$$
$$X2 := X2 <<< 22$$
$$Bi+1 := X0, X1, X2, X3$$

Where <<< meansLeft Rotation and << means LeftShift. In the last round, this linear transform is replaced by an additional key mixing.

$$B32 := SS7(B31 \oplus K31) \oplus K32$$

3. THE PROPOSED SERPENT DECRYPTION ALGORITHM

Decryption is different from encryption in that the inverse of the S-boxes must be used in the reverse order, as well as the inverse linear transformation and reverse order of the sub keys Fig.3.

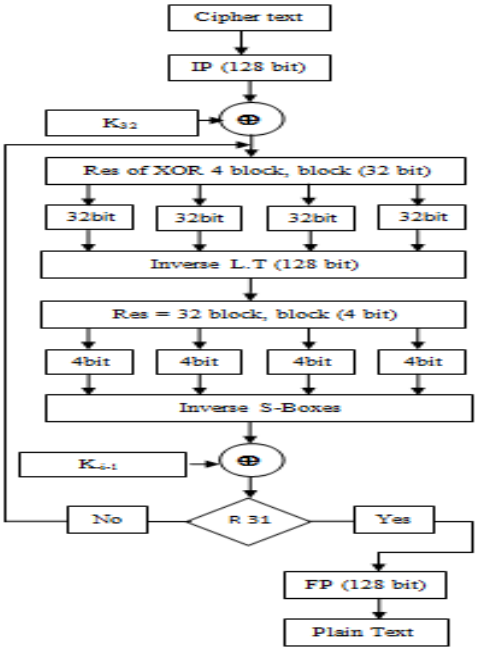


Fig 3: Decryption

3.1. Inverse Linear Transform

The 128-bit will be divided to four 32-bits and these words are linearly mixed by some shift, rotate and XOR operations. A complete round of Serpent is described as:

$$X0, X1, X2, X3 := (Bi \oplus Ki)$$
$$X2 := X2 >>> 22$$
$$X0 := X0 >>> 5$$
$$X2 := X2 \oplus X3 \oplus (X1 << 7)$$
$$X0 := X0 \oplus X1 \oplus X3$$
$$X3 := X3 >>> 7$$
$$X1 := X1 >>> 1$$
$$X3 := X3 \oplus X2 \oplus (X0 << 3)$$
$$X1 := X1 \oplus X0 \oplus X2$$
$$X2 := X2 >>> 3$$
$$X0 := X0 <<< 13$$
$$Bi+1 := X0, X1, X2, X3$$

Where >>> means RightRotation and << means LeftShift.

4. SPECIFICS OF FPGA IMPLEMENTATION

Field-programmable gate arrays (FPGAs) are reprogram- able silicon chips. Ross Freeman, the cofounder of Xilinx, invented the first FPGA in 1985.

FPGA devices are a highly promising alternative for implementing private-key cryptographic algorithms. Compared with software-based implementations, FPGA implementations can achieve superior performance [7].

For serpent algorithm to access effective implementation in both software and hardware, some specific aspects of FPGA architecture must be chosen. In all FPGA devices from this producer,

so called *Look-Up Table* (LUT) is the element located in every logic cell which is provided for generation of any combinational function. Xilinx Spartan-6 XC6SLX45 is chosen.
In Spartan-6 architecture, in turn, every LUT table has the total capacity of 64b being sufficient for generation of a 6-input Boolean function but, alternatively, can be configured for generation of two different 5-input functions of the same variables. These configuration nuances will have significant impact on implementation of the cipher transformations[8].

VHDL stands for VHSIC Hardware Description Language, and VHSIC in turn stands for Very High Speed Integrated Circuits. VHDL is an acronym for Very High Speed Integrated Circuit Hardware Description Language which is a programming language used to describe a logic circuit by function, data flow behaviour, or structure[9].

VHDL aims at modeling or documenting electronics systems. Due to the nature of hardware components which are always running, VHDL is a highly concurrent language, built upon an event-based timing model. A VHDL program can be transformed with a synthesis tool into a netlist, that is, a detailed gate-level implementation[10].

Serpent code was implemented in Xilinx ISE Design suite version 14.2.

5. COMPARISON BETWEEN 128-BITS SYMMETRIC BLOCK CIPHERS

In our implementations, we focused on the safety factor performance because the security is the most important factor in the evaluation. Our throughput and Area results are compared with the FPGA-based results inNIST final report [11]. Only the cryptographic core of serpent algorithm was implemented using FPGA.

