FPGA Implementation of Optimized the 64-BIT RC5 & RC6 Cryptography Encryption Algorithm

Rubina Soni, Sarabjit Singh

Abstract- ***In today's era there is a great demand for secure communications systems, which in turns demand for real-time implementation of cryptographic algorithms. In this paper we present a hardware implementation of the RC6 & RC5 algorithm using VHDL Hardware Description Language. We also explore the competence of RC6 & RC5 from the hardware implementation perspective with Field Programmable Gate Arrays (FPGAs) as the end technology. 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. Our analysis and synthesis studies of the ciphers will suggest that RC5 & RC6 are fast block ciphers, developed by RSA Security, which exploits data rotation to achieve a high level of nonlinearity. 128 bit key and 12 rounds of operation in the design will provide great security to the user's data. The optimization of area, timing and power is done during physical design process.***

Index Terms—Cryptographic Algorithms, RC5, RC6.

I. INTRODUCTION

With the wireless communication being the emerging technology, the need to have secure data transmission is of great importance. Today, it is important that data is sent confidentially over the network without fear of hackers or unauthorized access to it. This makes implementing security in networks a vital demand. There are a numerous encryption algorithms that are now commonly used in computation. Here RC5 & RC6 cipher patented by RSA Security is presented and implemented. A hardware system design is proposed to improve its performance. RC5, a fast block cipher, was proposed in 1994, which exploits data rotation to achieve high level of nonlinearity. RC5 symmetric cipher provides data protection via the use of a secret key only known to the encryption and decryption ends of the communication path[1]. RC6 is a symmetric key block cipher derived from RC5. It was designed by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin to meet the requirements of the Advanced Encryption Standard (AES) competition by the National Institute of Standards and Technology (NIST). The algorithm was one of the five finalists, and was also submitted to the NESSIE CRYPTREC projects [2]. Though the algorithm was not eventually selected, RC6 remains a good choice for security applications. It is proprietary of RSA Security. Generally, implementing ciphers in software based on its speed in terms of computation is not efficient and hence the use of hardware devices is an alternative.

Revised Version Manuscript Received on May 28, 2015.

Student Rubina Soni, ECE, PTU/ Guru Gobind Singh College of Engg. &Technology Talwandi Sabo/Bathinda, India.

Astt. Prof. Sarabjit Singh, ECE Guru Gobind Singh College of Engg. &Tech, Talwandi Sabo. /Bathinda, India.

The market for consumer and wireless devices is changing rapidly with the convergence of applications, standards and usage. The complexity challenges are forcing the design of devices at the sub-micron technology. The VLSI industry made this feasible using ASIC and FPGA technology with the help of on-chip memories. This not only has the advantage of added security but also increases speed of operation as it decreases the data transaction latency between the core and the memory. The RC5 & RC6 algorithms are written in VHDL and are simulated using Modelsim. After generating gate level netlist using Xilinx, FPGA implementation is carried out with Xilinx.

II. BRIEF OVERVIEW OF RC5 ALGORITHM

RC5 is a symmetric block cipher with data dependent cyclic rotation which provides high degree of nonlinearity. Notation of particular RC5 algorithm is RC5-w/r/b, in this design RC5-32/12/16 is employed which indicates 32-bit words, 12 rounds and 16-byte secret key. The input and output blocks will be 2^w=64bits. RC5 has three modules: key expansion, encryption and decryption. It is the latest in a family of secret key cryptographic methods; RC5 is more secure than RC4 [3 17ami] but is slower. Generally, implementing ciphers in software is not efficient based on its speed in terms of computation and hence the use of hardware devices is an alternative [4 9ami]. The RC5 algorithm uses three primitive operations and their inverses. These are:

(1) Addition/subtraction of words modulo 2W, where w is the word size.

(2) Bit-wise exclusive-OR of words denoted by XOR.

(3) Rotation: the rotation of word x left by y bits is denoted x<<<y. The inverse operation is the rotation of word x right by y bits, denoted by x>>>y.

A. Key Expansion

The b-byte secret key is stored into an array L [0…c-1] of c=b/u words, where u=w/8 is the number of bytes per word. The two magic constants P_w and Q_w are used to generate an array S [0…t-1] where t=2r+2.

