Implementation of Low Power RC5 Algorithm in XILINX FPGA

V Chaitanya Tummalapalli *, MD. Khwaja Muinnuddin Chisti **
(Department of Electronics and Communication Engineering, GITAM University, Visakhapatnam-530 045)

ABSTRACT: Nowadays data transmission through a channel requires more security. Security gaining more importance than simply transmission. Secure transmission require cryptographic algorithm. The requirements of hardware implementation of these algorithms are less power consumption, allocation of resources, re-configurability, and architecture efficiency and cost efficiency. The rc5 encryption had requirements of low power and low complexity and low delay. This paper aims to the low power and complexity. It's organized as brief introduction about algorithm, System on Chip architecture, pipelined architecture, low power architecture and results. The proposed design is described with VHDL, synthesized by Xilinx Synthesis Technology (XST) and implemented in Xilinx FPGA Spartan™-IIE devices.

KEY WORDS: RC5 algorithm, SOC, low power architecture, FPGA implementation.

1. INTRODUCTION

This paper aims to the low power and complexity this paper gives brief induction to rc5 convectional structure and reconfigurable architecture and pipeline structure. Finally low power and low complexity architecture. For the synthesis propose The parameters taken are w = 64, r=12and b=128The proposed design is described with VHDL, synthesized by Xilinx Synthesis Technology (XST) and implemented in Xilinx FPGA Spartan™-IIE device.

.

2. RC5 ALGORITHM

A particular RC5 is exactly designated as RC5-w/r/b, where the variable parameters w, r, and b respectively denote the word size (in bits), the number of rounds, and the length of secret key (in bytes). The allowable value of w is 16, 32 and 64; the allowable values of r and b range from 0 to 255. The parameter of RC5-32/12/16 is commonly chosen

There are three routines in RC5: key expansion, encryption, and decryption. These routines consist of three primitive operations (and their inverse): words addition , bitwise XOR, and data-dependent left rotation of x by y denoted by x <<< y. Note that only the log2(w) low order bits of y affect this rotation. In the key-expansion routine, the user-provided secret key is expanded to fill a key table whose size depends on the number of rounds. The key table is then used in both encryption and decryption.

The decryption follows the same scheme as encryption except that it requires words subtraction and rotation to the right. The description of the encryption algorithm is given.

```
$A=A+S[0]$
$B=B+S[1]$
for i=1 to r do
{
    $A = ((A \oplus B) <<< B) + S[2i]$
    $B = ((B \oplus A) <<< A) + S[2i+1]$
}
```

The RC5 sub key generation is quite complex which Produces a final array S of t = 2 * r + 2 number of sub keys from b-byte secret key and predefined magic constants Pw and Qw. This includes three steps. First, the secret key array K [0... b-1] in byte is copied into an array L [0... c-1]. Second, array S is initialized using an arithmetic addition determined by the Pw and Qw. At last, a mix in the secret key in three passes over the array S and L is performed as given below.

```
$i = j = X = Y = 0$
for i=1 to 3   max (t,c) do
{
    $X = S[i] = (S[i] + X + Y) <<< 3$
    $i = (i + 1) \bmod (t)$
    $Y = L[j] = (L[j] + X + Y) <<< (X + Y)$
    $j = (j + 1) \bmod (c)$
}
```

3. SYSTEM ON CHIP ARCHITETCURE

The RC5 algorithm consists of three main components: the key expansion algorithm, an encryption algorithm and a decryption algorithm. RC5 has three main parameters: w; the word size in bits, r; the number of rounds and b; the number of bytes in the user's secret key. The system on chip RC5 Architecture is shown in Figure1.The internal memory unit is implementated internally in the RC5 core. Internal memory internal to RC5 core such that the key is not accessible. The key expansion module starts its key expansion when key_in is high. The next session describe the key expansion module and the encryption module in detail.

