VLSI Architecture For Cipher In 5G New Radio

D. Malathi , S.Suvetha, S.Shanmuganathan ,K.Vignesh

Abstract: Security and privacy are of prime concern in the emerging technologies like internet of things (IoT) and cyber-physical systems (CPS) based applications. Lightweight cryptography plays an major role in securing the data in this emerging pervasive computing environment. The main objective of this project is to implement High performance and Area efficient VLSI architecture with 64-bit data path for present Cipher and to compare the results between present Cipher. Cipher is an algorithm for encryption and decryption operation. It is based on the concept of Substitution-Permutation network. These networks contains both S-boxes and P-boxes which has specified algorithm for each that converts input bits as blocks into output bits .The block runs for 9 clock cycles to get encrypted output and also to get decrypted output. Simulations is done on Model Sim software and synthesis is on Xilinx FPGA device. It gives the throughput of around 3712 Mbps and Efficiency of around 11.56%, this architecture gives the better results when compared to existing Cipher..

Index Terms: Cryptography, High performance, Area, Substitution-Permutation network, Throughput, Efficiency

1. INTRODUCTION

In cryptology, a cipher is an algorithm for encrypting and decrypting data. Symmetric key encryption, also known as secret key encryption, based on the use of ciphers, which operate symmetrically. A cipher converts data by processing the original, plaintext characters (or other data) into cipher text, which should seems to be random data. Commonly, ciphers used two main types of transformation: transposition ciphers, which keep all the original bits of data in a byte but mix their order, and substitution ciphers, which replace specific data sequences with other specific data sequences. For example, one type of substitution would be to convert all bits with a value of 1 to a value of 0, and vice versa. The data output by either of the method is called the cipher text. Modern ciphers enable private communication in many different networking protocols, including the Transport Layer. Security (TLS) protocol also offer encryption of network traffic. Many communication technologies, including phones, digital television and ATMs, depends on ciphers to maintain security and privacy.

2 RELATED WORK

Bassam J. Mohd et.al [2] had presented a comprehensive survey of state-of-the-art research development in lightweight block ciphers' implementation by presenting the taxonomy of the cipher design space and precisely they explained the scope of lightweight block ciphers for low resource devices. But it is little difficult process for hardware and software implementations. A. Bogdanov et.al [4] had described an PRESENT- ultra-lightweight block cipher. They gave equal importance to both security and Hardware design during the design of the cipher and at 1570 GE the hardware that is needed for Cipher is competitive with current leading compact stream ciphers. But it consumes more area because of more number of implementations. Mohamad Sbeiti et.al [5] explored the performance of the PRESENT block cipher on FPGAs and alsothey had provided the implementation results of an efficiency that is, throughput per slice and differentiated them with other block ciphers. Though this Cipher is well suited for high-speed and high-throughput applications, it consumes more power. Carsten Rolfeset.all [6] presented three different types of architecture of the present ultra-lightweight algorithm and they pointed their suitability for both active and passive smart devices. Their implementations require only maximum of 1000 Gate Equivalents. Even though it achieves a relatively high throughput rate; it requires 50% more area. Elif Bilge Kavun and TolgaYalcin [8] proposed the FPGA implementations on two different PRESENT lightweight Ciphers by making use of existing RAM blocks in FPGAs for storage of internal states so that the slice count will be reduced. It does not provide a better throughput and power consumption is more.Francois-Xavier Standaert et.al [11] presented the FPGA implementations of ICEBERG, a block cipher established for reconfigurable hardware implementations and it is presented at FSE 2004. This Cipher implementation results provides a better improvement for hardware efficiency.

