Optimized S-Box Architectures of PRESENT Cipher for Resource Constrained Applications

Ruby Mishra, Manish Okade, Kamalakanta Mahapatra
Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela, India
rubymishrabgr1@gmail.com, okadem@nitrkl.ac.in, kkm@nitrkl.ac.in

***Abstract*—This paper presents the design of three different types of S-box architectures for the PRESENT cipher with an aim of optimizing the design parameters for resource constrained applications. Three variants of S-box design are proposed here. The first includes minimization of the existing Boolean function using logic minimization and K-maps, while the other two designs focus on MUX-based design optimizations. The proposed designs give us an idea about the trade-offs involved in the design and performance metrics. FPGA implementations of the proposed design which includes modifying the substitution layer, reduces the slices used by around 27% and also achieves an enhancement of throughput when compared with related state-of-the-art architectures. These results are tailor made to meet the design specifications for IoT enabled devices which to the best of our knowledge has not been investigated earlier.**

***Index Terms*—cipher, substitution layer, permutation layer, round key, IoT applications**

I. Introduction

Increase in demand for pervasive computing opens the scope of development of IoT devices. Design of these devices is a challenge in all aspects because IoT enabled applications aim to work in a constrained environment. Among all the design challenges for such kind of devices, security issues play a crucial role especially from the hardware point of view. Many security algorithms are being implemented on hardware to prevent attacks by intelligent adversaries who primarily aim at system failure. Securing the devices against the attacks of intelligent adversaries falls under the broad domain of cryptography. This is typically carried out by the ciphers and accomplished in hardware by designing cipher architectures. Optimization of these cipher architectures is highly essential for compact applications. PRESENT, HEIGHT, CLEFIA, KATAN, KLIEN, XTEA, LED, PRINCE, Piccolo, SIMON and SPECK, etc. are few examples of ciphers whose architectures can be explored for IoT devices. Major targets of resource constraint ciphers are RFID tags and readers, sensor networks which include electronic payments, e-cash, healthcare monitoring, etc. These applications need compact security algorithms, and hence cipher architecture design gains importance which is the focus of this work.

The context of symmetric block cipher is typically chosen here for the design of substitution box (S-box). A block cipher comprises of many rounds operating on the plaintext [1]. The strength of a crypto algorithm depends on the strength of its S-box. A substitution layer is designed using S-boxes which are functional representation of the input plaintext. The ciphertext depends on this confusion created by the S-box such that the probability of an attack is reduced. The basic design criteria for S-boxes include increased non-linearity, less uniformity in differential properties and property of bijectivity [2]. Many ciphers exist in literature which are suitable for applications requiring low power, smaller area, etc. However, among them the PRESENT cipher proposed in [1] is standardised, simple, along with its suitability for both hardware and software implementations. Hence in this work we concentrate on proposing efficient S-box implementations for the PRESENT cipher along with studying its performance for resource constrained applications.

II. Related Work

In this section we discuss the literature with regard to the different types of S-box implementations proposed for PRESENT cipher. Tay et al. [3] proposed using an LUT based S-box instead of a Boolean S-box wherein they optimized the design by K-map factorization especially for smaller resource utilization. In their simplified design 26 AND gates and 17 OR gates were needed. Sbeiti et al. [4] investigated a S-box from Boolean function using "espresso" tool in which critical path delay is reduced as compared to LUT based S-box implementation. But in this work the design parameters for the ciphers compared are not on the same FPGA platform and also the maximum frequency of operation is less for the design. Kavun et al. [5] explored another approach where they stored the S-box in the RAM for FPGA implementation which yielded less area occupancy and a suitable throughput. But this approach has more number of operation cycles and also has more complex control circuitry. In [6] three different architectures for PRESENT were explored, among which the serial implementation of the design had less resources for hardware implementations but requires more number of processing cycles. A novel, low-cost hardware architecture for PRESENT was proposed in [7] which utilized reduced bit size for substitution and permutation layers. This work was later improved in [8] by the introduction of two new proposals, enabling its use in IoT applications. Iterative and serialized hardware architectures for PRESENT were investigated in [9].

