A VLSI Architecture for the PRESENT Block Cipher with FPGA and ASIC Implementations

Jai Gopal Pandey[1], Tarun Goel[2], Mausam Nayak[1,3], Chhavi Mitharwal[1,3], Sajid Khan[4], Santosh K. Vishvakarma[4], Abhijit Karmakar[1], and Raj Singh[1]

[1] CSIR- Central Electronics Engineering Research Institute, Pilani, India- 333031
{jai,abhijit,raj}@ceeri.res.in
[2] Central Research Laboratory, Bharat Electronics Ltd., Bangalore, India-560013
{tarungoel.com}@gmail.com
[3] Banasthali Vidyapith, Vanasthali, Rajasthan, India-304022
{nayak.mausam,chhavimitharwal}@gmail.com
[4] Indian Institute of Technology Indore, Simrol, Indore, India-453552
{phd1601102015,skvishvakarma}@iiti.ac.in

Abstract. The infrastructure of internet-of-things (IoT) and cyber-physical systems (CPS) is based on the security of communicated data. Here, lightweight cryptography plays a vital role in IoT/CPS resource-constrained environments. In this paper, we propose an architecture for the PRESENT lightweight block cipher and its VLSI implementation in an FPGA and ASIC. The input-output ports of the architecture are registered and data-path is based on 8-bit. It requires 49 clock cycles for processing of 64-bit *plaintext* with 80-bit user key. The FPGA implementation of the proposed architecture is done in Xilinx Virtex-5 device in comparison to an existing design improved performance has been obtained. Further, an ASIC implementation of the architecture is done in SCL 180 nm technology where gate equivalent (GE) of the design is 1608 GEs and area of chip is 1.55 mm^2. At 100 MHz operating frequency, total power consumption of the chip is 0.228 mW. A throughput of 130.612 Mbps, energy 112.15 nJ, energy/bit 14.018 nJ /bit, and 0.813 efficiency is obtained.

Keywords: PRESENT block cipher· Lightweight cryptography· VLSI architecture· FPGA · ASIC.

1 Introduction

The foundation of internet-of-things (IoT) [1], cyber-physical systems (CPS) [2], [3] and edge computing [2] technologies heavily rely on communicated data. As shown in Fig. 1, data can be between human-to-machines and their any combination [4]. In this fast-growing computing infrastructure, small computing devices are deployed for sensing, control, communication and computation needs. Subsequently, deployment trend of devices for IoT/CPS applications, bring a drastic change in designing of electronic circuits and associated systems. Here, the design metrics are being optimized for resource, latency, power and bandwidth [1].

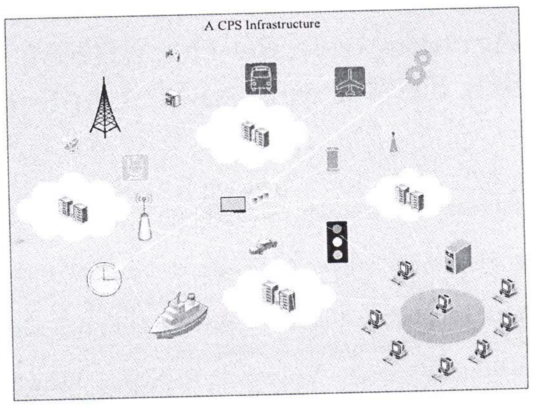

Fig. 1. A modern infrastructure of IoT, CPS and edge computing technologies.

Secure communication is very essential for restricting access of information to unauthorized persons or machines [3], [5], [6]. In this concern, cryptography plays a vital role. In cryptography, the encryption operation is used to convert data into a secure form that is known as ciphertext. Hardware-based security solutions with symmetric key cryptography algorithms are ideally suited to meet the IoT/CPS security challenges [5]. In a survey of lightweight-cryptography ciphers [7], it has been mentioned that, efficient implementation of the ciphers are closely dependent on the selection of appropriate architectures for achieving low implementation complexity and high performance. In the context of lightweight cryptography, ISO/IEC 29192-2 has standardized symmetric block cipher algorithm PRESENT in the year 2012 [8]. The algorithm provides adequate security goals along with hardware-oriented performance attributes which makes it a prominent choice for developing lightweight cryptographic applications [9].