F. Macéet.all [12] had explored the performance of scalable encryption algorithm in current field-programmable gate array (FPGA) devices which is initially designed for software implementations in smart cards, controllers, or processors. But its performance is low when compared to other algorithms. Ashraf A.M. Khalaf [13] had analysed the problem in security and presented a triple hill cipher algorithm and implemented in FPGAs to encrypt binary data’s such that images, videos etc. to increase the security level. Though it provides better security but it has complex architecture to achieve the results. Lara-Nino et.al [14] presented a novel based FPGA-based design for the lightweight block cipher PRESENT aiming at obtaining a low-cost design in terms of area. It achieves minimal latency and reduced area but it is critical for the efficiency when the operation frequency follows standardized specifications.

3 PRESENT CIPHER

3.1 Building Blocks

Present cipher is realized with the use of some basic building blocks such as s box, p box, register and round key

- *D.Malathi, Professor, Electronics and communication engineering, Kongu engineering college, Erode, Tamilnadu , India,*
  E-mail: malathid2001@gmail.com
- *S.Svetha is currently pursuing Undergraduate degree program in Electronics and Communication Engineering in Kongu engineering college, Erode, Tamilnadu , India,.*
  E-mail: suvetha99s@gmail.com
- *S.Shanmuganathan is currently pursuing Undergraduate degree program in Electronics and Communication Engineering in Kongu engineering college, Erode, Tamilnadu , India,.*
  E-mail: shanmuganathans33s@gmail.com
- *K.Vignesh is currently pursuing Undergraduate degree program in Electronics and Communication Engineering in Kongu engineering college, Erode, Tamilnadu , India,.*
  E-mail: vigneshkesavan599@gmail.com

scheduling blocks. Brief descriptions of each of these blocks with respect to present cipher are given below.

3.1.1 S –Box

A S-box also known as Substitution box it is a basic accent of4 symmetric key algorithms as in Fig.1. .It under goes substitution process and put back a small block of bits i.e the input bit of the S-box by another block of bits i.eS-box output bit. This method must be one-to-one, to ensure inverse operation(hence decryption).

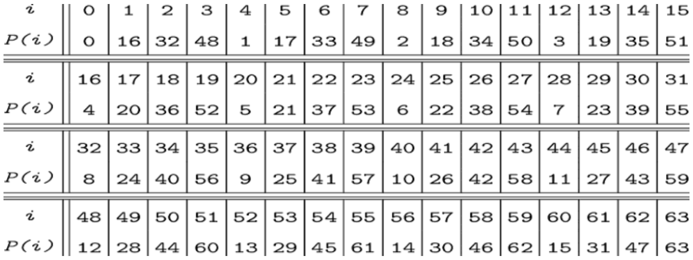
Table 1 *Substitution box for 16-bits.*

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	5	6	B	9	0	A	D	3	E	F	8	4	7	2	1

3.1.1 P –Box

A P-box is also called as Permutation box. It is a permutation operation of all the bits that is ,it takes the outputs of all the S-boxes of one round, undergoes permutation operation, and then ingest them into the S-boxes of the next round. The output bits of any S-box are spreaded as input bits of as many S-box. It is the main property of P-box.

Table 2
Permutation Box for 64 bits



i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

3.1.1 OPERATION OF PRESENT CIPHER

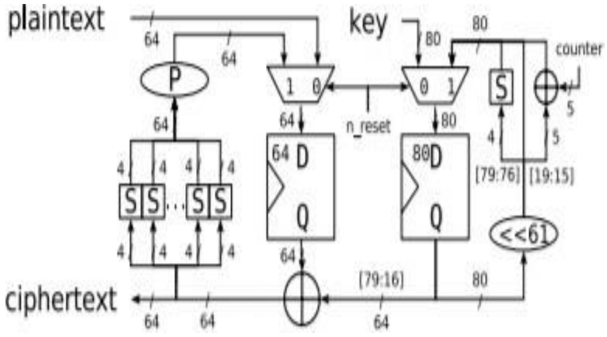


Fig. 1 *The data path of an area-optimized version of the Present-80 encryption*

Fig.1 shows the The data path of an area-optimized version of the Present-80 encryption.The entire cipher control logic was implemented as a 3- state finite-state machine. After reset the first round begins and the two inputs of the algorithm, plaintext and user-supplied key are read from the corresponding registers. The 64-bit and 80-bit multiplexers select the appropriate input depending on the value of the round counter, i.e. initial values for plaintext and key are valid only in round 1. Both 64- bit and 80-bit D-flip- flops are used for round synchronization between the round function output and the output of the key schedule. A Part of the round key undergoes Xor operation with plaintext. Key schedule operation and round function runs in parallel for each round $1 \leq i \leq 31$.

