FPGA Simulation of Type-3 Feistel Network of The 128 bits Block Size Improved Blowfish Cryptographic Encryption

Ashwaq Talib Hashim*, Yousra A. Mohammed* & Dr.Ekhlas H. Karam***

Received on:13/5/2009
Accepted on:7/1/2010

Abstract

Reprogrammable devices such as Field Programmable Gate Arrays (FPGAs) are highly attractive options for hardware implementations of encryption algorithms as they provide cryptographic algorithm agility, physical security, and potentially much higher performance than software solutions , therefore this paper investigates a hardware design to efficiently implement block ciphers in VHDL based on FPGAs. This hardware design is applied to the new secret-key block cipher called 128-bits improved Blowfish is proposed which is an evolutionary improvement of 64-bits Blowfish designed to meet the requirements of the Advanced Encryption Standard (AES) to increase security and to improve performance. The proposed algorithm will be used a variable key size up to 192 bytes. It is a Type-3 Feistel network iterated simple function 16 times.

The resources used to implement the design just described are: the VHDL hardware description language, an FPGA platform from Xilinx and the Xilinx Synthesis Technology (XST) software synthesis tools that belong to ISE 9.2i package. The device of choice is the XCV600-4fg680 belonging to the Virtex family of devices.

In this paper, a pipeline and sequential methods are used to get a high througput (2.893Gbps) and a low area hardware design respectively.

Keywords: Cryptographic, Blowfish, VHDL, and FPGA.

محاكاة FPGA لشبكة فيستيل نوع 3 وحجم كتلة 128- bits لخوارزمية التشفير المطورة Blowfish

الخلاصة

الماديات القابلة للبرمجة مثل رقاقة المصفوفات المبرمجة (FPGA) خياراتَ جذّابة جداً لتطبيقاتِ خوارزمياتِ التشفير كما يُوقّرونَ أمن طبيعي ، وأداء أعلى بكثير فعلاً مِنْ حلول البرامج ، لذا هذه المقالة تَتحرّى تطبيق تصميم مادي بشكل كفؤء لخوارمية تشفير جديدة بأستخدام لغة وصف الماديات (VHDL) المستندة على رقاقة المصفوفات المبرمجة. هذا التصميم المادي يطبق على خوارزمية تشفير كتلية ذات المفتاح السري مسماة 128-bits Blowfish وهو تحسين تطويري

*Control and Systems Engineering Department, University of Technology/ Baghdad
**Electrical & Electronic Engineering Department, University of Technology/ Baghdad
*** College Engineering, University of Al-Musttansira / Baghdad

https://doi.org/10.30684/etj.28.9.1
2412-0758/University of Technology-Iraq, Baghdad, Iraq

الى 64-bits Blowfish صممت لتحقيق متطلبات معيار التشفير المتقدم(AES) لزيادة الامنية وتحسين الاداء. الخوارزمية المقترحة سوف تستخدم مفتاح متغيّر يقدّر حجمه إلى حد 192 بايت. وهي شبكة فيستيل نوع 3 يكرر وظيفة بسيطة 16 مرة.
المصادرالتي تُستَعملُ لتَطبيق التصميمِ المَوْصُوف: لغة وصفِ الماديات رقاقة المصفوفات المبرمجة مِنْ Xilinx (XCV600-4fg680) وتقنية تأليفِ Xilinx (إكس إس تي) أدوات تأليفِ برامج الذي يَعُودان إلى برنامج ISE9.2i.

1.Introduction

FPGA (**Field Programmable Gate Arrays)** is the chosen implementation platform due to its proven advantages, such as fast prototyping and advanced reconfigurability. From several FPGA families available on the market, in this paper high performance Virtex family from Xilinx, Inc. Is chosen the targeted FPGA for this implementation is VIRTEX XCV600-4fg680 FPGA.