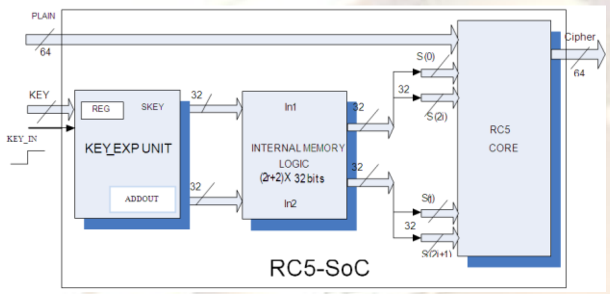

Figure 1 system on chip architecture

4. KEY EXPANSION MODULE

The proposed key expansion module with the 128 length key is presented in Fig. 2. The module includes four parts: the ROM for the magic number constants storage, the array for the immediate sub key storage, the arithmetic unit to mix, and the Address Generator for generating size of the s array and L array respectively.

The S array and L array correspond to array S and L in the algorithm respectively. A ROM is used to store the magic constants of *Pw* and *Qw*. For w=32, the *Pw* requires XOR 1 to make it odd when it is read from ROM. The key expansion module shows results in three operations. In the first operation, the L array is erased to zero before the secret keys are stores into the L. Transformation of the secret key of byte array K into word array L is done. In the second operation, the magic constants in the ROM are read into the Adder module to done the addition and every sum is stored word by word into the 32K-bit S array

In the third process, the words read from the S array and the L array are mixed through addition and left rotation operations. Then the results are written back to the S array and the L array. The key expansion unit output will be the s array contents.

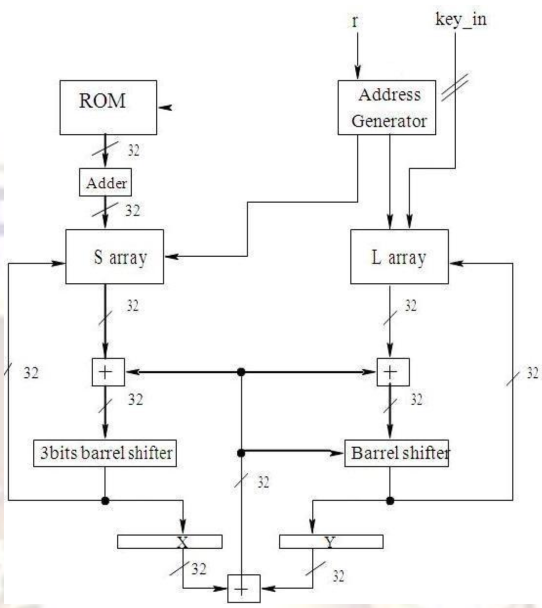

Figure 2.key expansion process

5. ENCRYPTION DECRYPTION MODULE

Encryption process is as shown in figure. Figure shows the expanded rounds from 1 to r. Each round consists of three primitive operations. The decryption process is reverse operation of encryption. The RTL schematic as shown in figure 11.Synthesis by Xilinx Synthesis Technology (XST) and implemented in Xilinx FPGA Spartan™-IIE device.

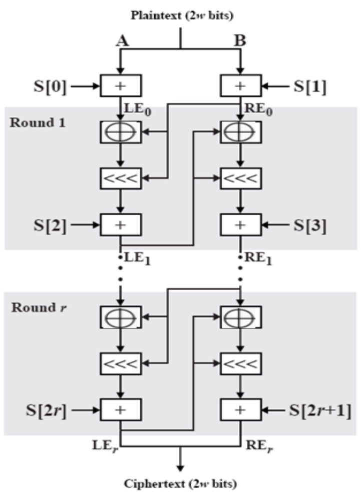

Figure 3. Encryption operation

6. PIPELINED IMPLEMETATION

The encryption and decryption process can be implemented in pipeline format to increase throughput. The pipeline registers can be placed between each round such that in each round output is saved in between registers. The architecture of proposed system is presented in Figure 6. The initial unit represents the plain text divided into two half and addition with s (0) and s (1) respectively. The control signals are not shown for simplicity. Between each round placed a register and data register such that outputs rounds are pace in the register and the inputs are taken from the data. In each clock register data placed in data register from register. In final round the cipher txt is taken out. Decryption process is inverse of above process from round r to round 1. Ecryption is as shown in figure 4.

