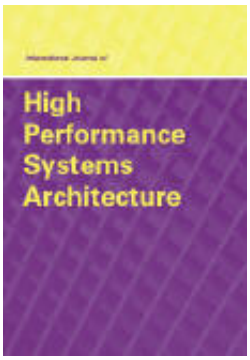


International Journal of High Performance Systems Architecture

ISSN online: 1751-6536 - ISSN print: 1751-6528
https://www.inderscience.com/ijhpsa

Efficient hardware implementation of SIMECK lightweight block cipher

Shweta Kumari, Zeesha Mishra, Bibhudendra Acharya

DOI: 10.1504/IJHPSA.2023.10054397

Article History:

Received:	11 July 2022
Last revised:	08 August 2022
Accepted:	05 September 2022
Published online:	06 April 2023

Efficient hardware implementation of SIMECK lightweight block cipher

Shweta Kumari

Department of Electronics and Communication Engineering,
National Institute of Technology Raipur,
CG, 492010, India
Email: shwetabhagat59@gmail.com

Zeesha Mishra

Department of Microelectronics and VLSI,
CSVTU Bhilai,
Durg, 491107, India
Email: zmishra.phd2016.etc@nitrr.ac.in

Bibhudendra Acharya*

Department of Electronics and Communication Engineering,
National Institute of Technology Raipur,
CG, 492010, India
Email: bacharya.etc@nitrr.ac.in
*Corresponding author

Abstract: The internet of things (IoT) has recently expanded, resulting in a new world of smart gadgets with substantial security consequences. For their vital security role, lightweight block ciphers have gained a significant amount of development in low resource devices (LRDs). SIMECK is a new lightweight block cipher family that incorporates the finest aspect of both SIMON and SPECK. SIMECK is a more efficient block cipher than SIMON and SPECK cipher. These lightweight ciphers are especially referred to as an alternative to the AES for RCD. In this study, area optimised architecture is implemented for SIMECK lightweight block cipher with sizes: 64/128. For implementation on different platforms such as Sparton-6, Sparton-3, Virtex-7, Virtex-6, Virtex-5 and Virtex-4 FPGA are used to examine several properties such as block size, key scheduling, and throughput, among others. The proposed area optimised architecture have attained a maximum operating frequency of 496.429 MHz with 61 slices and a high throughput of 706.032 Mbps on the Virtex-7 platform.

Keywords: lightweight cryptography; resource constrained devices; IoT; internet of things; low resource devices; FPGA; field programmable gate array; advanced encryption standard.

Reference to this paper should be made as follows: Kumari, S., Mishra, Z. and Acharya, B. (2023) 'Efficient hardware implementation of SIMECK lightweight block cipher', *Int. J. High Performance Systems Architecture*, Vol. 11, No. 3, pp.129–136.

Biographical notes: Shweta Kumari received her Bachelor's degree in Electronics and Communication Engineering from the National Institute of Technology Raipur, India. Currently, she is an MTech student pursuing a degree in VLSI and Embedded System from the National Institute of Technology Raipur, India. Her research interests include efficient hardware implementation and optimisation of lightweight cryptographic ciphers for resource-constrained environments, verification, and VLSI system design.

Zeesha Mishra has earned her MTech (2015) in VLSI Design from Chhattisgarh Swami Vivekananda Technical University, Bhilai India and PhD in Electronics and Communication Engineering from National Institute of Technology Raipur India. She is currently serving as an Associate Professor in the Microelectronics and VLSI Department, Chhattisgarh Swami Vivekanand Technical University Bhilai, India. She research interests include cryptography, high-performance, architectures, hardware security, and FPGA-based designs. She has more than 18 research publications in national/international journals and conferences.

Bibhudendra Acharya received his MTech in Telematics and Signal Processing from National Institute of Technology Rourkela, India and PhD in Electronics and Communication Engineering from National Institute of Technology Rourkela, India. He is currently serving as an Associate

Professor in the Electronics and Communication Engineering Department, National Institute of Technology Raipur, India. His research interests include cryptography, signal processing, high-performance architectures, hardware security, and FPGA-based designs. He has more than 100 research publications in national/international journals and conferences.

1 Introduction

In the realm of cryptography, lightweight cryptography is among the fastest developing fields of study. It's a subset of cryptography that focuses on providing security solutions for resource-constrained applications like radio-frequency identification (RFID) tags, wireless sensor networks (WSNs), and medical equipment (Patro and Acharya, 2019). Lightweight cryptography has grown in importance during the last many years. It is inspired by the requirement for unique primitives that can function on devices having extremely little processing capability. This encryption mechanism attempts to strike a balance between security, area, and performance. The key length determines a trade-off between area and safety. The amount of encryption rounds aids in achieving the optimal performance and security compromise. The architecture's cost and performance are determined by the type of architecture developed (Mishra et al., 2020). The most frequent forms of lightweight Ciphers are block and stream Ciphers, which are explored in the next section.

