TELKOMNIKA, Vol.15, No.1, March 2017, pp. 494~503
ISSN: 1693-6930, accredited **A** by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v15i1.4713

High throughput FPGA Implementation of Advanced Encryption Standard Algorithm

Soufiane Oukili[1], Seddik Bri[2]
Materials and Instrumentation (MIN), High School of Technology, Moulay Ismail University, Meknes, Morocco.
Corresponding author, e-mail: soufiane.oukili@gmail.com[1], briseddik@gmail.com[2]

Abstract

The growth of computer systems and electronic communications and transactions has meant that the need for effective security and reliability of data communication, processing and storage is more important than ever. In this context, cryptography is a high priority research area in engineering. The Advanced Encryption Standard (AES) is a symmetric-key criptographic algorithm for protecting sensitive information and is one of the most widely secure and used algorithm today. High-throughput, low power and compactness have always been topic of interest for implementing this type of algorithm. In this paper, we are interested on the development of high throughput architecture and implementation of AES algorithm, using the least amount of hardware possible. We have adopted a pipeline approach in order to reduce the critical path and achieve competitive performances in terms of throughput and efficiency. This approach is effectively tested on the AES S-Box substitution. The latter is a complex transformation and the key point to improve architecture performances. Considering the high delay and hardware required for this transformation, we proposed 7-stage pipelined S-box by using composite field in order to deal with the critical path and the occupied area resources. In addition, efficient AES key expansion architecture suitable for our proposed pipelined AES is presented. The implementation had been successfully done on Virtex-5 XC5VLX85 and Virtex-6 XC6VLX75T Field Programmable Gate Array (FPGA) devices using Xilinx ISE v14.7. Our AES design achieved a data encryption rate of 108.69 Gbps and used only 6361 slices ressource. Compared to the best previous work, this implementation improves data throughput by 5.6% and reduces the used slices to 77.69%.

Keywords*: security, advanced encryption standard, high throughput, pipeline, S-box, FPGA*

1. Introduction

Cryptography is the study of methods for transmitting data in confidential manner. It is essentially based on mathematical, computer and physical concepts, which study the set of techniques to encrypt information and make it unintelligible except for its recipient. Cryptography encompasses many problems: encryption, authentication and key distribution to name a few. An encryption algorithm, or cipher, ensures transforming plaintext into ciphertext under the control of a secret or public key [1, 2]. The Data Encryption Standard (DES) was the first modern secret key algorithm. It had been developed at IBM in 1970s and adopted as a Federal Information Processing Standard (FIPS) by the National Institute of Standards and Technology (NIST) in 1977 [3]. In order to replace the DES, which its short key and especially the small size of its blocks made it insecure with regard to the technological advances and the volume of data to be secured, the NIST announced a competition for a new encryption algorithm. After all reviews, an algorithm known as Rijndael, developed by two Belgian cryptographers: Dr. Joan Daemen and Dr. Vincent Rijmen, had been selected. In November 2001, the Advanced Encryption Standard (standardized version of Rijndael) became a FIPS standard (FIPS-197) [4, 5].

High throughput, low power, and compactness have always been topic of interest for AES software and hardware implementation. As compared to software implementation, hardware implementation provides greater physical security and higher speed [6, 7]. The main goal of this paper is to implement high throughput AES design in FPGA device, using as less hardware as possible.

S-box substitution is the only non-linear transformation in the four transforms of AES arithmetic and is the key point to improve architecture performances. The most traditionally S-box implementation uses various kinds of memories such as ROMs, BRAMs, and LUTs to store

Received September 1, 2016; Revised October 18, 2016; Accepted November 21, 2016

all predefined 256 8-bit values. Unfortunately, it suffers from an unbreakable delay of memories that leads to a reduction in throughput [8, 9]. S-box can be implemented using normal basis in composite field arithmetic, where it is possible to use pipelining and sub-pipelining techniques in order to decrease the critical path delay and increase the throughput [10, 11].

