A Comparative Analysis of Pipelined and Area Optimized Blowfish Algorithm in FPGA for embedded tiny Sensor Node

S.Roy Chatterjee[#1], M.Chakraborty[*2]

[#]Electronics And Communication Engineering,Netaji Subhash Engineering College, Kolkata, India
[*]Information Technology, Institute Of Engineering And Management,Kolkata,India
[1]rcswagata@gmail.com
[2]mohuyacb@iemcal.com

***Abstract*—With the unprecedented growth of modern technology the need for protecting data has increased vehemently in wireless sensor network. This need is not only restricted to the military but also subsumes daily lives as well. Hence safeguarding of data in wireless sensor network is one of the leading fields of research in modern times. Area has been a major stumbling block faced while designing encryption algorithms on hardware for sensor network as it involves tiny sensor nodes. The Blowfish algorithm is one of the fastest block ciphers and may be an efficient alternative to the existing encryption algorithms like AES, DES, IDEA etc. In this paper, pipelined hardware architecture (for enhancement of the speed) and area-optimized hardware architecture (for minimization of the size) are both designed for Blowfish algorithm and a comparative performance analysis has been made to trade off between the space and time complexities for embedded tiny sensor node.**

The architectures were implemented by verilog which were synthesized, placed and routed in Spartan3e chip XC3s500e-5fg320 using ISE Design Suite 12.1. The result indicates that area is minimized up to 17.3% from the normal architecture and 19% from the pipelined architecture. Throughput per slice for the area optimized architecture is enhanced 1.01 Mbps as compared to a lower throughput per slice of 0.267 Mbps for the normal architecture of Blowfish algorithm while it is slightly less than the throughput per slice of 1.955Mbps of pipelined architecture.

Keyword- Blowfish Algorithm, Cognitive Radio Sensor Network, Field programmable gate array, Hardware Architecture

1. INTRODUCTION

In the modern era of Cognitive Radio Sensor Network (CRSN), several emergent applications demand a strong encryption technique to ensure hidden data transmission from the unintended person [1,2]. According to [3], a CRSN consists of tiny low power sensor nodes embedded into mobile phones that adapts to the changing environment of the existing cellular network by analyzing the RF surroundings and adjusting the spectrum use appropriately to send sensory data of explosive traces within a defined territory to the respective service provider, which in turn would inform the law and enforcement agency or Police to combat terrorism. The hardware implementation of any encryption scheme for this type of application requires crucial analysis between time and space complexity of the chosen algorithm for the additional embedded hardware and computational overhead which in turn affects the speed and size of respectively of the embedded tiny sensor nodes [4,5].

Smaller the size and higher the speed better is the performance with regard to portability and throughput respectively. The idea is to design hardware architecture for an encryption algorithm and analyze its space and time complexities. Several symmetric key algorithms are designed, among which DES, Triple DES, AES, Blowfish are widely used. Bruce Schneier designed blowfish in 1993 as a fast, free alternative to existing encryption algorithms [6,7]. Since then this algorithm has been used considerably, and it is slowly gaining acceptance as a strong encryption algorithm.

The Blowfish algorithm based on Feistel Network consists of 64 bit input data where same algorithmic operations are performed in each iteration round, except last round and has many advantages. The iteration rounds are suitable and efficient for hardware implementation using simple operations like exclusive-OR and addition on 32 bit word. The relative strength of an encryption algorithm depends on the key length. Blowfish uses variable length (32-448 bits) key and it is less complex and provides higher speed as compared to other existing algorithms [8, 9]. So it is suitable for CRSN applications which exchanges small sized packets. In this paper, pipelined hardware architecture (for enhancement of the speed) and area-optimized hardware architecture (for minimization of the size) are both designed for Blowfish algorithm and a comparative performance analysis has been made to trade off between the space and time complexities.

In pipelined architecture seventeen hardware module blocks are designed for each round of iteration and all hardware units operate in parallel. The first hardware module block operates on the input raw data and the rest of the hardware module blocks operate on the computed output data from their immediate previous hardware module block. Hence the first hardware module block is made to operate on the second set of input raw data simultaneously with the second round of iteration operating on the first set of data.

