Novel Hardware Architectures for PRESENT Block Cipher and its FPGA Realizations

1st Ruby Mishra
School of Electronics
Kalinga Institute of Industrial Technology
Bhubaneswar, India
rubymishrabgr1@gmail.com

2nd Manish Okade
Dept. of ECE
National Institute of Technology
Rourkela, India
okadem@nitrkl.ac.in

3rd Kamalakanta Mahapatra
Dept. of ECE
National Institute of Technology
Rourkela, India
kkm@nitrkl.ac.in

***Abstract*—The paper investigates novel hardware architectures for PRESENT Block Cipher with the motivation of its applicability to IoT applications. PRESENT has been chosen for two reasons: firstly, it belongs to the lightweight cipher category, and secondly, existing works haven't fully focused their attention on power metric optimization of this cipher. The Substitution Permutation Network (SPN) module of PRESENT cipher is optimized by modifying its datapath and utilizing additional hardware units that significantly reduce power consumption and achieve high throughput. The novel aspect of the SPN module design is the input selection and feeding technique to the substitution and permutation layers via the hardware units comprising multiplexers. The optimized SPN module is then included in the overall encryption architecture of PRESENT for performance analysis. The proposed architectures have been evaluated on NEXYS4 DDR FPGA at an RFID operating frequency of 13.56 MHz, making them suitable for IoT applications. Additionally, the paper also throws light on how a designer can optimally harness the resources available in an FPGA architecture to achieve improvement in the performance of the cipher architecture. Comparative analysis with state-of-the-art shows dynamic power reduction by 28.57% and a reduction of 32.81% in the area for the proposed architectures. Besides, performance parameters like the throughput of the proposed design have been significantly improved while maintaining an optimized energy consumption when compared with state-of-the-art architectures.**

***Index Terms*—lightweight cipher, PRESENT, FPGA, low-power, datapath optimization.**

I. Introduction

The rise in demand for RFID-based consumer gadgets has led to a thirst for computation anywhere and anytime. IoT has been beneficial in establishing effective communication between the devices and the user, and the nature of computation is pervasive. The devices deployed in an IoT environment are constantly involved in handling sensitive data and communicating with each other and end users. Although the accomplished connectivity improves people's lifestyles, increasing the need for IoT applications, it also creates data security and privacy concerns. As a result, there is a considerable need to secure the data from intelligent adversaries. Furthermore, because of their limited size, power, and lower computational capacities, the applications in this category are referred to as resource-constrained. Consequently, balancing security and design optimization trade-offs to improve the performance of resource-constrained applications gains importance. This calls for encryption architectures to be lightweight and satisfy the required performance yardsticks, unlike traditional cryptography techniques. Another parameter that needs to be addressed by the designers is the size or area occupied by the hardware. FPGAs are mostly used for prototyping of a design. Though the technology node has been shrinking, the design implemented on an FPGA should utilize the hardware resources efficiently.

The Substitution Permutation Network (SPN) module is an important module in a cipher architecture. It comprises the substitution layer with S-boxes and the permutation layer. The S-boxes are the crucial units that determine the security and performance of the encryption algorithm. The encryption process involves feeding the plaintext and the key to the architecture, which passes through the SPN module and the key scheduling module for certain rounds, generating the ciphertext. This calls for optimising the SPN module containing the nonlinear substitution and permutation layers. After the optimization of the SPN module, it also needs to be plugged into the overall cipher architecture to gauge its performance. Hardware architectures can be designed by adopting design strategies like parallel, loop unrolling, pipelining, iterative, or serialization. To achieve this, there are several challenges because the change in size and style of the datapath changes the design and performance metrics. Parallel architectures increase the speed of operation, but the hardware utilization is very high. In loop unrolling, the identical copy of the architectures is replicated depending on the unrolling factor, increasing the design's area consumption. Pipelined architectures are obtained by adding registers to the parallel architectures, which achieves increased throughput, but using registers increases the power consumption. Serialized architectures no doubt reduce the area and power of the design, but the latency increases considerably. Additionally, round-based design is mainly used for lightweight ciphers because they simultaneously have less delay and area. Once the hardware architecture option is fixed, the size of inputs and datapath are decided to meet the specifications.