FPGA devices from this family consist of thousands of universal building blocks, known as *Configurable Logic Blocks* (*CLBs*), connected using programmable interconnects. Reconfiguration is able to change a function of each CLB and connections among them, leading to a functionally new digital circuit. Each CLB slice contains a small block of combinational logic, implemented using programmable look-up tables, and two one-bit registers [1]. Additionally, Virtex FPGAs are SRAM-based, and are customized by loading configuration data into internal memory cells. In some modes, the FPGA reads its own configuration data from an external PROM (master serial mode). Otherwise, the configuration data is written into the FPGA (Select-MAP™, slave serial, and JTAG modes). Finally, Virtex devices provide better performance than previous generations of FPGA. Designs can achieve synchronous system clock rates up to 200 MHz including I/O [2].

This paper presents some existing VHDL implementation. G. Umamaheswari and Dr. A.Shanmugam [3] address efficient hardware implementation approaches for the AES (Advanced Encryption Standard) algorithm and describes the design and performance testing Rijndael algorithm. Krishnamurthy G.N [4] attempts to develop a simple, stronger and safer cryptographic algorithm which is called "Blow-CAST-Fish". VHDL implementation is used and tested to show percentage improvement in the performance of the modified Blow-CAST-Fish. . Faez F. Shareef [5] presents the implementation of RC6 block cipher that involve the encryption and decryption on FPGA Vertix II device with highly compact architecture through reuse the same units for the identical operation in both algorithms.

2. Hardware Implementation using VHDL

VHDL (**V**ery High Speed Integrated Circuit **H**ardware **D**escription **L**anguage) was chosen as a language used to describe the improvement suggested to the

algorithm stated above. VHDL has many features appropriate for describing the behavior of electronic components ranging from simple logic gates to complete microprocessors and custom chips. Features of VHDL allow electrical aspects of circuit behavior (such as rise and fall times of signals, delays through gates, and functional operation) to be precisely described [6].

This paper also is presenting a mix of VHDL architecture (structural and behavioral) in order to write the deign code of the 16 rounds encryption and decryption that describe the 128-bits block size improved blowfish algorithm. The presented architecture allows keeping the flexibility of the algorithm by taking advantage of generic VHDL coding. It executes one round per clock cycle, computes the round and the key round in parallel and supports both encryption and decryption at a minimal cost.

3. Type-3 Feistel Network of The 128-bits Block Size Improved Blowfish Encryption Algorithm

The algorithm takes four 32-bit plaintext data words A, B, C, D as input and produces four 32-bit ciphertext data words A, B, C, and D. The cipher is word-oriented, in that all the internal operations are performed on 32-bit words. This algorithm is a type-3 Feistel network iterated simple function 16 times (See figure 1).

This cipher uses a variety of operations to provide a combination of high security, high speed, and implementation flexibility. It uses also four key dependent (S-box) tables of 255 32-bit words to provide good resistance against linear and differential attacks, as well as good avalanche of data and key bits.

In each round the output of F-function is the input to E-function then, one data word will be used as the input to the E-function and the three output words from the E-function are added or XORed to the other three data words. In addition, the source word is rotated by 13 positions to the left. The algorithm uses the same structure of F-function of previous Blowfish algorithm [7].

3.1 The E-function

The E-function takes as input one data word and uses two more key words to produce three output words. In this function three temporary variables will be used, denoted below by L, M and R (for left, middle and right). Below it is also refer to these variables as the three "lines" in the function. Initially, R will be set to hold the value of the source word rotated by 13 positions to the left, and M will be set to hold the sum of the source word and the first key word. Then the lowest nine bits of M will be viewed as an index to S-boxes (the S-box is chosen by using this function (*number of round mod 4*), and set L to hold the value of the corresponding S-box entry.

The second key word is multiplied (constrained to contain an odd integer) into R and then rotate R by 5 positions to the left (so the 5 highest bits of the product becomes the 5 lowest bits of R after the rotation). Then R XORed into L, and also view the five lowest bits of R as a rotation amount between 0 and 31, and rotate

M to the left by this amount. Next, R will be rotated R by 5 more positions to the left and XOR it into L. Finally, again the five lowest bits of R will be viewed as a rotation amount and rotate L to the left by this amount. The first output word of the E-function is L, the second is M and the third is R as shown in figure 2 [7].