P_w=odd $((e-2)2^w)$

Q_w=odd $((\phi-2)2^w)$

where e= 2.718281828459(base of natural logarithms) ϕ=1.618033988749(golden ratio)

Algorithm for key expansion is given as

```
S[0]=$P_w$
for i = 1 to t-1 do
S[i] = S[i-1] + $Q_w$
i = j = 0;
A = B = 0;
do 3 * max(t, c) times:
A = S[i] = (S[i] + A + B ) <<< 3;
B = L[j] =(L[j] +A+B) <<< (A + B);
i = (i + 1) mod (t);
j = (j + 1) mod(c);
```

B. Encryption

The input block to RC5 consists of two w-bit words given in two registers, A and B. The output is also placed in the registers A and B. As explained in [6],

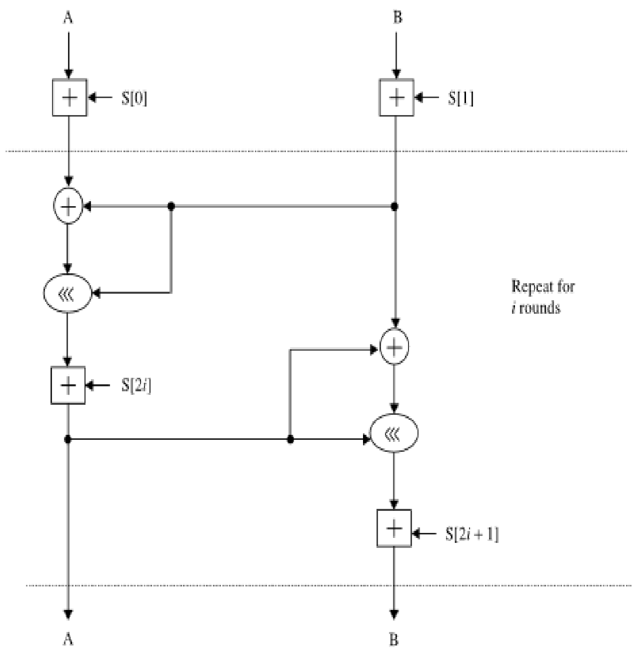

Fig 1: RC5 Encryption

RC5 uses an expanded key table, S[0, 1, . . . , t − 1], consisting of t = 2(r + 1) words. The key-expansion algorithm initializes S from the user's given secret key parameter K. However, the S table in RC5 encryption is not like an S-box used by DES. The encryption algorithm is given in the pseudo code as shown below:

```
A = A + S[0];
B = B + S[1];
for i = 1 to r do
A = ((A ı B) <<< B) + S[2i];
B = ((B ı A) <<< A) + S[2i + 1];
The output is in the registers A and B.
```

C. Decryption

The decryption process takes a cipher-text as the input and produces a plaintext as the output. The decryption routine is easily derived from the encryption routine. In this routine the inverse formulation of encryption routine is processed.

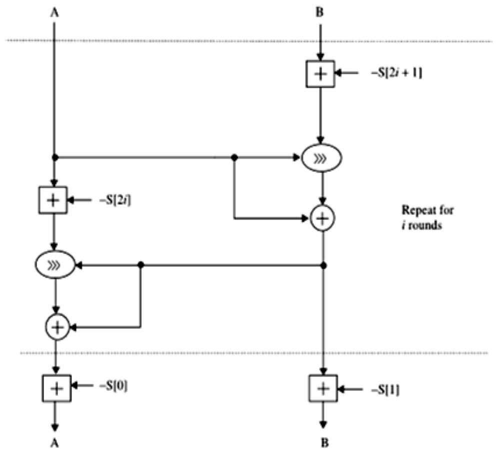

Fig. 2 : RC5 Decription

III. STRUCTURE OF THE RC6 CIPHER ALGORITHM

A. Basic Operations

RC6-*w/r/b* operates on units of four w-bit words using the following six basic operations[5]. The base-two logarithm of w will be denoted by lg w.

- a + b integer addition modulo 2^w
- a - b integer subtraction modulo 2^w
- a $\oplus$b bitwise exclusive-or of w-bit words
- a X b integer multiplication modulo 2^w
- a<<<b rotate the w-bit word a to the left by the amount given by the least significant lg w bits of b
- a>>>b rotate the w-bit word a to the right by the amount given by the least significant lg w bits of b

B. Key Schedule

The user supplies a key of b bytes. From this key, 2r + 4 words (w bits each) are derived and stored in the array S [0, 2r+ 3]. This array is used in both encryption and decryption.