The investigations carried out in this paper focus on the architectural design of lightweight symmetric block ciphers to significantly improve the design and performance metrics so

TABLE I
DIFFERENT ARCHITECTURES FOR PRESENT CIPHER

Works	Datapath Size	Key Size	#S-boxes	FPGA	ASIC
[2]	64	80/128	16	Spartan-III	
[2]	64	80/128	16		180nm
[3]	16	80/128	4	Virtex-V, Spartan-6	-
[4]	16	80/128	4	Virtex-II,V, Spartan-6	-
[5]	8	128	2	Virtex-II,V	-
[6]	64	80/128	16	-	180nm
[7]	64	80/128	16	-	180nm
[8]	64	80/128	16	Virtex-V	-

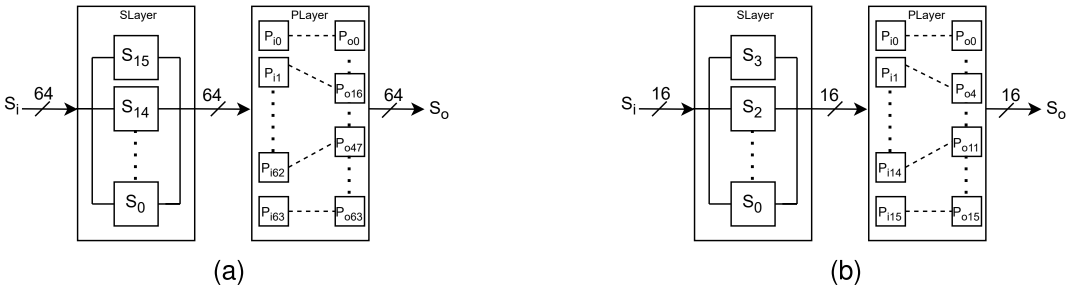

Fig. 1. General Architectures for SPN Module of PRESENT cipher. (a) Original Architecture A_0 with 64-bit SPN Module (b) Serial Architecture A_1 with 16-bit SPN Module

that they can be utilized for secure IoT applications. To demonstrate the usefulness of the proposed optimization techniques, the PRESENT lightweight cipher is utilized for prototyping purposes. The SPN module of PRESENT cipher is optimized by modifying its datapath and utilizing additional hardware units to achieve low power consumption. The optimized SPN module is then included in the overall encryption architecture of PRESENT cipher for performance analysis. The paper highlights how the resources available in an FPGA architecture can be optimally occupied by a design, thereby improving the performance of an encryption architecture. Specifically, the NEXYS4 DDR FPGA is used for design implementation. Therefore, special design attention is focused on optimally harnessing the F7 and F8 MUXes that are available in Artix-7 FPGA architecture. Besides, since the area and performance metrics depend on the architecture of the individual modules in the datapath [1], its optimizations are explored along with fully exploiting the targetted FPGA architecture.

II. LITERATURE REVIEW

PRESENT is a symmetric block cipher suitable for IoT-based applications proposed by Bogdanov et al. [2]. PRESENT cipher was designed to achieve low area when implemented in hardware platforms and is suitable for resource-constrained applications like RFID tags and sensor networks. Since then, there have been a lot of developments in optimizing the hardware architecture of PRESENT lightweight cipher for area, power, and delay parameters. A few variants of the standard PRESENT cipher optimized in previous works, implemented on either FPGA or ASIC platforms are outlined in Table I. As observed from Table I, the PRESENT cipher architectures have been designed in a parallel, pipelining, or iterative fashion.The evaluation and comparison of existing architectures are either not for the same datapath size or not on the same platform. The iterative 64-bit architecture proposed in [2] was evaluated on both FPGA and ASIC platforms. However, their design was evaluated on older FPGA families, leaving enough scope for optimization in terms of the design and performance metrics on newer FPGA family variants. In [4], a parallel and iterative architecture of PRESENT was proposed that consumed more area and had high latency. A 16-bit datapath architecture for PRESENT proposed in [3] has more latency due to using four S-boxes in the 16-bit datapath. A serial architecture having an 8-bit datapath for a 128-bit PRESENT cipher was proposed in [5], which consumed more slices and achieved less throughput. Loop-unrolling technique-based architecture was proposed in [6]. However, it occupied more area which was its major drawback. The architecture proposed in [8] utilized BRAM in the key-scheduling module, which consumed more area and power. Another built-in self-test for PRESENT cipher was proposed in [9], but the hardware utilization of the design was higher. PRESENT cipher has been chosen for design and evaluation because it has been standardized under ISO/IEC 29192-2 [10] and motivates the growing research for optimization of design and performance metrics of encryption architectures. Two notable PRESENT architectures, namely the original PRESENT architecture proposed by Bogdanov et al. [2] and serial architecture proposed by Lara et al.