4 PROPOSED METHOD

4.1 4 .1 Present algorithm

The principle of PRESENT cipher is based on the concept of substitution and permutation network (SPN). It works on a block size of 64-bit and supports key length variants of 80-bit. There are total 31 rounds and each round consists of an XOR operation, which is required to introduce a round key Ki, for 0≤i≤31, where K_{31} is used for post-whitening operation. A nonlinear substitution layer operation is performed in each round and this layer consists of a 4-bit S-box which is applied 16- times in parallel. Further, there is a linear bitwise permutation layer.

4.1.1 Key schedule

The cipher requires a unique round key (K_i) in each round, where the input key is stored in a key register $K(K_{79}k_{78}$ .... $k_0)$ for 80-bit key.

4.1.2 Round key operation

With the current state $b_{63}\ldots b_0$ and for the given leftmost 64-bit of the round Key $k_i = k_{63}^i \ldots k_0^i$ $0 \le i \le 31$ the Add Round Key operation is defined as $b_j = b_j \oplus k_j$ for $0 \le j \le 63$.

4.1.3 S Box and Inverse S Box

The PRESENT algorithm requires a 4-bit-to-4-bit S-box (S) as $F_2^4 \to F_2^4$. The 64-bit current state $b_{63}\ldots b_0$ is taken as 16 4-bit words $W_{15}\ldots W_0$, where $W_i = b_{4xi+3} \parallel b_{4xi+2} \parallel b_{4xi+1} \parallel b_{4xi}$ for $0 \le i \le 15$. The output S[wi] provides the updated state values as per [9]. The S-boxes are used in each of the rounds and in the key scheduling operation. For inverse S- box, the S-box does not satisfy S(S(x)) = x; thus, same S-box cannot be used in encryption and decryption both. For the PRESENT cipher, a relation between the S-box and inverse S- box is given by expression $S^{-1}(S(x)) = x$.

4.1.4 P Layer and inverse P layer

The bit permutation layer is used to move bit i of the state to bit position P(i) and it is given by the following expression,

$$P(i) = \begin{cases} i.16 \bmod 63, & i\hat{}\{0,\ldots.62\} \\ 63, & i=63 \end{cases} \qquad (4.1)$$

$$P^{-1}(i) = \begin{cases} i.16\bmod 63, & i\{0\ldots..62\} \\ 63, & i=63 \end{cases} = i \qquad (4.2)$$

4.2 Operation of cipher

To implement the PRESENT block cipher a 64-bit data path is chosen, mainly to implement the permutation operation efficiently. In the proposed architecture there are three main components which are using encryption/decryption engine, key scheduling unit and controller. There is a 1-bit input signal 'enc_dec' which is used to select the encryption or decryption operations. If 'enc_dec' is at logic '1' level then encryption operation is performed, else the decryption operation is executed. An up-down counter facilitates the integrated encryption/decryption operation.