The internet of things (IoT), which is a collection of technologies with energy constraints and real-time requirements, integrates multiple autonomous embedded sensory items with communication intelligence. Many degrees of security can affect the IoT: access to intellectual property (IP), cyber terrorism, and sabotage in complex structures such as industrial automation and traffic monitoring (Yang et al., 2015). Compromise is at the heart of cryptographic design, and there is a desire to use components with strong cryptographic features for robust security and to build in a high margin of security by taking a method many times longer than appears necessary (Lara-Nino et al., 2017). Lightweight cryptography is designed for applications such as RFID, IoT, wireless sensor networks, and wireless body area networks (WBANs). Cryptographic researchers will develop security solutions for these widely used gadgets. Since it allows a message to be delivered via an unsecured channel, these ciphers are the foundation of secure systems. Symmetric-key encryption is performed by converting a block of input plaintext to an output ciphertext via a secret key (Ying et al., 2020). Although the AES algorithm gives a high-security level but every application does not require this much high security. This is one of the primary motivations behind the creation of new lightweight block ciphers and other advanced algorithms (Hanley and O'Neill, 2012). The US National Security Agency (NSA) introduces the SIMON and SPECK lightweight block ciphers based on addition rotation exclusive-OR (ARX) (Sheikpour et al., 2019). SIMECK is another new lightweight block cipher provides the maximum security along with smaller area as compare to SIMON (Philipcris et al., 2020). Lightweight block ciphers are suitable for low resource devices (LRDs) and have applications in various fields like RFID, smartcards, IoT, WBAN and much more (Maene and Verbauwhede, 2015).

1.1 Previous work

A group of experts at the US National Security Agency submitted the SIMON and SPECK ciphers publicly in June 2013 (Lara-Nino et al., 2017). Both the ciphers are lightweight in nature and have a family of key and block size ranges from 64 to 256 bits and 32 to 128 bits respectively, making them appropriate for low-area cryptographic systems and hardware implementations. In paper (Yang et al., 2015), optimised hardware architectures were implemented on different Spartan and Zynq families having 32-bit serial data path of LED and parallel data path of SIMECK and SIMON with 64/128 algorithms. In paper, Ying et al. (2020), author proposed a flexible architecture in different FPGA platforms, where the compactness of the cipher is preserved and providing a versatile architecture that serves adaptive security using a variable key-size. Paper (Abed et al., 2019), presented serial and parallel architectures having a trade-off between the area and the throughput. Paper (Beaulieu et al., 2015) examines loop unrolled hardware implementations of 6 lightweight cipher, as well as an AES implementation as a baseline, in order to meet the requirements for low latency, single-cycle with a small footprint.

1.2 Contribution

- In the serial datapath architecture, the design is reduced, which results in reduced area footprint while providing the better trade-off between the area and the performance.
- It provides a detailed analysis of the proposed design's area and throughput. All proposed architectures are applications for low resource application and low power consumption.
- Verilog HDL is used to model the hardware architectures and implementation is done in a variety of FPGA families. The comparison was made with other cryptographic algorithms, and the metrics utilised is area and speed.

1.3 Structure of the paper

Five sections are present in this paper, the second of which describes the algorithm of SIMECK lightweight block ciphers. Section 3 explained the proposed architectures and Section 4 presents the efficient hardware implementation and comparison against the other lightweight block ciphers. Section 5 is all about the conclusion part.

2 Theoretical background

Block ciphers may be created using Feistel ciphers, which are symmetric networks. The structure provides the advantages of similar encryption and decryption operations with a minor change in the key scheduling mechanism. The architecture or size of the code needed for implementation is reduced as an outcome of the common structure. Feistel structures are composed of a round function that is iterative in nature. Each cycle of plain text processing in the encryption process involves substitution and permutation. In a balanced Feistel network, the input block is split into two halves, with the right side of the block generally staying unchanged (Lara-Nino et al., 2017). The actions on the left side are dependent on the right side of the block and encryption key.

In typical Feistel networks, only a fraction of the input key is utilised in operations, while the entire key is used to produce sub-keys. The left and right sections of the data are exchanged in each round, and in the final round, the integrate parts of the left and right blocks serve as the cipher text for a certain input plain text. In the decryption process, the encrypted text is utilised as input to the Feistel network, which is quite identical to the encryption process. The distinction in the decryption technique is demonstrated by the sub keys, that are used in reverse order. Unlike SPNs, Feistel networks do not require the round function to be invertible. The structures can be balanced or unbalanced, having equal or uneven left and right textual parts.

