Int. J. Chem. Sci.: 14(S3), 2016, 725-732
ISSN 0972-768X
www.sadgurupublications.com

LOW POWER ASIC IMPLEMENTATION OF RC5 ALGORITHM

J. HARISH, S. J. MADHURI, V. YASWANTH* and K. JAGANNADHA NAIDU

Department of Micro and Nanoelectronics, SENSE, VIT University, VELLORE (T.N.) INDIA

ABSTRACT

RC5 is a fast symmetric block cipher algorithm known for its simplicity in hardware and software implementations. A novel feature of RC5 is that it provides a variable length secret key (0 to 255 bytes), a variable word size in bits (16/32/64) and a variable number of rounds (0 to 255), hence providing flexibility in security to the user. RC5 also heavily makes use of data dependent rotations, making it difficult for crypt attacks. In this paper, low power, high throughput RC5 architecture is investigated. Multi V_{DD} technique has been adapted for low power synthesis. Comparisons were made with normal synthesis ie., without using low power constraints. Applying multi V_{DD} constraints has shown remarkable reduction in total power by 94.9% and 95.1% for encryption and decryption respectively.

Key words: RC5, Multi V_{DD}, Data dependent rotation.

INTRODUCTION

Providing security to the user data in a potentially alien and hostile environment has always been a primary concern among the peers, working in the field of computer systems, communication networks and e-commerce transactions. Cryptography is a popular field which emerged as a solution to offer information security to the confidential data being transmitted over wireless communication channels.

The major cryptographic techniques are :

(a) Secret or symmetric key cryptography: A single secret key is shared for encryption and decryption.

*Author for correspondence; E-mail: harishjeevakala@gmail.com, madhuri.janney@gmail.com, yaswanthvedagiri@gmail.com, jagannadhanaidu.k@vit.ac.in

(b) Public key cryptography: Different keys are used for encryption and decryption.

Secret or symmetric key cryptography

There are two types of symmetric key ciphers namely, stream and block ciphers. Stream ciphers operate on a single bit of data, equivalent to saying that the block size is 1 bit. Feedback structures are used in stream ciphers to provide a different key for each bit to be encrypted. Major stream ciphers are RC4, FISH, ISAAC, SEAL, SNOW etc. Block ciphers operate on a block of data, for eg, 64 bit, 128 bit and so on. Unlike stream ciphers, a single key is used to encrypt each block of data. They have Fiestel like structure and are relatively complex. Although the whole process is said to be slower, they ensure higher degree of security compared to stream ciphers. Example block ciphers are RC5, DES, Blowfish etc. Out of existing block cipher algorithms, RC5 provides the user with flexibility in choosing the size of data and key. The number of rounds for encryption/decryption is not fixed. Therefore user can customize the security level needed according to the application. The rest of the paper is organized as follows- section II describes about the literature survey and related works. Section III explains about RC5 algorithm in detail, including key expansion, encryption, decryption algorithms and their architectures. Section IV discusses the low power design approach adapted in the proposed work. In the section V, the calculation of throughput is presented. Section VI includes results and discussions. Conclusion of the proposed work is discussed in section VII.

Related works

Different hardware architectures were proposed by earlier works for implementing RC5 algorithm for optimizing different parameters such as area, speed, throughput etc. Ruhan Bevi et al.[2] discussed about FPGA based pipelined architecture in order to increase through put.

The RC5 algorithm

R. L. Rivest[1] proposed the cryptographic algorithm RC5, which is popularly known as value transformation block cipher. Three major steps involved in implementing RC5 algorithm are key expansion, encryption and decryption. These steps undergo three primary operations: words addition, bitwise XOR and data-dependent cyclic left/right rotation of A and B denoted as, A <<< B/A >>> B. The proposed work considers plain text (2w) of size 64 bit, number of rounds as 12 and number of bytes in key as 16. (w/r/b = 32/12/16) to

provide optimum security and speed[1]. The plain text of size 2w is equally divided to size w and stored in registers A and B. All operations are carried out on these two data and finally concatenated to form a 64 bit ciphered text.

Key expansion

The user defined key array of size K[0,1....b-1] is expanded based on the number of rounds to fill the expanded key array S[0.....t-1], where t = 2(r+1). This makes use of two magic constants P_w and Q_w defined as –

$$P_w = \text{odd}((e-2)2^W); \quad Q_w = \text{odd}((\quad -2)2^W)$$

Where, e = 2.718281828459 (base of natural logarithms), = 1.618033988749 (golden ratio) and odd (x) is the odd integer nearest to x.

Step I: Covert the user defined secret key of size b bytes to a word array L [0…c-1], where c=b/u words, u=w/8 is the number of bytes per word.