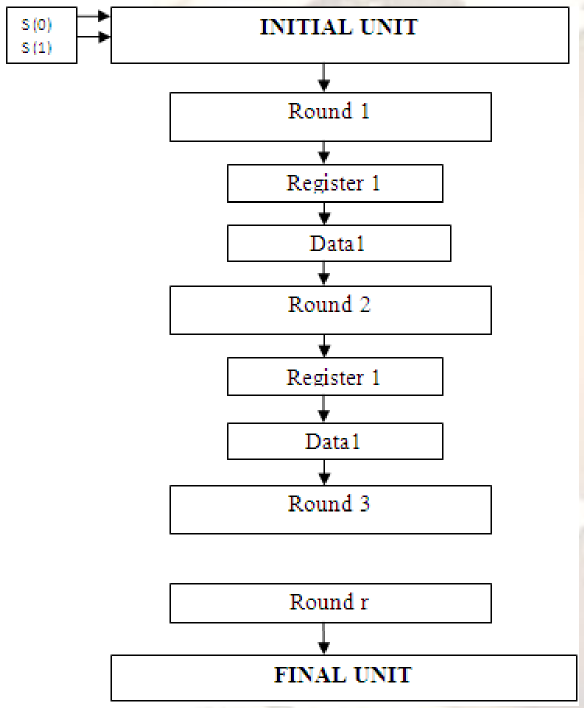

Figure 4. Pipelined implementation of encryption process

In the proposed system the size of maximum block is considered to be 128-bit. The system can support the coding of 32, 64, 96 and 128 bits. The case of 96 bits rejected because it is not supported from RC5 standard the main disadvantage of this design is hard ware complexity is more. We can reuse the same round to reduce the hardware by using the hardware. We can switch to the design called as low power design to reduce the power and hardware complexity.

7. LOW POWER ARCHITECTURE

The figure below shows the proposed encryption/decryption design i.e. one round operation, comprises 2 registers, one Circular shifter, one XOR block, and an add/sub to execute addition or subtraction. Notably, this architecture combines encryption and decryption into one core by sharing the same circuit instead of two cores in pipelined architecture. Besides, replace one add/sub with two multiplexers to reduce area and power.

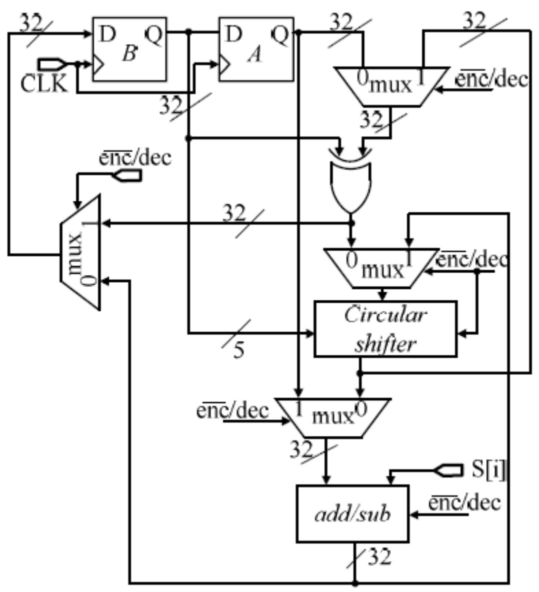

Figure 5. One round of low power rc5 architecture

The architecture above encryption and decryption process can be selected by the enc/dec signal. If it's one the decryption is activated and if zero encryption is activated. Total compress of two clock cycles. In the first clock cycle, the data stored in register *A* is XOR with *B*, and the data stored in register *B* is shifted to *A*. At this moment, only *A*, half plaintext, is encrypted. The other half plaintext, *B*, which has already been shifted to *A*, will operate in the second clock cycle

The bi-directional Circular shifter is made of 32 32-to- 1 multiplexers with a 5-bit selection signal. The rotational direction is determined by enc/dec. the shifter can be implemented as the butterfly shifter. An *n*-bit butterfly network consists of lg(*n*) stages. In each stage, *n* bits are considered as *n*/2 pairs. Determined by a single control bit, two bits in a pair can go directly to the same location at the output, or exchange position with the other bit. So *n*/2 control bits are needed in each stage to specify the path for *n*/2 data pairs. The example of the 16 bit butterfly network each small box in the figure can be considered as a 2:1 multiplexer (MUX), where one of two bits in a pair is selected. In the first stage, the distance between two bits in a pair is 16/2 = 8. So bit 0 is paired with bit 8, bit 1 with bit 9, and so on. In the last stage, the distance between two paired bits is one, i.e., two bits are next to each other.