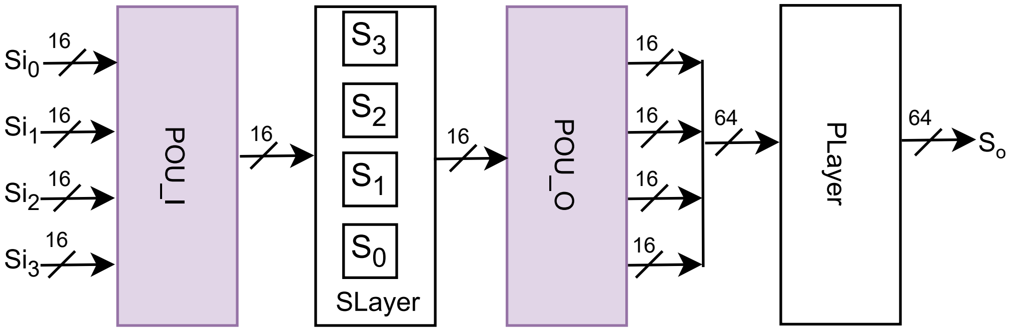

Fig. 2. Proposed Architecture for SPN Module of PRESENT, PD_{SPN}

[3], are briefly described below since they are utilized for comparative analysis with the proposed architectures. These two architectures have been preferably selected in this work because [2] originally proposed the standard architecture for PRESENT cipher while the architecture in [3] has similar datapath size with the proposed architecture.

1) The SPN module for the PRESENT cipher proposed by Bogdanov et al. [2] is denoted as A_0 in Fig. 1a and refers to the original architecture. The architecture has a 64-bit substitution layer (SLayer) and a permutation layer (PLayer). SLayer has a 4-bit S-box; thus, sixteen S-boxes are present in the SPN structure. The output bits of the SLayer are then permuted in the PLayer, and the mappings are shown in the figure. The advantage of this architecture is that all the outputs can be retrieved by passing through the SPN layer in one iteration only.
2) The serial architecture proposed by Lara et al. [3] denoted as A_1, in Fig. 1b, has a 16-bit datapath with a 16-bit SLayer and PLayer. MUX21 and registers are utilized to pass the input bits into the SPN module. The SLayer comprises four S-boxes to meet the 16-bit I/O requirement of the SPN structure. Therefore, four iterations are required to obtain the final output of the substitution function. The PLayer consists of the sixteen input pins mapped to the output pins.

III. Key Contributions

1) Hardware architecture of substitution permutation network for PRESENT Cipher has been proposed to achieve better performance.
2) The architecture consume less power and exhibit high throughput when compared with the state-of-the-art, due to the proposed power optimization unit added to the architectures.
3) The proposed architectures have been evaluated for a frequency of 13.56 MHz which validates the functionality for IoT environment.
4) The evaluation and comparison of proposed architecture has been carried out for both FPGA and ASIC platform to exemplify the usage.