SIMON and SPECK ciphers are a family of lightweight block ciphers, developed to meet the requirements of future IoT devices in a resource-constrained environment (Lara-Nino et al., 2017). Both the ciphers offer platform flexibility and perform very well with hardware and software devices however, SIMON cipher is specially optimised for hardware device performance and SPECK cipher is optimised for software device performance. SIMEK cipher is a new family of lightweight block ciphers that combines the finest design principles from SIMON and SPECK to generate more efficient and compact block ciphers that are perfect for resource-constrained devices (RCD) (Abed et al., 2019), Different block and key sizes are supported by these ciphers ranging from 32 to 128 and 64 to 256 bits which are eventually used for the hardware implementation of cryptographic applications. NSA developed these cryptographic block ciphers to offer security and flexibility in lightweight cryptographic applications (Ying et al., 2020).

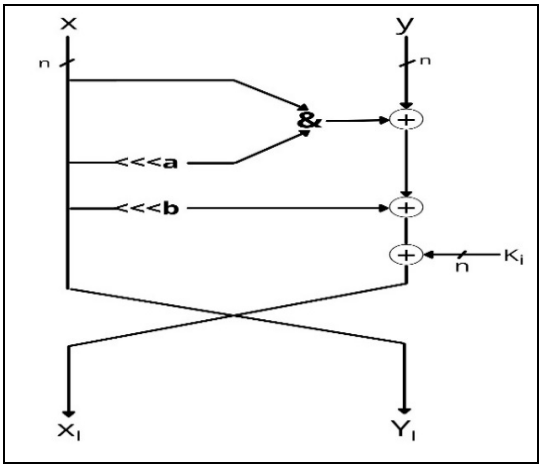
2.1 SIMECK cipher

SIMECK is a Feistel framework-based cipher, which was originally introduced at CHES in 2015. SIMECK provides a variety of input plaintext (from 32 to 128) represented by 2*n* in Table 1. SIMECK cipher intended to have a tiny hardware footprint, while yet being compatible in software implementations (Abed et al., 2019). Figure 1 shows the SIMECK one round Feistel structure.

Table 1 Parameters of SIMECK cipher

Block size 2n	*Key size mn*	*Word size n*	*Keywords m*	*Number of rounds*
32	64	16	4	32
48	(72) (96)	24	(3) (4)	(36) (36)
64	(96) (128)	32	(3) (4)	(42) (44)
96	(96) (144)	48	(2) (3)	(52) (54)
128	(128) (192) (256)	64	(2) (3) (4)	(68) (69) (72)

Figure 1 SIMECK one round Feistel structure



2.1.1 Round function

The Pseudo-code for the implementation of this cipher is detailed below.

Encryption routine of SIMECK cipher for 64/128 configuration:

Input: xp (64), kl (128)

Output: yp (64)

Process to encrypt a block of data

{

Divide xp into Xp and Yp

//Left and right plain text

Divide kl into kp1, kp2, kp3, kp4.

for (i = 1) to total number of rounds (i.e., 44)

{

(Xpi, Ypi) ← (Ypi XOR (Xpi AND left shift (Xpi,5))

Divide kl into kp1, kp2, kp3, kp4.

}

for (i = 1) to total number of rounds (i.e., 44)

{

(Xpi, Ypi) ← (Ypi XOR (Xpi AND left shift

(Xpi,5))XOR (left shift (Xpi,1) XOR kpi, Xpi))