4. Design Principle

This section describes the main techniques conceived to implement the hardware design of the 128-bits block size improved blowfish algorithm.

4.1 Implementation of Encryption and Decryption

In many cases, a sequential (pipeline) design allows time sharing of the same part of the circuit between various functions, which can significantly reduce the amount of area consumed by this design. Moreover the sequential design may increase the encryption speed directly proportionally to the number of pipelined stages. Our project is consists of 16 *round* circuit modules. The *round* circuit describes exactly one round of the improved cipher algorithm, including encryption and decryption in order to give a better conceive to the benefit achieved from the hardware implementation that exploit resources between encryption and decryption algorithms. The *round* can be run repeatedly to perform entire encryption or decryption process.

The sequence of data flow operations of the round circuit is shown in figure (3), and example of functional simulation of this circuit is shown in figure (4), this simulation is done by ModleSim simulator program.

4.2 Subkeys Implementation

The 128-bits block size improved blowfish uses a large number of subkeys. These keys must be precomputed before any data encryption or decryption. In this paper, to simplify hardware design, the key scheduling is assumed to be performed off-chip.

1-The key array also called P-array consists of 48 32-bit subkeys: P0, P1, and P47. These keys are stored as a part of encryption and decryption modules.

2-There are four 32-bit S-boxes with 256 entries each:

Substitution is an important operation in symmetric block ciphers to add maximum non-linearity. In fact the strength of DES algorithm is based on its substitution boxes (SP-Boxes).

In this paper RAM solution is chosen to store the contents of S-box (S1, S2, S3 and S4) as shown in figure (5). The input to this RAM are five 8_bit address and the outputs are five 32_bit data four of them are used to computed the output of F_function and the other is used in the E_function to computed L output.

4.3 The Pipelined Datapath

In this paper, at every clock cycle the architecture receives a plaintext block as input and processes it as it goes through the stages that make up the pipeline to produce a ciphertext block. The design can process different blocks simultaneously in the different stages of its datapath. At every clock cycle a ciphertext block leaves the pipeline

from the last stage and the first stage receives a new input plaintext block. This architecture reaches the best performance owing to the exploitation of temporal and spatial parallelism when processing plaintext blocks.

Pipelining has been shown to be an effective method of dramatically increasing the throughput capabilities of a given algorithm. However, it comes at the expense of limiting the number of cryptographic modes that can be supported at the maximum throughput rate [8].

For this paper XST reports an estimated operational clock frequency of 22.599 MHz for the whole design. From this information it is possible to compute the throughput, which turns out to be 2.893 Gbps.

The general block diagram, schematic diagram, and the detail design diagram of the hardware implementation of 128-bits block size improved blowfish algorithm are shown in figures (6),(7) and (8) respectively.

4.4 Inputs and Outputs

Table (1) shows the description of the inputs and outputs of the hardware proposed design and how they are used.

5. Implementation results

The 128-bit improved blowfish block cipher is implemented Xilinx VIRTEX XV600-4fg680 Field Programmable Gate Array (FPGA). All the architectures are modeled in VHDL, verified the functionality using Modelsim VHDL simulator and synthesized the data paths using ISE 9.2i software synthesis tool and Xilinx place and route tool. Table (2) demonstrated device utilization as provided by implementation reports. While the timing and delay specification is summarized by the synthesis report are shown in table (3).

6. Conclusions

This paper is carried out a design for implementing the 16 rounds of Type-3 Feistel Network of the *128-bits block size Improved Blowfish Cryptographic Encryption* and pipeline them in order to optimize the frequency (22.599MHz) and throughput (2.893 Gbps) results. Another strong focus is placed on low area circuits and to perform this sequential designs with very low area requirements are implemented.