IV. Proposed Architecture for SPN Module of PRESENT Cipher

The proposed architecture, denoted as PD_{SPN}, is shown in Fig. 2. The SPN module optimization is carried out in two ways. Firstly, by datapath optimization and secondly, by including novel hardware units called power optimization units (POU) at the input and the output to reduce power consumption. The PD_{SPN} architecture consists of four 16-bit input pins (Si_0, Si_1, $\ldots Si_3$) and one 64-bit output (S_0). The input pins are fed to the substitution layer (SLayer) via an input power optimization unit (POU_I). Similarly, the output of the SLayer is fed to an output power optimization unit (POU_O) that gives the four 16-bit output. A single SLayer, with only four S-boxes, is utilized instead of sixteen S-boxes for 64-bit input requirement as in original PRESENT [2] architecture. The 64-bit PLayer is unchanged and permuted according to the original architecture of PRESENT [2] PLayer. The POU_I and POU_O are designed using adaptable multiplexers and demultiplexers, respectively, as described below:

The power optimization unit (POU) utilizes multiplexers as its basic design component. POU is based on selectively applying the inputs to a multi-bit MUX. Though writing an HDL code for a MUX is a trivial task, a flexible architecture suitable for passing the inputs through it highlights the significance of the POU. A generic formulation for determining the number of input variables and zero logic values to be assigned to a MUX is illustrated in Eq. (1) and Eq. (2), which can be applied to create a POU architecture for given dimensions.

$$\#(Input\ variables) = \begin{cases} 1 & : \ k = 1 \\ N \times k & : \ k > 1 \end{cases} \tag{1}$$

$$\#(Zero\ values) = \begin{cases} T - 1 & : \ k = 1 \\ ((T \times k) - (N \times k)) & : \ k > 1 \end{cases} \tag{2}$$

where (N) is the number of inputs in a specific architecture, the size of each input is k, and T represents the size of the MUX to be utilized, i.e. number of inputs to a MUX. The choice of T solely depends on the available hardware

TABLE II
INPUT MATRIX FOR 16-BIT MUX81 FOR FOUR INPUT DESIGNS

MUX81 Inputs ⟶	I[0]	I[1]	I[2]	I[3]	I[4]	I[5]	I[6]	I[7]
MUX0	Si_0[0]	Si_1[0]	Si_2[0]	Si_3[0]	0	0	0	0
MUX1	Si_0[1]	Si_1[1]	Si_2[1]	Si_3[1]	0	0	0	0
MUX2	Si_0[3]	Si_1[3]	Si_2[3]	Si_3[3]	0	0	0	0
.	.	.	.	.	.	.	.	.
MUX15	Si_0[15]	Si_1[15]	Si_2[15]	Si_3[15]	0	0	0	0

resource on the FPGA or standard cells or specifications of target applications. In this paper, NEXYS4 DDR FPGA is used for design implementation, and therefore, a 16-bit multiplexer (MUX) of size 8:1 (MUX81) is chosen since F7 and F8 MUXes are present in Artix-7 FPGA architecture. Hence, here *k*=16 and *T*=8. The POU architecture design is based on the idea of first assigning MUX81 inputs with the design's input variables and assigning the remaining inputs to logic '0' values. After carrying out an exhaustive evaluation of possible cases, it is observed that for a *k*-bit MUX81, the total number of input variables assigned should be $N \times k$, and the input pins assigned with logic '0' is ($T \times k$) - ($N \times k$). Assigning logic '0' values to the input reduces switching activity at these nodes, because the zero-to-one and one-to-zero transitions reduces. This results in reducing the dynamic power consumption and utilizing hardware efficiently.