In area-optimized hardware architecture a single hardware unit is designed and reused for sixteen different rounds of iterations that in turn minimize the hardware requirement for encryption. Both the hardware architectures are designed in Verilog Hardware Description language (HDL) and implemented in Spartan 3E XC3s500e-5fg320 Field programmable gate array (FPGA).

FPGAs are hardware devices whose functions are not fixed and can be programmed in-system. It provides an attractive solution to developers needing custom logic for short time-to-market products [10, 11]. It has got high acceptance as a promising alternative for the implementation of block ciphers.

After the introduction in section I, an overview of Blowfish algorithm is provided in section II. Section III provides the pipelined hardware architecture of the Blowfish algorithm followed by the area optimized Blowfish algorithm in section IV. A comparative analysis of the normal, pipelined and area-optimized Blowfish hardware architecture is made in section V. Section VI concludes the paper with some highlights on future developments.

2. OVERVIEW OF BLOWFISH ALGORITHM

The Blowfish encryption algorithm is a 64 bits block cipher with a variable key length. Each round consists of a key-dependent permutation, and substitution. There is a P-array and four 32-bit S-boxes [B. Schneier.1994 ;Lin et al.2000]. The P-array contains of eighteen numbers of 32-bits subkeys, while each S-box contains 256 entries. The algorithm consists of two parts:

A key-expansion part and a data-encryption part.

Key expansion converts a key of at most 448 bits into several sub key arrays. These sub keys must be pre-computed before any data encryption or decryption.

The process of Subkey generation is illustrated as follows-

Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3). For example:

P1 = 0x243f6a88

P2 = 0x85a308d3

P3 = 0x13198a2e

P4 = 0x03707344

XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key, until the entire P-array has been XORed with key bits. Similarly value of all S boxes is initialized with the sub keys value.

The encryption algorithm for Blowfish is illustrated in Figure 1.

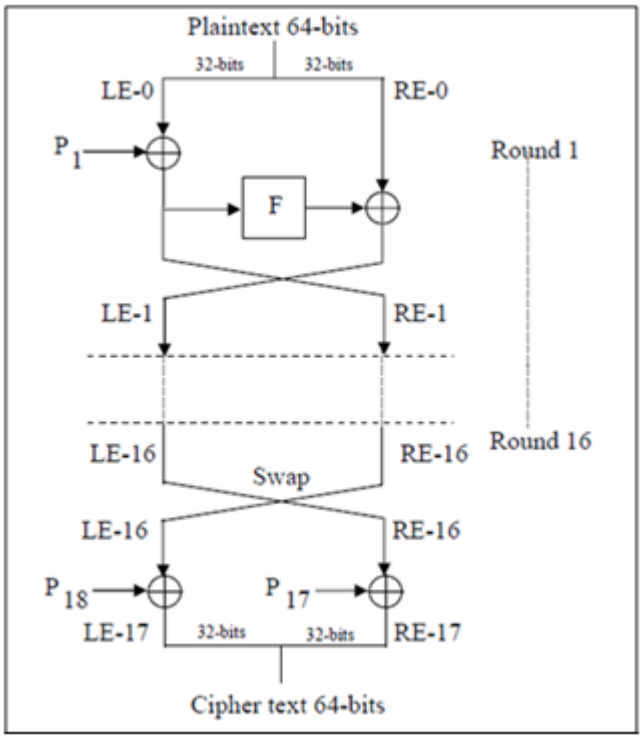

Fig. 1. Flow diagram of encryption process of Blowfish algorithm

For encryption, the 64-bits plaintext is separated into a left and right half each consisting of 32-bits. The encryption routine consists of a 16 round Feistel network. In the first round, an exclusive-or operation is performed between the left 32-bits (LE-0) and the 32-bits P1 of the P-array that has already initialized with key. This value becomes the next 32-bits right value (RE-1) that is feed into the F function block. Each byte of the RE-1 is then used for table lookup in their respective S-Boxes. The internal operation of the F function box is illustrated in Figure 2.