In the hardware design both encryption and decryption use the same blocks. The only functional difference, for encryption the key quarter is being shifted to the left. Decryption as a reverse process should start at the point where encryption has stopped and then go through the same key schedule backwards. Naturally it assumes shifting the registers in the opposite direction. This circuit would have 259 pins. Only input message and output cipher block give 2×128= 256 pins but the others are control pins.

7. References

[1] M.McLoone and J.V.McCanny, *Single-Chip FPGA Implementation of the Advanced Encryption Standard Algorithm*, in the proceedings of FPL: The Field Programmable Logic Conference, Lecture Notes in Computer Science, LNCS 2147, p.152ff, 2002.

[2] Xilinx, Inc., "Virtex 2.5 V Field Programmable Gate Arrays", April 2,

2001. Available at http:// www.xilinx.com
[3] Mrs.G.Umamaheswari and Dr.A.Shanmugam "Efficient VLSI implementation of the block cipher Rijndael algorithm",Academic Open Internet Journal Volume 12, 2004 http:// www.acadjournal.com
[4] Krishnamurthy G.N , V. Ramaswamy , Leela G.H and Ashalatha M.E," Blow-CAST-Fish: A New 64-bit Block Cipher", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.4, p. 282,April 2008.
[5] Faez F. Shareef, Ashwaq Talib Hashim and waleed F. Shareef, "Compact Hardware Implementation of FPGA Based RC6 Block Cipher", Journal of Engineering and Applied Sciences: 3(7):589-601,2008 http:// medwellonline@gmail.com
[6] Altium, Inc., "VHDL Language Reference"; Technical Reference; TR0114 (v1.2) September, 2005.
[7] Ashwaq T. Hashim "Type-3 Feistel Network of The 128-bits Block Size Improved Blowfish Cryptographic Encryption", Eng & Tech, Journal, Vol. 27, No. 2, 2009
[8] W. Semancik, L. Mercer, T. Hoehn, G. Rowe, M. Smith-Luther, R. Agee, D. Fowlkes, and J. Ingle ,"Cell Level Encryption for ATM Networks and Some Results from Initial Testing," , *DoD Fiber Optics Conference*, March 19

Table (1) Description of the Inputs and Outputs

Inputs	
Clk	**1-bit input clock signal**
Input data	**The input data is 128 bits. In encryption mode, this is the plaintext. In decryption mode, this is the ciphertext.**
RESET	**1-bit input signal. This should not be asserted at the same time as ENC_DEC.**
ENC_DEC	**1bit - This signal toggles between encryption and decryption operation. 1 means encrypt, 0 means decrypt.**
Outputs	
Output data	**The output data is 64 bits. In encryption mode, this is the ciphertext In decryption mode, this is the plaintext.**

Table (2) FPGA device utilization

Category	Amount of Elements used	Total Amount of Elements available	Percentage Of use
Clcks	**1**	**4**	**25%**
Slices	**6370**	**6912**	**92%**
Slice Flip flops	**2048**	**13824**	**14%**
IOs	**259**	**512**	**50%**
4-input LUTs	**11489**	**13824**	**87%**

Table (3) timing and delay specification

Timing Summary	
Speed Grade	-4
Minimum perioc	44.250ns
Maximum Frequency	22.599MHz
Minimum input arrival time before cloc	63.981ns
Maximum output required time after cloc	8.426ns
Maximum combinational path dela	No path found
Throughput	2.893 Gbps

