ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 2, February 2013

Design of RC5 Algorithm using Pipelined Architecture

R.Sanju Abraham[1], A.Arun[2]

Abstract- **Data transmission through a channel requires more security, so security gaining is more importance than simply transmission. In this document RC5 algorithm is used for encryption and decryption of data to have secure data transmission for communication purposes. The information should be sent confidentially over the network without fear of hackers or unauthorized access to it. This makes security implementation in networks a crucial demand. Symmetric Encryption Cores provide data protection via the use of secret key only known to the encryption and decryption ends of the communication path. Secure transmission require cryptographic algorithm. This document describes the RC5 encryption algorithm, a fast symmetric block cipher suitable for hardware or software implementations. A novel feature of RC5 is the heavy use of data dependent rotations. RC5 has a variable word size, variable number of rounds and a variable length secret key. The encryption and decryption algorithms are exceptionally simple. The requirements of hardware implementation of these algorithms are less power consumption, allocation of resources, re-configurability, architecture efficiency and cost efficiency. This paper aims to the speed, improve performance and throughput. It's organized as brief introduction about algorithm, RC5 algorithm, System on Chip architecture, pipelined architecture, and results.**

Index Terms – **RC5, Pipeline, System on Chip (SoC).**

I. INTRODUCTION

Nowadays the rapid growth of wireless communication provides lot of data services. On the other hand, we are facing lot of security threats and attacks. So the secure transmission of data is mandatory. The primary method used for protecting valuable data is encryption.

Many encryption schemes constitute the area of cryptography. Two types of encryption are commonly used: conventional or symmetric encryption and public-key or asymmetric encryption. Five ingredients of symmetric encryption are plaintext, encryption algorithm, secret key, cipher text and decryption algorithm. The way in which the plaintext is processed is important. A block cipher processes input one block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time.For secure use of conventional encryption, we have two requirements: a strong encryption algorithm and transmission and reception of secret key in secure fashion. Here we used RC5 encryption algorithm which is symmetric block cipher.

RC5 has a variable word size, a variable number of rounds and a variable length secret key. RC5 is exactly designated as RC5-w/r/b, where w denotes word size in bits, the standard value is 16,32and 64 bits; r denotes number of rounds and allowable value ranges from 0 to 255; b denotes length of user's secret key in bytes and the allowable value ranges from 0 to 255. The parameters we have used is RC5-32/12/16. RC5 consists of three components: key expansion, encryption and decryption algorithm. This uses three primitive operations and their inverse.

1. Addition of words "+". This is modulo-2^w addition and the inverse operation subtraction of words"-".
2. Bit wise exclusive OR (XOR) of words
3. 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 is denoted x>>>y.

Manuscript received Feb, 2013.

R.SANJU ABRAHAM,*ELECTRONICS AND COMMUNICATION , Saveetha Engineering College, Chennai, India, Mobile No-9994138630*

A.ARUN*, ELECTRONICS AND COMMUNICATION , Saveetha Engineering College, Mobile No-9444905943.*

II. RC5 ALGORITHM

A. Key Expansion

This routine expands the user's secret key K to fill the expanded key array S, S resembles an array of t=2(r+1) random binary words determined by K. It uses two word-sized binary constants P_w and Q_w. They are defined as,

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

(1)

$$Q_w = Odd((\phi-1)2^w)$$

(2)

where

$e = 2.718281828459...$(base of natural logarithms)

$\phi = 1.618033988749...$(golden ratio)

The three steps of key expansion are as follows:

1.Converting the secret from bytes to words

The expansion process first copy the secret key K[0…b-1] into an array L[0…c-1] of c=[b/u] words, where u=[w/8] is the number of bytes/word.