In this paper we propose a hardware architecture for the PRESENT block cipher and its VLSI implementations for a filed-programmable gate array (FPGA) device and an application-specific integrated circuit (ASIC). The architecture is based on 8-bit datapath, works on the 64-bit input and of 80-bit user key and provides 64-bit ciphertext in 49 clock cycles. The FPGA implementation of the architecture is done in Xilinx Virtex xc5vlx50 device and the architecture is proposed with an existing 8-bit datapath based architecture. In comparison, the proposed architecture outperforms in performance. The ASIC implementation of the proposed architecture is done in SCL 0.18 m technology. Here, the chip operates on 3.3V and the core works at 1.8V. The total power consumption of the chip is 2.45 mW, where, the dynamic component is 0.211 mW and the static one is 0.016 mW. With 100 MHz operating frequency the obtained a throughput 130.162 Mbps, energy 112.15 nJ, energy/bit 14.018 nJ/bit is obtained. The efficiency of the design is 0.813. Total area of chip is 1.55 mm^2 with 1608 two input *NAND* gate equivalent (GE).

Rest of this paper is structured as: a brief introduction of the PRESENT algorithm is provided in Section 2. Section 3 discusses about some of the related work. An architecture for the PRESENT cipher is proposed in Section 4. An experimental setup and detailed about the used EDA tools are described in Section 5. Section 6 provides experimental results and comparison with an established architecture. Finally, conclusions are given in Section 7.

2 The PRESENT Algorithm

The PRESENT algorithm works on block sizes of 64-bit with 80/128-bit user keys and requires 31 internal rounds [9]. Each of the 31 round consist of an XOR operation, which is required to introduce a round key K_i for $0 \leqslant i \leqslant 31$, in which K_{31} is used for post-whitening operation. Further, there is a linear bit-wise permutation layer and a non-linear substitution layer. The non-linear layer uses a single 4-bit S-box. The algorithmic steps is depicted in Fig. 2. It requires mainly four functions, which are explained in the following subsections.

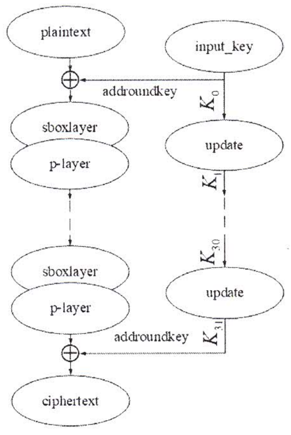

Fig. 2. A top-level algorithmic description of PRESENT cipher.

2.1 The Key Schedule

The cipher requires a unique round key (K_i) in each round, where the input key is $K(k_{79}k_{78}...k_0)$ for 80-bit key or $K(k_{127}k_{126}...k_0)$ for 128-bit key length.

2.2 Add Roundkey Operation (AddRoundKey)

With current state $b_{63}...b_0$ and for the given leftmost 64-bit of the round key $K_i = k^i_{63} \dots k^i_0$ (or $K_i = k^i_{127} \dots k^i_{64}$);$0 \leqslant i \leqslant 31$, the *AddRoundKey* operation is defined as $b_j = b_j \oplus k^i_j$ for $0 \leqslant j \leqslant 63$ [9].

2.3 Substitution Box Layer

The cipher requires a 4-bit S-box (S) as $\mathbb{F}_2^4 \rightarrow \mathbb{F}_2^4$ [9]. 64-bit current state $b_{63}...b_0$ is taken as sixteen 4-bit word $w_{15}...w_0$, where, $w_i = w_{4\times i+3} \parallel w_{4\times i+2} \parallel w_{4\times i+1} \parallel w_{4\times i}$ for i $0 \leq i \leq 15$. $S[w_i]$ provides updated state value [9]. S-box is used in each rounds and key scheduling operation.

2.4 Bit Permutation (*p-layer*)

Bit permutation layer is used to move bit i of the *State* to bit position $P(i)$ [9].

3 Some of the Existing VLSI Architectures for the PRESENT Block Cipher

Architectural space exploration with iterative, parallel and serial variants are discussed in [10] and [11]. The serial architecture provided by [12], requires 303 clock cycles for complete encryption. A rough estimate of the gate count is around 1100, 2-input *NAND* gates, known as gate equivalents (GEs). Iterative type of architectures are reported in [10], [11] and [13]. The architectures of [10] and [11] take 250 clock cycles for complete processing with 1100 GEs. The architecture of [13] is based on 64-bit datapath, 33 clock cycle latency and requires 57 slices for 64-bit key and 69 slices for 128-bit on Xilinx xc5vfx70t FPGA device.