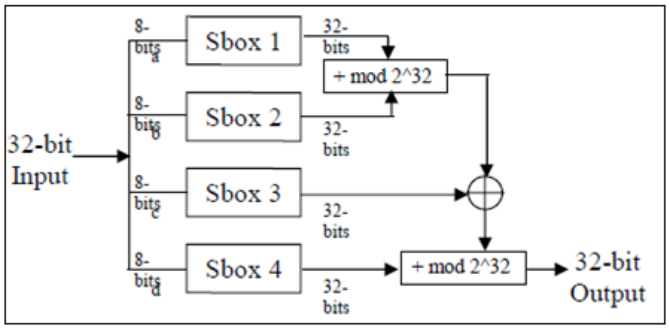

Fig. 2. The internal operations in F function box

Before performing the operation, all the S boxes are initialized with the sub key values and all operation are performed with modulus 2^{32}. The 32-bits value of 'S-box 1' is added to the 32-bits value of 'S-box 2'. The result is taken as the input for the exclusive-or operation to perform with the 32-bits value of 'S-box 3'. The output of the exclusive-or operation is then added to the 32- bits value from 'S-box 4' to get the final output of the function block.

A bitwise exclusive-or operation is performed between the 32-bits output from the F function and the right half of the data (RE-0) to generate (LE-1) for the next round as illustrated in figure 1. Round 2 is then performed with inputs LE-1 and RE-1. This process is repeated for a total of 16 rounds. After completing the 16 rounds, LE-16 and RE-16 values are swapped. An exclusive-or operation is performed between the swapped LE-16 and P18 and also between RE-16 and P17 to obtain LE-17 and RE-17, respectively. The 32-bits values of LE-17 and RE-16 are combined to obtain the 64-bit cipher text.

Decryption is exactly the same as encryption, except that P1, P2, ..., P18 are used in the reverse order.

3. PIPELINED HARDWARE ARCHITECTURE OF BLOWFISH ALGORITHM

In a pipelined architecture, if a series of instructions are provided as input to a pipelined processor, then the second instruction will start executing before the first has been finished [Schmit et al. 2000; Tessier et al. 2001]. The final results of each instruction emerge at the end of the pipeline in rapid succession. Figure 3 shows the instruction execution with pipelining.

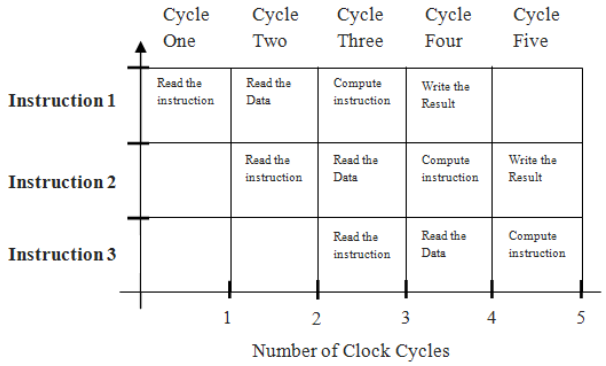

Fig. 3. Instruction execution with pipelining

As illustrated in Figure 3, the execution of the Instruction 2 starts before the completion of Instruction 1.This allows a more efficient use of processor and minimizes the time of execution of a large number of instructions. Here main objective is to design the pipelined for encryption, so key expansion is not taken into consideration.

The pipelined hardware architecture of the blowfish algorithm is shown in Figure 4. There are sixteen hardware module blocks are designed to execute each stage of iteration in parallel and among of which only first hardware module block operates on the input raw data and rest hardware module blocks operate on the computed output data from the previous module block [12].

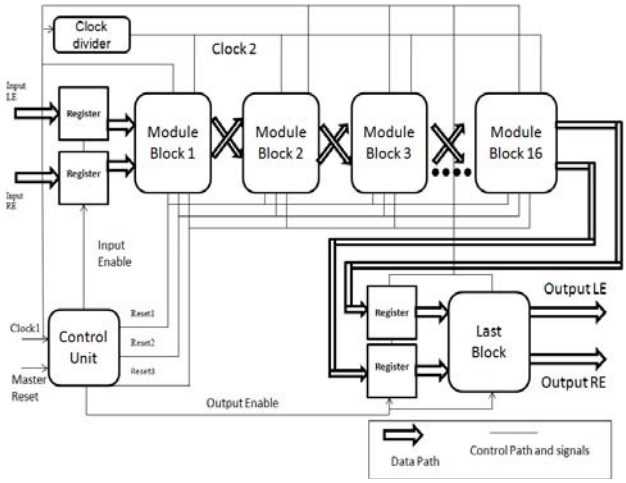

Fig.4. Pipelined Design of Blowfish architecture

The data path is controlled by a hardware control unit that in turn facilitate the work of each hardware module blocks in parallel with different sets of data at a time. A clock divider is also required to effectively synchronize the working of F function boxes of all the hardware module blocks.

