AES-32: An FPGA implementation of lightweight AES for IoT Devices

Sumit Singh Dhanda[1, 2], Brahmjit Singh[1] and Poonam Jindal[1]

[1] *Department of Electronics and Communication, National Institute of Technology, Kurukshetra, India*
School of Computing Science and Engineering, Galgotias University, Greater Noida, India

E-mail address: dhandasumit@gmail.com, brahmjit.s@gmail.com, poonamjindal81@nitkkr.ac.in

Received ## Mon.20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: IoT is marked by the resource-constrained devices. Information security is the main challenge that arise due to wireless transmission of data by ubiquitous sensors. The phenomenal growth of resource constrained devices in IoT setups has motivated for the research of lightweight solutions for information security. In this work, an optimized implementation of AES for high throughput has been presented. The data path of the AES is compressed to 32-bit. Implementation has been carried out on different FPGA families. Data path compression and use of BRAMs has led to improved throughput with savings in resource consumption. Loop-unrolled AES results in the consumption of 2669 slices which 12 times as big as this design. While 32-bit AES with 128-bit data path consumes 4 times more resources than proposed design which uses 223 slices and 5 BRAMs on Artix-7 FPGA. The proposed design delivers throughput in the range of 2.2 to 3.5 Gbps and achieves efficiency of 1.75 Mbps-7.8 Mbps per slice on different FPGAs. It outperforms different lightweight ciphers and constrained AES implementations in existing literature.

Keywords: *Internet of Things (IoT), Data path, Advance Encryption Scheme (AES), Field Programmable Gate Arrays (FPGA), Information security.*

1. INTRODUCTION

Cisco estimates that the number of connected devices will rise from 50 billion by 2020 to reach 500 billion by 2025 [1]. To ensure the information security in Internet of Things (IoT) small sensors and devices need to be safeguarded against sniffing attacks. Smart grids are also an application of IoT. Enormous data exchange and openness of resource sharing among smart meters in smart grid have also generated challenges of data security [2]. Data privacy and data leakage are also an important concern at cloud level as well [3]. Confidentiality of the information can be enhanced with the help of Block ciphers. These are used for ensuring information security [4-5] in various standards. Blockchain technology is another important area where cryptographic algorithms serve as the base of security [6]. Advanced Encryption Standard (AES) is considered to be the most secure block cipher that can be used for the purpose. National Institute of Standards and Technology (NIST) [7], released FIPS-197 in which AES was adopted as a standard symmetric cipher. It ensures confidentiality at two levels

i. For high throughput applications such as e-commerce or in case of trunk communication.
ii. For lower data rates it can be used for resource constrained devices.

Software and hardware implementations of AES are utilized for these purposes. Hardware implementation of AES is preferred as compared to software implementation for high throughput applications. These implementations are carried out either on field programmable gate arrays (FPGA) or on application specific integrated chips (ASIC). Major research areas of AES implementation are

highlighted in Fig 1. To minimize the delay highly pipelined architectures are implemented. Area reduction is achieved by iterative architecture. Several optimizations in the basic operations such as SubBytes or Mix-Columns, arithmetic operations etc. are also used for the same. Further, resource sharing [8] has also been used to minimize the area and increase the speed of the architecture while maintain the integrity of the cipher. Data-path reduction [9] is one of the resource sharing techniques to achieve the smaller area implementation of AES. Due to the ever-increasing demand of security solutions for resource constrained devices researchers are still working in the direction of developing new architectures of AES.

Various attempts are reported in literature towards optimizations are broadly focused in two categories:

i) Pipelined (fully or partially) architecture for implementation of high speed.
ii) Compact and low-power architecture for the low resources or low-cost devices and feedback mode of operations.

Major contribution of this work is to explore the adaption of AES-128 to low-cost devices in IoT by effective resource utilization. A two-step approach is applied to minimize the latency and resource consumption. First step considers the compression of data path to 32-bits. Use of BRAMs available in FPGA, maximizes the utilization of available resources. Although, this design utilizes 32-bit data path like others [10-12] but optimum utilization resources enable it to stand out among these designs. Efficient use of available block RAMs has enabled it to minimize the resources. Use of BRAMs for the implementations of the S-boxes yielded heavy reduction in the resource consumption.

The contribution of this work are as follows:

i. In this paper, a new high performance constrained architecture has been presented for resource constrained devices.
ii. Architecture makes use of BRAMs to minimize the resource consumption on FPGAs. It achieves the optimum utilization of FPGA resources.
iii. It also presents the state of the art in the field of research.
iv. The result is compared with existing designs and lightweight implementations for IoT.

Rest of the paper is organized as follows: Various contemporary implementations are presented alongside older ones in Section 2. Section 3 provides the implementation details. Implementation results of the proposed design are presented, compared and discussed while identifying its applications in Section 4. In the end Section 5, draws conclusions and provides the future work directions.