In this paper, we present efficient high throughput architecture and implementation of 128-bit key AES algorithm. We have adopted pipeline approach who modifies the critical path by increasing the operating frequency. It consists of parallelizing input/output data with the processing. Consequently, the algorithm is divided into stages and pipeline registers are placed. By incrementing the number of these stages, the critical path and the clock pulse width of the system can be decreased and as a result the throughput is increased. We have inserted pipeline registers before each AES round and even between round operations to further reduce the critical path. In addition, we have proposed 7-stage pipelined S-box based on composite field, in which field operations are implemented in lower order fields and by lower cost subfield operations, in order to avoid the unbreakable delay of memories and to achieve any further increase in processing speed. Moreover, efficient AES key expansion architecture suitable for pipelined AES is presented. Our proposed AES architecture was implemented on Xilinx Virtex-5 and Virtex-6 FPGA devices. The FPGAs offer the advantage of hardware speed and software flexibility and programmability.

This paper is organized as follows. Section 2 presents a brief background of AES algorithm. Section 3 reviews relevant works of various authors reported in stat-of-art. Section 4 presents our proposed AES architecture. Results and comparisons between our implementation and different reported approaches are provided in Section 5. Finally, conclusion and references are given respectively.

2. Background of AES Algorithm

AES algorithm is a symmetric block cipher, in which both the sender and receiver use a same key for encryption and decryption. The size of the input block is fixed to be 128 bits, regardless of the key size, which can be 128, 192, or 256 bits. Like most block ciphers, the global function is an iteration of rounds. The number of rounds, respectively 10, 12 and 14, depends on the key size. For this paper, 128-bit key is chosen, which requires 10 rounds of encryption. The 128-bit data block is processed internally on a two-dimensional byte table called "State". It consists of a matrix of four rows and four columns of bytes. Each round executs four different byte-oriented transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey, except for the last round, in which MixColumn transformation is not performed. Apart from this, there is an initial round at the start that consists of only AddRoundKey transformation. For each round, 128-bit input data and 128-bit key are required and the output serves as input for the next one [4]. Figure 1 shows the 128 bit-key AES algorithm.

a. **SubBytes:** operates independently on each byte of the state using an alternative S-Box table. The S-Box is built by the composition of two transformations: multiplicative inverse over GF (2^8) and combining the inverse function with an invertible affine transformation.
b. **Shiftrows:** changes the position of bytes in the state by left rotating each row by its index in order to give a new state table.
c. **MixColumns:** multiplies each column of the state by a fixed polynomial, so that each input byte affects all four output bytes.
d. **AddRoundKey:** adds the corresponding round key to the current state matrix. It performs a bit by bit XOR between the two elements.

The decryption structure of AES algorithm can be derived by inverting the encryption one directly and the operations of rounds are replaced by their inverse (InvSubBytes, InvShiftRows, InvMixColumns).