The hardware Module Block is designed by inserting seven 32 bit registers in order to allow pipelining as shown in Figure 5.

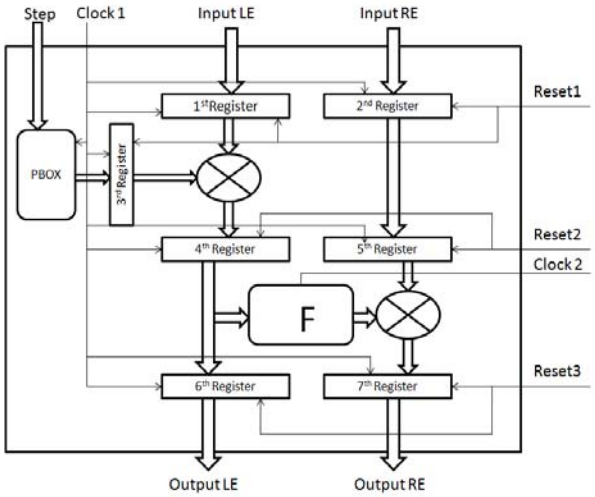

Fig.5. Architecture of an individual module block

This hardware module block is replicated 16 times in the entire design, to perform sixteen iteration rounds of the blowfish algorithm. The designed architecture differs from the original architecture in its working due to the addition of the last pair of registers and selective switching of all the registers. The first, second and third registers operate simultaneously in the first clock cycle and holds the input data during the entire clock cycle when the P box data is fetched and an exclusive-or operation is performed between the left 32-bits (LE-0) and the 32-bits of the P-array that has already initialized with key. The right hand input data doesn't perform any computation during this cycle, and hence holds the original data. The fourth and fifth registers operate in pair and hold the data needed for operation of the F block and second exclusive-or operation. The sixth and seventh registers store the computed final data of a module block from the subsequent blocks until the time of the next clock cycle when the next module block is ready to accept the data

The hardware control unit is designed based on finite state machine to generate several control signals as shown in Figure 4.The control signal named 'Reset1, Reset2, and Reset3'are utilized for selective switching of the registers. Each of the sixteen module blocks are connected to the same set of 'reset' signals and hence have similar operation in parallel but with different set of data. The 'Output Enable' control signal enables the last block to compute all the mathematical operations of the 17th iterative round of the Blowfish algorithm. The state diagram of the control unit is shown in Figure 6.

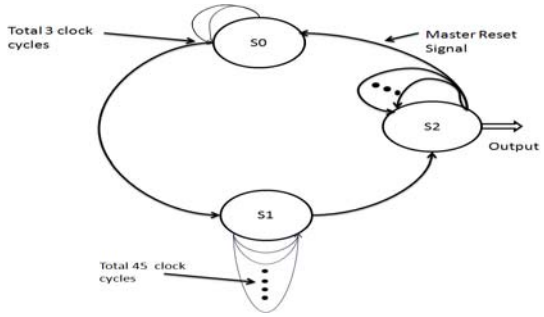

Fig.6. State Diagram of the Control Unit of Pipelined Architecture

Master reset signal is set to logic one to start the operation of control unit. The 1st state S0 is used to start the blowfish operation with first raw input data word of 64 bit and S0 state needs 3 clock cycles for its operation. After 3rd clock cycle, the state changes to state S1. The execution of the 2nd state S1 needs three clock cycle each time and it repeats fifteen times for complete the iteration rounds. In the state S0, only first hardware module block executes the operations whereas in the state S1 all the sixteen module blocks execute their operations in parallel. The 3rd state S2 is the final working state, which enable the last block and also all the sixteen hardware module blocks operate simultaneously with the last block. The state S2is performed repeatedly depending upon the total number of input data words. The Master reset signal is reset to logic zero to return back to the state of S0 at the end of the input data words.