Architectures provided in [10] and [12] are based on 8-bit datapath. To bring external 80-bit *plaintext* and 128-bit key, 16 clock cycles are needed. To process the intermediate State for the encryption operation, 31 clock cycles are required. As the output of the architecture is of 8-bit, it also needed an 8 clock cycles for producing the output. Thus, the total latency of the architecture is 55 clock cycles with 1800 GEs. The architecture of [12] when implemented in Xilinx Virtex-5 xc5vlx50 FPGA device takes 87 slices [12] with 47 clocks latency. Maximum clock frequency 221.64 MHz and throughput is 341.64 Mbps.

4 A VLSI Architecture for the PRESENT Lightweight Block Cipher

A proposed architecture for the PRESENT block cipher is shown in Fig. 3. Here, 8-bit datapath has been selected, it provides an optimal trade-off in terms of power and performance. The three main components of the architecture are: encryption engine, key scheduling and a controller. The key scheduling block takes 80-bit input key and generates thirty one intermediate round keys for the 31 individual rounds of the cipher. The permutation is a simple bit-transposition, which needs connecting wires only. The main building blocks of the proposed architecture have been arranged in different subsections that are described below.

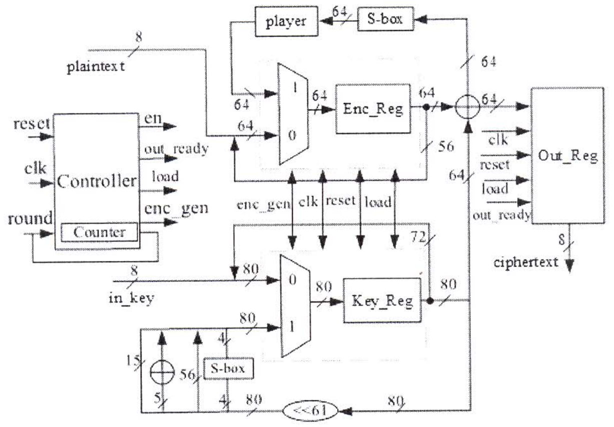

Fig. 3. Proposed architecture for the PRESENT cipher.

4.1 Datapath of the Proposed Architecture for the Encryption Operation in the PRESENT Cipher

The proposed architecture is based on 8-bit datapath consists of a 64-bit *Enc_Reg* and an 80-bit *Key_Reg* register. These registers are used for for storing internal states of encryption operation and intermediate round keys respectively. Two 64-bit and an 80-bit (or a 128 bit) multiplexers are used to switch the data between load and round computation phases. The datapath contains sboxlayer (16 S-boxes) and one S-box for 80-bit (two S-boxes for 128-bit key) for the key scheduling operation. Along with this, one 64-bit XOR gate, 5-bit XOR gate and a 3-bit and one 5-bit up-counters are also used. The *plaintext* is loaded in the first clock cycle. In the next clock cycle multiplexer switches data and then for next 31 cycles all intermediate states are computed.

Data is available at the *Enc_Reg* and XORed with intermediate round key. Further, the mixed *State* is passed to the *sboxlayer*, which provides 64-bit data concurrently to the *P-layer*. Subsequently, the data is passed to the *Enc_Reg* through a multiplexer shown in Fig. 3. In last clock cycle, ciphertext is available at the *Out_Reg* register. Thus, a total of $10 + 31 + 8 = 49$ clock cycles are required to encrypt a single block of 64-bit *plaintext*. The key processing unit works *on-the-fly* with each round. A 64-bit register is used to store the round key, the first leftmost 64-bit of key register is XORed with the intermediate state. The *sboxlayer* is implemented by area-optimized combinational logic circuit.

4.2 A Controller for the Proposed Architecture

A controller as shown in Fig. 4 is designed to generate various required signals for controlling the key generation and encryption engine. In the figure, There are six states in the FSM, it generates four control signals which are: *en*, *io_load*, *en_gen* and *out_ready*.

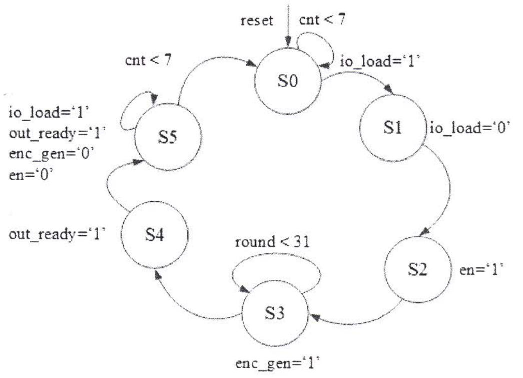

Fig. 4. FSM for the PRESENT cipher.