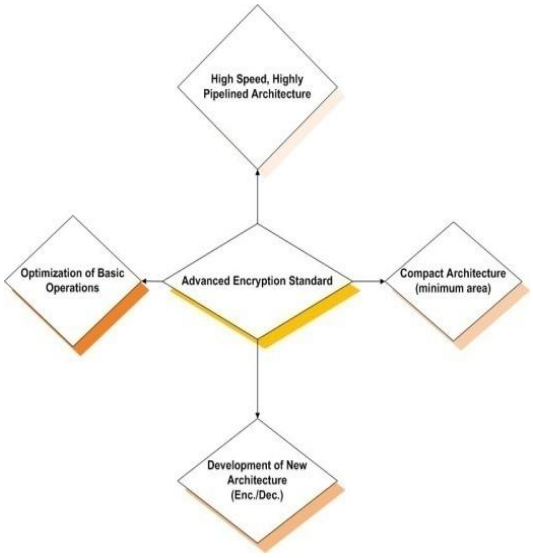

Fig 1. Research directions in AES

2. RELATED WORK

In [8], authors used the 32-bit data path, resource sharing between these encryption and decryption units and subfield arithmetic to minimize the hardware requirement. In [13], a low power AES architecture with an optimized S-box have been implemented on an FPGA with 128-, 192- and 256-bit keys. An ASIC implementation for the AES processor has been carried out in [14] which is capable of delivering the throughput of 2.29Gbps. In [15], authors carried out 32-bit implementation on FPGA with the help of pre-computed key expansion for FPGA. S-box is implemented as LUTs. The design used the dedicated memory blocks that were available on FPGA. Shift rows is performed with addressing logic. It is made possible by arranging the state- bytes in such a manner that were efficiently stored in shift registers. The same method has been used in [16] to reduce storage requirements and implement data paths of various sizes. In [10], authors have improved the FPGA resource consumption using T-box method. In [17], a theoretical design for the AES architecture was presented to optimize the resource consumption. In [9], authors carried out a fully parallel and

loop unrolled implementation of AES using composite field arithmetic and LUT based T-Boxes. It was carried out for two different architectures one was 8-bit S-box based while the other was 32-bit data path. The architectures were optimized for high speed and low latency. The theoretical architecture presented by in [17] was utilized in [18] with a core added with decryption functionality and 8-bit data path. Data path contains S-box implementation in combinatorial logic. A study focused on the IoT devices and their design was presented in [19] but they left small devices. The power consumption for AES has been reported 42 mW in this study which is not appropriate for the constrained IoT devices. Hence, the 128-bit architectures mentioned in [19-21] are not suitable for the implementation in constrained devices due to power requirements. Similarly, [22] utilizes 32-bit data path and has power consumption in micro-watt level but the area requirements make it unsuitable for the small sensors. In [23], an asynchronous design has been presented for 128-bit data-path AES that consumes lesser power but the area requirements are high for the small devices and power consumption is still a concern for resource constrained devices. In [24], a lightweight AES algorithm is implemented on FPGA. Mixcolumns step is removed to achieve minimum delay in an adhoc voice link.

In [25], a new AES crypto-hardware accelerator was presented for the devices such as Bluetooth controller. It uses power efficient designs for S-box, MixColumns, Shiftrows and their inverses. The area occupied is 3120 GE for the 130 nm CMOS technology. In [26], a new design named nano-AES was presented utilizing 8-bit data path. It was an ASIC implementation which achieved 35-2.4% improvements over previous works. In [27], have presented 8-bit architecture for the SILC, CLOC, AES-JAMBU, and COLM authenticated ciphers. All of these are designed by modifying AES core. AES-JAMBU used the least resources among all of these. A crypto-engine for AES-GCM was purposed in [28], which generates the throughput of 100 Gbps. It can be utilized in optical transport networks. It is designed using 40 nm library. AES has been adapted to design a chaos-based algorithm for the encryption in [29]. It provides security for images and data. Authors have tested the scheme for different tests and attacks and high resistance has been reported against such attacks. Security issues of AES based designs are highlighted in next few works. True random number generators (TRNGs) have a statistical weakness due to physical randomness. A post-processing method can be used to solve this issue. An S-box based solution have been proposed in [30]. In [31], a correlation scan attack against XOR compaction is proposed. In [32], LC-FARES was presented. It has the capability to identify injected-faults. Sixteen 8-bit registers are used, in a 32-bit architecture, for implementing ShiftRows. A flexible AES design, that can choose from different defense mechanisms, key sizes and mode of operations etc., is presented in [33] using an agile approach. It uses Chisel framework to achieve reduced code size. Authors have designed an advanced crypto-hardware for AES. It supports variable key sizes in multiple modes [34]. The designs are synthesized using 7 nm CMOS technology. In [35], authors have presented a lightweight cipher using Lorentz-chaotic system (LCS). It occupies only 27 slices and uses feistel structure. LCS has been used to generate the random numbers which are used in the key.

Numbers of works have been reported on the AES optimizations. There is need for the reduction in resource consumption of AES. Data path compression is one of the popular strategies for the area minimization. But only few works have been reported for lightweight-AES for the IoT applications. It is a big clearly highlights a gap in the literature and motivated us to adopt following methodology:

i. AES-128 has been adapted to AES-32 by data path compression. BRAMs are used to further minimize the resource consumption.
ii. Verilog is used for coding the design which is synthesized on PlanAhead software.
iii. Thereafter, it was implemented on different FPGAs.
iv. Based on these FPGA implementation design is compared with existing designs.

Proposed design outperforms existing works in throughput and area.

3 32-bit Data Path Implementation

This 32-bit iterative architecture was designed for high throughput with minimized resource-consumption. It is shown in Fig 2 below. MixColumns are 32-bit in size just like main data path. A separate S-box is used for on-the-fly key generation. It reduces the delay and enhance the performance of the design. Initially, a 128-bit input is provided to AES-32 module. This input is converted into 32-bit blocks for the SubBytes operation.