2.Initializing the array S

The second process is to initialize array S to a pseudo random bit pattern using arithmetic progression by constant values P_w and Q_w.

```
S[0]=$P_w$;
For i=1 to t-1 do
S[i]=S[i-1]+$Q_w$;
```

3.Mixing in the secret key

The third process is to mix in the user's secret key in the array S and L array.

```
i=j=0;
A=B=0;
Do 3*max(t,c) times;
A=S[i]=(S[i]+A+B)<<< 3;
B=L[i]=(L[i]+A+B)<<< (A+B);
i= (i+1)mod (t);
j= (j+1)mod (c);
```

B.Encrption

The input block is given in two w-bit registers A and B. Key expansion has been already performed, so that array S[0…t-1] has been computed.

```
A=A+ S[0];
B= B+ S[1];
For i=1 to r do
A= ((A $\oplus$ B) <<< B) + S[2*i];
B= ((B $\oplus$ A) <<< A) + S[2*i +1];
```

The decryption is the inverse process of encryption routine.

III.RELATED WORKS

Pipelining is an implementation technique where numerous instructions are overlapped in execution. It takes advantage of parallelism that exists among the actions needed to execute an instruction. In present, pipelining is the key implementation technique used to make fast CPUs. Here for the same data rc5 fulfil repeated number of operation which needs to compute large number of iterations for encryption and decryption.

In RC5 encryption and decryption process can be implemented in pipeline technique to improve performance and to increase throughput. Between each rounds a register and data register are placed. When a round completes the output will be stored in a register and the inputs are taken from the data register. The plaintext inputs will be given from the initial unit and when one round is completed, the values will be stored in the register. In every single clock register data is placed in the data register from the register. Finally the cipher text is taken from final unit after all rounds are completed.

IV. PROPOSED METHOD

System-on-Chip (SoC) model contains three main components. First, the expansion unit in which the user secret key is recorded and expanded in size. Here the size depends upon number of rounds. The second is expanded keys that are stored in internal memory unit. The third is RC5 core. It needs to read expanded key in sequential way in order to encrypt the plain text. The encryption and decryption processes are performed here.

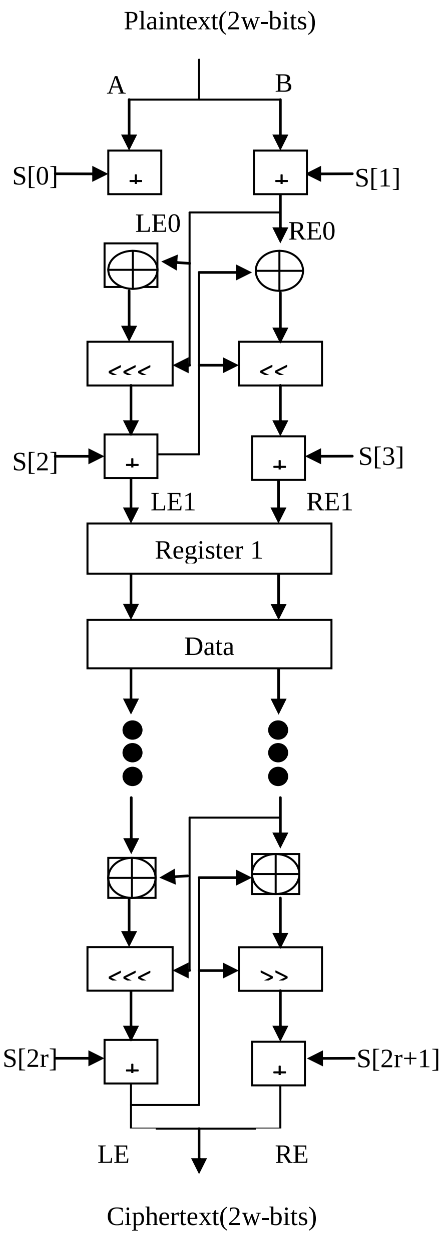

Figure 1: Pipeline Encryption Operation

To avoid memory tapping attacks, the memory is made as a part of the chip. This system architecture is made feasible using FPGA technology with on-chip memories. This system uses RC5 32/12/16 parameters. That is two 32-bit inputs and outputs, 12 rounds and 16-byte (128-bit) secret key.These parameters which are given as inputs to the circuit made the system flexible. These parameters can be modified according to user's application.