}

yp← Xpi OR Ypi

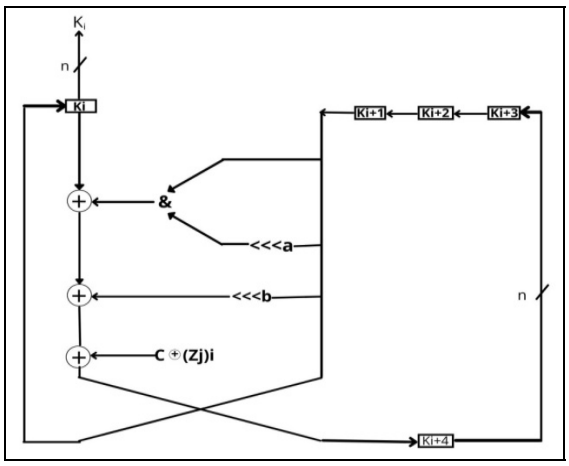
}

The round function of the SIMECK cipher is based on Feistel structure that includes circular left shift along with XOR and AND operation (Encarnacion and Gerardo, 2020). Input plaintext is divided into X and Y of size n-bits. Round key Ki is applied in each round for further operation. It is similar to the round function of SIMON. Suppose X and Y refer to the input data bit with upper and lower word of size n-bit and k_i is the round key of ith round, here the number of rounds is from 0 to r–1. $X <<< I$ stand for a left shift operation of X by I-bits, XOR operation is represented by ⊕ which is bit-wise and and stands for AND operation of bit-wise nature. In this algorithm, we convert a $2n$-bit plaintext $[2n–1:0] = Xo \parallel Yo$ into a $2n$-bit ciphertext $[2n–1:0] = Xr \parallel Yr$. The values of and b in this case, are 8 and 2, respectively. Each round is provided the key Ki, which is generated using key scheduling. In this paper, we have considered three different block sizes that are 32, 48, and 64. Each block size consists a different number of rounds, according to that the keys will be generated.

2.1.2 Key schedule

The Key Schedule is meant to create a key in each round. Key scheduling of simeck cipher is shows in Figure 2. Key schedule techniques focused on operations like shift right, XORing with a constant, which is generally very simple and lighter than round function (Gulcan et al., 2015). Key scheduling takes a master key K and divide it into 4 parts which are K_0, K_1, K_2 and K_3 for the first round. Subkeys for each round are generated from these master keys. Constants C and $(Z_j)i$ which are predefined, used for the key generation. To remove sliding characteristics and symmetry in circular shifting for various keys used for round operation, the SIMECK cipher key scheduling utilises a succession of single bit rounding constants $Z_j[i]$ with i from 0 to 4 and j from 0 to r–1. It also describes 5 sequences: Z_4, Z_3, Z_2, Z_1, and Z_0, which give cryptographic distinction between various SIMECK cipher versions with a similar block size (Ying et al., 2020). The values of p and q are 3 and 1, respectively, here in this case. Furthermore, the key scheme employs constant $C = 2^n – 4$ of n bits (Ying et al., 2020).

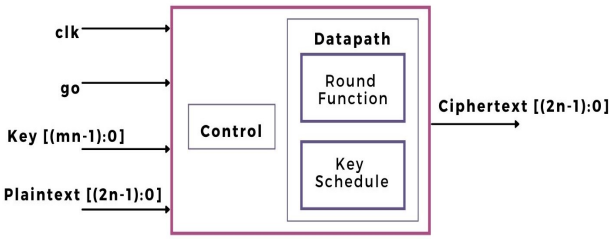
Figure 2 SIMECK cipher key schedule



3 SIMECK cipher proposed architecture

The proposed hardware architectures for implementing SIMECK ciphers are presented in this part. Figure 3 presents the input-output interface between the outside environment and the cipher. Flexible structures are developed that handle different block and key sizes. These three sizes (32/64, 48/96, and 64/128) are intended for use in a wide range of embedded systems, including RFID systems. Figure 6 present the Serial datapath for the key scheduling of Simeck cipher. It depicted the interaction between cryptography and the outer world in Figure 7. In this figure, the cipher is functioning and controlled by the outside signal go in our proposed architecture. We have two options here: load phase and run phase. When the go signal is low, initial data is loaded from the input key and plaintext. When the go signal goes high, the cipher enters into the run phase, and the ciphertext is obtained by the conclusion of the run phase.

Figure 3 The input-output interface between the outside environment and the cipher (see online version for colours)



The SPECK and SIMON are combined in the SIMECK cipher. SIMECK is a novel cipher, especially used for RCDs (Abed et al., 2019). After a shift operation, registers are used to save the values. SIMECK cipher has some similarities with SIMON and SPECK ciphers. Its round function is similar to the SIMON round and the key scheduling part is similar to SPECK as it reuses the round function. The proposed architecture is depicted in Figures 4

and 5 for 64/128 block size and key size. Figure 4 presents the Top module block diagram of SIMECK cipher. Serial datapath of round function is presented in Figure 5. In this serial datapath all the multiplexers and registers are of 1-bit width which serves the purpose to save area. To perform cyclic shift operation, two more multiplexers M_1 and M_5 are used here. M_1 MUX is used to shift left by 1 bit and M_5 MUX is used to shift left by 5 bits. When serial counter value equals to 0, M_1 selects q_{n-1} else p_{n-1} in case of value greater than 0. Similarly, M_5 selects q_{n-1} if serial counter value is less than or equal to 4 else p_{n-1}.