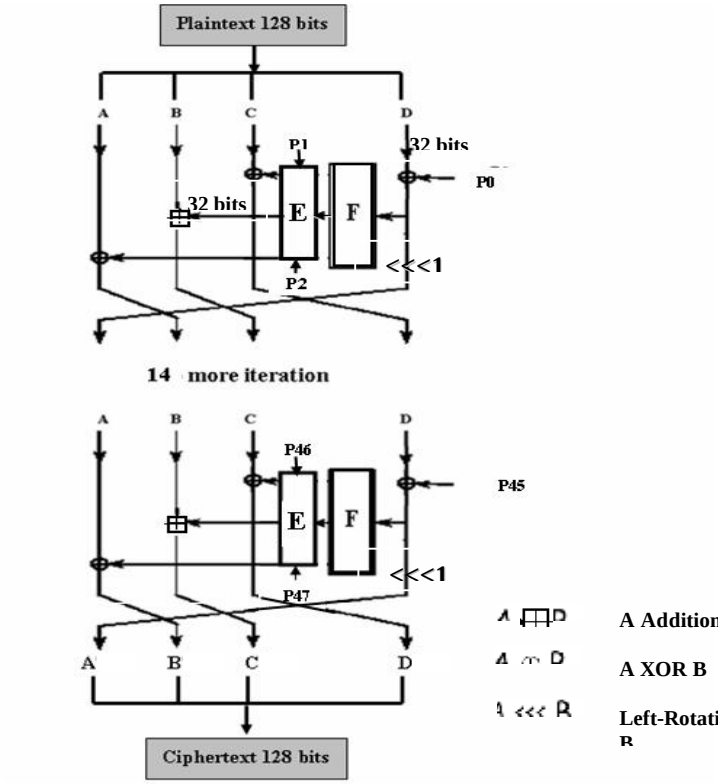

Figure (1) Type-3 Feistel Network of The 128-bits block size Improved Blowfish Encryption Algorithm