The number of rounds affects both encryption speed and security. For example, in credit card transaction security is the main thing and speed is not important. So the user could go for larger value of round. For some other applications, high speed is mainly required. In such situations, the user can choose small value of rounds. This makes RC5 algorithm flexible. Another thing which affects speed and security is word size. Having constant set of parameter values, the application may be affected.

At a particular timing of a setkey signal the architecture needs a registration of the user secret key into the system. For providing the secret key , parallel lines has to be assumed for one step registration. Here the input can be changed by the setkey pulse if the secrete key do not stay as input in some case. Therefore the plaintext inputs are given and the core reads the memory logic unit in a sequential manner for getting the ciphertext.

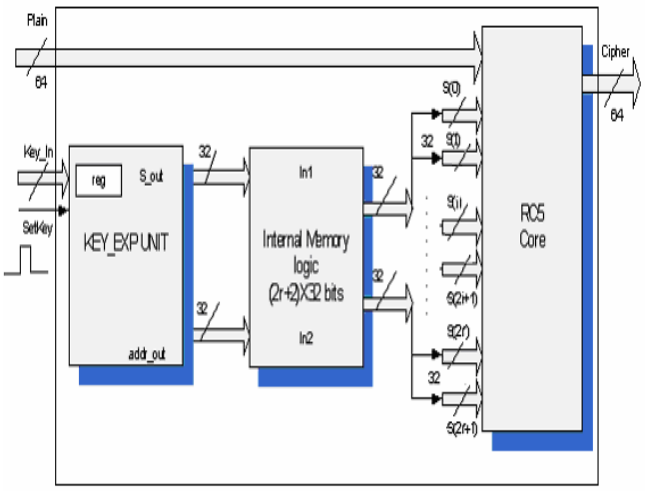

Figure 2:Architecture of RC5 Encryption System

V.DESIGN TECHNIQUES

Nios II processor, which is a soft processor that can be instantiated on an Altera FPGA device. This is taken for its flexibility and more features. As defined in Hardware description language the NIOS II processor can be implemented in the FPGA by using Quartus II tool. To add the necessary functional units such as memories, I/O interfaces and timers to the Nios II processor, the System-on-a-programmable-chip (SOPC) Builder software was used. To form a complete system Nios II processor can be used with a variety of other components.

These components comprise a number of standard peripherals, and it is also possible to define custom peripherals. The Nios II processor has a Reduced Instruction Set Computer (RISC) architecture.

VI.VLSI SYNTHESIS RESULTS

Both conventional and proposed method has been described in Verilog, synthesized by Quartus II tool and for application level calculation; we attached SOPC builder Nios II processor peripheral. The parameters taken for encryption and decryption are w=64, r=12, b=128. The plaintext is given as input to the encryption core and then ciphertext is obtained. The same ciphertext is given to decryption core, then the plaintext is obtained as output. The RTL schematic is given below.