In state S0 the signals *io_load*='1', which enables the *Enc_Reg* register for storing first byte of *plaintext* and the user key is also stored in the same clock cycle. The state remains in S0 for eight clock cycles, by this complete 64-bit of *plaintext* and 64-bit of 80-bit user key are stored in the respective registers. The remaining two bytes of data is stored in subsequent two clock cycles of states S1 and S2. In state S2, the signal *en*='1', enables a 5-bit up-counter for counting 31 rounds of the cipher. In state S3, multiplexers are switched and the *enc_gen* signal is used to start the intermediate operations by enabling the encryption and key registers. The state remains in S3 until counter value reaches 31. Then, the state is switched to state S4 where counter is disabled by *en*='0'and out_ready signal is switched to logic '1'. First byte of ciphertext is available after 42 clock cycles further in state S5. the state remains in state S5, for next eight clock cycles and ciphertext is available through the output register (*Out_Reg*). Thus, the architecture requires a total of 49 clock cycles to produce 64-bit registered ciphertext for the 80-bit user key. Similarly, for 128-bit user key the controller needs six more clock cycles to produce a registered output.

5 Experimental Setup & used EDA Tools

The proposed architecture is implemented in VHDL language and simulated in *Modelsim* simulator. This simulator is also utilized to generate switching activity interchange format (SAIF) file with one hundred random vectors. for computation of power dissipation using Xilinx *XPower Analyzer* [14] tool. This tool also requires two additional files, which are device specific. The first file is a physical representation of the design mapped to components in native circuit description

(NCD) format. The second file is a physical constraints file (PCF). These files are generated by synthesizing design for Xilinx Virtex-5 xc5vlx50 FPGAs device using Xilinx *ISE Design Suite*[15]. After validating the proposed architecture in FPGA device, an ASIC implementation of the proposed architecture in SCL 180 nm technology [16] is done. Here, Synopsys *Design Compiler* is used to synthesize the architecture. To perform physical design and power analysis, Synopsys *IC Compiler* and *Power Compiler* tools are utilized [17]. Performance of the proposed architecture is evaluated in terms of throughput, maximum frequency, power, energy, energy/bit and efficiency. Throughput, energy and efficiency are computed as:

$$Throughput = \frac{(\text{max}.\ freq. \times total\ bits)}{latency} \tag{1}$$

$$Energy = \frac{(power \times latency)}{operating\ freq.} \tag{2}$$

$$Efficiency = \frac{throughput}{gate\ equivalent} \tag{3}$$

The gate equivalent (GE) is replaced by total number of consumed slices for the FPGA implementation. The VLSI implementation results are provided in following subsections.

6 Experimental Results and Discussion

Experimental results for simulation, synthesis and an ASIC implementation are provided. These results are arranged in different subsections and described below.

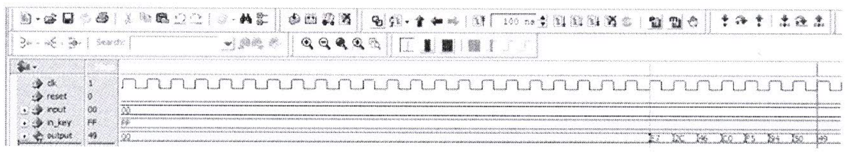

Fig. 5. A simulation result for the proposed architecture with 64-bit plaintext and 80-bit key in *Modelsim* simulator.

6.1 Simulation Result of the Proposed Architecture

A snapshot of the simulation result using *Modelsim simulator* is shown in Fig. 5. Here 64-bit X"000...0" data as a *plaintext* and 80-bit data as user key X"*FF*...*F*" are provided to the proposed architecture. Each data of the *plaintext* and key is sent in byte-wise fashion. First byte of output X"*E*72*C*46*C*0*F*5945049" is obtained a registered output after 42 clock cycles, which is shown at the first cursor in Fig. 5. In subsequent seven cycles all the output bytes are available which is pointed-out by second cursor in the figure.

6.2 Results for the FPGA Implementation and Comparison with an Existing Architecture

An FPGA synthesis result for the proposed architecture is performed and the design is compared with an existing architecture. In comparison, the selected design metrics are: LUTs, registers and total number of consumed slices for resource comparison. Performance of the design is compared for latency, maximum operating frequency, throughput (1), energy (2) and efficiency (3). The FPGA device synthesis and comparison result is shown in Fig. 6.