C. Encryption

Input:

- Plain text stored in four w-bit input registers A, B, C, D
- Number r of rounds
- w-bit round keys S [0, … ,2r + 3]

Output:

- Cipher text stored in A, B, C, D

Procedure:

```
B = B + S [0]
D = D + S [1]
For  i = 1 to r do{
```

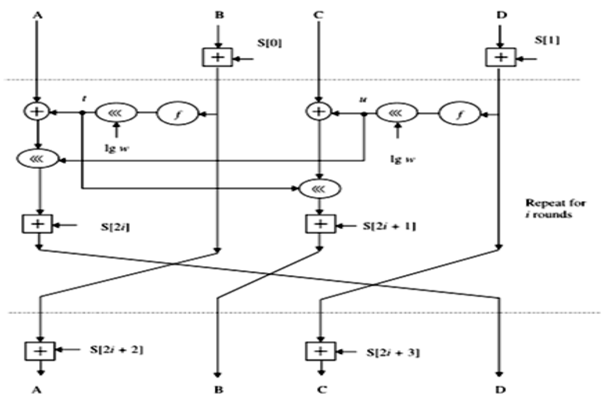

Fig 3: RC6 Encryption

```
t = (B X (2B + 1)) <<< lg w
u = (D X (2D + 1)) <<< lg w
A = ((A ⊕ t) <<< u) + S [2i]
C = ((C ⊕ u) <<< t) + S [2i+ 1]
(A, B, C, D) = (B, C, D, A)
}
A = A + S [2r + 2]
C = C + S [2r + 3]
```

D. Decryption

Input:

- Cipher text stored in four w-bit input registers A, B, C, D

- Number r of rounds

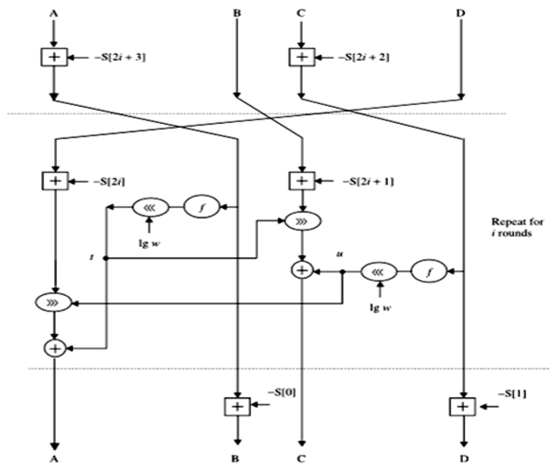

Fig 4: RC6 Decryption

- w-bit round keys S[0; … ; 2r + 3]

Output:

- Plaintext stored in A, B, C, D
- Procedure:

```
C = C – S [2r + 3]
A = A – S [2r + 2]
for i = r down to 1 do
{
    (A, B, C, D) = (D, A, B, C)
    u = (D X (2D + 1)) <<< lg w
    t = (B X (2B + 1)) <<< lg w
    C = ((C – S [2i + 1]) >>> t) Åu
    A = ((A – S [2i]) >>> u) Å t
}
D = D – S [1]
B = B – S [0]
```

IV. FPGA Vs. ASICS

Field Programmable Gate Arrays (FPGAs) consist of arrays of configurable logic blocks that implement logical functions of gates that are easily reconfigurable. In contrast, Application Specific Integrated Circuits (ASICs) provide only the functionality needed for a specific task. An ASIC chip will support a particular application for which it is designed, but not a modified version of the same application introduced after the ASIC design is completed. On the other hand, the configuration of an FPGA can be easily reprogrammed to accommodate a design modification. Other key factors that favour the use of FPGAs for hardware implementation of ciphers include faster turnaround design time, scalable security, and variable architecture parameters. For those reasons we have chosen FPGAs as the target technology.