Figure 4 Top module block diagram of SIMECK cipher

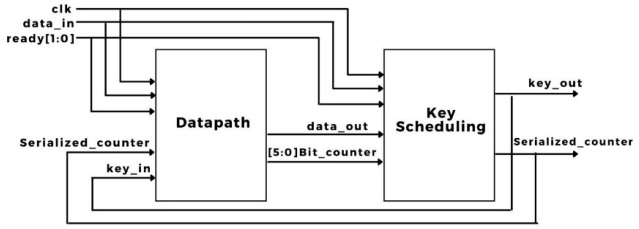


Figure 5 Serial datapath for the round function (see online version for colours)

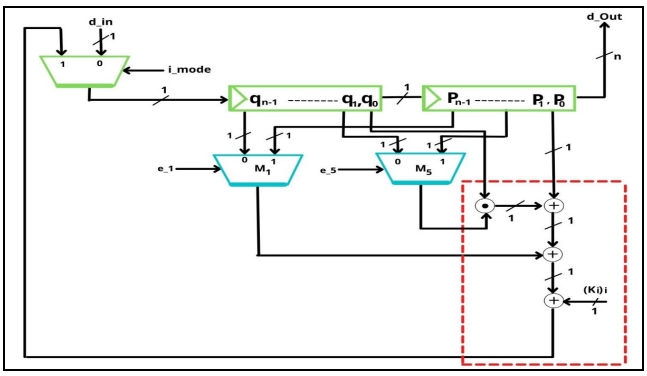
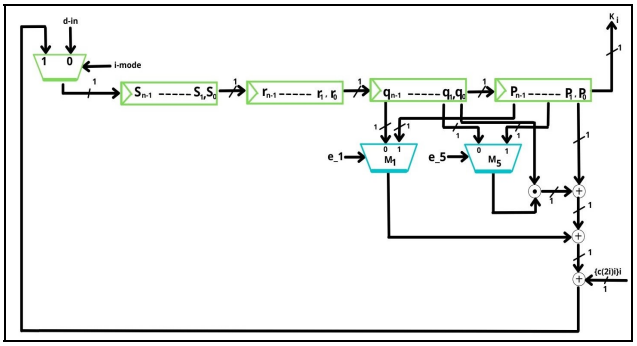


Figure 6 Serial datapath for the key scheduling (see online version for colours)



4 Hardware implementation and comparison

The FPGA platform is employed for hardware implementation in this case. FPGA is a unanimous choice for implementing the design in hardware, which is cost-effective and requires less time (Piyush et al., 2021). This paper presents a hardware implementation of SIMECK cipher with size 64/128 on different families of FPGA platform. Every implementation has different parameters that can be calculated in this platform, and FPGA is reconfigurable. Every implementation has its own set of parameters that must be calculated. Equations (1)–(3) represent the Throughput, Throughput per area (TPA), and Energy per bit respectively. Max Frequency denotes the highest frequency, whereas Cycle denotes the length of time.

$$Throughput = \frac{Max\,Frequency \times Block\,size}{Cycle} \tag{1}$$

$$Throughput\ per\ area\left(TPA\right) = \frac{Throughput}{Area\ /\ Slices} \tag{2}$$

$$Energy\ per\ bit = \frac{Power \times Cycle}{Max\,Frquency \times Block\,size} \tag{3}$$

The maximum frequency value, the design cycle, and the block sizes utilised in the design can all be used to calculate throughput in any hardware design. Verilog HDL is used for implementing the proposed pipelined architecture. Simulation of the cipher was done on Xilinx ISE. The interactive graphical application XPower Analyser is used for power analysis purposes, which determines the static and dynamic power consumption on different platforms. Other integrated measures for hardware performance factors like area, frequency, and power, are determined for comparing with other ciphers (Piyush et al., 2021). Table 2 represents the implementation of proposed architecture on Different FPGA Platforms for SIMECK ciphers. A comparison of different ciphers on various platforms has been shown in Table 3. The proposed area optimised architecture of SIMECK cipher was compared with other different ciphers in terms of area, throughput, and frequency. Sparton-3, Sparton-6, Virtex-4, Virtex-5, Virtex-6, and Virtex-7 platforms are used for the implementation and comparison purposes with various other ciphers.

Table 2 FPGA implementation of proposed architecture on different platforms

Performance Matrix	Sparton-3	Sparton-6	Virtex-4	Virtex-5	Virtex-6	Virtex-7
LUT	262	296	226	214	204	209
FFs	204	206	247	200	202	187
Slices	139	109	113	67	70	61
Frequency(MHz)	194.701	216.903	349.318	459.923	468.469	496.429
Throughput(Mbps)	276.908	308.484	496.807	654.110	666.267	706.032
Throughput/area	1.992	2.830	4.396	9.762	9.518	11.574
Energy(uJ)	1.566	0.846	1.962	2.488	2.349	0.774
Energy per bit (nJ/bit)	0.024	0.013	0.030	0.038	0.036	0.012
Static Power (mW)	0.029	0.071	0.085	0.035	0.072	0.098
Dynamic Power (mW)	0.143	0.117	0.351	0.518	0.450	0.250
Total Power (mW)	0.172	0.198	0.436	0.553	0.522	0.348