4.2.1 Data path of the Integrated Present Architecture

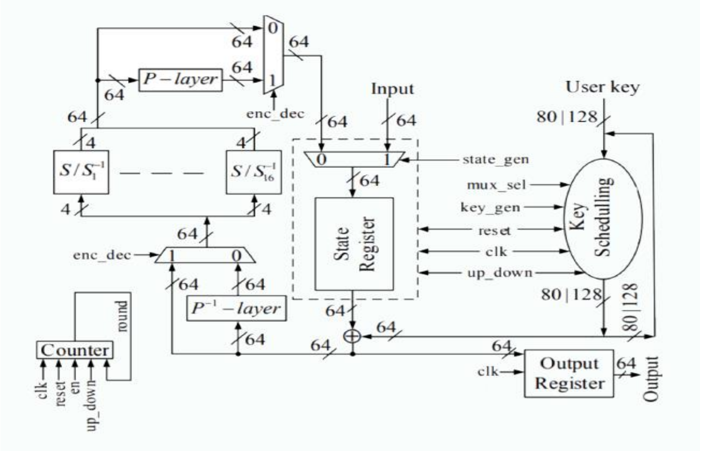


Fig.2 *PRESENT block cipher for an integrated encryption/decryption operation*

The architecture in Fig.2 consists of a set of multiplexers, XOR gates and registers. The bit permutation process and inverse bit permutation process are simple bit transposition operation, which require only simple wirings. There is one 64-bit multiplexer which is required to switch the state between the load phase (Input) and the intermediate state. Next, the multiplexer passes the state to a 64-bit state register. This register is used to store the intermediate state and passes it to the 64-bit XOR gate. This gate performs the XOR operation of intermediate state coming from the state register with 64 bit round key coming from the key scheduling unit. In the architecture, both the S-box (S) and inverse S-box are realized by the area-optimized combinational logic implementation. To differentiate between the encryption and decryption operations, two 64-bit multiplexers are deployed in the data path. In the proposed architecture, the inputs and outputs are registered. The output register is added to synchronize the output with the last round. In this architecture total of 9 clock cycles are required for the encryption operation to get the cipher text. Here, the computed keys have been simultaneously stored in a state register so that there is no need to compute the last round key for other blocks of input. Thus, only 9 clock cycles are required to decrypt the remaining blocks of cipher text. The advantage of using the integrated architecture is that there are some resources which can be used in both encryption and decryption operations.

4.2.2 Key schedule operation

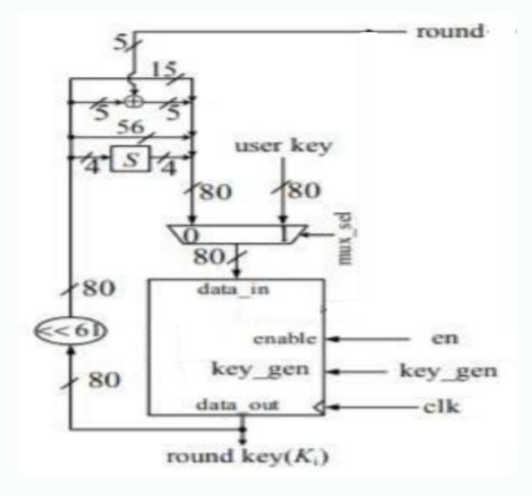


Fig. 3 *The key-scheduling process in the PRESENT Cipher*

Key scheduling unit in Fig.3 works in the storage mode. Here, computation of the round keys is performed only for the first block of data and the computed round keys are stored simultaneously in the BRAM. This computation mode offers a reduced number of clock cycles for the decryption operation. The key storage mode is also beneficial for processing a large chunk of data which contains multiple blocks that has to be encrypted or decrypted with the same key.

5 RESULTS

5. 1 Synthesis Results

Table 3 *Resource utilization*

Elements	Resource Utilization	
	Present Cipher	Cipher -64 bit
LUTs	348 out of 303600	266 out of 28800
Registers	126 out of 607200	321 out of 28800
Slices	126 out of 75900	321 out of 28800

From Table 3 the Resource utilization of LUTs, Registers and Slices for 64-bit Cipher is reduced when compared with existing Cipher (PRESENT Cipher)

5. 2 Synthesis Parameters

Table 4 *Synthesis Parameters*

Elements	RESULTS	
	Existing method	Proposed method
	Present Cipher	Cipher -64 bit
Latency	33 clock cycles	9 clock cycles
Throughput(Mbps)	417	3712
Efficiency(%)	3.32	11.56

From Table.4 the Synthesis Parameters i.e Latency has reduced and Throughput(Mbps), Efficiency is increased for 64-bit Cipher when compared to existing Cipher(PRESENT Cipher).

