Optimized architecture for SNOW 3G

N. B. Hulle[1], Prathiba B[2], Sarika R. Khope[3], K. Anuradha[4], Yogini Borole[5], D. Kotambkar[6]
[1,2,3,5]Department of Electronics and Telecommunication, G H Raisoni Institute of Engineering and Technology, India
[4]Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, India
[6]Department of Electronics Design Technology, Shri Ramdeobaba College of Engineering and Management, India

Article Info

Article history:

Received Apr 3, 2020
Revised Jun 19, 2020
Accepted Aug 11, 2020

Keywords:

Cryptography
FPGA
SNOW 3G
Stream cipher
VHDL
Wireless network security

ABSTRACT

SNOW 3G is a synchronous, word-oriented stream cipher used by the 3GPP standards as a confidentiality and integrity algorithms. It is used as first set in long term evolution (LTE) and as a second set in universal mobile telecommunications system (UMTS) networks. The cipher uses 128-bit key and 128 bit IV to produce 32-bit ciphertext. The paper presents two techniques for performance enhancement. The first technique uses novel CLA architecture to minimize the propagation delay of the 2^{32} modulo adders. The second technique uses novel architecture for S-box to minimize the chip area. The presented work uses VHDL language for coding. The same is implemented on the FPGA device Virtex xc5vfx100e manufactured by Xilinx. The presented architecture achieved a maximum frequency of 254.9 MHz and throughput of 7.2235 Gbps.

Corresponding Author:

N. B. Hulle,
Department of Electronics and Telecommunication,
G H Raisoni Institute of Engineering and Technology,
Pune, India.
Email: nagnath.hulle@raisoni.net

1. INTRODUCTION

Security of the records is important in the systems where personal and financial matters are involved. Hiding of information from unauthorized users becomes essential in such systems and services. Cryptography is one of the widely used techniques for securing information from eavesdroppers. Considering the need to secure information many researchers are working in the area of information security. To maintain advanced network security, the concern network architecture must change from traditional security to advanced security. The same may be achieved by sinking holes in the security wall.

Cryptography algorithms and their associated key are more secure when it is implemented on a hardware platform [1]. Side-channel attacks and fault attacks may exist. However, developed algorithms must be fast enough to support autonomous protocols. These protocols use different encryption algorithms for a different session. Many recent autonomous protocols like secure sockets layer (SSL) and internet protocol security (IPsec) use different ciphers for different sessions.

Hardware implementation of the cryptographic algorithm on FPGA devices is attractive solutions because FPGAs are reconfigurable [2-8]. This property provides flexibility for dynamic system development and capable of implementing a wide range of functions/architectures/algorithms. It seems to be significant to emphasize FPGA based implementations of cryptographic algorithms, especially high throughput architectures [9]. SNOW 3G algorithm is the core of the 3rd generation partnership project (3GPP) algorithms UEA2 and UIA2. The 3GPP is a joint attempt between telecommunication associations (TG) to make globally applicable specifications for long term evolution (LTE) mobile phone systems [10, 11].

The presented work uses optimized architecture for SNOW 3G stream cipher. This architecture requires only 2K bytes of memory for implementation of S-box in place of 8K bytes of memory required for the existing SNOW 3G architectures [10-16]. The paper is arranged in the following sections. Section-2 provides initial versions of SNOW stream cipher [17]. Section-3 provides the working and design parameters of the SNOW 3G algorithm. Section-4 lists existing work related to presented techniques. Section-5 presents optimized SNOW 3G architecture and its analysis. The results are discussed in Section-6 and Section-7 concludes the presented work.

2. INITIAL VERSIONS OF SNOW

The researcher Patrik Ekdahl et. al. proposed a stream cipher SNOW (SNOW 1.0) in the year 2000 [18], after two years Hawkes et. al. described a new attack known as a guess-and-determine attack [19] on SNOW 1.0. SNOW 1.0 has two limitations. The first limitation was finite state machine (FSM) has a single input, which allows the attacker to disturb the working procedure in FSM and second SNOW 1.0 was little unlucky in choosing feedback polynomial. This allows creating bitwise correspondence in FSM and which is the base of distinguishing attack.

Patrik Ekdahl et. al. the proposed a new version of SNOW cipher as SNOW 2.0 [20] with modifications in SNOW 1.0 [18]. They provided two inputs to FSM and modified feedback polynomial in the new version SNOW 2.0. The two inputs to FSM in SNOW 2.0 makes the guess-and-determine attack more difficult because FSM update registers R1 and R2 do not depend only on FSM output. The polynomial selection in SNOW 1.0 was made to speed up the multiplication by left shift operation in LFSR. This allows the result of multiplication to appear at many places as a bit shifted version of the original word. Such a selection of polynomial provides a base for correlation attack in the initial version [18]. SNOW 2.0 provides better distribution of the bits in feedback function by defining field (F_2^32) as an extension over the field (F_2^8). Each multiplication was implemented as shifting the content by one byte and unconditional XOR with 256 possible patterns. So SNOW 2.0 [20] is strong against correlation attack as compared to Snow 1.0 [18]. During the evaluation of The European Telecommunications Standards Institute (ETSI)/Security Algorithms Group of Experts (SAGE), the SNOW 2.0 was further modified to increase its resistance against algebraic attacks and the new design named as SNOW 3G [10].

3. SPECIFICATIONS AND WORKING OF SNOW 3G

SNOW 3G generates a 32-bit ciphertext per clock cycle with the help of a 128-bit key and 128-bit initialization vector (IV) as shown in Figure 1. It consists of the main four modules initial operations, linear feedback shift register (LFSR), finite state machine (FSM), and a feedback path. The initial operations will divide 128 bit Key into four blocks as per equations (01), (02), (03), and (04). Similarly, it also divides 128 bit IV into four blocks as per equations (05), (06), (07), and (08) [10].

$$K3 = k[0] \parallel k[1] \parallel k[2] \parallel \ldots \parallel k[31] \tag{1}$$

$$K2 = k[32] \parallel k[33] \parallel k[34] \parallel \ldots \parallel k[63] \tag{2}$$

$$K1 = k[64] \parallel k[65] \parallel k[66] \parallel \ldots \parallel k[95] \tag{3}$$

$$K0 = k[96] \parallel k[97] \parallel k[98] \parallel \ldots \parallel k[127] \tag{4}$$

$$IV3 = iv[0] \parallel iv[1] \parallel iv[2] \parallel \ldots \parallel iv[31] \tag{5}$$

$$IV2 = iv[32] \parallel iv[33] \parallel iv[34] \parallel \ldots \parallel iv[63] \tag{6}$$

$$IV1 = iv[64] \parallel iv[65] \parallel iv[66] \parallel \ldots \parallel iv[95] \tag{7}$$

$$IV0 = iv[96] \parallel iv[97] \parallel iv[98] \parallel \ldots \parallel iv[127] \tag{8}$$

where $k[0]$, $iv[0]$ are LSB part and $k[127]$, $iv[127]$ are MSB part of the key and IV respectively.