TABLE I: PRESENT S-box

I/P: x	0	1	2	3	4	5	6	7
O/P:S(x)	C	5	6	B	9	0	A	D
I/P: x	8	9	A	B	C	D	E	F
O/P:S(x)	3	E	F	8	4	7	1	2

III. Review of Present Cipher

PRESENT is a symmetric block cipher suitable for IoT based applications proposed in [1] which was standardized under ISO/IEC 29192-2 [10]. It accepts a 64-bit plaintext and supports a key size of 80-bit or 128-bit. In this paper PRESENT-80 is considered for all description and analysis. It consists of a substitution and permutation network and has 31 rounds along with a last round for addition of key. Each round involves the addition of the round keys and also combination of the substitution and permutation layer. Substitution layer which is a non-linear layer consists of sixteen 4×4 bit S-boxes shown in Table I. The permutation layer is a linear layer which requires minimum hardware resources for implementation. The key schedule includes left rotation of the key register by 61-bit, same S-box is used here again for passing the left-most four bits and a round counter including XOR operation. The pseudocode [1] for PRESENT algorithm is given below:

Algorithm 1 PRESENT

Input: (PLAINTEXT).
Output: (CIPHERTEXT).
1. Generate round keys
2. **for** i = 1 to 31 **do**
3. addRoundKey(STATE, K(i))
4. sBoxLayer(STATE)
5. pLayer(STATE)
6. **end for**
7. addRoundKey(STATE, K(32))

A. Review of existing S-Box designs for PRESENT

Existing PRESENT cipher consists of an S-box which is 4-bit to 4-bit function, i.e. S: $F_2^4 \rightarrow F_2^4$ as proposed in [1]. This means that the output S is a function of x, i.e. S(x), where x is 4-bit input and S(x) is also of 4-bit . The S-box in hexadecimal notation is given in Table I. The complete design architecture with various S-boxes added to the existing literature are described in the following sections. The designs denoted as D1, D2 and D3 exist in the literature and here it is revisited briefly;

Design-I, (D1): This design closely resembles the original architecture proposed in [1]. Here the function S(x) is obtained from Table I, in terms of 4-bit output, i.e., $S_0(x)$, $S_1(x)$, $S_2(x)$, $S_3(x)$. The S-box is re-designed in this work by directly implementing these Boolean functions in terms of logic gates. This is done to obtain all the results in the same design platform for fairer comparison with other designs. The symbolic representation of the architecture used to implement this design is shown in Fig. 1 where PT refers to input plaintext and CT refers to output cipher text. State is a register that refers to the intermediate data of the plaintext in various rounds of the encryption process. Key transformation performs the key scheduling of the input key. Round key is generated by the most-significant 64-bits of the output from the key transformation module. SPN refers to the substitution permutation network which consists of the substitution layer and the permutation layer. The round key output is XORed with the state output to get the desired ciphertext after completion of all the rounds.

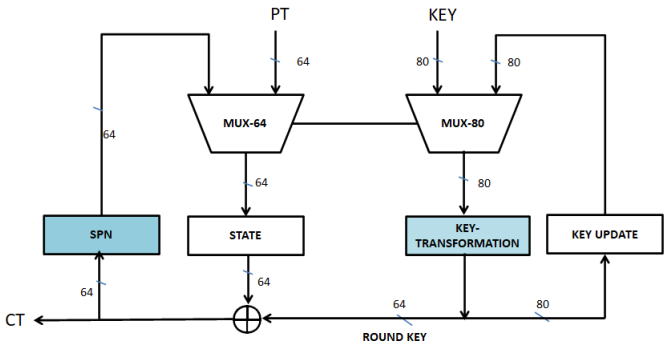

Fig. 1: Present Architecture

Design-II, (D2): It refers to a similar architecture as in D1 except that the dedicated MUX shown in Fig. 1 used for selection of inputs is no longer used in the design. The synthesis tool itself generates the hardware according to the code. Another change is that the substitution box used is LUT-based [1] with reference to Table I.