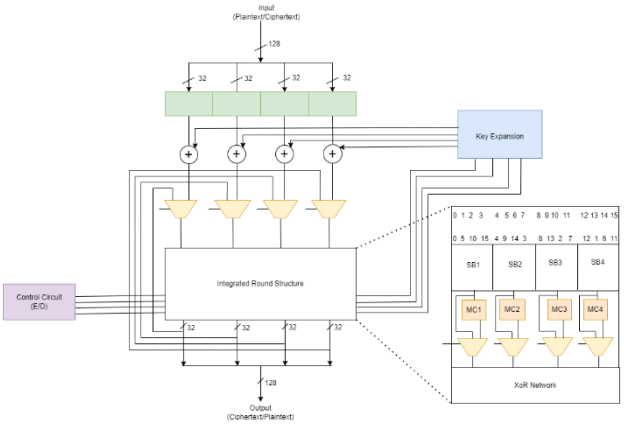

Fig 2. BRAM based 32-bit iterative AES architecture

ShiftRows operates on input bytes and arrange them into a fixed sequence. Every time input is fed this sequence is repeated. Considering this, sequence of output-bytes is changed in the state as per the standard design of Chadoweic and Gaj [15] (2003) and N. Pramstaller, et al. [16] (2004). After wards, MixColumns operation is performed which are 32-bit wide and four in number. Final operation is Add round key and a separate SubBytes unit is used for the same. Here, the calculation is done on the fly. The SubBytes operation utilizes either galois field (GF) arithmetic or the S-Box can be stored as look up table (LUT). Here, S-box is implemented as LUT which makes it a little more resource consuming compared to GF based design. But we have utilized the block RAM (BRAM) available in the FPGA to store the S-box entries. In this architecture, the encryption and decryption utilize the same data-path and resources. This has been enabled by the use of multiplexers. Key Expansion unit also utilizes the separate S-box but adapting FPGA to utilize the on-board block RAM has helped in cutting down the resource consumption. 5 BRAMS have been used in all in the Artix-7 FPGA. It has enabled the design to reduce the resource requirement heavily. Total 256 entries have been made for the byte substitution table. In this process, the 'case' statement has been utilized for the byte substitutions. It is an area consuming process but it will help in faster execution of the cipher. Due to 32-bit data path and sharing of S-box, the number of cycles required to implement one round now become 5 (4 cycles for main data and one cycle for key expansion). Multiplexers helps in sharing of resources.

The SubBytes architecture is shown in Fig 3. It is a 32-bit wide operation which is divided in four 8-bit wide operations individually. Hence, these are calculated individually and combined in the end. The Key expansion module utilizes a separate SubBytes module in this design and hence it is able to calculate the output in minimum cycles.

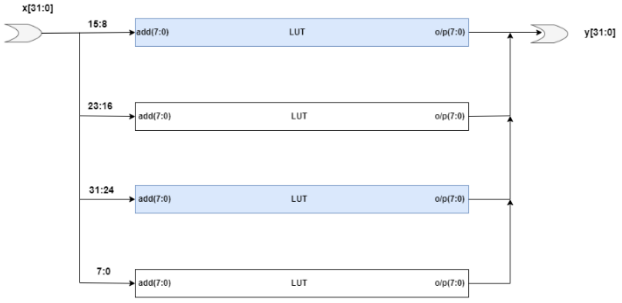

Fig 3: BRAM based SubBytes

Although, the cost is paid in terms of BRAMs and additional control circuitry. Inverse SubBytes is similar to SubBytes operations and uses same number of resources.

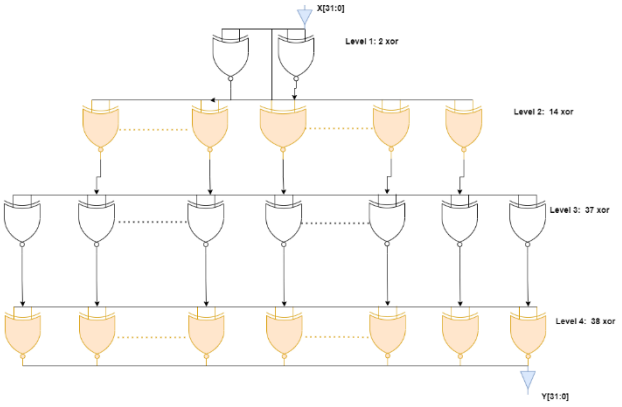

Fig 4: MixColumns structure

Four levels of logic constitutes the MixColumns operation. Fig 4 shows the different levels of logic used in the MixColumns design. There are total 4-levels of logic used in this design and a total of 91 XOR operations are needed. There are two XOR gates on first level, Fourteen XOR gates are present on level 2. While level 3 consist of thirty-seven gates and finally thirty-eight can be found in level 4. The inverse MixColumns operation is quite similar to the design but there are five levels of logic.

4 Result and Discussions

The initial implementation of the design is carried out with Xilinx Vivado software version 2014 and Artix-7 FPGA. On the other hand, for the comparison with exiting designs the synthesis is carried out using Xilinx PlanAhead software and implementation is carried out on different FPGAs. While mentioning the FPGAs, we have used 'V' for Virtex, 'K' for Kintex and 'Sp' for Spartan family while numeric values 5, 6, 7 or alphabet 'E' etc. represent the generation of the particular family.

This 7-series FPGA have two types of slices; slice-M and slice-L. Here, the advantage of using slice-M is that it can utilize its LUTs to configure distributed RAM (DRAM). It helps in better utilization of resources. Another strategy that we have adopted is to utilize the BRAM for S-box. BRAM on 7-series FPGAs has storage capacity up to 36 Kbits which makes it ideal suited for the S-box storage. It can also be used for other storage as well. Since S-box as LUT has 256 entries and each entry is a byte long, using slice resources or DRAM for the same will be a waste of resources. Fig 5 and 6 represent the top-level schematic of the AES-128 bit and AES-32 bit respectively. AES-128 has been implemented as loop unrolled architecture while as mentioned earlier; AES-32 is implemented as the iterative architecture.