Initial operations are performed on key and iv as per Table 1. The output of the initial operations block is loaded into LFSR before the first clock cycle [10]. The second module LFSR consists of sixteen stages each having parallel 32 bits. Contents of LFSR are shifted from MSB (S15) to LSB (S0) in each clock cycle. S15 receives new value from the feedback path at each clock cycle. Third module FSM consists of three parallel 32-bit update registers R1, R2, R3, two S-Boxes S1, S2 each of 4Kbytes, two 32 bit modulo adders and two 32 bit XOR gates. The final module is the feedback path which consists of functions MULα, DIVα, and many XOR operations. The two functions MULα and DIVα are implemented as lookup tables.

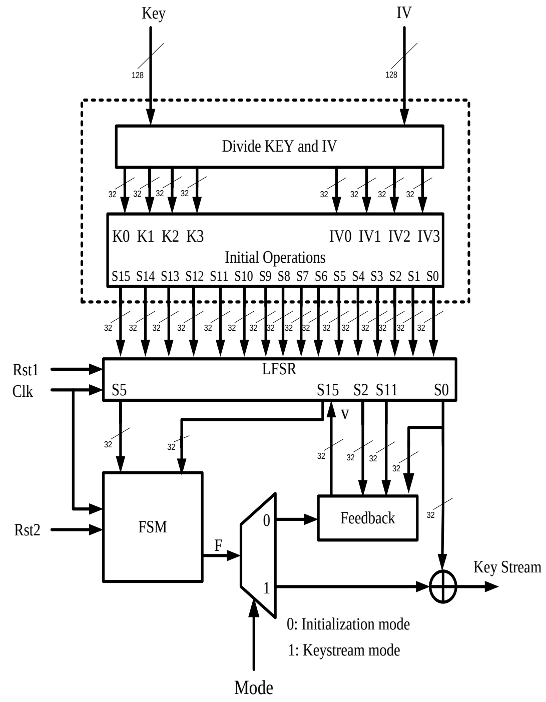


Figure 1. Existing SNOW 3G architectures

Table 1. LFSR initialization calculations

$S_0 = K_0 \oplus 1$	$S_4 = K_0$	$S_8 = K_0 \oplus 1$	$S_{12} = K_0 \oplus IV_1$
$S_1 = K_1 \oplus 1$	$S_5 = K_1$	$S_9 = K_1 \oplus 1 \oplus IV_3$	$S_{13} = K_1$
$S_2 = K_2 \oplus 1$	$S_6 = K_2$	$S_{10} = K_2 \oplus 1 \oplus IV_2$	$S_{14} = K_2$
$S_3 = K_3 \oplus 1$	$S_7 = K_3$	$S_{11} = K_3 \oplus 1$	$S_{15} = K_3 \oplus IV_0$

SNOW 3G works into two modes of operation, initialization mode and keystream mode. At the start of initialization, the model system should reset LFSR and FSM using terminals Rst1 and Rst2 respectively. In the first clock cycle values calculated in the initialization, a mode is loaded into sixteen stages of LFSR but FSM registers should remain in a reset state. In the second clock cycle, Rst2=0 and now LFSR is clocked. At each clock, 32-bit output F of FSM is combined with S0, S2 & S11 in the feedback path by selecting mode 0 from a select line of the multiplexer and applied to S15 as intermediate signal v. The following equation provides the intermediate signal v in the initialization mode [10].

$$v = (S0{,}1 \| S0{,}2 \| S0{,}3 \| 0x00) \oplus MUL_\alpha(S0{,}0) \oplus \mathrm{S2} \oplus (0x00 \| S11{,}0 \| S11{,}1 \| S11{,}2) \oplus DIV_\alpha(S11{,}3) \oplus \mathrm{F} \tag{9}$$

After 32 clock cycles, SNOW 3G enters into keystream mode. Operations in this mode are the same as initialization mode but the only difference is that output F of FSM is not combined in feedback path by making mode = 1 from the multiplexer. The intermediate signal in keystream mode is given by the following equation [10].

$$v = (S0{,}1 \|\ S0{,}2 \| S0{,}3 \| 0x00) \oplus MUL_\alpha(S0{,}0) \oplus \mathrm{S2} \oplus (0x00 \| S11{,}0 \| S11{,}1 \| S11{,}2) \oplus DIV_\alpha(S11{,}3) \tag{10}$$

In keystream mode, FSM is clocked for one clock cycle and its first output is discarded when it is clocked for n clock cycles to encrypt n number of 32-bit words, where n = number of 32-bit data words is to be encrypted [10].

4. RELATED WORK

The study of existing architectures of SNOW 3G evolved two challenges. One minimizing propagation delay of the 2^{32} modulo adders and other is minimizing the chip area of S-boxes. The researcher Kitsos *et al.* [12] realized S-boxes using 8 lookup tables. Each lookup table consumes 1 KB memory, so memory used for S-box realization is 8 KB. Jairaj *et al.* used symmetry of S-box lookup tables to minimize cache requirement in the software implementation of SNOW 3G [21]. Kitsos *et al.* [12] used conventional CLA for modulo adder implementation. The researcher Pai and Chen [22] presented a modified CLA design to minimize the propagation delay. Traboulsi *et al.* [23] implemented SNOW 3G on an embedded platform. The motive of the design was to minimize the memory required for S-box implementation. Researchers used 2 lookup tables in place of 8 lookup tables for implementation of 2 S-boxes. Eight-bit shifting with cache memory is used efficiently to minimize memory requirement.

5. PRESENTED SNOW 3G ARCHITECTURE

Considering the challenges of existing FSM, the proposed implementation uses the following refinements to improve the performance of the SNOW 3G algorithm.

- Use of novel modulo CLA architecture over 2^{32} to minimize propagation delay in FSM, which decides the critical delay of the algorithm
- Use of novel S-Box architecture to minimize chip area

5.1. Novel modulo CLA architecture over 2^{32}

Modulo adders are usually implemented by using ripple-carry adders, but this technique increases the propagation delay of the critical path. The propagation delay of n bit ripple carries adder is (2n+1) gate delays. Modulo adder over 2^{32} implemented by using ripple-carry adders will have delay of (2*32+1 = 65) 65 gates delay, assuming average gate delay of 10 ηs the total delay of one modulo adder will be 65*10 = 650 ηs. FSM consists of two such adders so a total delay of modulo adders for single computation will be 1300 ηs.

The propagation delay of modulo adders can be minimized by using CLA for its implementation. Existing CLAs are realized by using basic gates i.e. AND, XOR, and OR gates, but Pai *et al*. realized CLA by using universal gates i.e. NAND or NOR gates [22]. The same design minimized gate requirement as compared to existing architectures. At the same time, this CLA [22, 24] designs are faster than conventional CLA architectures. Adder architecture [25] developed for LILI-II cipher uses different approach for addiation.