Step II: In this step we initialise the S array using magic constants such that

$S[0] = P_w$

For I =1 to t-1

do $S[i] = S[i-1]+Q_w$;

Step III: Here we mix the user secret key in S and L array in three passes. S and L array will be of different sizes, the larger array among the two is processed three times, and the other array will be handled next[1]. The algorithm is given by –

i = j = 0;

A = B = 0;

do 3 * max(t, n) 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 (n);

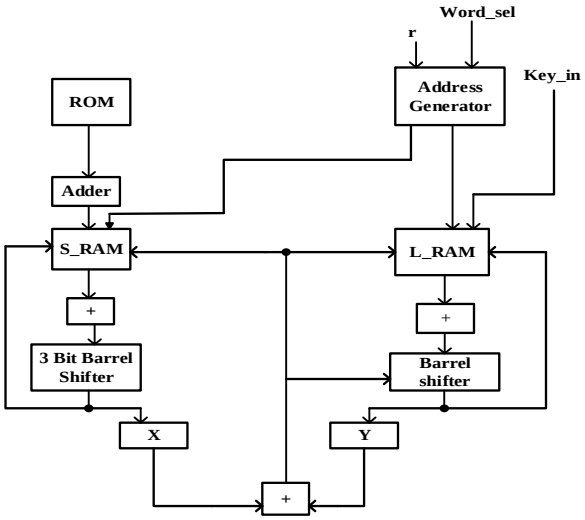

Fig. 1: Key expansion architecture

Encryption architecture

The input will be stored in two w bit registers A and B. Assuming that S array is ready, the new value of A and B are computed through an iterative process as shown in Fig. 3. The algorithm for encryption is given as –

A =A + S[0];

B = B + S[1];

For I =1 to r

do A = ((A⊕B) <<< B) + S[2*i];

B = ((B⊕A) <<< A) + S[2*i +1];

If the select line of the multiplexers is 1, then the plain text stored initially in the registers A and B, are added with the keys S[0] and S[1], respectively. If the select line is 0, iterative round of operation is carried out. In the iterative round, exclusive OR operation is performed on the data obtained from the two multiplexers. This data will be circularly left shifted. The rotation amount will be decided by least significant $\log_2 w$ bits of the data stored in the other register. The result is added with the next set of keys. After the completion iterative round, the result is loaded back to the registers A and B, which will contain the cipher text.

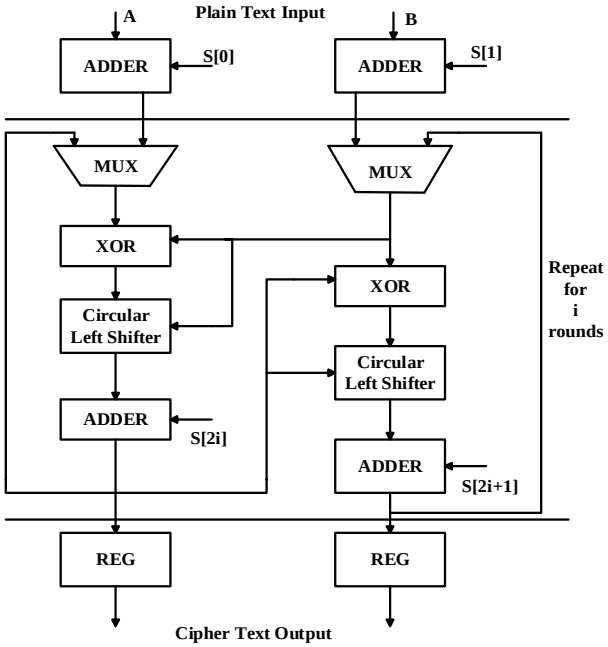

Fig. 2: Encryption architecture

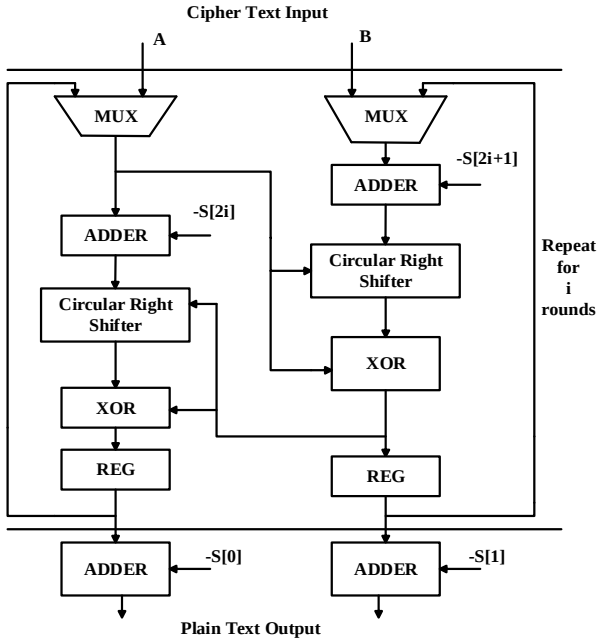

Fig. 3: Decryption architecture

Decryption architecture

The decryption routine can be obtained from the encryption routine[1] and is the reverse process of encryption given by the algorithm,