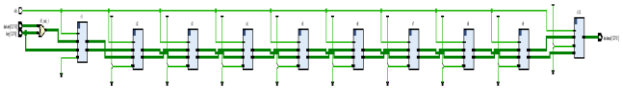

Fig 5. Top level Schematic for AES-128

Table 1 represents the resource consumption of the AES-32 on Artix-7 FPGA and its comparison with AES-128. It shows that total 568 LUTs have been used while the number of slices stands at 223. The design also utilizes 5 BRAMs available on FPGA. These BRAMs are used for the implementation of the S-boxes which are implemented as the LUT. It helps in the better resource utilization. There are two types of slices available on 7-series FPGAs slice-M and slice-L. Slice-M has advantage that these can be adapted to form the DRAMs which can further be used for the storage purposes while the software is using the optimizing strategies. In our design, out of 223 slices that have been used; 40 percent are slice-M and 60 percent are slice-L. But the main reduction in the resource consumption is achieved through the use of BRAMs. To emphasize on savings that our implementation has achieved we have compared it to AES-128 that has been implemented on the same Artix-7 FPGA. It has loop unrolled architecture. The comparison is shown in the Table 1 and depicted in Fig 7. It shows that AES-128 consumes 2668 slices and a total of 9571 LUTs. The results obtained for the utilization of the resources in the Artix-7 FPGA are as follows:

The total improvement in terms of resource consumption is 91.64 percent when it is compared to AES-32. Similarly, we have also implemented AES-32 with 128-bit data path. It uses 424 slices and 1231 LUTs. These results point out that just by compression of data path to 32-bit an area saving of 47.40 percent is achieved.

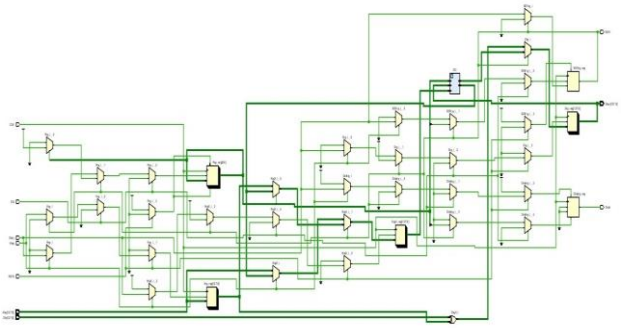

Fig 6. Schematic for AES-32 using BRAMs

Table 1. Comparison of three designs amongst each other

Design	Slices	LUTs	Improvement (%)
AES-128 (loop unrolled architecture)	2669	9571	91.64
AES-32-bit with 128-bit data path	424	1231	47.40
AES-32-bit	223	568	

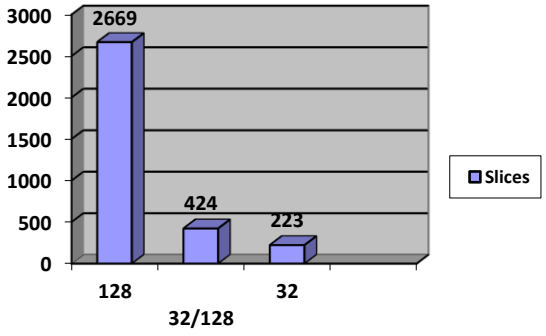

Fig 7. Comparison of three basic implementations of AES among each other

The same is represented in the bar chart in Fig 8. If this 32-bit implementation is compared with the existing implementations it can be seen that there is drastic reduction in terms of area due to the use of BRAMs for S-box implementation. This reduction helps in making design compact and better suited for the small devices.

The proposed design is compared with the existing ones based on three factors. The number of slices consumed for the implementation. The maximum frequency of operation (F_{max}) design has clocked on the FPGA. The throughput

delivered by the proposed design and its efficiency which is calculated as throughput per slice (TPS).

Mega Hertz (MHz) is the unit for calculating Maximum frequency of operation. It is the maximum value recorded when design is implemented on a specific FPGA. Throughput, T is recorded in mega-bits per second (Mbps). The formula for throughput is

$$T = \frac{B \times F}{N} \quad (1)$$

where, the number of bits processed at a time which forms block-size is represented by B, F is frequency of operation and number of clock-cycles utilized in the encryption of a complete block of data are shown by 'N'. (or decryption)

Equation (2) provides another parameter for analyzing the performance i.e. Throughput per slice, TPS, or efficiency:

$$TPS = \frac{T}{R} \quad (2)$$

Here, number of resources i.e. slices or LUTs, that are occupied by the design when implemented on a particular FPGA are denoted by R. Here, it represents slices. Utilization of resources can be rightly estimated with the help of Efficiency.

For the comparison with the existing designs, we have synthesized the design in Xilinx PlanAhead and implemented it on the different Xilinx FPGAs. It provided us with results to carry out a detailed comparison with existing designs. For comparison, we have adopted the strategy mentioned by [36] where the concept of 'equivalent slices' and a 'normalization method' was adopted to carry out the design comparison.

As the comparison is made among old and new devices, the concept of 'equivalent slices' is used for the same. As many designs are based on the look up tables which are stored in the BRAMs available in FPGA. Hence, the resource consumption has two parts: number of slices and BRAMs. The cost of a BRAM is calculated in term of slices and finally, added to number of slices to calculate the equivalent number of slices consumed by the design.

Detailed study of the literature suggests that different types of FPGA are used for implementation. FPGA used varies from Virtex-II to Virtex-7. Hence, a 'normalized TPS' calculation is provided for a fair comparison. Virtex-II, Virtex-II pro, Spartan-3, Virtex-E and Virtex-4 implementations are calculated under following assumptions:

i) As a slice in these FPGAs comprises 2 LUTs while FPGAs from Virtex-5 onwards comprise of 4 LUTs per slice. Therefore, the occupied area is divided by 2 in case of these FPGAs.

ii) In these FPGAs, one BRAM is equivalent to 64 slices (18 Kb BRAM) while Virtex-5 onwards the one BRAM is equivalent to 128 slices (36 Kb BRAM).

iii) Finally, to normalize the frequency of operation, a factor of 1.22 (550/450) is multiplied to the frequencies achieved by these FPGAs because these FPGAs can have maximum frequency of operation 450 MHz while Virtex-5 onwards the maximum frequency of operation is 550 MHz.

Based upon this normalization criteria, the normalized frequency, throughput and TPS are calculated for the FPGAs of older generation i.e. before Virtex-5. We can see the comparison of the purposed design with the designs available in literature.