Reduction in propagation delay and chip area is possible in existing architectures [12-16, 22], so the presented research work uses universal gates for CLA implementation and other techniques to minimize the number of gates required. Novel modulo CLA architecture over 2^{32} uses following three architectures in multilevel CLA designs for performance improvement

- 4 bit CLA at LSB (to calculate S0 to S3)
- 4 bit CLA at middle stages (to calculate S4 to S27)
- 4 bit CLA at MSB (to calculate S28 to S31)

Using the above CLA architectures novel architecture for modulo CLA over 2^{32} was designed as shown in Figure 2. Presented modulo adder architecture is an area, propagation delay, and energy-efficient as compared to existing modulo CLA architectures.

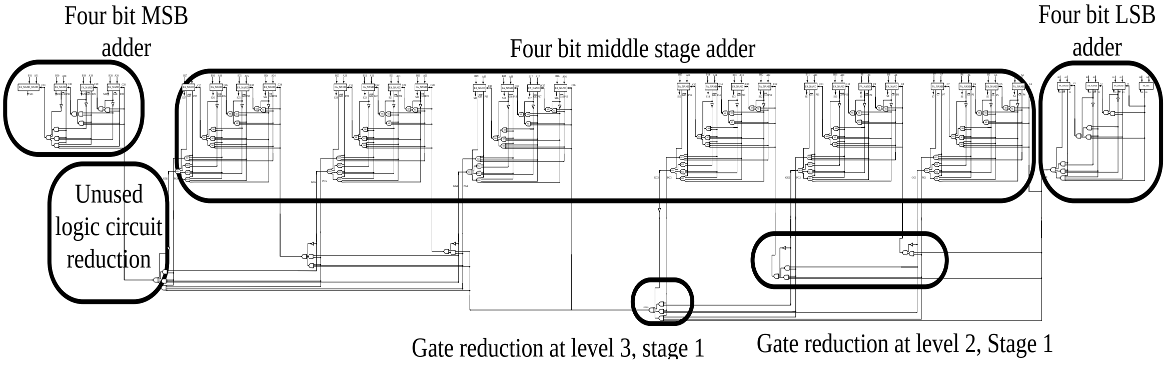

Four bit MSB adder
Four bit middle stage adder
Four bit LSB adder
Unused logic circuit reduction
Gate reduction at level 3, stage 1
Gate reduction at level 2, Stage 1

Figure 2. Novel modulo CLA architecture over 2^{32}

5.2. Novel S-Box architecture

Two S-boxes S1 & S2 are used in SNOW 3G architecture each requires memory of 4 KB. The lookup table of S1 is taken from the Rijndael substitution box and a lookup table of S2 is based on Dickson polynomial over GF-28. As per design specification, each S-box (S1 or S2) is implemented by using 4 lookup tables and each lookup table has 256 values each of 4 bytes. So the implementation of each lookup table requires (256x4 = 1024bytes of memory). Each S-box has 4 lookup tables, so total memory required for the implementation of S1 or S2 is (4x1024 = 4K) 4KB. The total memory needed for the realization of two S-boxes is 8KB. Existing implementation [10-16, 26-29] uses S-box architecture as shown in Figure 3.

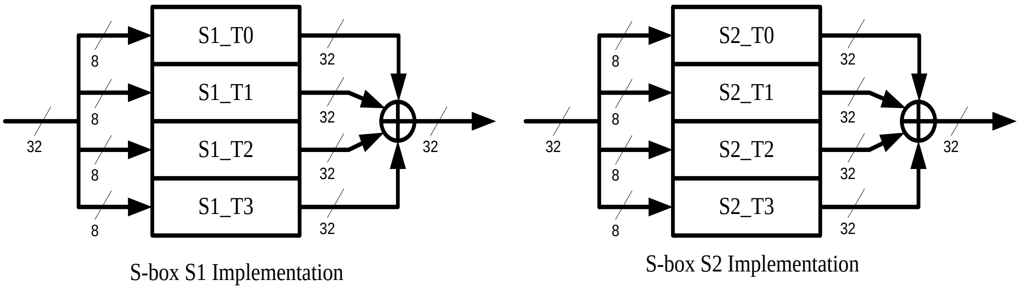


Figure 3. Existing S-boxes architecture

The four lookup tables of S1 i.e. S1_T0 to S1_T3 as shown in Figure 3 has the same content but exist in 8 bit shifted form. Analogous is the case of S-box S2. Presented novel S-box architectures use a single lookup table for implementation of S-box (S1 or S2). Presented research work uses two architectures for S-box implementation. First architecture as shown in Figure 4, consumes fewer resources but useful to low-frequency applications only. Second architecture as shown in Figure 5, consumes fewer resources as compared to existing architectures but required more resources as compared to Novel S-box architecture-1.

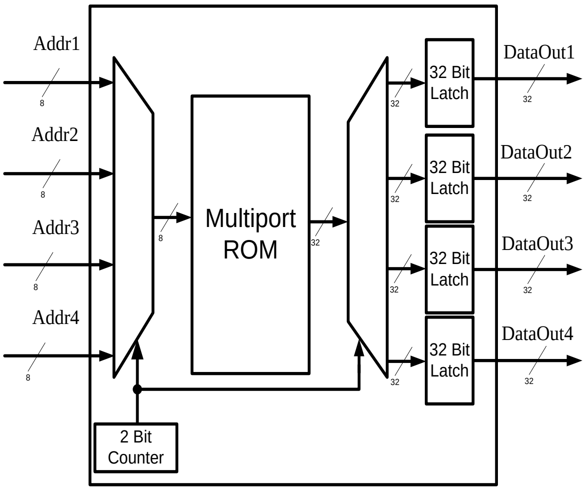


Figure 4. Novel S-Box architecture1

Presented designs require 2 KB of memory for the realization of S-boxes S1 & S2. These designs save 6 KB of memory as compared to existing designs. S-box architecture-1 saves 6 KB memory at the cost of some additional hardware (Single 2-bit counter, two 4 I/p multiplexers, and four 32 bit latches). This architecture is 4 times slower than conventional architectures and useful for low-frequency applications. S-box architecture-2 has the same speed as conventional architectures but uses 4 additional 256:1 multiplexers. The second architecture can be used for low and high-speed applications depending on cost and speed tradeoffs