In this work, the proposed PD_{SPN} architecture has a 16-bit datapath, similar to Lara et al. [3] except for the inclusion of the POU, which reduces the power consumption considerably. The POU is also used in the inner core of the SLayer for input selection. From the above formulations, to feed the four inputs (Si_0, Si_1, Si_2, or Si_3) of the SPN, the input values of the POU are customized. POU_I unit shown in the SPN architecture comprises sixteen MUX81. By following the techniques highlighted earlier with regard to the POU, the four input variables are applied to the first four inputs of each MUX81. The remaining inputs of each MUX81 are applied with '0' values. Therefore, the total number of input variables applied are (4×16=64), and the total number of '0' values applied are (128−64=64), which is obtained using Eq. (1) and Eq. (2), with *N*=4, *k*=16, and *T*=8. The concept is demonstrated using Table II, where each row represents the input variables and '0' values that are applied to the sixteen number of MUX81 (MUX0-MUX15) inputs, shown as I[0]-I[7]. Apart from these inputs the select lines are of 3-bit, since *T*=8 and depending on the select values the inputs are selected simultaneously for each MUX81. The 16-bit output of POU_I are then utilized to feed the SLayers inputs simultaneously. This is beneficial in selecting the inputs depending on the select lines and hence consumes less power. Similarly, the POU_O is customized using demultiplexers, where four outputs are utilized, and others are filled with zeros. The four 16-bit demultiplexer outputs are combined to form 64-bit, which is then fed to the PLayer. In total, four iterations are required in this case to obtain the output since four inputs must be selected. However, in general the number of iterations will depend on the value of *N*, The number of iterations to achieve the final output and other features of the proposed architecture and existing SPN architectures is shown in Table III. The idea of splitting the 64-bit input into 16-bit is to harness the advantages of POU. The inputs are fed parallel to the architecture with the optimized area, and at the same time, due to the POU structure, the inputs are selected all at a time, thereby reducing the latency. The size of inputs, outputs, datapath, and the POU can be altered according to the requirement for a specific design.

TABLE III
FEATURES OF THE ARCHITECTURES FOR SPN MODULE

Designs	I/O Size	#S-boxes	#Iterations
A_0 [2]	64	16	1
A_1 [3]	16	4	4
PD_{SPN}	16/64	4	4(for *N*=4)

V. OVERALL ARCHITECTURES FOR PRESENT CIPHER

This section outlines the complete PRESENT cipher encryption architectures (PA) by plugging the proposed PD_{SPN} SPN module. The key size chosen is 80-bit. The plaintext and ciphertext size of the cipher architecture is split into four 16-bit inputs to match with the SPN I/Os and inner core datapath. Comparative analysis is carried out with the original PRESENT architecture [2] and PRESENT architecture proposed by Lara et al. [3] as enumerated below with specific notations.

1) A_{A0}: The original complete PRESENT architecture given by Bogdanov et al. [2] that utilizes the SPN module A_0.
2) A_{A1}: Modified complete PRESENT architecture proposed by Lara et al. [3] that utilizes the SPN module A_1.
3) PA_{SPN}: Proposed complete PRESENT encryption architecture using proposed PD_{SPN} SPN module.

All the above architectures are designed and simulated on the same FPGA platform for fair comparison. A detailed description of the results is given in the subsequent sections.

TABLE IV
DESIGN AND PERFORMANCE METRICS EVALUATION OF SPN MODULE ON NEXYS4 DDR FPGA

Designs (SPN module)	LUT (#)	Slices (#)	Total Power (W)	Dynamic Power (W)	CPD (ns)
A_0 [2]	64	16	0.124	0.032	9.97
A_1 [3]	8	4	0.15	0.008	7.89
PD_{SPN}	17	5	0.093	0.002	8.87

TABLE V
DESIGN AND PERFORMANCE METRICS EVALUATION OF PRESENT ENCRYPTION ARCHITECTURE

Designs (PRESENT Arch.)	FPGA Family	LUT (#)	Slices (#)	FF (#)	Total Power (W)	Dynamic Power (W)	CPD (ns)
A_{A0}, [2]	Artix-7	169	64	68	0.098	0.007	4.79
A_{A1}, [3]	Artix-7	133	40	92	0.096	0.005	5.45
[6]	Virtex-5	-	81	-	-	-	1.56
[9]	Kintex-7	215	97	154	0.073	0.013	1.44
[9]	Spartan-6	225	68	156	-	-	3.16
[8]	Virtex-5	177	55	151	0.687	0.126	1.75
PA_{SPN}	Artix-7	94	43	52	0.097	0.005	3.87