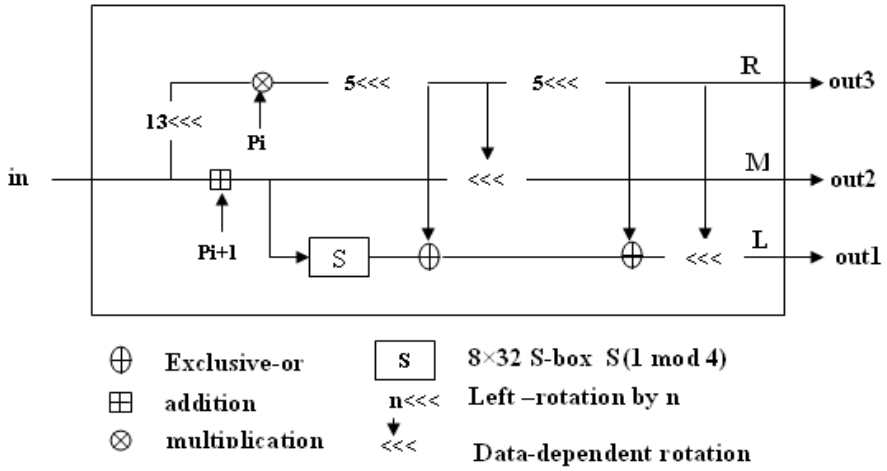

Figure (2) The E-function

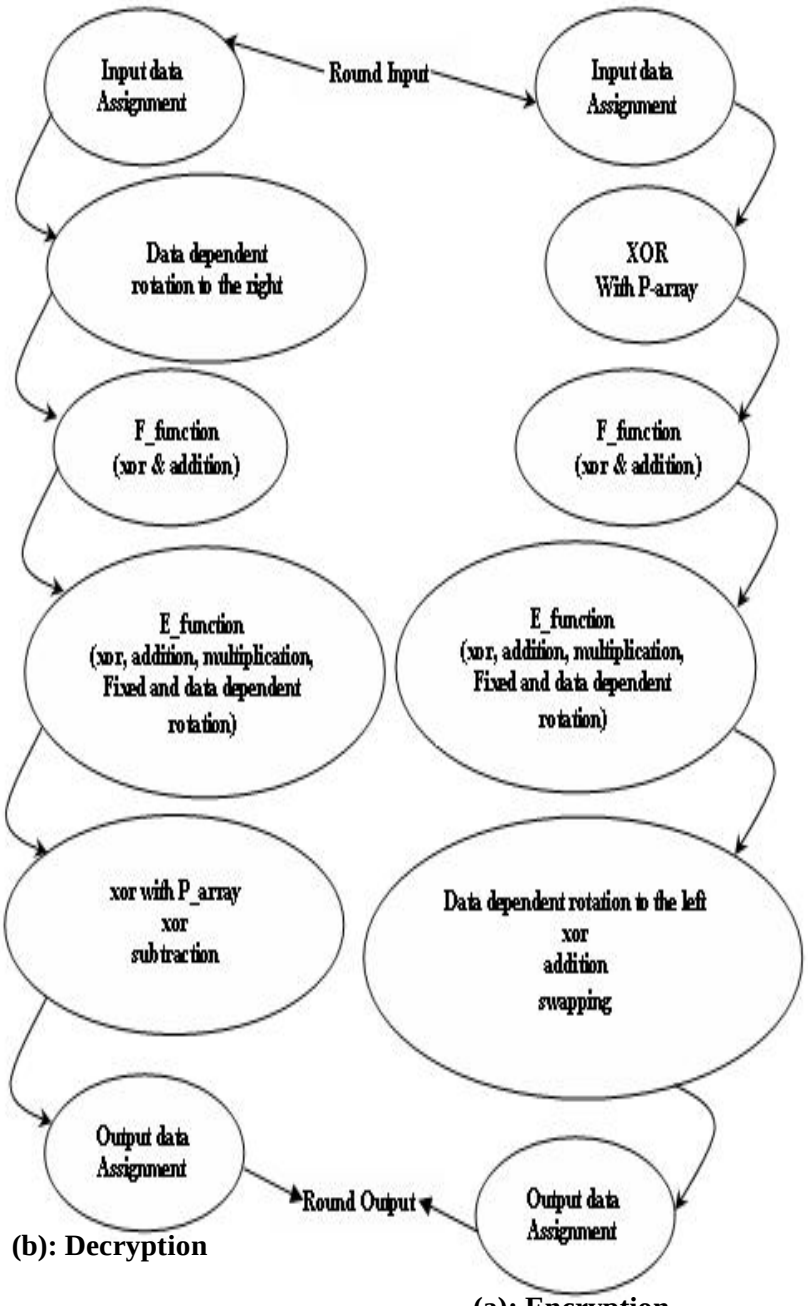

Figure (3) Data flow operations of hardware round circuit (Encrypt& Decrypt Algorithms)