For i = r down to 1 do

B = ((B- S[2*i +1]) >>> A)⊕A

A = ((A- S[2*i]) >>> B)⊕ B

B = B-S[1]; A= A- S[0];

The following Fig. 4 describes the decryption architecture.

RESULTS AND DISCUSSION

The proposed work is verified using Verilog HDL using synopsys VCS compiler. During synthesis, constraint file is given as an input to meet the desired functionality and timing values. Multi V_{DD} is a popular technique used to reduce power. Owing to the fact that dynamic power is directly proportional to V_{DD}^2, it is possible to selectively choose the blocks which can work with lower supply voltage and thereby help in reducing total power consumption. However, this will increase the delay of the system. In this design, key expansion is the module which is computationally complex and time consuming. 32 nm technology file has been linked to the proposed design and 0.95V supply voltage is given to the key expansion block. For the encryption and decryption modules, a slightly lesser voltage of 0.7 V has been applied. Working at a clock frequency of 100MHz, the proposed design requires 12 cycles to encrypt or decrypt one block of data. Throughput is calculated using the formula,

$$\text{Throughput} = \frac{\frac{(64)\text{bits}}{\text{Clock}}}{(12)\frac{\text{Cycles}}{\text{Block}} * \left(\frac{1}{100\text{Mclocks/sec}}\right)}$$

The throughput achieved for encryption as well as decryption is 533 Mbps. It is to be noted that there exists a significant trade off between the through put and the power dissipation. The following Tables 1 and 2 gives the details of power and chip area consumed for encryption and decryption respectively. It can be observed from the tables that power greatly reduces after applying multi V_{DD} technique to the proposed design.

Table 1: Encryption-power and area details

Encryption	Normal synthesis	Low power synthesis
Total dynamic power (mW)	3.1579	1.2126
Cell leakage power (mW)	48.4832	1.4135
Total power (mW)	51.641	2.6261
Total area (sq.µm)	35255.6471	37625.285359

Table 2: Decryption-power and area details

Decryption	Normal synthesis	Low power synthesis
Total dynamic power (mW)	3.1452	1.1985
Cell leakage power (mW)	45.8727	1.2010
Total power (mW)	49.018	2.3995
Total area (sq.µm)	32509.9496	33905.185786

CONCLUSION

This paper has explored the effect of applying multi V_{DD} technique to RC5 algorithm, using Synopsys DC compiler. Exceptional reduction in total power has been observed. Decrease in total power is by 94.9% and 95.1% for encryption and decryption respectively. With the clock frequency of 100 MHz, the throughput achieved was 533 Mbps.

REFERENCES

1. R. L. Rivest, The RC5 Encryption Algorithm, Proceedings of the 1994 Leuven Workshop on Fast Software Encryption, Springer-Verlag (1995) pp. 86-96.
2. Ruhan Bevi, S. S. V. Sheshu and S. Malarvizhi, FPGA Based Pipelined Architecture for RC5 Encryption, IEEE, Digital Information and Communication Technology and its Application (DICTAP), 214-219 (2012).
3. O. Elkeelany and S. Nimmagadda, Effect of Loop-Unrolling in Hardware Reconfigurable Implementations of RC5-192 Encryption Algorithm, IEEE Region 5 Conference (2008).

4. M. Yoshikawa and K. Sakaue, Dedicated Hardware for RC5 Cryptography and its Implementation, World Congress in Computer Science, Computer Engg. Appl. Computing (WORLDCOM'11), USA (2011).

5. H. Li, J. Li and J. Yang, An Efficient and Reconfigurable Architecture for RC5, IEEE, CCECE/CCGEI, Saskatoon, 0-7803- 8886-0 (2005).

6. J. Liang et al., An Area Optimized Implementation of Cryptographic Algorithm RC5, 5th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (2009).

7. M. Pillmeier, M. Schulte and E. Walters III, Design Alternatives for Barrel Shifters, Proc. SPIE Advanced Signal Processing Algorithms, Architectures and Implementations (2002) pp. 436-447.

Accepted : 11.10.2016