VI. RESULTS AND DISCUSSION

A. Simulation Results for proposed SPN architectures of PRESENT cipher

The SPN architectures described are all simulated on the NEXYS4 DDR (Artix-7) FPGA platform. FPGAs, have been chosen for their compatibility to parallel computations and high flexibility. The proposed architectures have been constrained with a clock frequency of 13.56 MHz, suitable for IoT and RFID applications. The evaluated parameters include the area in terms of LUTs, slices and flip flops (FF), critical path delay (CPD), total power consumed, and dynamic power consumption, which indicates the switching activity of the design. The evaluated values in Table IV show the proposed PD_{SPN} SPN module consumes the least total power and exhibits the least dynamic power consumption compared to state-of-the-art designs A_0 ($\downarrow$ 25%) and A_1 ($\downarrow$ 38%). The reasons for achieving the optimization in power metric in PD_{SPN} is due to the usage of POU_I and POU_O in its architecture. These selector modules prove very beneficial while passing and retrieving multi-bit input to the SPN structure, whereas A_0 (64-bit), A_1 (16-bit) does not inculcate any power efficient multiplexers. Hence, depending on the requirement of the target application and the availability of the I/O size, a particular design can be selected with one or more optimized designs and performance parameters.

B. Simulation Results for Complete PRESENT Architectures

The proposed architectures for the complete PRESENT cipher are evaluated on the FPGA platform as illustrated in Table V. The PA_{SPN} architecture has an almost comparable number of slices as [3], but there is a reduction of LUTs by 29% and FFs by 43%. It is also observed that the total power and the dynamic power consumption are more or less equal to [3] due to the 16-bit I/Os and the overall architectural similarity. Due to the inclusion of POU, similar areas and power consumption are achievable with more speed in the proposed designs due to a decrease in the latency of the architecture. PA_{SPN} architecture has a reduction of 32.81% slices, 28.57% dynamic power, and 19.21% CPD when compared with A_{A0}. This is because A_{A0} is a 64-bit datapath architecture without any inclusion of multiplexer-based selector modules. Comparing PA_{SPN} with [6] i.e. 64-bit datapath, has a trade-off for speed but is efficient for low area requirements because the number of slices is decreased by 46.91%. This illustrates that the MUX-based designs are primarily suitable for high-speed applications due to mapping the multiplexers with appropriate FPGA resources. Besides, it has been observed that the architecture in [9] has more slices on Kintex-7 and Spartan-6 FPGA, but the CPD is less compared to the proposed PRESENT architecture in this paper. Further, the PRESENT architecture implemented on Virtex-5 [8] has more slices and higher power consumption compared to PA_{SPN} but has an advantage of less CPD. Therefore, it can be observed that the design and performance metrics solely depend on the core architecture style. However, the number is also dependent on the FPGA family being utilized.

C. Performance Analysis

In this section, the performance metrics have been illustrated as a proof of concept. Three parameters considered in this work are switching activity in terms of toggle count,

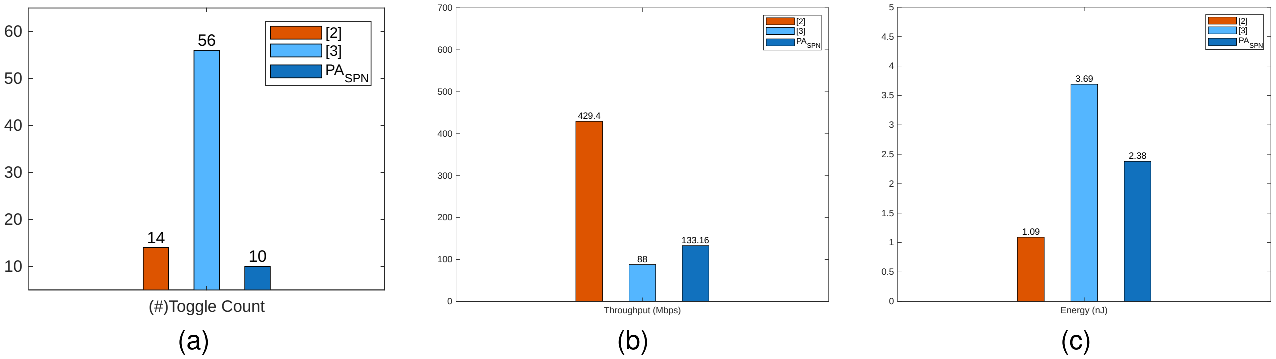

Fig. 3. Comparison of Performance Metrics for PRESENT Encryption Architecture(a)Switching Activity. (b)Throughput at $Fmax$. (c) Energy at $Fmax$