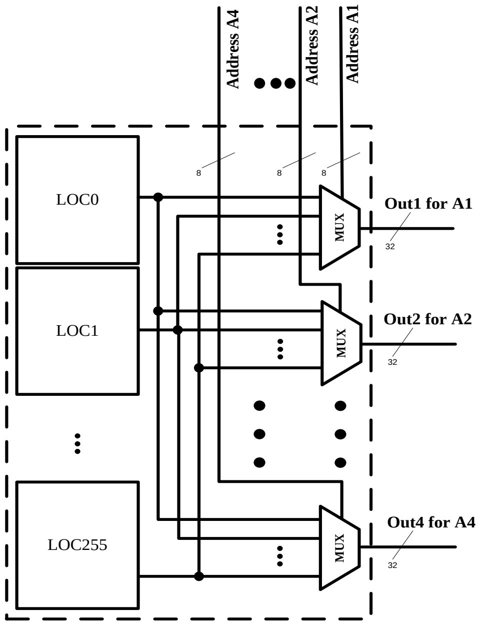


Figure 5. Novel S-Box architecture 2

5.3. Optimized SNOW 3G architecture

Optimized SNOW 3G architecture as shown in Figure 6 is designed using novel modulo CLA architecture and novel S-Box architecture as discussed in the previous section. SNOW 3G architecture designed using S-Box architecture-1 is used for low-frequency applications and needs two clock arrangements. Whereas SNOW 3G architecture designed using S-Box architecture-2 is used for high-frequency applications and needs a single clock. Internal block diagram of optimized SNOW 3G architecture as shown in Figure 7.

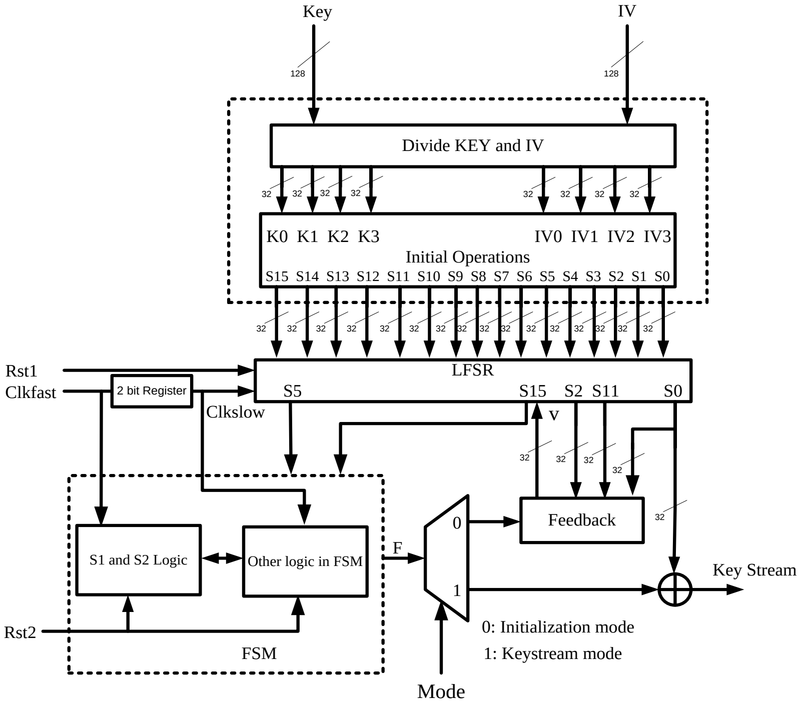


Figure 6. Top module of refined SNOW 3G architecture

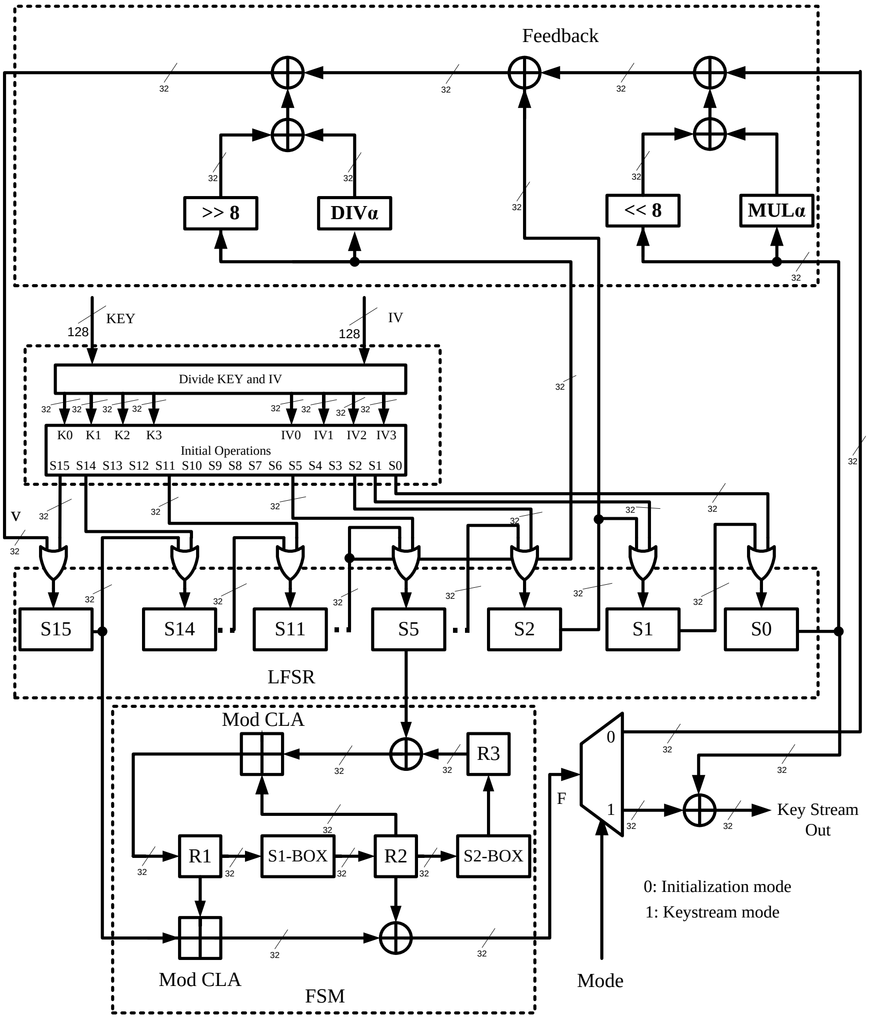


Figure 7. Internal block diagram of optimized SNOW 3G architecture

FSM of SNOW 3G architecture consists of two modulo adders and two S-Boxes. Two modulo adders will decide the speed of the algorithm and two S-boxes will decide hardware utilization of the algorithm. The use of novel modulo CLA over 2^{32} minimizes propagation delay and the use of novel S-box architecture minimizes hardware utilization. These refinements help to improve the performance of the SNOW 3G algorithm in terms of throughput and area.

Optimized SNOW 3G architecture uses VHDL language for coding. The same is implemented on the FPGA device Virtex xc5vfx100e manufactured by Xilinx [30]. The presented architecture achieved a maximum frequency of 254.9 MHz and throughput of 7.2235 Gbps. Table 2 shows particulars about the technology used. Figure 8 and Figure 9 show RTL schematic and output waveform of the presented architecture respectively.