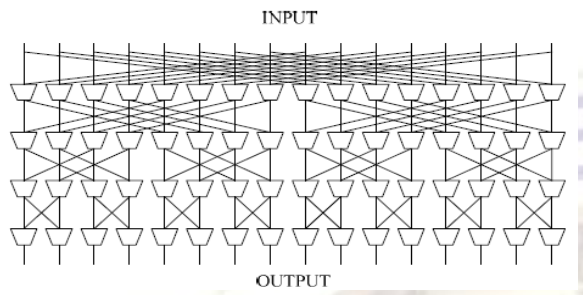

Figure 6. 16 bit butterfly network

The one round can be reused to get the results. i.e., encryption and decryption process. The proposed architecture for this is represented below.

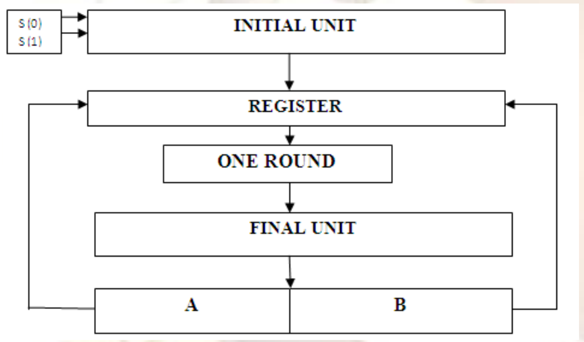

Figure 7. Low power encryption

In initial round the operation of addition or sub with s (0) and s (1) is done. And this given to the round and it calculate the value of A and B for one round given to the final unit register. After the final unit the values are stored in the registers A and B and given to the register for the next round operation. So on this goes on until the rounds is completed. After rounds are completed. The final values of cipher are presents at final unit register. Decryption is exact reverse iteration. Final value is present at initial unit. Total no of clocks required are 4r where r is round number.

8. RESULTS

The proposed design is described with VHDL, synthesized by Xilinx Synthesis Technology (XST) and implemented in Xilinx FPGA Spartan™-IIE device. The encryption and decryption is done in XILINX FPGA. The parameters taken are w = 64, r=12and b=128. The plain text is given as input to encryption core then Cipher is obtained. The same cipher is given to decryption core then the plain text is available at output. The RTL schematic of various block are shown.