Table 3 Resource usage comparison of proposed architecture on different devices

Block Ciphers	Block Size	Key Size	Device	Area (Resources)			Speed	
				FFs	LUTs	Slices	Max.freq. (MHz)	Throughput (Mbps)
Shadow [3]	64	128	Virtex-5	227	156	199	-	743.038
Present [2	64	128	Sparton-3	264	201	151	194.63	91.59
Present [2]	64	128	Sparton-6	220	201	61	210.66	99.13
Present [2]	64	128	Virtex-5	239	201	73	431.78	203.19
Present [2]	64	128	Virtex-4	265	201	152	364.56	171.56
AES [4]	64	128	Virtex-5	258	286	113	113.25	90.60
Clefia [4]	64	128	Virtex-5	467	329	155	93.96	44.22
Present [4]	32	128	Virtex-5	237	203	70	245.76	53.32
Simeck[TW]	64	128	Sparton-3	262	204	139	194.70	132.56
Simeck[TW]	64	128	Sparton-6	296	296	109	216.90	147.68
Simeck[TW]	64	128	Virtex-4	326	247	113	349.32	237.83
Simeck[TW]	64	128	Virtex-5	214	200	67	459.92	313.14
Simeck[TW]	64	128	Virtex-6	204	202	70	468.47	318.95
Simeck[TW]	64	128	Virtex-7	209	187	61	496.43	351.61

TW: This work.

Figure 7 represents the resource usage (LUTs, FFs and Slices) on different FPGA platforms. SIMECK cipher requires only 61 slices on virtex-7 FPGA platforms which indicates an extremely area efficient architecture. Figure 8 is a graphical representation of power consumption. Figure 9 gives the throughput in different FPGA families which follow equation (1).

Figure 7 Resource usage comparison of proposed architecture on different (see online version for colours)

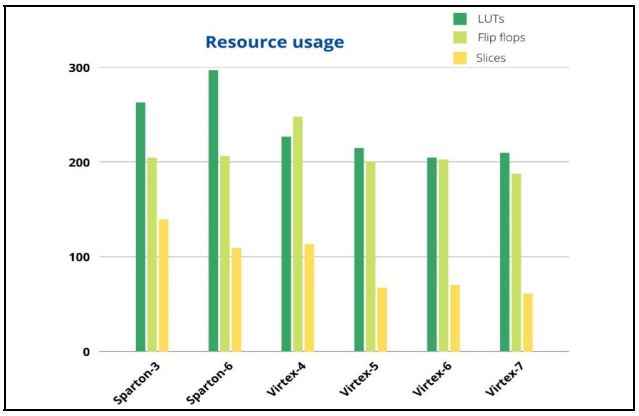


Figure 8 Power consumption comparison of proposed architecture on different (see online version for colours)

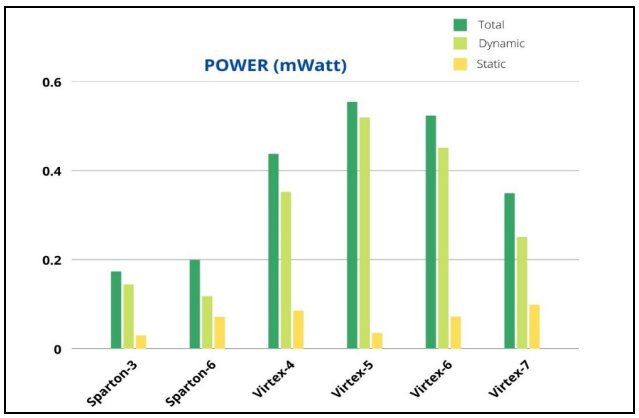


Figure 9 Throughput comparison of proposed architecture on different devices (see online version for colours)

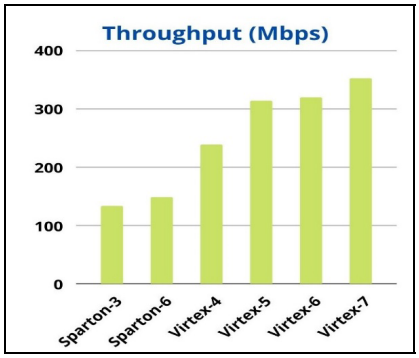
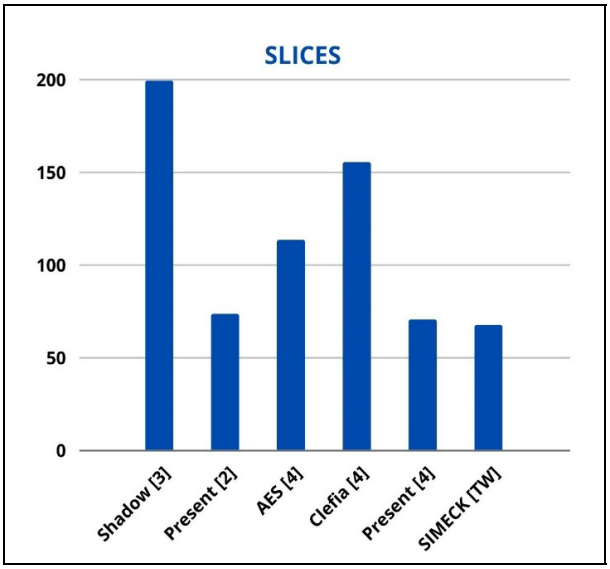