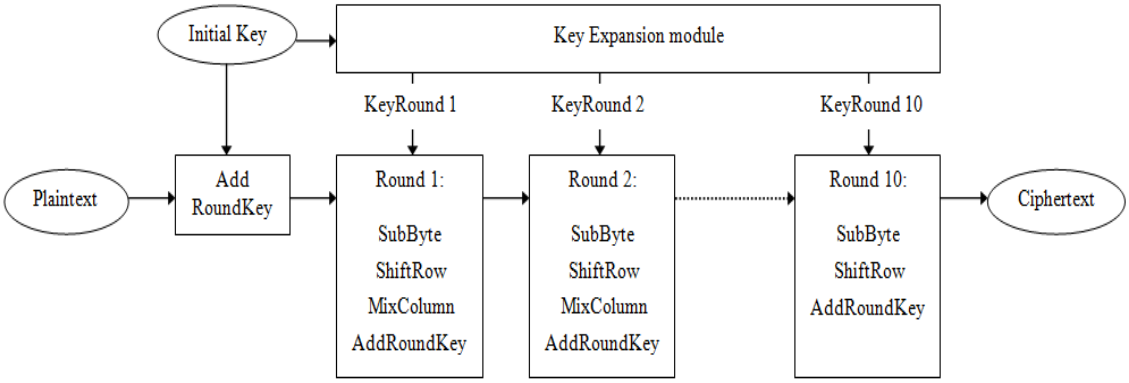

Figure 1. 128-bit key AES Algorithm

AES algorithm takes the main key and executs an expansion function to generate the round keys. This function is an iterative algorithm with the same round numbers as AES encryption and the output of each round represents the entry of the next one. In case of 128-bit key AES, it generates a total of 11 16-byte round key, taking into account that the first one represents the main key. At each round, the first four bytes of the input constitute the word w0, the next four bytes, the word w1, and so on. The final word bytes (w4) are left rotated by one position, and then each byte goes through the SubWord (S-box) substitution function. The result is XORed with an RCon constant dependent on the round number. Finally, the words are added to generate a new 128-bit round key. The key expansion is designed to be resistant to known cryptanalytic attacks, the inclusion of a round-dependent constant eliminates the symmetry and similarity between the ways round keys are generated [4]. Figure 2 shows one round of key expansion module.

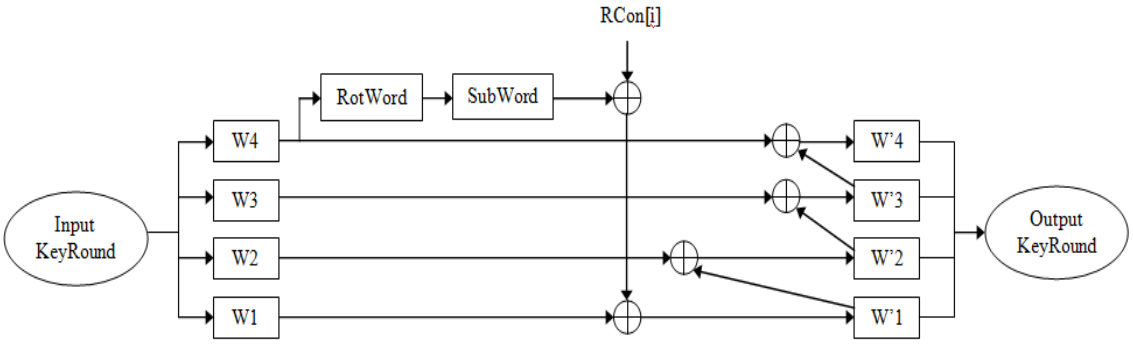

Figure 2. Round i of Key Expansion Module

3. Previous Works

AES algorithm can be implemented using different methods and contributions which can be categorized as follows. In the first one, loop unrolling and iterative techniques are used to increase throughput to area ratio and decrease area cost [12, 13]. The second category includes the designs which use pipelining and full-pipelining techniques to increase operational frequency and throughput [6], [14]. Many scientists have reported their development of AES architectures in order to achieve optimum performances in terms of throughput, efficiency and area ressources.

Rais and Qasim [15] proposed an iterative AES architecture using high performance S-Box design based on reduced residue of prime numbers. Their goal was to use a novel S-Box based on LUT whose entries are set as residue of prime number. Shakil et al [16] presented an