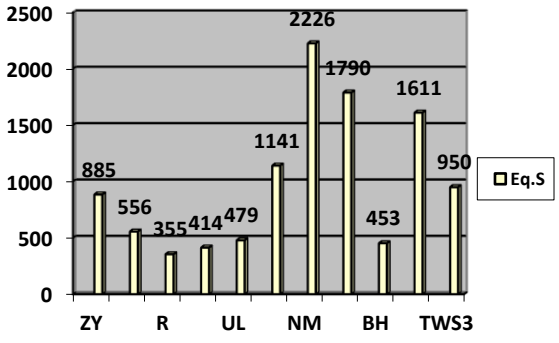

Fig 8. Comparison of Resource Consumption among Designs

Fig 8 presents the resource consumption of all the designs. The design by [10] is the most resource constrained. While [15] are second most constrained implementation. Our design is fourth among these designs in terms of resource consumption. The results are depicted the equivalent slice calculations. Hence, the BRAM occupancy increases the number of slices. But being the part of the FPGA architecture, their use increases the utilization of available resources which would be wasted otherwise. This use of BRAMs helps the design to achieve high operating frequencies.

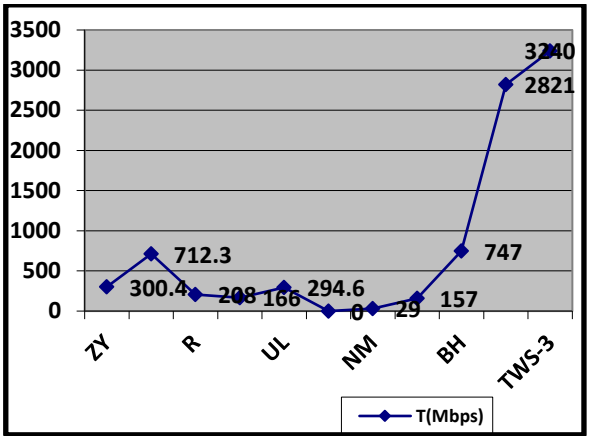

Fig 9. Throughput comparison among designs

TABLE 2: COMPARISON WITH DIFFERENT DESIGNS WITH NORMALIZATION

Work	FPGA	Slice + BRAM	Equivalent Slices	Fmax (MHz)	T (Mbps)
[37] ZY	V-5	885 4992	885 4992	103.3 116	300.4 1350
[12] NB	V-5	556	556	256	712.3
[10] R	Sp-3	163+3	355	71	208
[15] CG	Sp-2	222+3	414	60	166
[11] UL	Sp-3	287+3	479	123.464 (101.2)	294.4
[38] NK	V-4	2281	1141	167.14 (137)	-
[39] NM	V800-4	4452	2226	28.06 (23)	29
[40] LO	V-E	3580	1790	-	157.07
[41] RBH	V-5	69+3	453	257	747
This work	V-5	459+9	1611	220	2821
	Sp-3	619(/2) + 10	950	253.15 (207.5)	3240.32

TABLE 3: COMPARISON OF NORMALIZED THROUGHPUT AND TPS AMONG DIFFERENT DESIGNS

Work	T (Mbps)	TPS (Mbps/ Slice)
[37] ZY	300.4 1350	0.339 0.270
[12] NB	712.3	1.28
[10] R	208	0.70
[15] CG	166	0.32
[11] UL	294.4	0.61
[38] NK	-	-
[39] NM	29	0.013
[40] LO	157.07	0.08774
[41] RBH	747	1.65
This work V-5	2821	1.75
Sp-3	3240.32	3.414

Fig 9 depicts the comparison of the designs on the basis of throughput which is measured in megabits per second (Mbps). Both implementations of the proposed design provide the best throughput among presented designs with 2821 and 3240 Mbps. It shows that in terms of performance proposed design provides superior throughput.

Fig 10 shows the comparison on the basis of maximum frequency of operation. These all are normalized results and designs by [41] and [12] occupies the first and second place in terms of maximum frequency. The proposed design occupies third place on Spartan-3 FPGA while fourth place is for the Virtex-5 FPGA implementation of this design. Hence, design performs satisfactorily on this parameter.

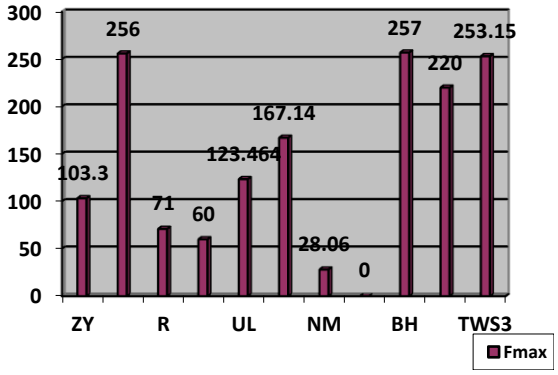

Fig 10. Frequency comparison among designs

Apart from the comparison with the existing designs Table 4 compares the proposed design with the latest lightweight ciphers proposed for the IoT requirements.

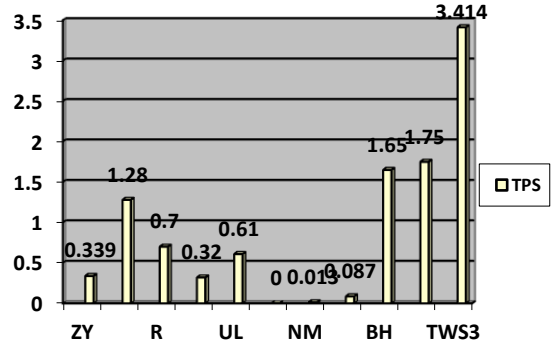

Fig 11. TPS comparison among designs

TABLE 4: A COMPARISON WITH OTHER LIGHTWEIGHT PRIMITIVES

Design	FPGA	Slices	Fmax	Throughput (Mbps)	TPS (Mbps/slice)
PRESENT [42]	V-6	157	186.3	372.6	2.37
LED [43]	V-7	217	169.09	338.18	1.55
HIGHT [43]	V-7	252	372.3	744.6	2.95
SIMON [44]	V-7	95	219	292	3.07
XTEA [45]	K-7	228	345.9	26.42	0.115
[48]	V-6	657	170	2188.7872	3.33
	V-7	666	213	2734.5792	4.11
	K-7	536	213	2734.5792	5.10
LC-FARES S. [32]	V-6	283	166	241.4	0.85
	V-7	277	310	450.9	1.62
	K-7	271	330	480	1.77
This work	V-6	289+9	176.305	2256	7.8086
	V-7	489+5	240.269	3075	6.28924
	K-7	460+5	274.907	3518	7.64958