Design-III, (D3): Another design is implemented by using the technique of elimination of common sub-expression of the original equations for the S-box given in [1]. This approach was given in [3] for 128-bit key; so we have obtained unique common min-terms to reduce the resources and used it for 80-bit implementation of the cipher.

IV. Key Contributions

The contributions of this work are listed below;

1) Special emphasis is given on designing optimized S-box architectures for PRESENT cipher specially keeping in mind the constraints of IoT applications. Three variants of S-box design are proposed here. The first includes minimization of the existing Boolean function using logic minimization and K-maps, while the other two designs focus on MUX-based design optimizations. Since in this work, emphasis is on S-box, the other modules like permutation layer, key scheduling and encryption process is untouched and is same as in [1].
2) In order to carry out a fair comparison between existing S-box designs and proposed three design variants, same FPGA platform is used along with maintaining the key size to be 80-bit. This to the best of our knowledge has not been carried out earlier since in [3], a 128-bit key was used. Additionally, we carried out detailed analysis

TABLE II: Minimised S-Function

S(x)	Function in terms of input x
S_0(x)	$(\bar{x}_2 + x_1) \odot (x_3 \oplus x_0)$
S_1(x)	$\bar{x}_3 x_1 (x_0 \oplus x_2) + x_3 x_0 (x_2 \odot x_1) + \bar{x}_2 (x_3 \oplus x_1) + x_3 (x_2 \odot x_0) + x_1 \bar{x}_0 (\bar{x}_3 + \bar{x}_2)$
S_2(x)	$(x_3 \odot x_2)\bar{x}_1 + x_0 x_2 (x_3 \oplus x_1) + \bar{x}_2 x_1 \bar{x}_0 + x_3 \bar{x}_1 x_0$
S_3(x)	$\bar{x}_3 (x_1 \odot x_0) + (x_3 \oplus x_2) x_1 + x_3 \bar{x}_2 x_0$

showing the improvements obtained in area as well as the reduction in overall resources for the proposed S-box designs.

V. Proposed S-Box Design, (PD)

The architecture is retained same as the base architecture of PRESENT cipher as shown in Fig. 1. The modules highlighted in blue shade indicate the S-box function in SPN network and the key transformation module which is modified and three variants of this implementation are proposed as listed below;

PD-a: In this design, the function used for the S-box is minimized using K-map and then optimized at gate level to obtain minimum number of gates using Boolean theorems. The input and the corresponding output data is first taken from the Table I. The idea is then to design the function using logic gates, and in this work the function is designed specifically using AND, OR, NOT and XOR gates. The Boolean functions obtained for the S-box obtained after all type of minimizations is shown in the Table II. Here the input bits are considered to be $x_0,\ x_1,\ x_2,\ x_3$ and output bits are denoted as $S_0(x),\ S_1(x),\ S_2(x),\ S_3(x)$.

PD-b: In the second variant, the S-box designed here is based on utilization of 4:1 MUX. Here one 4:1 MUX is needed to obtain each output bit of the S-function. To obtain all the four bits of the output, four 4:1 MUXes are required. The inputs to each of the MUX used in the architecture achieves the required functionality for the S-box of the cipher. The truth table is first obtained for the 4-bit input and 4-bit output with reference to the data given in the Table I. If we analyse the combinations for the input bits, $x_0,\ x_1,\ x_2,\ x_3$, we can generate a set of relevant inputs. Hence it is inferred here that the input to the MUX is decided by observing x_1x_0 for constant values of x_2 and x_3 for all input combinations. So each individual 4:1 MUX has x_1x_0 as its select lines, and x_3, x_2 along with some of their logic combination as the input lines. The input bits determined for each 4:1 MUX is given in Table III.