throughput($(Fmax \times Bs)/latency$) in terms of Mbps, and energy ($(latency \times powerconsumed)/Fmax$) in terms of nJ. The maximum frequency of operation, $Fmax$, is evaluated by utilizing the CPD values, and Bs is the block size of the architecture. The switching activity affects the architecture's dynamic power consumption. Further, if we consider the side-channel resiliency of a design, the lesser the switching activity, the lesser the information extracted by the attackers. Therefore, the toggle count information representing the switching activity for the same input patterns was obtained from the SAIF files. The proposed PRESENT encryption architectures have better performance metrics, as shown in Fig. 3 as compared with [3] which has a similar datapath. Regarding the architecture in [2], it has more throughput and less energy consumption even if the $Fmax$ is less when compared to the proposed architecture. The reason is the latency of the architecture in [2] is 31 cycles whereas for the proposed architecture the latency is 124 cycles. However, [2] has more switching activity and dynamic power consumption compared with the proposed architectures. The two reference architectures were chosen for comparison because these have been re-evaluated on the NEXYS4 DDR FPGA at 13.56 MHz. The optimized metrics were improved for the proposed architectures because of the utilization of the POU module. Specifically, toggle count depends on the architecture, throughput values are affected by the $Fmax$ and latency, which are further dependent on the delay values, and finally, the power consumption and delay values affect the energy consumption. The performance metrics have been optimized, making the proposed architectures suitable for IoT applications.

VII. Conclusion

The paper focused on the architectural design optimization of PRESENT cipher to significantly improve the design and performance metrics so that it can be utilized for secure IoT applications. PRESENT cipher was chosen since it belonged to the lightweight symmetric block cipher category, making it suitable for the IoT environment. The SPN module was optimized by modifying its datapath and including hardware POUs at the input and output paths to reduce power consumption. The POUs adopt a unique technique for reducing power consumption rather than a conventional low-power optimisation technique. The novel aspect of the SPN module design is the input selection and feeding technique to the substitution and permutation layers. The optimized SPN module was then included in the overall PRESENT cipher architecture for performance analysis. As demonstrated in the results section, the proposed architectures meet the design and performance requirements to a great extent when compared with state-of-the-art architectures. Side-channel analysis of the proposed architectures and application of the POU modules to other lightweight ciphers or digital circuits can be possible future work of this paper.

Acknowledgment

This work is supported by SERB, Government of India, under grant number MTR/2023/000106.

References

[1] D. Mukhopadhyay and R. S. Chakraborty, *Hardware security: design, threats, and safeguards*. CRC Press, 2014.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] B. Rashidi, "High-Throughput and Lightweight Hardware Structures of HIGHT and PRESENT Block Ciphers," *Microelectronics Journal*, vol. 90, pp. 232–252, 2019.

[7] ——, "Efficient and High-Throughput Application-Specific Integrated Circuit Implementations of HIGHT and PRESENT Block Ciphers," *IET Circuits, Devices & Systems*, vol. 13, no. 6, pp. 731–740, 2019.

[8] J. G. Pandey, T. Goel, and A. Karmakar, "Hardware Architectures for PRESENT Block Cipher and their FPGA Implementations," *IET Circuits, Devices & Systems*, vol. 13, no. 7, pp. 958–969, 2019.

[9] Z. Haider, K. Javeed, M. Song, and X. Wang, "A low-cost self-test architecture integrated with present cipher core," *IEEE Access*, vol. 7, pp. 46 045–46 058, 2019.

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