Table 4 presents a comparison among different lightweight ciphers. These primitives are chosen for comparison on the basis of implementation of slices, frequency of operation, throughput and TPS. As can be observed from the table 4, proposed design is not as constrained in implementation as the other ciphers. It consumes maximum resources among all the ciphers. The main reason behind this is the block size processed in these ciphers. Proposed lightweight-AES algorithm processes 128-bit block in comparison to the 64-bit block size of LED, XTEA, SIMON and HIGHT. Small block size means that size of the ciphers and processing time will be small. But iterative architecture of AES-32, data path compression reduces the area required. Use of BRAMs further minimize the resource consumption. AES-32 has been compared with the design in [48] which does not utilize BRAMs and s-box has been implemented in galois field. It clearly highlights that proposed design achieves the maximum utilization of available FPGA resources. Comparison with [48] further highlights that implementation of s-box in BRAMs results in 50 to 90% improvement in the throughput. The number of cycles remained 10 due to a separate s-box for key-schedule. Hence, AES-32 is able to achieve higher throughput with small area.

AES-32 can be operated in nearly same frequencies. The throughput of the design underlines the performance of the design. It is best among all the designs in the table. TPS (efficiency) data shows that the design is able to achieve the optimum utilization of FPGA resources. It presents maximum performance per unit resource among all the designs. A thorough comparison reflects that slightly more consumption of resources by the purposed design enables it to deliver best performance among all. It helps in delivering highest throughput and maximum TPS. The TPS results obtained by the AES-32 reflects the optimum utilization of resources of the FPGA along with best per slice performance. Use of BRAMS in the implementation has enabled the faster processing and low-resource consumption of the proposed design. Another reason for the optimum performance is the use of separate 'SubBytes' for key-schedule has helped in delivering high throughput and exhibit low latency. It also consumes low resources which make it a suitable option for the different use cases in IoT such as smart buildings, smart lighting,

AC control and surveillance etc. Low latency makes it desirable for faster response applications such as smart-grid applications.

5 Conclusion and Future Work

In this work, we have adapted AES-128 to AES-32 employing data path compression strategy. Sharing the resources between encryption and decryption path, the LUTs requirement is minimized. Effective utilization of FPGA resources has led to further reduction in the number of slices and improvement in throughput over existing designs. AES-32 which is nearly 6.9 times smaller in comparison to loop- unrolled AES-128 is more suitable for small IoT devices. Utilizing five on-board block RAMs overall consumption of LUTs is remarkably reduced. It results in lesser slices requirements (223 slices for encryption) for AES implementation. The proposed design achieves a throughput of 3.2 Gbps on Spartan-3 device while it remains between 2.2 to 3.5 Gbps in other Xilinx FPGAs. Efficiency of the design also verifies the optimum utilization of the resources by the design. It ranges from 1.75 Mbps – 7.8 Mbps per slice. High throughput and low resources make it a suitable option for the different use cases in IoT such as smart buildings, smart lighting, AC control and surveillance etc. Low latency may make it desirable for faster response applications such as smart-grid applications. BRAMS are good way for resource reduction in FPGA. But its implementation in gates occupies large area. In future, effort will be on further reduction of slice consumption and developing more lightweight-ciphers for IoT applications.

References

[1] Alnefaie, S., Alshehri, S. and Cherif, A. (2021) 'A survey on access control in IoT: models, architectures and research opportunities', *Int. J. Security and Networks*, Vol. 16, No. 1, pp.60–76.

[2] Dari, E.Y., Bendahmane, A. and Essaaidi, M. (2021) 'Verification-based data integrity mechanism in smart grid network', *Int. J. Security and Networks*, Vol. 16, No. 1, pp.1–11.

[3] Ghorbel, A., Ghorbel, M. and Jmaiel, M. (2021) 'A model-based approach for multi-level privacy policies derivation for cloud services', *Int. J. Security and Networks*, Vol. 16, No. 1, pp.12–27.

[4] Jindal P, Singh B, (2017) Quantitative analysis of the security performance in wireless LANs, Journal of King Saud University-Computer and Information Sciences, 29(3) 246-268.

[5] Jindal P, Singh B (2015) Analyzing security-performance tradeoff in block ciphers, *International Conference on Computing, Communication & Automation*, 2015, pp. 326-331, doi: 10.1109/CCAA.2015.7148425.

[6] Echchabi, A., Omar, M.M.S. and Ayedh, A.M. (2021) 'Factors influencing Bitcoin investment intention: the case of Oman', *Int. J. Internet Technology and Secured Transactions*, Vol. 11, No. 1, pp.1–15.

[7] NIST Advanced Encryption Standard (AES), FIPS PUBS 197, National Institute of Standards and Technology, November 2001.

[8] Satoh A., Morioka S., Takano K., and Munetoh S. (2001) A compact Rijndael hardware architecture with S-Box optimization", In *Proc. Theory and Application of Cryptology and Information Security (ASIACRYPT'01)*, LNCS, vol. 2248, pp. 239–254. Springer-Verlag.