V. SELECTION OF A TARGET FPGA

Scalability and cost must be considered. We believe that the chosen FPGA should be the best chip available, capable of providing the largest amount of hardware resources as well as being highly flexible so as to yield optimal performance. Unfortunately, the cost associated with current high-end FPGAs is relatively high (several hundred US dollars per device). However, it is important to note that the FPGA market has historically evolved at an extremely rapid pace, with larger and faster devices being released to industry at a constant rate. This evolution has resulted in FPGA cost-curves that decrease sharply over relatively short periods of time[6]. Based on the mentioned considerations, the Xilinx Virtex XCV1000BG560-4 FPGA was chosen as the target device. The XCV1000 has 128K bits of embedded RAM divided among thirty-two RAM blocks that are separate from the main body of the FPGA. The 560-pin ball grid array package provides 512 usable I/O pins. The XCV1000 is comprised of a 64 x 96 array of look-uptable based Configurable Logic Blocks (CLBs), each of which acts as a 4-bit element comprised of two 2-bit slices for a total of 12288 CLB slices[7]. This type of configuration results in a highly flexible architecture that will accommodate the round functions' use of wide operand functions.

VI. DEVELOPMENT ENVIRONMENT

For this project we use VHDL Hardware Descriptive Language to simulate the hardware implementation of the RC6 algorithm on FPGAs. We used the CAD tools available at GKU.

The design process can be divided into the following stages:

- Use Xilinx project navigator to generate the VHDL descriptions of the RC5 & RC6 cipher using both behavioural and structural architectures, as well as the functional VHDL simulation waveforms.
- Use Modelsim for Gate-level synthesis and logic optimization to produce the schematic in hardware.
- Use Xilinx ISE 10.1 to implement the synthesized design and determine the number of slices used and the efficiency of the algorithm when implemented in hardware.

VII. COMPARISION AND ANALYSIS

A relative analysis of RC6e with RC5 & RC6 is performed to present some measurements on the encryption and decryption. Effects of some parameters such as number of rounds, block size and the length of secret key on the performance evaluation criteria are inspected.

A. Parametric Comparison

Table 1 recapitulate the comparison between RC5, RC6 for different design considerations such as word size, block size, number of rounds and secret key size [8][9].

Parameters	**Algorithm Type**	
	RC5	**RC6**
b (key length in bytes)	0 – 255 (standard 16)	0 – 255 (standard 16)
r (no of rounds)	0 – 255 (standard 20)	0 – 255 (standard 20)
No of round keys	2r+2	2r+4
Block size in words	2w	4w
w (word size in bits)	16, 32, 64 (standard 32)	16, 32, 64 (standard 32)

Block size in bits	32, 64, 128 (standard 64)	64, 128, 256 (standard 128)
Transformation Function	Does not exist	F(x) = x(2x+1) mod 2w
Used Operation	+, -, ⊕, <<<, >>>	+, -, *, <<<, ⊕, >>>

Table 1: Comparison On The Basis Of Parameters

B. Result s

The waveform analysis of cryptographic techniques i.e. RC-5 and RC-6 is shown in the following figures. These waveforms verify the simulation as well behaviour level of the approach designed. This simulation can be done over Model Sim simulator or it can be effectively performed within Xilinx ISE suite.