Table 2. Technology used details

SNOW3GNEW Project Status (05/03/2020 - 11:34:41)			
Project File:	SNOW3GOPT.xise	Parser Errors:	No Errors
Module Name:	SNOW3G	Implementation State:	Synthesized
Target Device:	xc5vfx100t-3ff1136	• Errors:	No Errors
Product Version:	ISE 13.2	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

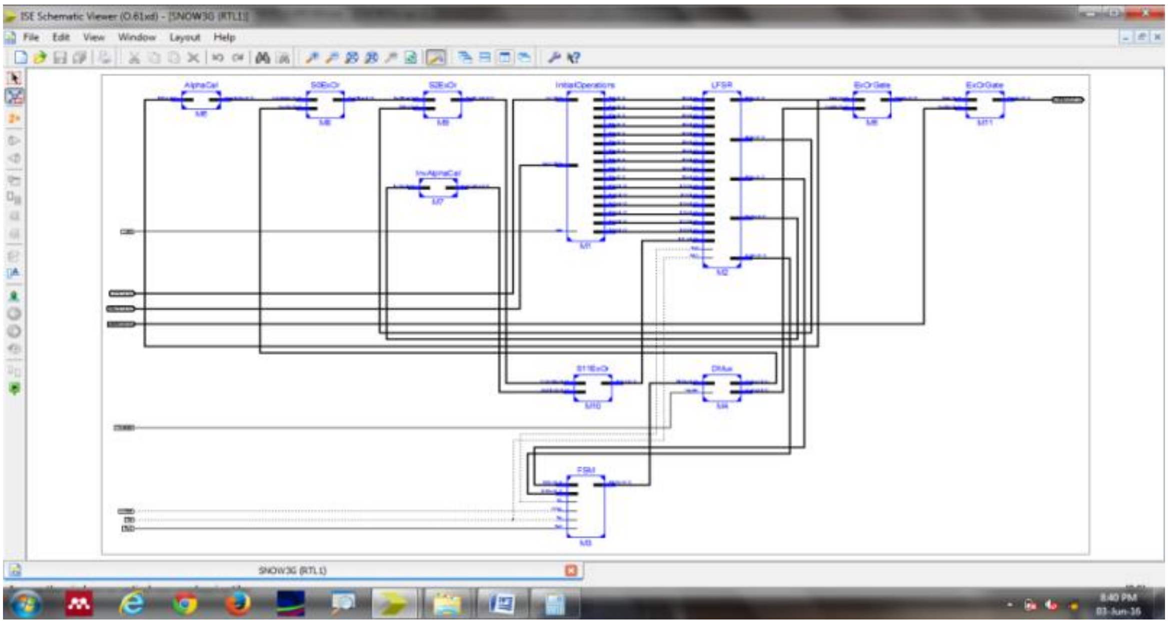
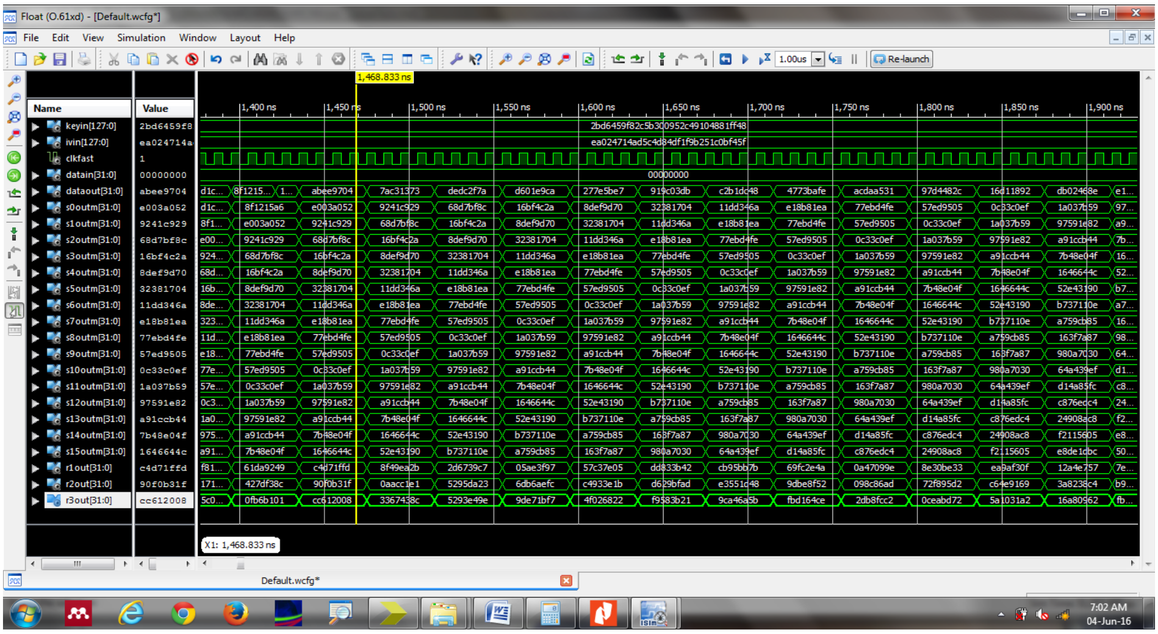


Figure 8. RTL schematic



Name	Value
keyin[127:0]	2bd6459f8
ivin[127:0]	ea024714a
clkfast	1
datain[31:0]	00000000
dataout[31:0]	abee9704
s0outm[31:0]	e003a052
s1outm[31:0]	9241c929
s2outm[31:0]	68d7bf8c
s3outm[31:0]	16bf4c2a
s4outm[31:0]	8def9d70
s5outm[31:0]	32381704
s6outm[31:0]	11dd346a
s7outm[31:0]	e18b81ea
s8outm[31:0]	77ebd4fe
s9outm[31:0]	57ed9505
s10outm[31:0]	0c33c0ef
s11outm[31:0]	1a037b59
s12outm[31:0]	97591e82
s13outm[31:0]	a91ccb44
s14outm[31:0]	7b48e04f
s15outm[31:0]	1646644c
r1out[31:0]	c4d71ffd
r2out[31:0]	90f0b31f
r3out[31:0]	cc612008

Figure 9. Output waveform

6. RESULT AND DISCUSSIONS

The following section discusses the result in terms of area, propagation delay, throughput, and memory utilized for presented SNOW 3G architecture.

6.1. The area

6.1.1. Novel modulo CLA architecture over 2^{32}

Presented novel modulo CLAs are used as modulo adders over 2^{32} in Optimized SNOW 3G architecture. A comparison of device utilization of existing [13, 22] and presented architectures is shown in Figure 10.