[9] Tim Good, Mohammed Benaissa, Very Small FPGA Application-Specific Instruction Processor for AES, IEEE Transactions on Circuits and Systems-I:Regular Papers, vol. 53(7), July 2006, pp. 1477–1486

[10] Rouvroy G., Standaert F. X., Quisquater J. J., and Legat J. D. (2004) Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications, in *Proc. ITCC'04*, Apr. 2004, vol. 2, pp. 583–587.

[11] Legat U., Biasizzo A., Novak F. (2011) A compact AES core with online error-detection for FPGA application with modest hardware resources", Microprocessors & Microsystems, 35(4), 405–416.

[12] Benhadjyoussef N., Karmani M., Machhout M., Hamdi B. (2020) A hybrid countermeasure-based fault-resistant AES implementation", Journal of Circuits, Systems, and Computers, 29(3) (2020) 2050044 (17 pages) DOI: 10.1142/S0218126620500449

[13] J¨arvinen K.U., Tommiska M.T., Skytt¨a J.O. (2003) *A fully pipelined memoryless 17.8Gbps AES-128 encryptor*, International Symposium on Field-Programmable GateArrays (FPGA 2003), Monterey, CA, 2003

[14] Verbauwhede, I., Schaumont, P., and Kuo, H. (2003) Design and performance testing of a 2.29 Gb/s Rijndael processor', IEEE Journal of Solid-State. Circuits, pp. 569–572.

[15] Chodowiec, P. and Gaj, K. (2003) Very compact FPGA implementation of the AES algorithm". In C¸.K. Koc¸ and C. Paar, editors, *Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES'03)*, LNCS vol. 2779, pp. 319–333 Springer-Verlag.

[16] Pramstaller, N., Mangard, S., Dominikus, S., and Wolkerstorfer, J. (2004) Efficient AES implementations on ASICs and FPGAs, In Dobbertin, H., Rijmen, V., and Sowa, A. (Eds): Proc. Fourth Workshop on the Advanced Encryption Standard ''AES - State of the Crypto Analysis'', AES 2004, LNCS, 3373, pp. 98–112.

[17] Järvinen T., Salmela P., Hämäläinen P., and Takala J. (2005) Efficient byte permutation realizations for compact AES implementations, In Proc. 13th European Signal Processing Conf.(EUSIPCO 2005), Antalya, Turkey.

[18] Hamalainen P., Alho T., Hannikainen M., and Hamalainen T. (2006) Design and Implementation of Low-area and Low-power AES encryption hardware core, in *IEEE pro. DSD*, pp. 577 - 583.

[19] Agwa, S., Yahya, E. & Ismail, Y. (2017) Power efficient AES core for IoT constrained devices implemented in 130nm CMOS. In Proc. 2017 IEEE ISCAS, Baltimore, pp. 1–4.

[20] Wang, Y., & Ha, Y. (2013) FPGA-based 40.9-Gbits/s masked AES with area optimization for storage area network. IEEE Transactions on Circuits and Systems II: Express Briefs, 60(1), 36–40.

[21] Wong, M. M., Wong, M. L. D., Nandi, A. K., & Hijazin, I. (2012). Construction of optimum composite Field architecture for compact high-throughput AES S-boxes. IEEE VLSI Syst., 20(6), 1151–1155.

[22] Bui, D., Puschini, D., Bacles-Min, S., Beigne, E. & Tran, X. (2017) AES Datapath optimization strategies for low-power low- energy multisecurity-level internet-of-things applications. IEEE VLSI Syst., 25(12), 3281–3290.

[23] El-Meligy, N., Anin, M., Yahya, E., & Ismail, Y. (2017) 130-nm low power asynchronous AES Core. In Proc. 2017 IEEE ISCAS, Baltimore, pp. 1–4.

[24] Kumar K, Kumar K.R., Kaur A. (2021) A lightweight AES algorithm implementation for encrypting voice messages using field programmable gate arrays, Journal of King Saud University,

Computer and Information Sciences https://doi.org/10.1016/j.jksuci.2020.08.005

[25] Nabihah Ahmad, S.M.Rezaul Hasan, (2021) "A new ASIC implementation of an advanced encryption standard (AES) crypto-hardware accelerator", Microelectronics Journal 117 105255

[26] Shahbazi K., Seok Bum Ko (2021) Area-Efficient Nano-AES implementation for Internet-of-Things Devices, IEEE Transactions on Very Large-Scale Integration (VLSI) systems, 29(1), 136-148.

[27] Mohsen Jahanbani, Nasour Bagheri, Zeinolabedin Norozi, (2020) "Lightweight implementation of SILC, CLOC, AES-JAMBU and COLM authenticated ciphers", Microprocessors and Microsystems 72 102925

[28] Eduardo Mobilon, Dalton Soares Arantes, (2021) "100 Gbit/s AES-GCM Cryptography Engine for Optical Transport Network Systems: Architecture, Design and 40 nm Silicon Prototyping", Microelectronics Journal 116 105229

[29] M. Maazouz, A. Toubal, B. Bengherbia et al., (2021) FPGA implementation of a chaos-based image encryption algorithm, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2021.12.022

[30] Ali Murat Garipcan, Ebubekir Erdem, (2021) "Design, FPGA implementation and statistical analysis of a high-speed and low-area TRNG based on an AES s-box post-processing technique", ISA Transactions 117 Page 160–171

[31] Yogendra Sao, Sk Subidh Ali, Dipojjwal Ray, Siddharth Singh, Santosh Biswas, (2021) "Co-relation scan attack analysis (COSAA) on AES: A comprehensive approach", Microelectronics Reliability 123 114216

[32] Saeideh Sheikhpour , Seok-Bum Ko, Ali Mahani, (2021) "A low cost fault-attack resilient AES for IoT applications", Microelectronics Reliability 123 114202

[33] Xinfei Guo, Mohamed El-Hadedy, Sergiu Mosanu, Xiangdong Wei, Kevin Skadron, Mircea R. Stan, (2022) "Agile-AES: Implementation of configurable AES primitive with agile design approach", INTEGRATION, the VLSI journal 85 87–96

[34] Pietro Nannipieri, Stefano Di Matteo, Luca Baldanzi, Luca Crocetti, Luca Zulberti, Sergio Saponara and Luca Fanucci, (2022) "VLSI Design of Advanced-Features AES Crypto-processor in the Framework of the European Processor Initiative", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 30, No. 2, February Page 177-186

[35] Shailaja, A. and Gorappa Ningappa, K. (2021) 'A low area VLSI implementation of extended tiny encryption algorithm using Lorenz chaotic system', *Int. J. Information and Computer Security*, Vol. 14, No. 1, pp.3–19.