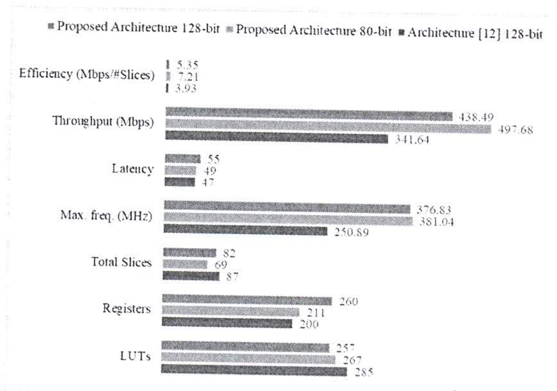

Fig. 6. An architectural comparison between the proposed architecture and the architecture [12] in Xilinx Virtex-5 xc5vlx50-1ff324 FPGA device.

Xilinx *XPower Analyzer* tool [14] with its default environment settings is used for performing power analysis with Virtex-5 xc5vlx50-1ff324 FPGA device. Power consumption of the proposed architecture at the maximum operating frequency is given in Table 1. Here, power distribution in terms of dynamic power and static power are provided. As the dynamic power consumption depends on switching activity, we used one hundred random vectors for the computation of power. The associated components of the dynamic power also have been provided. Total dynamic power dissipation is 49.63 mW for 64-bit key and 27.73 mW for the 128-bit key.

Apart from computing power dissipation at the maximum operating frequencies, it has also been computed at a constant frequency of 13.56 MHz. This particular operating frequency is most commonly used in RFID applications and in IoT transmitters [11]. Table 2 shows the computed power, energy and energy/bit for both 80 and 128-bit user keys.

Table 1. Power Consumption at Maximum Operating Frequency for Xilinx Virtex-5 xc5vlx50-1ff324 FPGA Device

Elements		Value	
		80-bit	128-bit
Dynamic Power (mW)	Clock	14.86	19.60
	Logic	14.86	3.21
	Signals	19.26	4.34
	IOs	0.85	0.59
	Total (PD)	49.63	27.73
Static Power (PS) (mW)		422.06	421.81
Total Power (PD + PS) (mW)		471.69	449.54
Energy (nJ)		60.66	65.61
Energy/bit (nJ/bit)		0.95	1.03

Table 2. Power Consumption at 13.56 MHz Operating Frequency for Xilinx Virtex-5 xc5vlx50-1ff324 FPGA Device

Elements		Value	
		80-bit	128-bit
Dynamic Power (mW)	Clock	1.68	2.87
	Logic	4.65	4.77
	Signals	9.41	9.37
	IOs	0.33	0.33
	Total (PD)	16.07	17.33
Static Power (PS) (mW)		421.68	421.69
Total Power (PD + PS) (mW)		437.45	439.03
Energy (μJ)		1.58	1.78
Energy/bit (μJ/bit)		0.025	0.028

6.3 Results for the ASIC Implementation in SCL 180 nm Technology

The ASIC implementation of the proposed architecture is done in SCL 180 nm technology, a view of the chip layout is shown in Fig. 7. Here, all the data is obtained after completing*placement and routing* (*PnR*) of the design. The core requires, clock (clk), reset, 8-bit input data, 8-bit user key and 8-bit output. Where, *pc3d01* is used as an input pad; *pc3o01* for output pad and *pc3c01* for the clock pad, where *pfrelr* is used as corner pads. For supplying 3.3 V power *pvda* pad is used with *pv0a* as ground. Similarly, for 1.8 V power *pvdi* pad is used with *pv0i* as ground. Area of the complete chip is 1.55 mm^2 and it takes 1608 two input *NAND* gate equivalent of standard cells. The standard cell area is 0.029594 mm^2, where, pad cell area of 0.770 mm^2, core size area 0.296283 mm^2 and pad core size area is 0.555293 mm^2.