Table 1: Differences between 128 symmetric block ciphers

Cipher	Serpent	RC6	Rijndael	Twofish	Mars
Structure	Substitution permutation network	Feistel network	Substitution-permutation network	Feistel network	Feistel network
Key Size	128, 192, 256-bits	128, 192, 256-bits	128, 192, 256-bits	Up to 256-bits	Up to 448-bits
No.round	32	20	10	16	32
No.SubKeys	33	44	11	42	40
No.SBoxes	8	None	1	8	2
Size.SBoxes	64 bits (16 x 4)	-	2048 bits (256 x 8)	64 bits (16 x 4)	8192 bits (256 x 32)
Total ROM bits	512	-	2048	512	16384
Speed	69	43	20	18	34
Safety Factor	3.56	1.18	1.56	2.67	1.90
Area [Kgates]	504	1,643	613	432	2,936
Through-put [Mbit/s]	932	204	1,950	394	226
Key Setup Time(µs)	0.09	0.15	0.20	0.25	3.12

6. EXPERIMENTAL RESULTS

Serpent architectures were implemented in Spartan-6 XC6SLX45 from Xilinx which was selected as a representative test platform and it served this role very well.
Briefly the Implementation of the Serpent algorithm based on FPGA has the following steps:

- Create design used VHDL codeusing FPGA Adv8.1(Mentor Graphics tools).
- SimulateRTL(Register Transfer Level) with ModelSim SE 6.3.
- Synthesize RTL design to target technology with precision synthesis.
- Design flow to the Xilinx – Project Navigator, ISE 14.2, entry using EDIF file (electronic data interchange format) from Mentor Graphics tools.
- Implement design (Transfer, map, place and route).Once a design is implemented, generate a bitstream file, then bit file can be downloaded.

The simulation results are shown in Fig.4, 5. The RTL results for SBOX and Linear Transformation are shown in Fig6, 7, 8, 9. RTL for final encryption block in Xilinx ISE is shown in Fig10.

Advantages of Spartan-6:

Spartan-6 FPGA increased performance, density, evolutionary feature enhancements and Dramatic cost and power reductions[12].

Table 2: Differences between Xilinx families

Feature	Spartan-3A (90nm)	Spartan-6 (45nm)
Logic Cells(Kbit)	Up to 55k	Up to 150k
LUT Design	4-input LUT +2FF	6-input LUT +2FF
Block RAM(Mbit)	Up to 2 Mbit	Up to 5 Mbit
Transceiver count / Speed	No	Up to 8 / Up to 3.125 Gbps
Memory Interface	400Mbps	DDR3 800Mbps
Max Differential IO	640 Mbps	1050 Mbps
Multipliers/DSP	Up to 126 Multipliers / DSP	Up to 184 DSP 48 Blocks
Memory Controllers	No	Up to 4 Hard Blocks
Clock Management	DCM only	DCM & PLL
Security	Device DNA only	Device DNA & AES

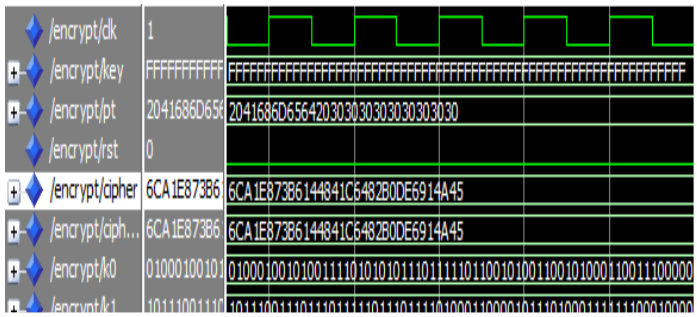
1-Text encryption:
Plaintext:

Mai Hossam Taher Ahmed

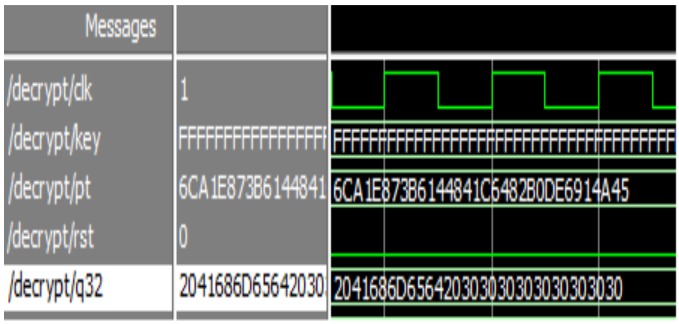
Ciphertext:

¾yI2é-)yÐF°ùö“´l¡ès¶HAÆH+
æ‘JE+ 6ï nGO´øEØî–+ 6ï nGO´øEØî–+ 6ï nGO´øEØî–+ 6ï nGO´øEØî–
+ 6ï nGO´øEØî–+ 6ï nGO´øEØî–

Simulation Results:



(a)



(b)

Fig 4: Pre-Synthesis for text using serpent (a) Encryption, and (b) Decryption

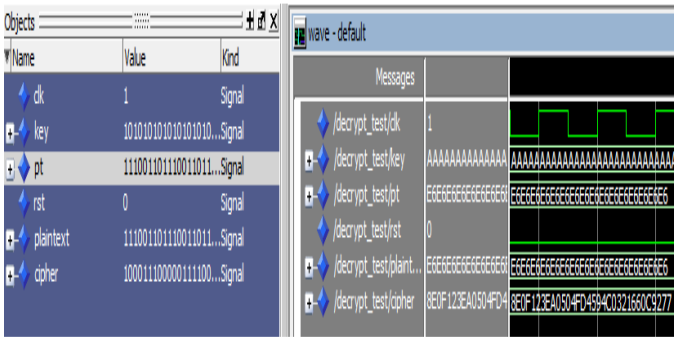


Fig 5: Pre-Synthesis for Image using Serpent Encryption Decryption Block

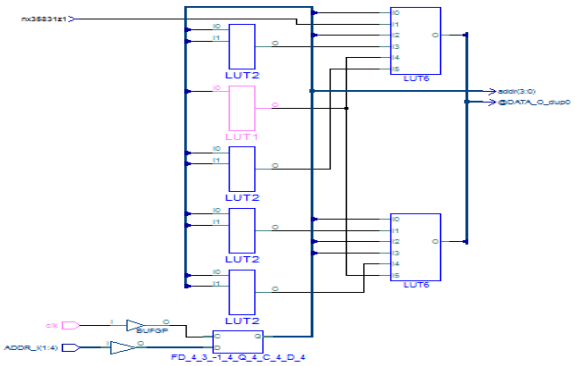


Fig 6: RTL Schematic for Serpent Sbox

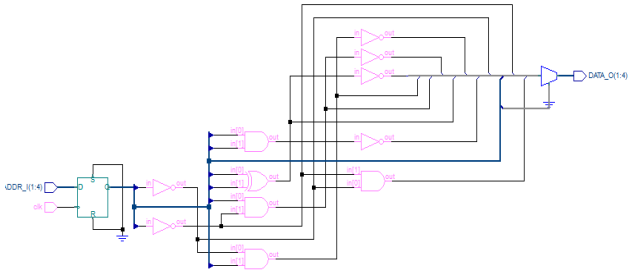


Fig 7: Technology Schematic for Serpent Sbox

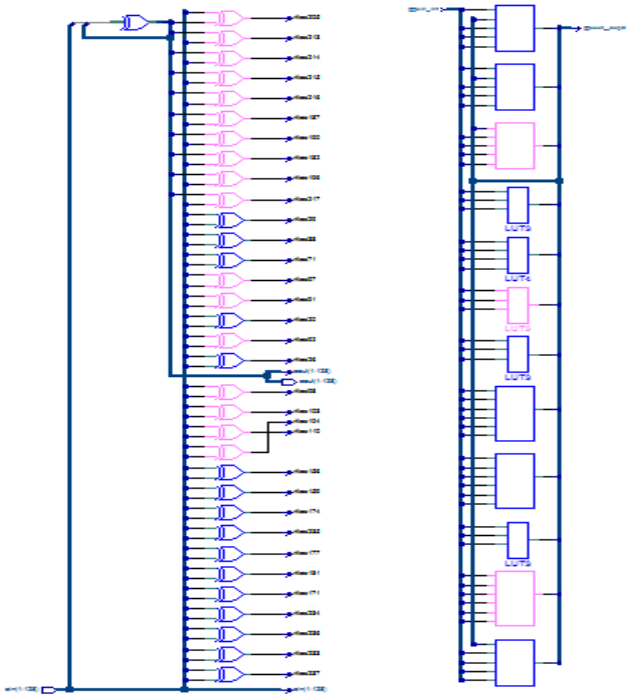


Fig 8: RTL Schematic and Technology Schematic of Serpent Linear Transformation

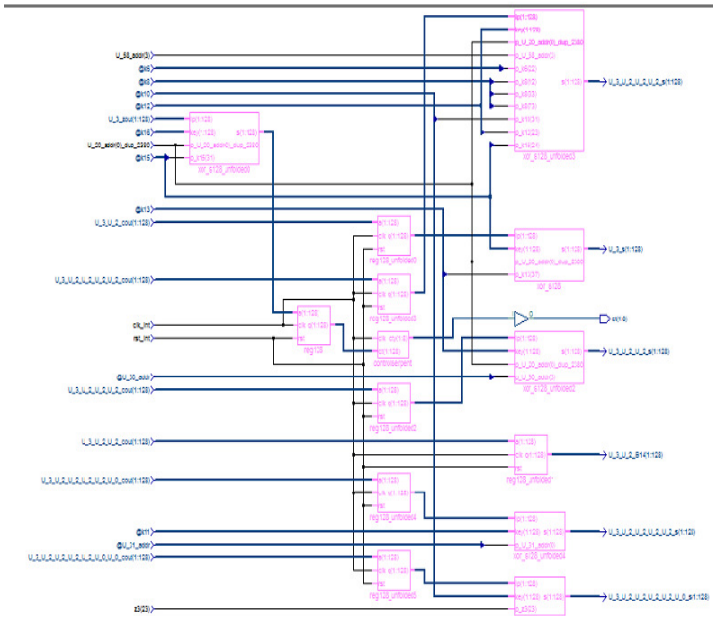


Fig 9: Mentor Graphics Technology Schematic serpent Encryption