4. AREA OPTIMIZED HARDWARE ARCHITECTURE OF BLOWFISH ALGORITHM

Area optimization is an added challenging issue in the hardware implementation of Blowfish algorithm for CSRN as it deals with very small sized sensor nodes embedded in mobile phones. This section is concerned with optimization of area consumption by designing hardware for only one round and reusing the same hardware to complete sixteen rounds of the encryption algorithm that employ same mathematical operations unlike normal Blowfish architecture where seventeen such hardware module blocks are used for seventeen iterations thereby increasing the space complexity. The architectural model for minimization of area is illustrated in Figure 7. Two data path hardware units are designed to perform the whole algorithm. Recursive hardware unit is used for rounds 1 through 16 and non recursive hardware unit is used solely to perform the 17th round. These two data units are controlled through a state machine called controller hardware unit in the architecture model.

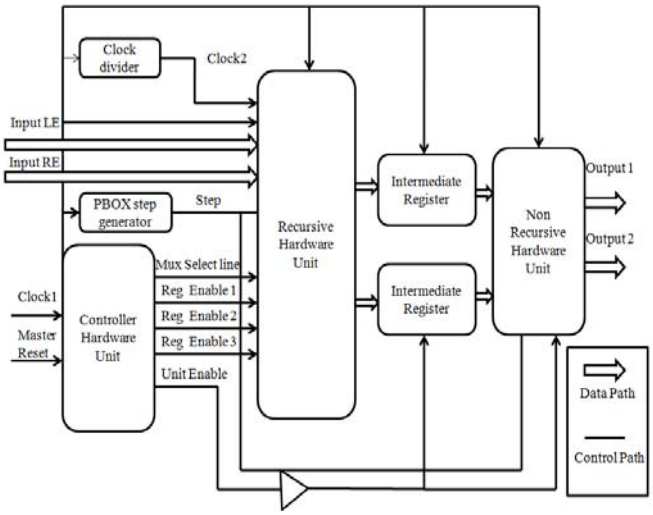

Fig.7. Hardware Architecture of Area Optimized Blowfish Algorithm

The recursive hardware unit as shown in Figure 8 is designed to operate only for a single iterative round at a particular time. It consists of three distinct sub modules isolated by a pair of 32-bit registers (Regn, n=1…7) to hold the intermediate data. This in turn helps in the synchronous operation of the left and right arms of the recursive hardware unit. The multiplexer in the recursive data unit is utilized to distinguish between the plaintext and feedback data. Each sub module requires one clock cycles to complete the mathematical operations and hence a single iterative round is executed in three clock cycle.

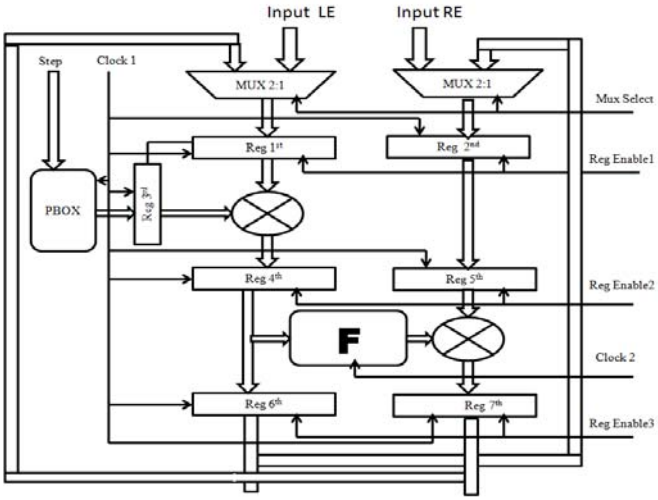

Fig.8.The Recursive Hardware Unit

The controller hardware unit generates several control signals to facilitate the iteration process. In Figure 8,' Mux Select' control signal is utilized to distinguish between plain text and feedback data. The registers are selectively reset and set by using three control signals named 'Reg Enablen '(n=1--3) signals in order to maintain the isolation. The 'Unit Enable' control signal enables the non recursive hardware unit to compute all the mathematical operations of the 17th iterative round of the Blowfish algorithm. This signal only turns true (logic value 1) when the 16 iterations are performed. The clock divider produces an additional clock referred

'clock2' from the master clock referred as 'clock1' in the architecture model. The 'clock2' is utilized for synchronously control the S-BOX of the F function block. The 'PBOX step generator' hardware unit produces the 'step' signal to fetch the P values synchronously from the P box. The state diagram for the entire process is shown in Figure 9.