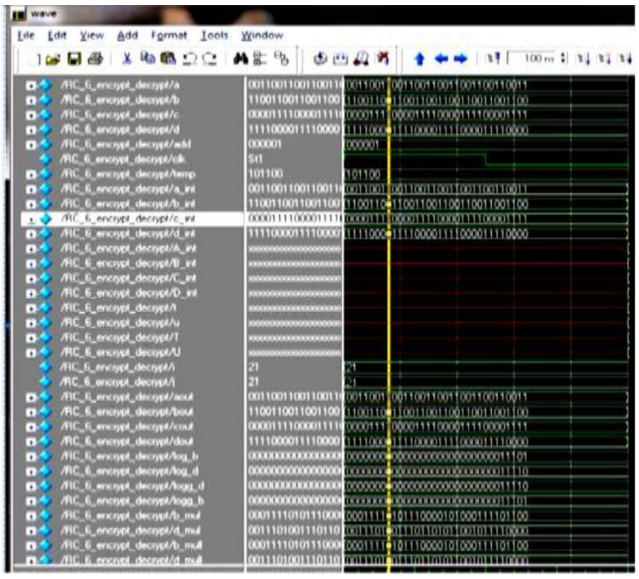

Fig. 5: RC 6 waveform Analysis

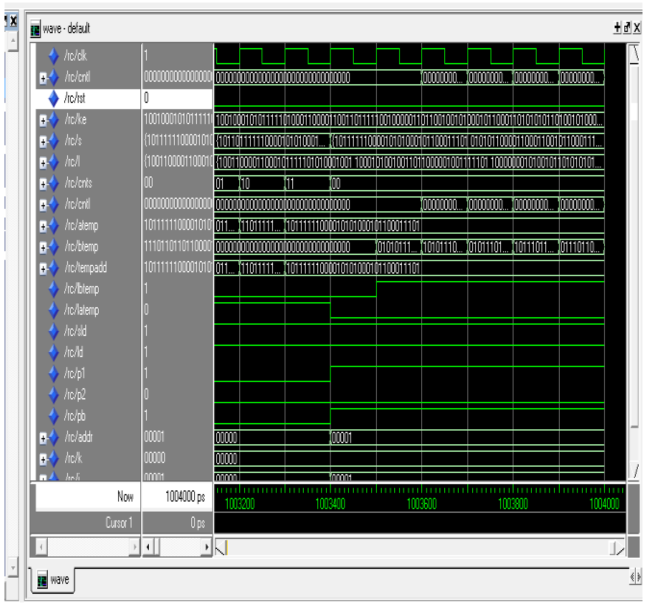

Fig 6: RC 5 waveform Analysis

VIII. CONCLUSION

The aim of this work is to develop a high performance system for data encryption using the RC5 & RC6 algorithm. An FPGA-based RC6 design operating at a maximum frequency of 43.7 MHz was implemented. Our initial intent was to show that:

1. FPGA implementations have much higher performance than software solutions.

2. A simple cipher design, such as the RC5 & RC6, makes use of simple operations (addition, subtraction, multiplication and rotation) that not only possess good cryptographic properties, but also make the overall cipher design efficient from the hardware implementation perspective. However we decided to take the design one step further and implement key scheduling on the FPGA which affected the expected performance quite a bit. Our design included all of key schedule, encryption and decryption modules. The key schedule module however had the biggest impact on the performance of the design. The advantages and disadvantages of this approach as opposed to the other approach of just loading the round keys to the FPGA

REFERENCES

1. N. Sklavos, C. Machs, and O. Koufopavlou, “Area optimized architecture and VLSI implementation of RC5 encryption algorithm,” in Proc. 10th IEEE International Conference on Electronics, Circuits and Systems, vol. 1. December 2003.
2. R.L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L.Yin, “The RC6 Block Cipher," available at website http://theory.csail.mit.edu/~rivest/rc6.pdf
3. M.Rahman, I.R.Rokon, "Efficient hardware implementation of RSA cryptography", Proc. of International Conference on Anti-counterfeiting, Security, and Identification in Communication, pp.316-319, 2009.
4. H.Rahaman, J.Mathew, A.Jabir, D.K.Pradhan, "Ctestable S-box implementation for secure advanced encryption standard", Proc .of IEEE International On-Line Testing Symposium, pp.210-211, 2009.
5. Olabisi, A. and Elkeelany, O. “Integrated design of RC5 algorithm.” In “The IEEE 39th Southeastern Symposium on System Theory,” , 2007.
6. Nimmagadda, S. and Elkeelany, O. “Performance evaluation of different hardware models of RC5 algorithm.” In “The IEEE 39th Southeastern Symposium on System Theory,”, 2007.
7. “XCV-1000 FPGA Board v. 1.0 User Manual,” *Xess Coporation*, available at website http://www.xess.com/manuals/xcv-1000-manual-1_0.pdf. 2005.
8. Abdul Hamid M. Ragab, Nabil A. Ismail, Senior Member IEEE and Osama S. Farag Allah, “Enhancements and Implementation of RC6TM Block Cipher for Data Security”, IEEE Catalogue No. 01 CH37239-0-7803-7101-1/01
9. Asma Belhaj Mohamed, Ghada Zaibi, Abdennaceur Kachouri “IMPLEMENTATION OF RC5 AND RC6 BLOCK CIPHERS ON DIGITAL IMAGES”, 978-1-4577-0411-6/11