Figure 10 is all about Area (slices) Comparison for the proposed architectures of SIMECK 64/128 with different lightweight block ciphers.

Figure 10 Slices comparison with different lightweight block ciphers (see online version for colours)



5 Result

Table 2 shows the FPGA Implementation of Proposed Architecture on Different Platforms, that includes the area, throughput and power analysis. Result obtained on Virtex-7 Platform is more optimised as compare to the other FPGA platform. Table 3 shows the Resource usage Comparison of proposed architecture on different devices. A comparison is done between different ciphers including Shadow, Present, AES, Clefia and SIMECK. The SIMECK cipher's serial architecture speeds up the encryption process as compare to other ciphers, as illustrated in Table 3. It delivers a high throughput of 706.032 Mbps for the Virtex-7 family and 666.267 Mbps for the Virtex-6 family. Furthermore, the proposed architecture has occupied 61 slices on Virtex-7 platform which is lesser than other ciphers. The proposed design yields ideal results for maintaining a trade-off between performance and resource usage.

6 Conclusion

SIMECK is a family of lightweight block cipher, specially designed for resource-constraint devices. It is suitable for both hardware and software applications.

This paper has proposed the area optimised architecture of SIMECK cipher. It is a family of lightweight block ciphers which is dedicated to software and hardware applications, especially for RCD and RFID applications. In this paper, serial architecture is implemented that improves the operating frequency. Hardware implementation is done on different FPGA and ASIC platforms. In comparison to existing ciphers, parameter metrics such as slices, LUTs, throughput, etc. have shown remarkable improvements. The area is represented in terms of slices for all the ciphers. There is a trade-off between throughput and the area. The maximum operating frequency obtained is 496.429 MHz and the highest throughput value is 706.032 Mbps along with 61 slices. FPGA Virtex-7 platform utilises 209 LUTS, 187 FFS and 61 slices, along with maximum frequency of 496.429 MHz and total power consumption of 0.348 mWatt. On Sparton-6 platform it utilises 296 LUTs, 206 FFS, 109 slices with 216.903 MHz maximum frequency and 308.484 Mbps throughput.

References

Abed, S., Jaffal, R., Mohd, B.J. and Alshayeji, M. (2019) 'FPGA modeling and optimization of a SIMON lightweight block cipher' MDPI', *Sensors*, Vol. 19, pp.1420–8220, doi: 10.3390/s19040913.

Beaulieu, R., Shors, D., Smith, J., Treatman-clark, S., Weeks, B. and Wingers, L. (2015) 'The SIMON and SPECK lightweight block ciphers', *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp.1–6, doi: 10.1145/2744769.2747946.

Encarnacion, P.C. and Gerardo, B.D. (2020) 'Performance analysis on enhanced round function of SIMECK block cipher', *12th International Conference on Communication Software and Networks*, Chongqing, China, Vol. 20, pp.978–1–7281, doi: 10.1109/ICCSN49894.2020.9139059.

Gulcan, E., Aysu, A. and Schaumont, P. (2015) 'A flexible and compact hardware architecture for the SIMON block cipher', *International Workshop on Lightweight Cryptography for Security and Privacy*, *Third International Workshop, LightSec 2014*, Springer, Istanbul, Turkey, pp.34–50, doi: 10.1007/978-3-319-16363-5.

Hanley, N. and O'Neill, M. (2012) 'Hardware comparison of the ISO/IEC 29192-2 block ciphers', *IEEE Computer Society Annual Symposium on VLSI*, Amherst, MA, USA, pp.19–21, doi: 10.1109/ISVLSI.2012.25.

Lara-Nino, C.A., Diaz-Perez, A. and Morales-Sandoval, M. (2017) 'Lightweight hardware architectures for the present cipher in FPGA', *IEEE Transactions on Circuits and Systems–I*, Vol. 64, No. 9, pp.2544–2555, doi: 10.1109/TCSI.2017.2686783.