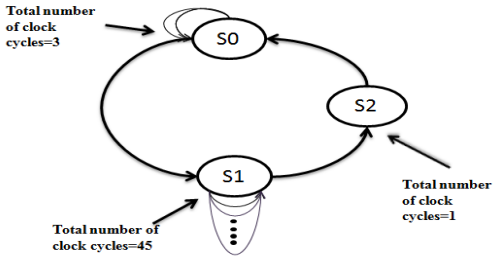

Fig.9. State diagram of Area Optimized Architecture

The state S0 is repeated three times to complete the first iteration round where recursive data unit fetches plaintext. The state S1 is executed fifteen times to complete the rounds 2 through 16 where the recursive data unit functions on the outputs of the previous iteration round. So a feedback path is used to feed the previous output to the recursive hardware unit for the next round. Lastly the state S2 is executed on one time to complete the last round utilizing the non recursive hardware unit.

5. COMPARATIVE ANALYSIS

The area optimized Blowfish algorithm has been implemented using Verilog HDL on ISE 12.1. The design has been routed in Spartan 3E XC3s500e-5fg320. A test bench has written where a plain text of 64 bits was provided as input to the synthesized design. The output was observed using ISim(O.61xd).

When evaluating a given implementation, the hardware resources required and the throughput of the implementation are usually considered the most critical parameters. No established metric exists to measure the hardware resource costs associated with the measured throughput of an FPGA implementation. Two area measurements are readily apparent—logic gates and configurable logic blocks (CLBs) slices. It is important to note that the logic gate count does not yield a true measure of how much of the FPGA is actually being used. Hardware resources within CLB slices may not be fully utilized by the place-and-route software so as to relieve routing congestion. To achieve a more accurate measure of chip utilization, CLB slice count was chosen as the most reliable area measurement. Therefore, to measure the hardware resource cost associated with an implementation's resultant throughput and the throughput per slice metrics are used.

The throughput of a design is the measure of the data computing capacity of a design. It is generally defined in terms of Bits / Second. The formula for computing throughput is given in equation (1).

$$\text{Throughput} = \frac{\text{Frequency} * \text{Input Word Size}}{\text{No.of Clock Cycles}} * \text{No. of Pipelining Stages} \quad (1)$$

Table I. Comparative Study of Area Optimized vs Normal and Pipelined Architectures of Blowfish Algorithm

Parameters	Area Optimized Design	Normal Blowfish Design	Pipelined Design
Number of slices	381	2203	3222
Critical path delay(ns)	3.40	3.40	3.383
Frequency(MHz)	294.131	294.131	295.63
Throughput(bits/sec)	384.77Mbps	588.255Mbps	6.3Gbps
Latency(clock cycles in nano second)	49	33	49
Throughput per Slice (Mbps)	1.01	0.267	1.955

Table I provides the comparative study between different hardware architectures of Blowfish algorithm. It indicates that area is minimized up to 17.3% from the normal architecture and 19% from the pipelined architecture. Throughput per slice for the area optimized architecture is enhanced 1.01 Mbps as compared to a lower throughput per slice of 0.267 Mbps for the normal architecture of Blowfish algorithm while it is slightly less than the throughput per slice of 1.955Mbps of pipelined architecture. Throughput for the area optimized design is 384.77 Mbps as compared to 588.255 Mbps for the normal architecture. However latency for area optimized and pipelined design has increased to 49 clock cycles from 33 clock cycles as provided in Table I. All the other parameters like critical path delay and frequency remaining almost same.

In pipelined architecture, the first input word made of 64 bit requires to be completely encrypted in 49 clock cycles and second word requires fifty two as first hardware module block is fetching second data word after three clock cycles. As a result, subsequent data words need extra 3 clock cycles each from its previous data word. But on the other hand, area optimized and normal architecture fetch the second data word for encryption after 49 and 33 clock cycles respectively. Hence we may conclude that area optimized architecture may be suitable for the application of CRSN where small size packets are transmitted and space complexity is much more predominant than speed whereas pipelined architecture may be suitable for high speed application with compromising the area.