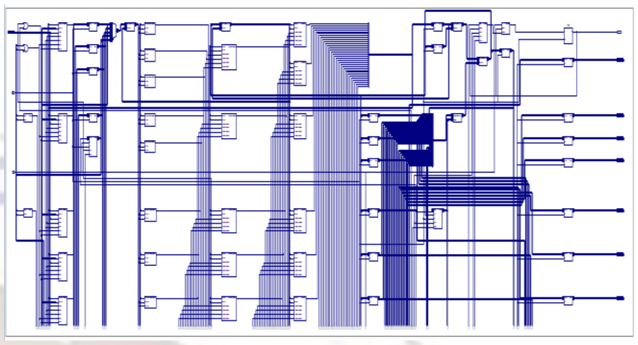

Figure8. RTL schematic of key expansion module

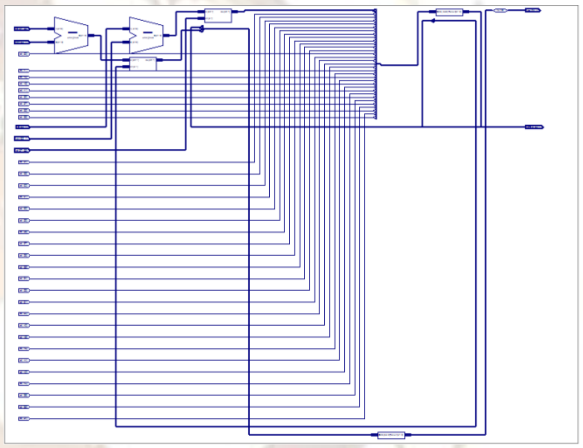

Figure9. RTL schematic of one round module

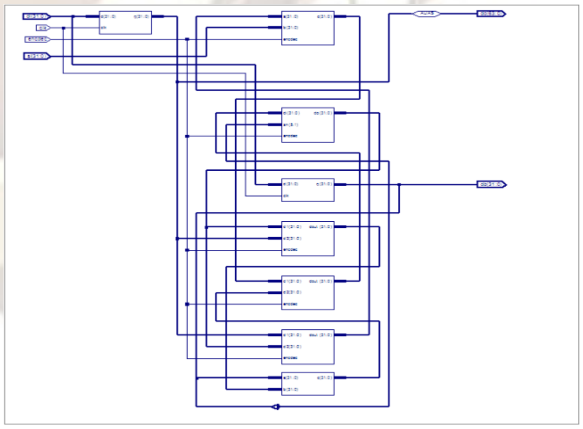

Figure10. RTL schematic of low power round module

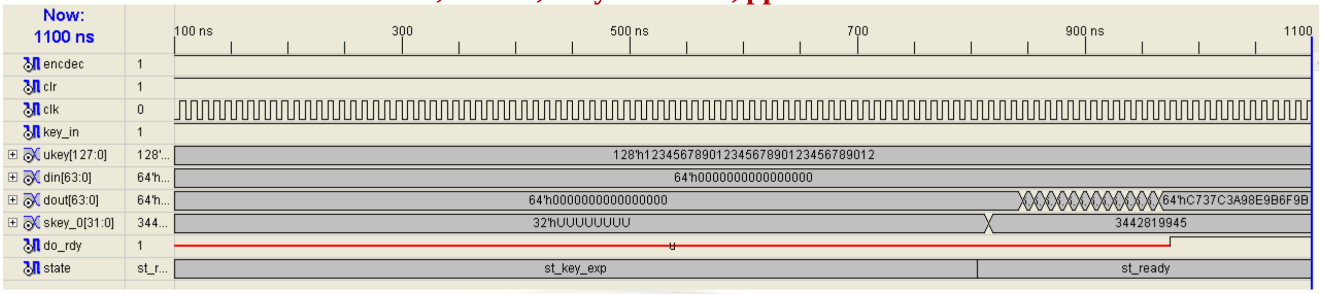

Figure 11. Encryption process

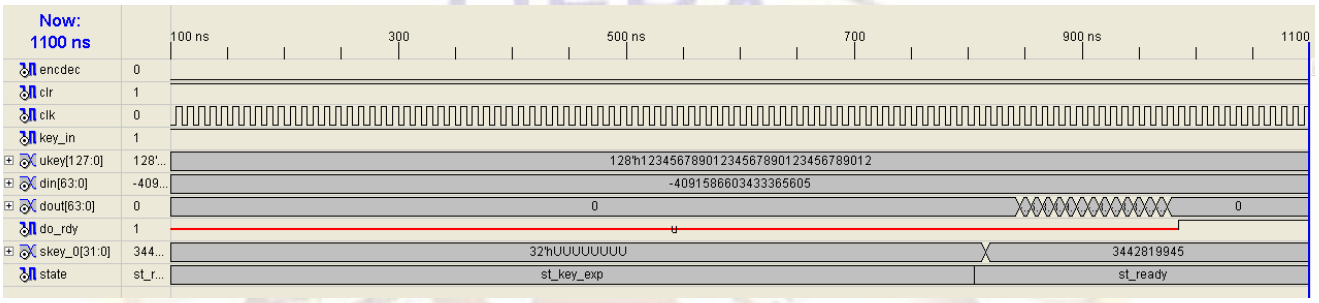

Figure 12. Decryption process

Power analysis is carried in Xilinx FPGA Spartan™-IIE power estimator. The results are given below

ARCHITECURE	CLB SLICES	POWER (mW)	THROUGHPUT (Mbps)
SOC	2608	25	9.69
PIPELINE ENCODER	4593	34	22.34
LOW POWER	451	7	26.40

Table 1 comparison of different architectures

8. CONCLUSION

In this paper represents RC5 algorithm and its low power implementation. In this paper various implementations of algorithm is taken The proposed design is described in VHDL, synthesized by Xilinx Synthesis Technology (XST) and implemented in Xilinx FPGA Spartan™-IIE device. Power calculated using XILINX power estimator. Results are tabulated above.

9. 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.