Performance of the proposed architecture in terms of power dissipation, throughput, energy, energy/bit and efficiency in SCL 180 nm Technology is given in Table 3. Here, Synopsys *Design Compiler*, *IC Compiler*, *Power Compiler* and *Prime Time* tools are used for physical design and power analysis. *Power Com-*

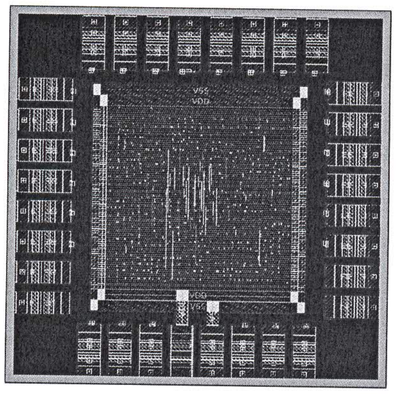

Fig. 7. A layout of the proposed architecture in SCL 180 nm technology.

piler tool requires *SAIF* and standard parasitic exchange format (*SPEF*) files. The *SPEF* file is generated by *IC compiler* tool after completing place and route (*PnR*) and the *SAIF* file is obtained by *ModelSim* tool. At 100 MHz clock fre-

Table 3. Performance of the Architecture in SCL 180 nm Technology.

Elements	**Utilization**	
Dynamic Power (mW)	IO pads	0.087
	Switching	0.051
	Cell internal	0.073
	Total dynamic power	0.211
Static Power (mW)		0.0163
Total Power (mW)		0.228
Throughput (Mbps)		130.612
Energy (nJ)		112.15
Energy/bit (nJ/bit)		14.018
Efficiency		0.813

quency, total power dissipation is 0.2288 mW for the 80-bit key and 0.27267 mW for the 128-bit key. *Design Compiler* and *IC compiler* tools provide netlist in verilog format at multiple stages of the design. The netlists generated by the tools are used to perform functional simulations using *ModelSim* simulator. The simulations results match with the simulation result of Fig. 5. Finally, in next section, conclusion of the paper is drawn.

7 Conclusion

We presented a VLSI architecture for the PRESENT block cipher algorithm. Datapath of the architecture is based on 8-bit and it requires 49 clock cycles for processing of 64-bit *plaintext*, 80-bit user key. An FPGA implementation of the architecture is done in Xilinx Virtex-5 xc5vlx50 device. In comparison to an existing architecture, the proposed architecture shows improvement in performance. Also, an ASIC implementation of the proposed architecture is done in SCL 180 nm technology. Area of chip is 1.55 mm^2, with 1608 gate equivalent (GE). At 100 MHz operating frequency, total power consumption of the chip is 0.228 mW. A throughput of 130.612 Mbps, energy 112.15 nJ, energy/bit 14.018 nJ /bit, and 0.813 efficiency is obtained. The proposed design can be utilized in IoT/CPS applications for data security.

References

1. M. Alioto. *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*. Springer, 2017.
2. W. Shi and S. Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016.
3. E. A. Lee and S. A Seshia. *Introduction to embedded systems: A cyber-physical systems approach*. MIT Press, 2016.
4. S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.
5. T. Xu, J. B. Wendt, and M. Potkonjak. Security of iot systems: Design challenges and opportunities. In *Proceedings of the 2014 IEEE/ACM Int'l Conf. on Computer-Aided Design*, pages 417–423. IEEE Press, 2014.
6. Al-Sakib K. Pathan. *Securing cyber-physical systems*. CRC Press, 2015.
7. T. Eisenbarth and S. Kumar. A survey of lightweight-cryptography implementations. *IEEE Design & Test of Computers*, 24(6), 2007.
8. ISO/IEC 29192-2:2012. Information technology – security techniques – lightweight cryptography – part 2: Block ciphers.
9. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, and et al. Present: An ultra-lightweight block cipher. In *CHES*, volume 4727, pages 450–466. Springer, 2007.
10. C. Rolfes, A. Poschmann, G. Leander, and C. Paar. Ultra-lightweight implementations for smart devices–security for 1000 gate equivalents. In *Int'l Conf. on Smart Card Research and Advanced Applications*, pages 89–103. Springer, 2008.
11. 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*, 2017.
12. N. Hanley and M. ONeill. Hardware comparison of the iso/iec 29192-2 block ciphers. In *VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symp. on*, pages 57–62. IEEE, 2012.
13. J. G. Pandey, T. Goel, and A. Karmakar. An efficient vlsi architecture for present block cipher and its fpga implementation. In *Int'l Symp. on VLSI Design and Test*, pages 270–278. Springer, 2017.
14. Xilinx. Xilinx power estimator user guide.
15. Xilinx. Ise design suite.
16. Semi-Conductor Laboratory (SCL). Semi-conductor laboratory (scl), gov. of india, 2018.
17. Synopsys. Synopsys products, 2018.