Maene, P. and Verbauwhede, V. (2015) 'Single-cycle implementations of block ciphers', *Proc. 4th International Workshop on Lightweight Cryptography for Security and Privacy*, Vol. 9542, pp.131–147, doi: 10.1007/978-3-319-29078-2_8.

Mishra, Z., Nath, P.K. and Acharya, B. (2020) 'High throughput unified architecture of LEA algorithm for image encryption', *Microprocessors and Microsystems*, Vol. 78, doi: 10.1016/j.micpro.2020.103214.

Patro, K.A.K. and Acharya, B. (2019) 'A novel multi-dimensional multiple image encryption technique', *Multimedia Tools and Applications* (*Springer*), Vol. 79, No. 19, pp.12959–12994, doi: org/10.1007/s11042-019-08470-8.

Patro, K.A.K., Acharya, B. and Nath, V. (2019) 'Various dimensional colour image encryption based on non-overlapping block-level diffusion operation', *Microsystem Technologies* (*Springer*), Vol. 26, pp.1437–1448, doi: org/10.1007/s00542-019-04676-w.

Patro, K.A.K. and Acharya, B. (2021) 'An efficient dual-layer cross-coupled chaotic map security-based multi-image encryption system', *Nonlinear Dynamics*, pp.2759–2805, doi: 10.1007/s11071-021-06409-z. Patro, K.A.K. and Acharya, B (2021) An efficient two-level image encryption system using chaotic maps', *International Journal of Information and Computer Security*, Online ISSN: 1744-1773, Print ISSN: 1744-1765.

Patro, K.A.K., Soni, A., Netam, P.K. and Acharya, B. (2020) 'Multiple grayscale image encryption using cross-coupled chaotic maps', *Journal of Information Security and Applications (Elsevier)*, Vol. 52, pp.102470, doi: org/10.1016/j. jisa.2020.102470.

Ramu, G., Mishra, Z., Singh, P. and Acharya, B. (2020) 'Performance optimised architectures of piccolo block cipher for low resource IoT applications', *International Journal of High Performance Systems Architecture International (IJHPSA)*, Vol. 9, No. 1, pp.49–57, doi: 10.1504/Ijhpsa.2020.107175.

Sheikpour, S., Hassani, M. and Mahani, S. (2019) 'High-throughput configurable SIMON architecture for flexible security', *Microelectronics Journal*, Vol. 113, pp.105–085, doi: 10.1016/j. mejo.2021.105085.

Yang, G., Zhu, B., Suder, V., Aagaard, M.D. and Gong, G. (2015) 'The SIMECK family of lightweight block ciphers', in Güneysu, T. and Handschuh, H. (Eds.): *Cryptographic Hardware and Embedded Systems–CHES 2015*, Vol. 9293, pp.307–329, doi: 10.1007/978-3-662-48324-4_16.

Bibliography

Guo, Y., Li, L. and Liu, B. (2020) 'Shadow: A lightweight block cipher for IoT nodes', *IEEE Internet of Things Journal*, Vol. 8, No. 16, pp.13014–13023, doi: 10.1109/JIOT.2021.3064203.

Mishra, Z. and Acharya, B. (2020) 'High throughput and low area architectures of secure IoT algorithm for medical image encryption', *Journal of Information Security and Applications*, Vol. 53, doi: 10.1016/j. jisa.2020.102533.

Mishra, Z. and Acharya, B. (2021a) 'High throughput novel architectures of TEA family for high speed IoT and RFID applications', *Journal of Information Security and Applications (Elsevier)*, Vol. 61, doi: 10.1016/j. jisa.2021.102906.

Mishra, Z. and Acharya, B. (2021b) 'Efficient hardware implementation of TEA, XTEA and XXTEA lightweight ciphers for low resource IoT applications', *International Journal of High Performance Systems Architecture*, Vol. 10, No. 2, pp.80–88, doi: 10.1504/IJHPSA.2021.119150.

Modi, P., Singh, P. and Acharya, B. (2021) 'Effective hardware architectures for LED and PRESENT ciphers for resource-constrained applications', *International Journal of High Performance Systems Architecture*, Vol. 10, No. 2, pp.89–104, doi: 10.1504/IJHPSA.2021.119151.

Shrivastava, N., Singh, P. and Acharya, B. (2020) 'Efficient hardware implementations of QTL cipher for RFID applications', *International Journal of High Performance Systems Architecture*, Vol. 9, No. 1, pp.1–10, doi: 10.1504/IJHPSA.2020.107173.

Singh, P., Acharya, B. and Chaurasiya, R.K. (2019) 'A comparative survey on lightweight block ciphers for resource constrained applications', *International Journal of High Performance Systems Architecture*, Vol. 8, No. 4, pp.250–270, doi: 10.1504/IJHPSA.2019.104953.