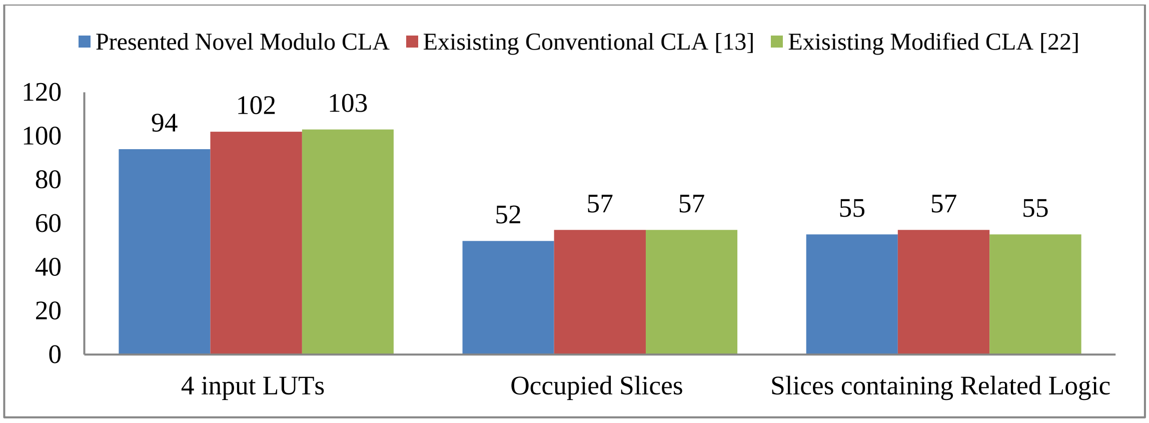


Figure 10. Comparisons of hardware utilization for CLA architectures

6.1.2. Novel S-Box architecture

Optimized SNOW 3G architecture uses Novel S-box architecture to avoid redundancy of lookup tables. Presented Novel S-Box architecture-1 is suitable for low-frequency applications and Novel S-Box architecture-2 is useful for high-frequency applications. The use of these novel architectures minimizes hardware requirement as shown in the Figure 11. The comparison shows that the hardware resources used in the presented architectures are less than existing architectures [12-16]. The reduction in area is possible because S-box is designed using one lookup table in place of four lookup tables.

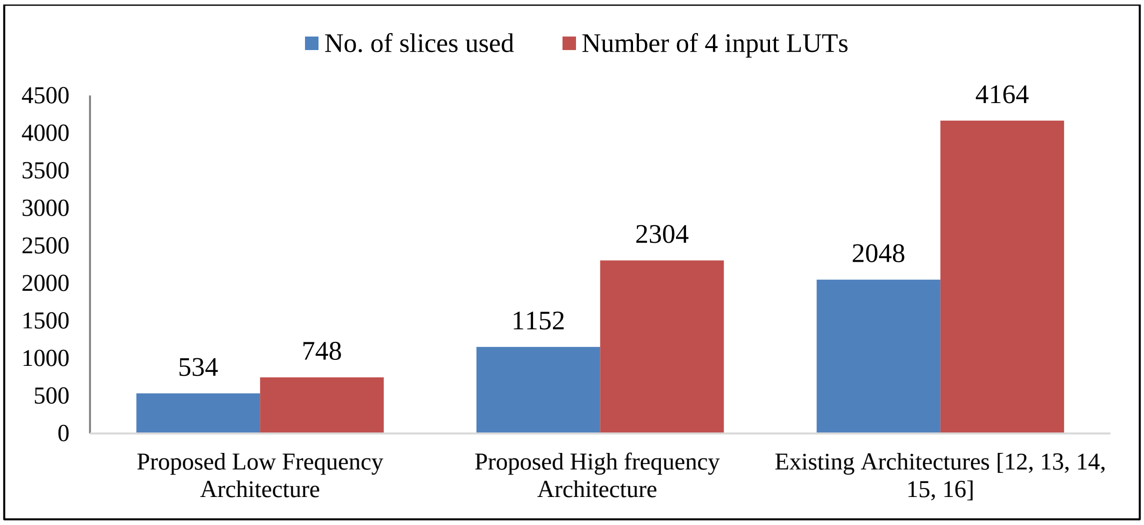


Figure 11. Comparisons of hardware utilization for S-box

6.1.3. Optimized SNOW 3G architecture

Optimized SNOW 3G architecture uses refined modulo CLA over 2^{32} and refined S-box to for performance improvement. Hardware resources used by optimized SNOW 3G architecture are presented in Table 3 and Table 4 shows comparisons of hardware resources used by optimized SNOW 3G and existing architectures [12-16].

The comparison shows that optimized SNOW 3G architecture utilizes minimum resources as compared to architecture presented Kitsos *et al*. [13], Madani and anougast [15] and Madani *et al*. [16]. The architecture presented by Kitsos *et al*. [12] is ASIC, so the comparison is difficult. The architecture presented by Zhang *et al*. [14] uses less hardware as compared to proposed refined architecture because only one mode implemented on hardware.

Table 3. Device utilization summary of optimized SNOW 3G architecture

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
1. Number of Slice Registers	870	64000	1%
2. Number of Slice LUTs	1208	64000	1%
3. Number of fully used LUT-FF pairs	680	1398	48%
4. Number of bonded IOBs	325	640	50%
5. Number of BUFG/BUFGCTRLs	10	32	31%

Table 4. Comparison of hardware resources for different architectures

Sr. No.	Architectures	Hardware resources used
1	Proposed Refined Architecture	870 Slice Registers and 1208 slice LUTs on Virtex 5
2	The architecture proposed by P. Kitsos et al. [12]	ASIC implementation used 25016 equivalent gates
3	The architecture proposed by P. Kitsos et al. [13]	Slices used 3559 on Spartan 3 Family
4	The architecture proposed by L. Zhang et al. [14]	Only one mode implemented on hardware to increase throughput with minimum hardware resources, 356 slices on Virtex 5
5	The architecture proposed by Mahdi Madani and Camel Tanougast [15]	1020 Slice Registers and 889 Slice LUTs on Virtex5
6	The architecture proposed Mahdi Madani, Ilyas Benkhaddra et al. [16]	912 Slice Registers and 1108 Slice LUTs on Virtex 5

6.2. Propagation delay

6.2.1. Novel modulo CLA over 2^{32}

Propagation delay comparison of proposed refined CLA and existing CLA architectures [13, 22] is shown in Figure 12. Propagation delay evaluation shows that delay of presented novel modulo CLA architecture is fewer than existing CLA architectures. The presented CLA architecture will help to improve the throughput of Optimized SNOW 3G architecture.