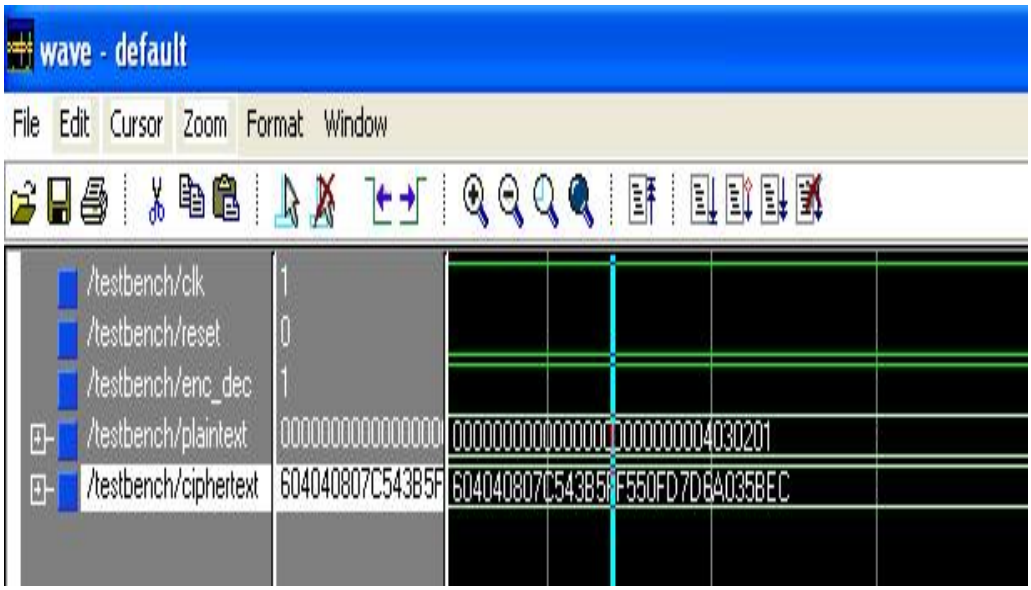

Figure (4-a) Encryption operation

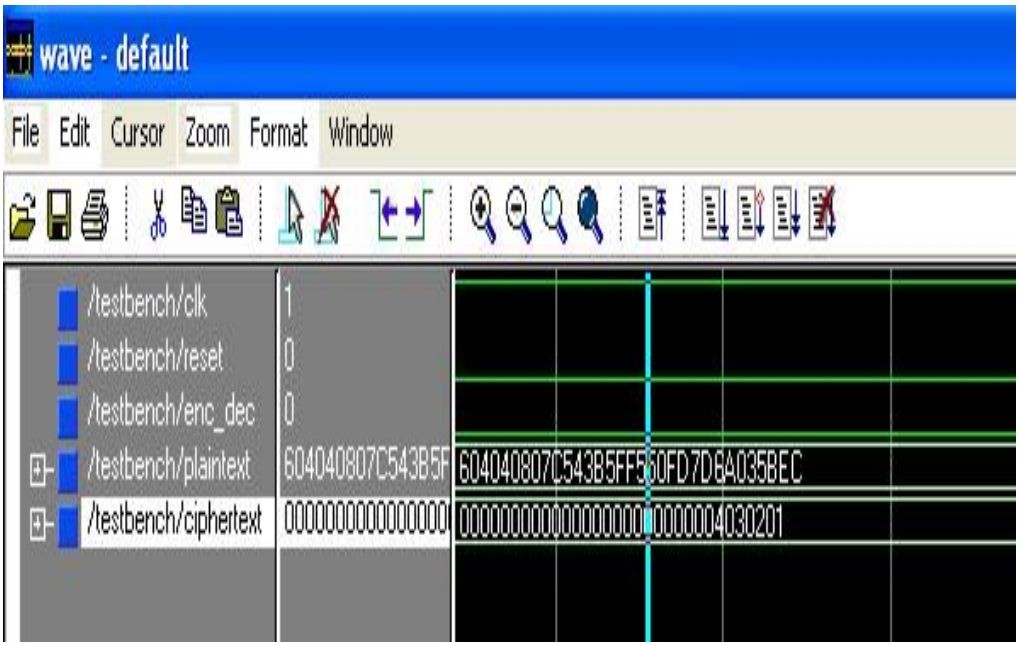

Figure (4-b) Decryption operation

Figure (4) Functional Simulation for one round ɑcuit: (a) Encryption and (b) Decryption

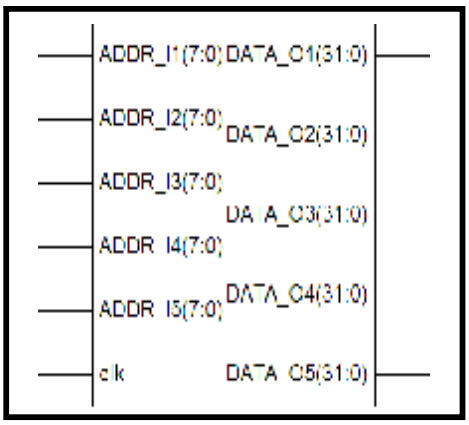

Figure (5) RAM Implementation for storing S-BOXS

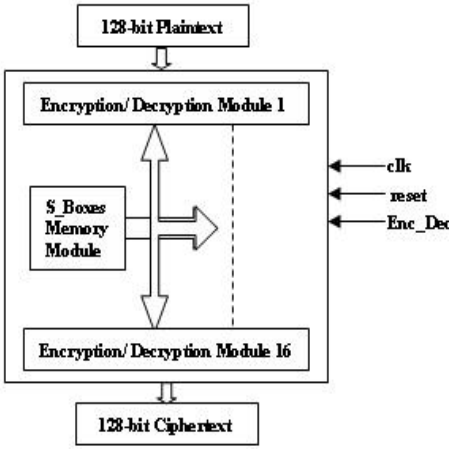

Figure (6) schematic block diagram of 128-bits block size Improved Blowfish Cryptographic Encryption