TABLE III: Inputs for 4:1 MUX-Based Design

O/P Bits	Inputs for 4:1 MUX			
	I_0	I_1	I_2	I_3
S_0	$x_3 \oplus x_2$	$x_3 \odot x_2$	x_3	$\bar{x}_3$
S_1	$x_3\bar{x}_2$	x_3	$\bar{x}_3 + \bar{x}_2$	$x_3 \odot x_2$
S_2	$x_3 \odot x_2$	$\bar{x}_2 + x_3$	$\bar{x}_2$	$\bar{x}_3 x_2$
S_3	$\bar{x}_3$	$x_3\bar{x}_2$	$x_3 \oplus x_2$	$\bar{x}_3 + \bar{x}_2$

PD-c: The S-box used here is a slight modification of the PD-b architecture where 8:1 MUX is used instead of 4:1 MUX to satisfy the output function given in Table I. The said design is implemented according to the Table IV. The inputs are obtained in the same manner as discussed in PD-b. One 8:1 MUX is used to get one bit of the output. Here x_2, x_1, and x_0 are the select lines and x_3 along with logic '0' and '1' is the input to the MUX.

TABLE IV: Inputs for 8:1 MUX-Based Design

O/P Bits	8:1 MUX Inputs							
	I_0	I_1	I_2	I_3	I_0	I_1	I_2	I_3
S_0	x_3	$\bar{x}_3$	x_3	$\bar{x}_3$	$\bar{x}_3$	x_3	x_3	$\bar{x}_3$
S_1	x_3	x_3	1	$\bar{x}_3$	0	x_3	$\bar{x}_3$	x_3
S_2	$\bar{x}_3$	1	1	0	x_3	x_3	0	x_3
S_3	$\bar{x}_3$	x_3	x_3	1	$\bar{x}_3$	0	$\bar{x}_3$	$\bar{x}_3$

Hence the the key aspects for the idea behind designing the S-box variants in this work is that, implementation of S-box module plays a crucial role in the results obtained for the design metrics in the overall design of a cipher. An optimised S-box results in total area reduction and increase in throughput of the PRESENT cipher.

VI. Results

A. Design Metrics

In this work, FPGA implementation of the design is presented due to its advantages like re-configurability, flexibility, low cost and shorter design time as compared to ASIC implementation. The design metrics are basically inferred from number of LUTs, flip flops (F/F) and these are contained in a slice whose utilization in the FPGA is given by the slice count (SC). Apart from these the other metrics required for determining the efficiency of ciphers for IoT based applications are maximum throughput and energy consumed [8], which are calculated as given in Eq. (1) and Eq. (2). Power consumed is used to determine the energy required for the encryption process.

$$Throughput(TP) = \frac{(N_b \times F_{max})}{Latency} \tag{1}$$

where, N_b=block size, in this case 64 bits, F_{max} is the maximum frequency of operation of the design and latency is the number of clock cycles required to obtain the 1st ciphertext for the given plaintext input.

TABLE V: Resource Utilization and Performance Report

VIRTEX-5 (XC5VLX50T-3-FF1136)							
SOURCE	D1	D2	D3	PD-a	PD-b	PD-c	[8]
#LUT	245	248	215	215	215	215	190
#F/F	149	149	149	149	149	149	153
SC	86	76	66	**67**	**66**	**68**	67
Fmax(MHz)	553.98	298.450	611.883	611.833	611.833	611.833	542.30

$$Energy = \frac{(Power \times Latency)}{F_{max}} \quad (2)$$

where, Power is the total power consumed by the design.

B. Implementation Details and Analysis

In this design we have used Virtex-V XC5VLX50T FPGA. The resource utilization and performance reports are generated using Xilinx-14.7 tool. The tables and other performance metrics are based on the results obtained in PAR (Place and Route) and Timing report. The detailed comparison of the architecture described in Section-V for PRESENT-80 is given in Table V. We can infer here that the slice count and number of LUTs for PD has been reduced significantly, i.e., by an average of 28% decrease in SC as compared to D1 and by 25% when compared to D2. Also there is an increase in maximum frequency as compared to D1 and D2. Among the three proposed designs of PD, PD-b with 4:1 MUX proves to be best as it has reduced slice count. Design in [8] has though reduced number of LUTs and slice count as compared to D1 and D2 but has increased number of flip flops when compared to PD, and also has less maximum frequency of operation.