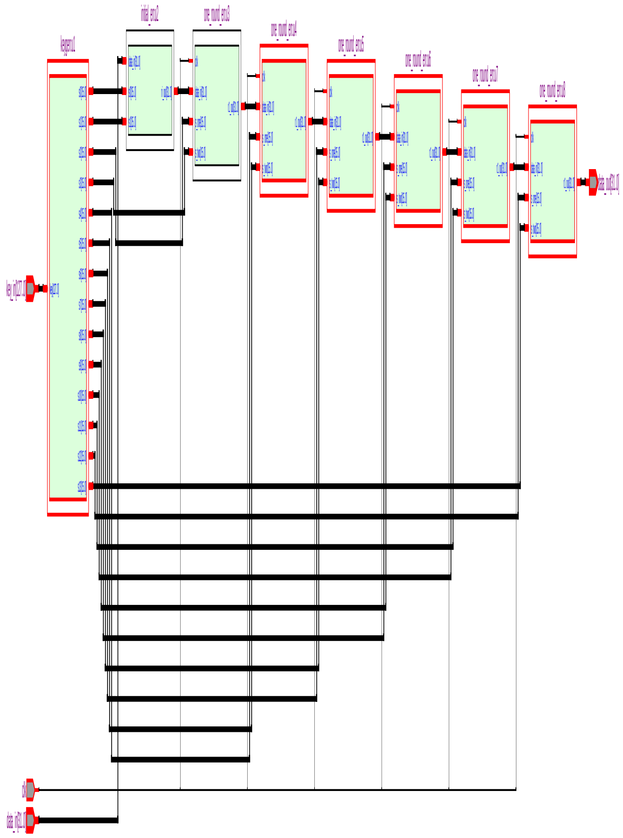

Figure 3:RTL schematic of encryption module

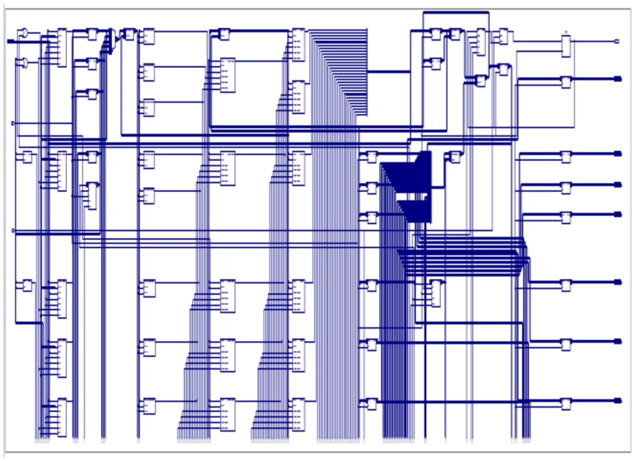

Figure 4:RTL schematic of key expansion module

Power analysis is carried in Quartus II tool and the results are given below.

.

TABLE 2: Power Analysis

Module	**Encryption**	**PipelineEncryption**
Total logic elements	3745	4005
Total combinational function	3745	3959
Dedicated Logic registers	0	840
Total register	0	840
Total thermal power dissipatio n	133.09mW	269.39mW
I/O thermal power dissipatio n	52.09mW	55.41mW

VII.CONCLUSION

This paper represents RC5 algorithm, pipeline technique and its speed. The proposed method describes in verilog, synthesized by Quartus II tool and implemented in NIOS II processor.

REFERENCES

[1] R.L. Rivest, "The RC5 encryption algorithm," Proceedings of the 1994 Leuven Workshop on Fast Software Encryption, pp. 86-96, Springer-Verlag, 1995

[2] Olabisi, O. Elkeelany, "Integrated design of RC5 algorithm," InProceedings of The IEEE 39th Southeastern Symposium on System Theory, 2007.

[3] Hua Li, Jianzhou Li, Jing Yang, "An efficient and reconfigurable architecture for RC5", Canadian Conference on Electrical and Computer Engineering, 2005

[4] Schubert and W. Anheier, "Efficient VLSI implementation of modern symmetric block ciphers," proceedings of ICECS'03, pp. 757-760, Pafos, Cyprus, 2003.

[5] S. Nimmagadda, O. Elkeelany, "Performance evaluation of different hardware models of RC5 algorithm," In the Proceeding of The IEEE 39th Southeastern Symposium on System Theory, 2007

[6] N. Sklavos, C. Machas and O. Koufopavlou, "Area optimized architecture and VLSI implementation of RC5 encryption algorithm," Proceedings of 10th IEEE International Conference on Electronics, Circuits and Systems, pp. 172-175, United Arab Emirates, 2003.

[7] Behrouz A. Forouzan," Cryptography and Network Security" THM publications, 2009.

[8] Janick Bergeron," Writing testbenches: Functional verification of HDL models", Kluwer Academic, 2000.

R.SANJU ABRAHAM recieved BE(ELECTRONICS AND COMMUNICATION) degree from VINS Christian College of Engineering in 2011,Pursuing my Master's Degree in APPLIED ELECTRONICS from Saveetha Engineering College, Thandalam Chennai

A.ARUN working as associate professor in Saveetha Engineering College, received Master's Degree from Anna University, Guindy, Pursuing Ph.D in the area of NOC from Anna University,guindy.Specialized in Medical electronics and have teaching experience for 10 years