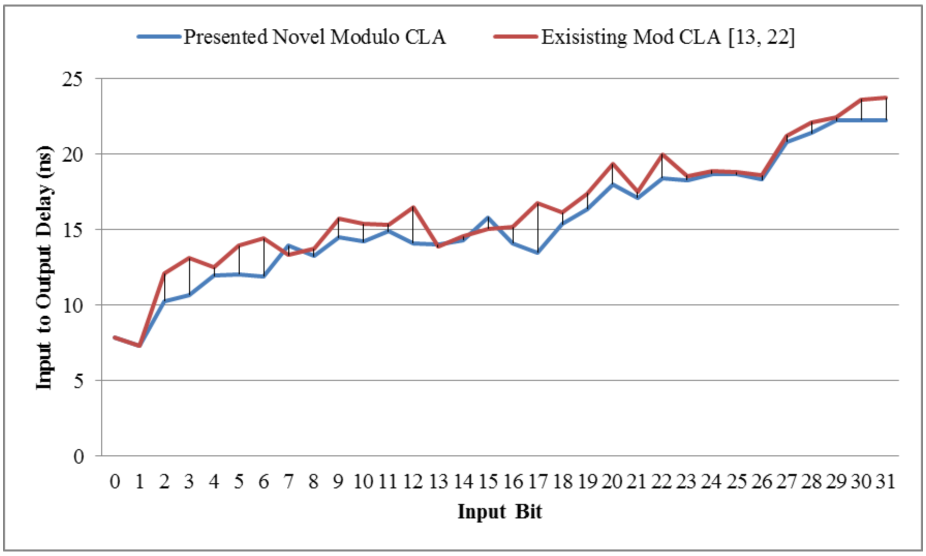


Figure 12. Delay comparisons of novel and existing CLA architectures

6.2.2. Novel S-Box architecture

The combinational path delay comparisons of proposed refined S-box architectures and existing S-box architectures [12-16] are shown in the Table 5. The comparison shows that the propagation delay of proposed low-frequency architecture is more as compared to other architectures, with less hardware. Similarly, the propagation delay of proposed high-frequency architecture is less as compared to other architectures with moderate hardware utilization. The path delay of existing architectures is more as compared to presented architecture 1 but less as compared to architecture 2. The hardware resources used by existing architecture are more as compared to other architectures.

Table 5. Hardware used and propagation delay comparison of S-box implementations

Sr. No.	S- Box architectures	No. of slices used	Number of 4 input LUTs	Propagation delay (ns)
1	Proposed Low-Frequency Architecture	534	748	9.92
2	Proposed High-frequency Architecture	1152	2304	9.12
3	Existing Architecture	2048	4164	9.34

6.3. Throughput and memory

Comparisons of throughput achieved and memory used for S-box realization of optimized SNOW 3G and existing SNOW 3G [12-16] architectures are shown in Figure 13. The comparison shows that throughput of optimized SNOW 3G is higher than architecture presented by Kitsos *et al*. [13], close to architecture presented by Kitsos *et al*. [12], but less than architecture presented by Zhang *et al*. [14], Madani and Tanougast [15] and Madani *et al*. [16]. This may be due to the use of more hardware resources.

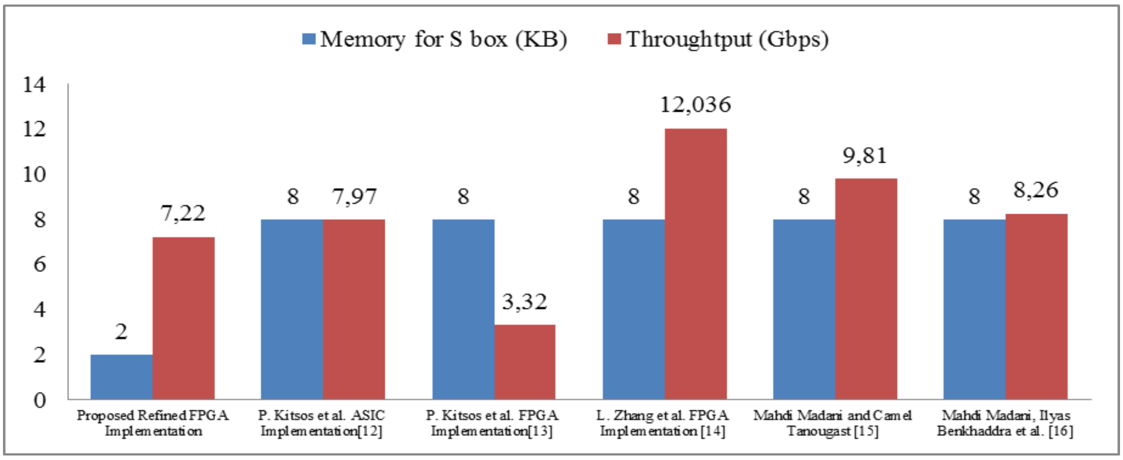


Figure 13. Comparisons throughput and memory used for the realization of S-boxes

7. CONCLUSION

Optimized SNOW 3G architecture is presented in the paper uses novel modulo CLA and novel S-box architecture. The use of novel CLA minimizes hardware required for modulo adders and minimizes propagation delay as compared to existing architectures. The use of novel S-box architecture minimizes 6 K bytes of memory as compared to existing architectures. The presented architecture uses 2K bytes of memory, whereas existing architectures 8 K bytes of memory for the same. The presented SNOW architecture attained throughput of 7.2463 Gbps at a clock frequency of 226.562 MHz. Presented architecture achieves throughput more than architecture and close to ASIC implementation.

The throughput of existing architectures is more than the presented architecture. It may be due to: (1) S-boxes used in these architectures use 8 KB memory for S-box realizations; (2) Architecture uses a software platform that helps to minimize hardware and to increase throughput; (3) Architecture is ASIC realization and ASIC designs are always faster than FPGA realizations.

REFERENCES

[1] C. Y. Yan and R. Xiao, “Study of block algorithms implement on hardware in an information security system,” in *Business Management and Electronic Information (BMEI)*, pp. 589-593, 2011.

[2] P. Leglise, et al., “Efficient Implementation of Recent Stream Ciphers on Reconfigurable Hardware Devices,” in *26th Symposium on Information Theory in the Benelux*, pp. 261-268, 2005.

[3] B. Prathiba, et al., “FPGA Implementation of Smart Cryptography Algorithm,” in *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 5, pp. 3017-3020, 2020.

[4] B. Wang and L. Liu, “A flexible and energy-efficient reconfigurable architecture for symmetric cipher processing,” in *IEEE International Symposium on Circuits and Systems*, vol. 14, pp. 1182-1185, 2015.

[5] Y. Chen, et al., “Research and Implementation of Reconfigurable Architectures of DES and ZUC,” in *Second Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 7, no. 1, pp. 216-220, 2017.

[6] N. B. Hulle, et al., “Compact Reconfigurable Architecture for Sosemanuk Stream Cipher,” in *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 3, pp. 607-611, 2020.