Table VI shows that throughput is high for PD i.e. 1223.67 Mbps. This value meets the frequency requirement of IoT devices which is 100 kbps to 1 Mbps. Also the energy required for the encryption of the plaintext of the cipher has reduced significantly when compared to D1, D2 and [8]. It is also to be noted that the total power consumption of the designs D1, D2, D3 and PD is 560 mW. The values for each of these design are almost same due to similar architectures whereas [8] reports the total power as 562.75 mW. In this work, we have used maximum operating frequency, Fmax and also 13.56 MHz frequency (for IoT applications) for calculation of throughput and energy using Eq.(1) and Eq.(2). Table VI

TABLE VI: Throughput and Energy Report

Source	D1	D2	D3	(PD-a-c)	[8]
Latency (Cycles)	32	32	32	32	133
Throughput(THR)at Fmax(Mbps)	1107.96	596.9	1223.67	**1223.67**	260.96
THR/SC at Fmax	12.88	7.85	18.54	≈**18.17**	3.89
THR(Kbps)/SC at 13.56 MHz	320	362.1	416.96	≈410	97.31
Energy(μJ)	0.03	0.06	0.02	**0.02**	0.138

shows that we have achieved the desired results. We have obtained a reduction in the number of resources and achieved high throughput for high speed applications as compared to D1, D2 and [8]. Based on the results in Table V and Table VI we can note that there always prevails a trade-off for area, cost and performance for VLSI designs which depends on the target application.

VII. Conclusions and Future Work

This paper investigated optimized S-box variants for the PRESENT cipher especially meant to be used in resource constrained environments like IoT devices. Three S-box have been designed and optimized here with an aim to reduce the number of resources and attain high throughput with less energy consumption to make the overall design suitable for IoT based applications. It is seen that MUX-based design of S-box proves to be efficient one. This work has also established a fair comparison among various designs by simulating the designs on the same platform. In future the designs can be further optimized specifically for low power applications and detailed cryptanalysis can be carried out which is also under consideration.

References

[1] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2007, pp. 450–466.

[2] L. Mariot, S. Picek, A. Leporati, and D. Jakobovic, "Cellular automata based s-boxes," *Cryptography and Communications*, vol. 11, no. 1, pp. 41–62, 2019.

[3] J. Tay, M. D. Wong, M. Wong, C. Zhang, and I. Hijazin, "Compact FPGA Implementation of PRESENT with Boolean S-Box," in *2015 6th Asia Symposium on Quality Electronic Design (ASQED)*. IEEE, 2015, pp. 144–148.

[4] M. Sbeiti, M. Silbermann, A. Poschmann, and C. Paar, "Design Space Exploration of Present Implementations for FPGAs," in *2009 5th Southern Conference on Programmable Logic (SPL)*. IEEE, 2009, pp. 141–145.

[5] E. B. Kavun and T. Yalcin, "Ram-Based Ultra-Lightweight FPGA Implementation of PRESENT," in *2011 International Conference on Reconfigurable Computing and FPGAs*. IEEE, 2011, pp. 280–285.

[6] C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Ultra-Lightweight Implementations for Smart Devices–Security for 1000 Gate Equivalents," in *International Conference on Smart Card Research and Advanced Applications*. Springer, 2008, pp. 89–103.

[7] C. A. Lara-Nino, M. Morales-Sandoval, and A. Diaz-Perez, "Novel FPGA-based low-cost Hardware Architecture for the PRESENT Block Cipher," in *2016 Euromicro Conference on Digital System Design (DSD)*. IEEE, 2016, pp. 646–650.

[8] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Lightweight Hardware Architectures for the Present Cipher in FPGA," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2544–2555, 2017.

[9] N. Hanley and M. ONeill, "Hardware Comparison of the ISO/IEC 29192-2 Block Ciphers," in *2012 IEEE Computer Society Annual Symposium on VLSI*. IEEE, 2012, pp. 57–62.

[10] I. 29192-2, "Information Technology – Security techniques –Lightweight cryptography – Part 2: Block ciphers," January 2012.