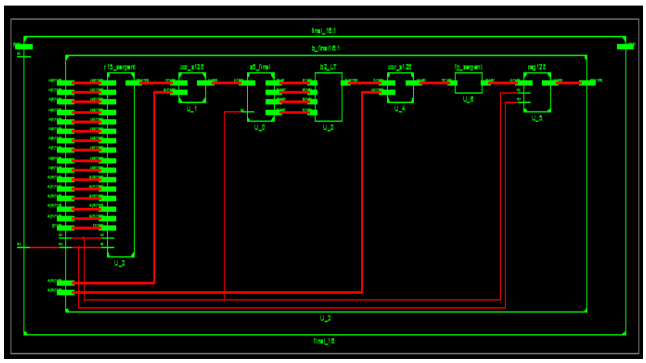
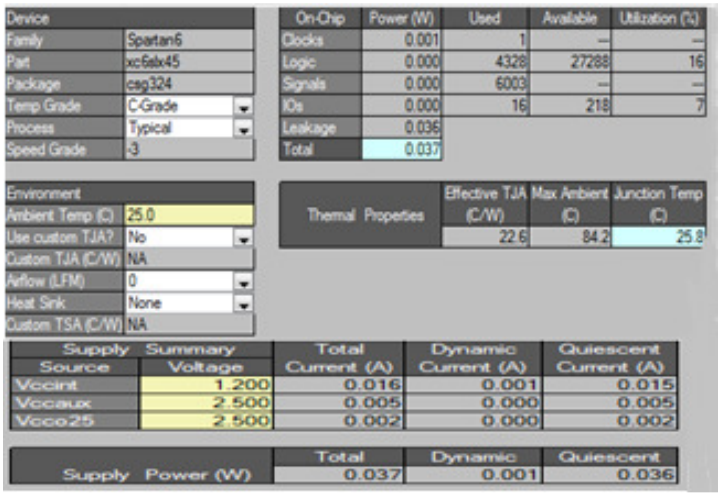


Fig 10: Xilinx RTL Schematic of Serpent Encryption

Table 3: Xilinx ISE XPower Analyzer for SERPENT Encryption Algorithm on Spartan6-XC6SLX45



Device		On-Chip	Power (W)	Used	Available	Utilization (%)
Family	Spartan6	Clocks	0.001	1	--	--
Part	xc6slx45	Logic	0.000	4328	27288	16
Package	csg324	Signals	0.000	6003	--	--
Temp Grade	C-Grade	IOs	0.000	16	218	7
Process	Typical	Leakage	0.036			
Speed Grade	-3	Total	0.037			

Environment		Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
Ambient Temp (C)	25.0		22.6	84.2	25.8
Use custom TJA?	No				
Custom TJA (C/W)	NA				
Airflow (LFM)	0				
Heat Sink	None				
Custom TSA (C/W)	NA				