Throughput is calculated by,
Throughput= (max. frequency x total no. of. bits)/latency
Efficiency is calculated by,
Efficiency=Throughput /no. of. slices

6 CONCLUSION

An integrated VLSI architecture for PRESENT lightweight block cipher of 64-bit is proposed to increases the performance. This architecture supports both the encryption and decryption operations with 80-bit key length. An another 16-bit Cipher architecture was implemented and its results was compared with 64-bit block cipher. The Simulation was carried out using Model Sim Software and synthesis was done in Xilinx -xc5v1x50t FPGA device, it gives the throughput of around 3712 Mbps and Efficiency of around 11.56% . Thus the Synthesis results shows that the necessary parameters required for the architecture gives better results than the existing designs.

7 RFERENCES

[1] Wendt T.Xu. J.B. and Potkonjak M. (2014) “Security of IoT systems: Design challenges and opportunities”, in IEEE/ACM Int'l Conf. on Comp.-Aided Design, San Jose, Califo , pp. 417-423.

[2] Mohd B. J, Hayajneh T and VasilakosA.V.(2015) “A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues”, Jour. of Network and Computer Appl., vol. 58, pp. 73-93.

[3] Eisenbarth T, Kumar S , Paar C, Poschmann Aand Uhsadel L(2007)“A survey of lightweight- cryptography implementations”, IEEE Design & TestofComputers, vol. 24, no. 6, pp. 522-533.

[4] BogdanovA , Knudsen L R , Leander G , Paar C, Poschmann A, Robshaw M.J. B, Seurin Y and VikkelsoeC.(2007) “PRESENT: An ultralightweight block cipher”, Springer, pp. 450-466.

[5] Sbeiti M, Michael S, Poschmann A and Paar C. (2009) “Design space exploration of present implementations for FPGAS”, Springer, pp. 141-145.

[6] Rolfes C, PoschmannA, Leander G and PaarC.(2008) “Ultra-lightweight implementations for smart devices-security for 1000 gate equivalents” Springer ,pp. 89-103.

[7] Biryukov and Alex (2011) "Substitution and Permutation (SP) Network" Encyclopedia of Cryptography and Security,Springer US, 1268-1268.

[8] Yalla P and Kaps J P.(2009) “Lightweight cryptography for FPGAs”, in IEEE Int'l Conf. on Reconfigurable Computing and FPGAs (ReConFig'09), pp. 225-230.

[9] KavunE.B and YalcinT(2011)“RAM-based ultra-lightweight FPGAimplementation of PRESENT”,Springer, pp. 280-285.

[10] Alippi C, Bogdanov A, and Regazzoni F(2014) “Lightweight Cryptography for Constrained Devices”, IEEE, pp.144-147.

[11] Standaert F.X, Piret G, Rouvroy G, and Quisquater J.J(2007)“FPGA implementations of the ICEBERG block cipher”, Integration, vol. 40, no. 1, pp. 20–27.

[12] Mace F, Standaert F.X, and Quisquater J.J(2008) “FPGA implementation(s) of a scalable encryption algorithm”, IEEE Trans. Very Large Scale Integr. Syst., vol. 16, no. 2, pp. 212–216.

[13] Bulens P, Standaert F.X, Quisquater J.J, Pellegrin P, and Rouvroy G(2008), “Implementation of the AES128 on Virtex-5 FPGAs”,in AFRICACRYPT Springer, pp. 16–26.

[14] Wang M(2008) “Differential Cryptanalysis of Reduced Round PRESENT” ser. LNCS, no. 5023. Springer-Verlag, 2008, pp. 40 –49.

[15] Collard B and Standaert F.X(2009), “A statistical saturation attack against the block cipher PRESENT”, in CT-RSA, ser. LNCS, vol. 5473. Springer,pp. 195–210.