[7] A. Khalid, et al., “RC4-AccSuite: A Hardware Acceleration Suite for RC4-Like Stream Ciphers,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 1072-1084, 2017.

[8] G. Paul and A. Chattopadhyay, “Three Snakes in One Hole : A 67 Gbps Flexible Hardware for SOSEMANUK with Optional Serpent and SNOW 2.0 Modes,” in *IACR Cryptology ePrint Archive*, pp. 1-19, 2013.

[9] M. Galanis, et al., “Comparison of the Hardware Implementation of Stream Ciphers,” *The International Arab Journal of Information Technology*, vol. 2, no. 4, pp. 267-274, 2005.

[10] “Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2, Document 2: SNOW 3G specification,” ETSI/SAGE Specification, Version 1.1, 2006.

[11] “Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2, Document 3: Implementor’s Test Data,” ETSI/SAGE Specification, Version 1.1, 2012.

[12] P. Kitsos, et al., “High-Performance ASIC Implementation of the SNOW 3G Stream Cipher,” in *IFIP/IEEE VLSI-SOC 2008 - International Conference on Very Large Scale Integration (VLSI SOC)*, Rhodes Island, Greece, pp. 1-4, 2008.

[13] P. Kitsos, et al., “FPGA-based performance analysis of stream ciphers ZUC, Snow3g, Grain V1, Mickey V2, Trivium and E0,” *Microprocessors and Microsystems*, vol. 37, no. 2, pp. 235-245, 2013.

[14] L. Zhang, et al., “Evaluating the Optimized Implementations of SNOW 3G and ZUC on FPGA,” in *Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 436-442, 2012.

[15] M. Madani and C. Tanougast, “Combined and Robust SNOW-ZUC Algorithm Based on Chaotic System,” in *2018 International Conference on Cyber Security and Protection of Digital Services, Cyber Security*, pp. 1-7, 2018.

[16] M. Madani, et al., “Digital Implementation of an Improved LTE Stream Cipher Snow-3G Based on Hyperchaotic PRNG,” *Security and Communication Networks*, vol. 2017, no. 2, pp. 1-15, 2017.

[17] R. D. Kharadkar and N. B. Hulle, “FPGA Implementation of Modulo (2^{31}-1) Adder,” in *7th International Conference on Emerging Trends in Engineering & Technology*, Kobe, Japan, pp. 85 -90, 2015.

[18] P. Ekdahl and T. Johansson, “SNOW - a new stream cipher,” in *RST open Nessie workshop*, Heverlee, Belgium, pp. 1-17, 2001.

[19] P. Hawkes and G. G. Rose, “Guess-and-Determine Attacks on SNOW,” in *SAC 2002 Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography*, pp. 37-46, 2002.

[20] P. Ekdahl and T. Johansson, “A New Version of the Stream Cipher SNOW,” in *International Workshop on Selected Areas in Cryptography*, pp. 47-61, 2002.

[21] V. Jairaj, et al., “High Performance Implementation of Snow3G Algorithm in Memory Limited Environments,” in *New Technologies, Mobility, and Security (NTMS)*, no. 5, pp. 1-4, 2011.

[22] Y. Pai and Y. Chen, “The Fastest Carry Lookahead Adder,” in *Second IEEE International Workshop on Electronic Design, Test and Applications*, no. 1, pp. 4-6, 2004.

[23] S. Traboulsi, et al., “An Optimized Parallel and Energy-Efficient Implementation of SNOW 3G for LTE Mobile Devices,” in *International Conference on Communication Technology (ICCT)*, pp. 535-538, 2010.

[24] N. B. Hulle, et al., “The Novel Architecture for Carry Lookahead Adder,” *Technical Journal of The Institution of Engineers (India)*, Pune Local Centre, vol. 36, no. 1, pp. 84-88, 2012.

[25] N. B. Hulle, et al., “High Performance Architecture for LILI-II Stream Cipher,” in *International Journal of Computer Applications*, vol. 107, no. 13, pp. 10-13, 2014.

[26] G. Orhanou, et al., “SNOW 3G Stream Cipher Operation and Complexity Study,” *Contemporary Engineering Sciences*, vol. 3, no. 3, pp. 97-111, 2010.

[27] S. Hessel, et al., “Implementation and Benchmarking of Hardware Accelerators for Ciphering in LTE Terminals,” in *Global Telecommunications Conference*, pp. 1-7, 2009.

[28] N. P. Maity and R. Maity, “Design and Modelling of Paralleled RAM Architecture,” in *International Conference on Future Information Technology*, vol. 13, pp. 98-102, 2011.

[29] A. Bikos and N. Sklavos, “Architecture Design of an Area Efficient High-Speed Crypto Processor for 4G LTE,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 729-741, 2018.

[30] xilinx.com, “XUPV5-LX110T,” 2015. [Online], Available: http://www.xilinx.com/univ/xupv5-lx110t.htm.

BIOGRAPHIES OF AUTHORS

Dr. N. B. Hulle is an Associate Professor at G H Raisoni Institute of Engineering and Technology, Pune, Maharashtra, India. He completed his Bachelor's and Masters from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India, and Ph.D. from Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur, Maharashtra, India. He has 21 years of teaching experience. He guides UG and PG students. His area of research is VLSI, cryptography & Wireless Network Security. He published papers in National & International Journals and conferences. Dr. N. B. Hulle is a life member of ISTE and IETE.

B. Prathiba is an Assistant Professor at G H Raisoni Institute of Engineering and Technology, Pune, Maharashtra, India. She has published articles in various international publications in the area of Wireless Sensor Networks. Her interested areas are Wireless Sensor Networks, Microcontrollers, and VLSI. B. Prathiba is a life member of ISTE and IETE.

Sarika R. Khope is an Assistant Professor at G H Raisoni Institute of Engineering and Technology, Pune, Maharashtra, India. She has published the articles in various international publications in the area of Image steganography, Image Fusion, and VLSI Her interested areas are Digital Systems Design, Machine Learning, and VLSI. Sarika Khope is a life member of ISTE.

Dr. K. Anuradha is a Professor and Dean ICT at Gokaraju Rangaraju Institute of Engineering & Technology, Hyderabad, Telangana, India. She is specialized in the areas of Data mining, Image Processing, Machine Learning, Network Security and Computer Networks.

Yogini Borole is an Assistant Professor at G H Raisoni Institute of Engineering and Technology, Pune, Maharashtra, India. She has published the articles in various international publications in the area of DSP and VLSI. Her interested areas are Digital Signal Processing, Power Electronics, and VLSI. Yogini Borole is a life member of ISTE and IETE.

Dr. D. Kotambkar is an Assistant Professor at Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India. She has published the articles in various international publications in the area of MIMO Wireless Communication.Her interested areas are Digital Signal Processing and Wireless Communication..She is a life member of IETE.