Supply Summary Source	Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.016	0.001	0.015
Vccaux	2.500	0.005	0.000	0.005
Vcco25	2.500	0.002	0.000	0.002

Supply Power (W)	Total	Dynamic	Quiescent
	0.037	0.001	0.036

Table 4: Resource used in the FPGA Implementation of AES using Spartan-3A, Virtex-II, Virtex-5, and Spartan-6

		Spartan-3A,	Virtex-5	Spartan-6
Process		90nm	65nm	45nm
Static Power Consumption (mw)		450	1209	370
IOS	**Used**	15	15	15
	Avail.	372	640	218
Global Buffers	**Used**	1	1	1
	Avail.	24	32	32
LUTs	**Used**	9071	7044	4278
	Avail.	11776	69120	27288
CLB Slices	**Used**	5279	1870	1494
	Avail.	5888	17280	6822
Block RAMs	**Used**	0	0	0
	Avail.	20	400	116

7. CONCLUSION

No doubt, the AES algorithm is faster, but the Serpent algorithm is more secure. Comparisons with other Symmetric Block Cipher have been investigated. In this paper, we have successfully implemented the Serpent algorithm using Spartan-6 FPGA device from Xilinx. Finally, the results have been shown that the hardware implementation is faster than the software as well as increase system security. In addition, it is noted that they save logic area and reduce the power consumption.

REFERENCES

[1] Richard A. Mollin, "An Introduction to Cryptography", Second Edition, ISBN: 1584886188 / 9781584886181, 2005.

[2] http://www.britanncia.com/EBchecked/topic/145058/cryptology/233467/The-Data-Encryption-Standard-and-the-Advanced-Encryption-Standard.

[3] Anderson, R., Biham, E. & Knudsen, L. (2000). The Case for Serpent. Proc. Third AES Candidate Conf.New York, http://csrc.nist.gov/archive/aes/ index.html (accessed April 2012).

[4] KGaj, P.Chodowiec, "Comparison of the hardware performance of the AES candidates using reconfigurable hardware", The Third Advanced Encryption Standard Candidate Conference, April 13–14, 2000, New York, USA (proceedings availablefromhttp://csrc.nist.gov/encryption/aes/round2/conf3/aes3conf.htm), 2000.

[5] Jürgen Becker, Marco Platzner, Serge Vernalde," Field Programmable Logic and Application", 14th Edition, ISBN: 978-3-540-22989-6, 2004.

[6] Anderson, R., Biham, E. & Knudsen, L. (1998).Serpent: A Proposal for the Advanced Encryption Standard. Proc. First Advanced Encryption Standard (AES) Candidate Conf. Ventura,California, http://www.cl.cam.ac.uk /~rja14/ serpent.html (accessed April 2012).

[7] Implementation of symmetric block ciphers in popular-grade FPGA Devices, Journal of Polish Safety and Reliability Association Summer Safety and Reliability Seminars, Volume 3, Number 2, 2012.

[8] JarosławSugier, "LOW-COST PROGRAMMABLE HARDWARE SOLUTIONS IMPROVING CONFIDENTIALITY IN COMPUTER SYSTEMS", Proceedings of the 13th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'13), 16–19 October 2013, Riga, Latvia, p. 387–396, ISBN 978-9984-818-58-0.

[9] Bibilo, P.N. and Avdeev, N.A., VHDL. Effektivnoeispol' zovaniepriproektirovaniitsifrovykh sistem (VHDL. Effective Use in Design of Digital Systems), Moscow: SOLON_Press, 2006.

[10] Peter J. Ashenden, "VHDL Tutorial", Elsevier Science (USA), 2004.

[11] Bryan Weeks, Mark Bean, Tom Rozylowicz, Chris Ficke, "Hardware Performance Simulations of Round 2 Advanced Encryption Standard Algorithms", National Security Agency (**NSA**).

[12] Xilinx, Inc. (2011). Spartan-6 Family Overview. DS160.PDF, www.xilinx.com (accessed April 2012).

[13] Rahul Jassal, "Wrapped RSA Cryptography Check on Window Executable using Reconfigurable Hardware", International Journal of Computer Engineering & Technology (IJCET), Volume 3, Issue 3, 2012, pp. 291 - 299, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.

[14] Ahmad Salameh Abusukhon, "Block Cipher Encryption for Text-To-Image Algorithm", International Journal of Computer Engineering & Technology (IJCET), Volume 4, Issue 3, 2013, pp. 50 - 59, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.