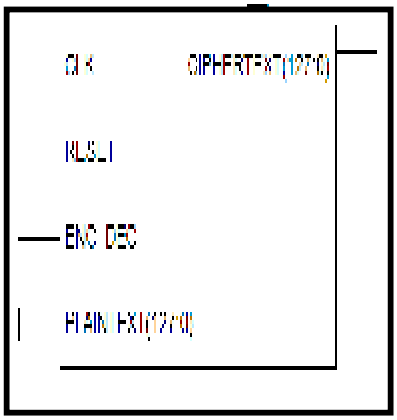

Figure (7) general block diagram of 128-bits block size Improved Blowfish Cryptographic Encryption

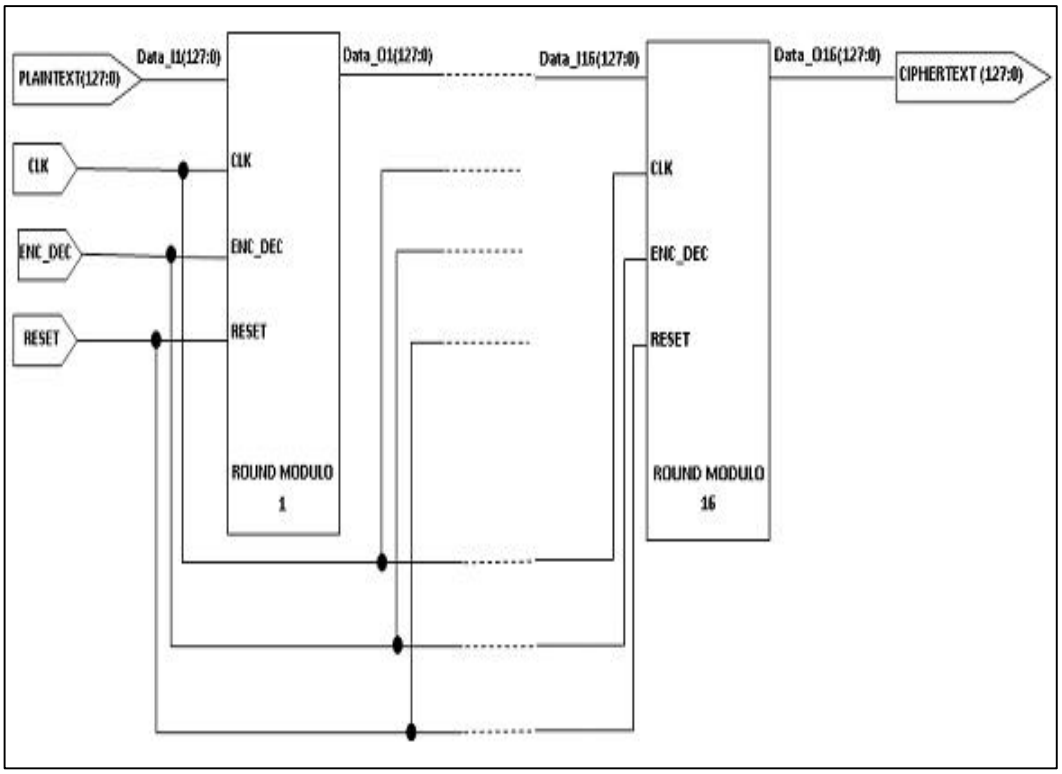

Figure (8) Pipeline implementation diagram of 128-bits block size improved Blowfish Cryptographic Encryption