6. CONCLUSION

The simulation results of the area optimized architecture for Blowfish algorithm in Verilog HDL show that number of slices requirement is minimized compared to normal and pipelined architecture that in turn minimized the space complexity. The architecture also improves throughput per slice that in turn minimize the cost of hardware design. However it provides slightly less throughput than the normal architecture. Here other performance parameters like frequency and critical path remain almost the same. Thus the area optimized architecture may be suitable for the application of CRSN that deals with small packet size whereas pipelined architecture may be utilized for CRSN applications that require high speed.

For future developments we desire to design a Feistel network-based encryption algorithm, which would optimize both time and space complexity of the embedded hardware module.

REFERENCES

[1] O. B. Akan, O.B. Karli and O.Ergul, 2009. Cognitive Radio Sensor Networks. IEEE Network(Aug 2009),vol 23,34-40 .doi:10.1109/MNET.2009.5191144.
[2] G. P. Joshi, S. Y. Nam and S. W. Kim,2013. Cognitive Radio Wireless Sensor Networks:Applications, Challenges and Research Trends. Sensors(2013), 13, 11196-11228. doi:10.3390/s130911196.
[3] S.Roy Chatterjee, M. Chakraborty , J. Chakraborty,2011. Cognitive Radio Sensor Node Empowered Mobile Phone for Explosive Trace Detection. Int. J. Communications, Network and System Sciences, 2011, 4, 33-41 doi:10.4236/ijcns.2011.41004
[4] A. Perrig, J.Stankovic and D. Wagner, 2004.Security In Wireless Sensor Networks. Communications Of The ACM, (June 2004),Vol. 47,53-57 .
[5] P. Ganesan, R. Venugopalan, P. Peddabachagari,2003.Analyzing and Modeling Encryption Overhead for Sensor Network Nodes. Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications. (Sept19, 2003) 151-159. ACM 1-58113-764-8/03/0009
[6] B. Schneier,1994.Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish) in Fast Software Encryption.Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, (1994),191-204.
[7] Michael C.-J. Lin, Youn-Long Lin,2000. A VLSI implementation of the Blowfish encryption/decryption algorithm. Proceeding ASP-DAC(2000) Proceeding of the Asia and South Pacific Design Automation Conference, (June 2000),1-2.
[8] A.Nadeem,2005. Performance Comparison of Data Encryption Algorithms. First International Conference on Information and Communication Technology, (27-28 Aug. 2005),84-89.
[9] J. Thakur, N.Kumar, 2011. DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. International Journal of Emerging Technology and Advanced Engineering, (2011)Vol 1, Issue 2.
[10] H. H. Schmit, S. Cadambi and M. Moe2000. Pipeline Reconfigurable FPGAs. Journal of VLSI Signal Processing Systems(2000) 24, 129–146, 2000.
[11] R.Tessier and W. Burleson, “Reconfigurable Computing for Digital Signal Processing: A Survey” Journal of VLSI Signal Processing 28, 7–27, 2001.
[12] S.RoyChatterjee, S.Majumder, B. Pramanik M.Chakraborty, “FPGA Implementation of Pipelined Blowfish Algorithm” , IEEE Fifth International Symposium on Electronic System Design,2014,pp.208-209

AUTHOR PROFILE

Prof. S. Roy Chatterjee presently holds the post of assistant professor in the department of Electronics & Communication Department in Netaji Subhash Engineering College, Kolkata, India. She completed her B.Tech (2004) and M.Tech (2006) from the department of Applied Physics, Calcutta University. Her research interests include communication, networking and cognitive radio. She has published six international journals and conference papers.

Dr. M. Chakraborty presently holds the post of professor and Head of the Department of Information Technology, Institute of Engineering & Management, Kolkata, India. She has done her PhD (Engg) in the field of mobile communication and networking from Jadavpur University (2007), Kolkata, India. Prior to that, she completed her B.Tech (1994) and M.Tech (2000) from the Institute of Radio Physics & Electronics, Calcutta University. Her research areas of interest include cryptographic techniques for network security, network intrusion detection and prevention techniques, sensor networks and cognitive radio. She has authored more than forty international journals and conference papers.