[36] D.-S. Kundi, Arshad Aziz, Nassar Ikram, A high performance ST-Box based unified AES encryption/decryption architecture on FPGA, Microprocessors and Microsystems 41 (2016) 37–46

[37] Yuan Zheng, Wang Y., Li J., Li R. and Zhao W., (2011) FPGA based optimization for masked AES implementation, *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2011, 1-4. doi: 10.1109/MWSCAS.2011.6026388.

[38] Kamoun N., Bossuet L. and Ghazel A. (2008) SRAM-FPGA implementation of masked S-Box based DPA countermeasure for AES, in IDT 2008. 74 – 77.

[39] Mentens N., Batina L., Preneel B. and Verbauwhede I. (2004) An FPGA implementation of Rijndael: trade-offs for side-channel security, Programmable Devices and Systems (IFAC 04), 2004, pp. 493-498.

[40] Ordu L. and Ors B., (2007) Power analysis resistant hardware implementations of AES, 14th IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2007, Marrakech, Morocco, December 11-14, 2007. IEEE 2007, ISBN 978-1-4244-1377-5, 1408-1411.

[41] Raed Bani-Hani, Khaldoon Mhaidat, Salah Harb, (2016) Very compact and efficient 32-bit aes core design using fpgas for small-footprint low-power embedded applications, J. Circuits Syst. Comput. 25 (07) (2016) 1650080.

[42] Wei Zhao, Yi Wang, Renfa Li (2012), A unified architecture for dpa-resistant present, in: 2012 International Conference on Innovations in Information Technology (IIT), IEEE, 2012, pp. 244–248.

[43] Srivatsan Subramanian, Mehran Mozaffari-Kermani, Reza Azarderakhsh, Mehrdad Nojoumian, Reliable hardware architectures for cryptographic block ciphers led and hight, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 36 (10) (2017) 1750–1758.

[44] Prashant Ahir, Mehran Mozaffari-Kermani, Reza Azarderakhsh, Lightweight architectures for reliable and fault detection simon and speck cryptographic algorithms on fpga, ACM Trans. Embed. Comput. Syst. 16 (4) (2017) 1–17.

[45] Kai Tian, Fault-resilient Lightweight Cryptographic Block Ciphers for Secure Embedded Systems, Thesis, 2014.

[46] Dhanda SS, Singh B, Jindal P (2020) Lightweight Cryptography: A solution to secure IoT, Wireless Personal Communication, 112(3), 1947-1980.

[47] Zodpe H., Sakpal A. (2020) An efficient AES implementation using FPGA with enhanced security features", Journal of King Saud University, Engineering Sciences 32, 115-122.

[48] Sumit Singh Dhanda, Brahmjit Singh, Poonam Jindal & Deepak Panwar (2022) A highly efficient FPGA implementation of AES for high throughput IoT applications, Journal of Discrete Mathematical Sciences and Cryptography, 25:7, 2029-2038, DOI: 10.1080/09720529.2022.2133242

Sumit Singh Dhanda received B.Tech and M.Tech degrees in Electronics and Communication Engineering from Kurukshetra University, Kurukshetra in 2005 and 2011 respectively. He has teaching experience of 10 years and currently pursuing his Doctoral Degree with ECE Department at National Institute of Technology, Kurukshetra, India. He has published 10 research papers in International/National conferences. His research interests include security algorithms for Internet of Things and wireless and mobile communication.

Brahmjit Singh Brahmjit Singh as completed Bachelor of Engineering in Electronics Engineering from Malaviya National Institute of Technology, Jaipur, Master of Engineering with specialization in Microwave and Radar from Indian Institute of Technology, Roorkee and Ph.D. degree from GGS Indraprastha University, Delhi. He is with the Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra working as Professor having 24 years of teaching and research experience. He is currently serving as Dean P&D and Regional Coordinator, Regional Academic Centre for Space at NIT Kurukshetra.

He has held several administrative and academic positions in NIT Kurukshetra which include Chairman ECE Department, Chairman Computer Engineering Department, Professor in-Charge Centre of Computing and Networking, and Member Planning and Development Board. He was also incharge of Siemens Centre of Excellence at NIT Kurukshetra. He has published 100 research papers in International/National Journals and conferences, organized several conferences and short-term courses. His current research interests include Wireless Sensor Networks, Cognitive Radio, and Security Algorithms for Wireless Networks and Mobility Management in wireless networks and planning & designing of Mobile Cellular Networks. He has been awarded The Best Research Paper Award on behalf of 'The Institution of Engineers (India)'. He is the member of IEEE, Life member of IETE, and Life Member of ISTE.

Poonam Jindal Poonam Jindal received B.E degree in Electronics and Communication Engineering from Punjab Technical University, Punjab in 2003, M.E degree in Electronics and Communication Engineering from Thapar University, Patiala in 2005 (India). She is working as Assistant Professor with Electronics and Communication Engineering Department, National Institute of Technology, Kurukshetra, India and completed her Doctoral Degree at National Institute of Technology, Kurukshetra, India. She has published 50 research papers in International/National journals and conferences. Her research interests include security algorithms for wireless networks and mobile communication. She is a member of IEEE.