implementation of AES-XTS on FPGA using memory based pipelined design. AES-XTS is designed for storage devices in order to protect them against cryptanalyst attacks. Sireesha et al [17] proposed AES FPGA implementation in which they use Dual-Port RAMs memory for storing the results of the operations and Digital Clock Manager to optimize the execution time and reduce design area. Jyrwa and Paily [18] developed an optimized iterative design for the AES algorithm with 128-bit keys. The S-box was generated using composit field to ensure high throughput and less area resources. In addition, each clock cycle was optimized to incorporate maximum number of operations. Gielata et al [19] proposed custom AES pipelined architecture in reconfigurable hardware in roder to achieve maximum speed and efficiency. The transformations were optimized in term of executing time and implemented as combinational logic. José et al [20] presented new methodology employing dynamic and partial reconfiguration with parallelism and pipelining to implement AES. This methodology combines the use of two hardware languages (Handel-C and VHDL) to achieve a very high-throughput implementation. Fan et al [21] proposed high throughput AES implementation with hardware sharing functional blocks. Efficient low-cost AddRoundKey architecture was used for real-time key generations and SubBytes transformation was implemented based on content-addressable memory (CAM) scheme to achieve high speed. Rizk et al [22] explored the tradeoffs of speed versus area in security processor design. Two implementations of the AES algorithm were introduced; the first one was based on the basic architecture of the AES and the second one on the sub-pipelined architecture. Banu et al [23] proposed high throughput hardware and software implementation of the AES algorithm. The hardware implementation was based on architectural optimization techniques like pipelining, loop unrolling and iterative design to increase the speed by processing multiple rounds simultaneously. Software implementation explored parallelization techniques with OpenMP to increase the speedup. Kaur et al [24] proposed an efficient FPGA implementation of AES in which S-box transformation was implemented as a look-up table (LUT). Issam et al [25] presented an efficient architecture for high-speed AES, where multistage sub-pipelined architecture was used to reach higher efficiency in terms of throughput and area. The S-box transformation is implemented using composite field arithmetic by merging and location rearrangement of different operations required in the steps of encryption and allowing higher efficiency. Samiee et al [26] proposed hardware AES implementation in wich they introduced new pipelined S-box architecture, for further time savings and higher throughput with highly efficiency in terms of area. Nalini et al [27] presented design and implementation of highly efficient Iterative and pipelined AES architectures. The iterative design was optimized for area and the pipelined one for speed. Naiem et al [28] proposed an area optimized design for AES in CBC mode where one round was implemented in order to reduce the required area and the latency. Mostafa et al [29] presented FastCrypto architecture which extended a general-purpose processor with an AES crypto coprocessor for encrypting/decrypting data with high throughput. In addition, they studied the trade-offs between FastCrypto performance and design parameters, including the number of stages per round and the number of parallel AES pipelines. Reza et al [30] presented a high throughput implementation of the AES based on the 2-slow retiming optimization technique in order to break the critical path of the design and improve the throughput. Shanxin et al [31] proposed high throughput pipelined AES architecture working on CTR mode by inserting registers in appropriate points and making the delay shortest. Rahimunnisa et al [32] presented high throughput AES architecture for hardware implementation targeted for low-cost embedded applications. It introduced parallel operation in the folded architecture to reach better throughput. Wang et al [33] proposed high-throughput masked AES architecture in order to protect data from differential power analysis attacks. They adopted unrolling technique and used FPGA block RAM to reduce hardware resources. Anwar et al [34] presented a crypto processor using a fully pipelined AES algorithm integrated with a 32-bit general purpose 5-stage pipelined MIPS processor. Rahimunnisa et al [35] presented an effective Parallel Sub-Pipelined AES architecture to achieve high throughput. Abolfazl et al [36] proposed three high-throughput AES implementations in ECB mode and one ultra-high throughput AES implementation in CTR mode. They used loop-unrolling, fully pipelining, and fully sub-pipelining techniques to increase throughput and performed area-delay efficient multiplier and multiplicative inverter over GF (2^8).

4. Proposed High Throughput AES Algorithm

4.1. Proposed AES Architecture

In this paper, we propose AES architecture that aime to achieve high throughput and use hardware resource as less as possible. Therefore, pipelining strategy is adopted. The pipelining divides the design into stages and registers are inserted in order to parallelize the data inputs and outputs with the processing. The critical path of the system can be decreased by incrementing the number of these stages and as a result the speed is increased. In order to achieve an area-throughput efficient design, it must be divided into optimum stages and inserting registers in appropriate placements. The proposed full-pipelining AES architecture is divided into 10 stages, where each round represents a stage. In addition, each operation of round is considered as stage. Therefore pipeline registers are introduced between rounds and between the operations of rounds. Proposed full-pipelining general block and round of AES are illustrated in Figure 3(a) and Figure 3(b), respectively.

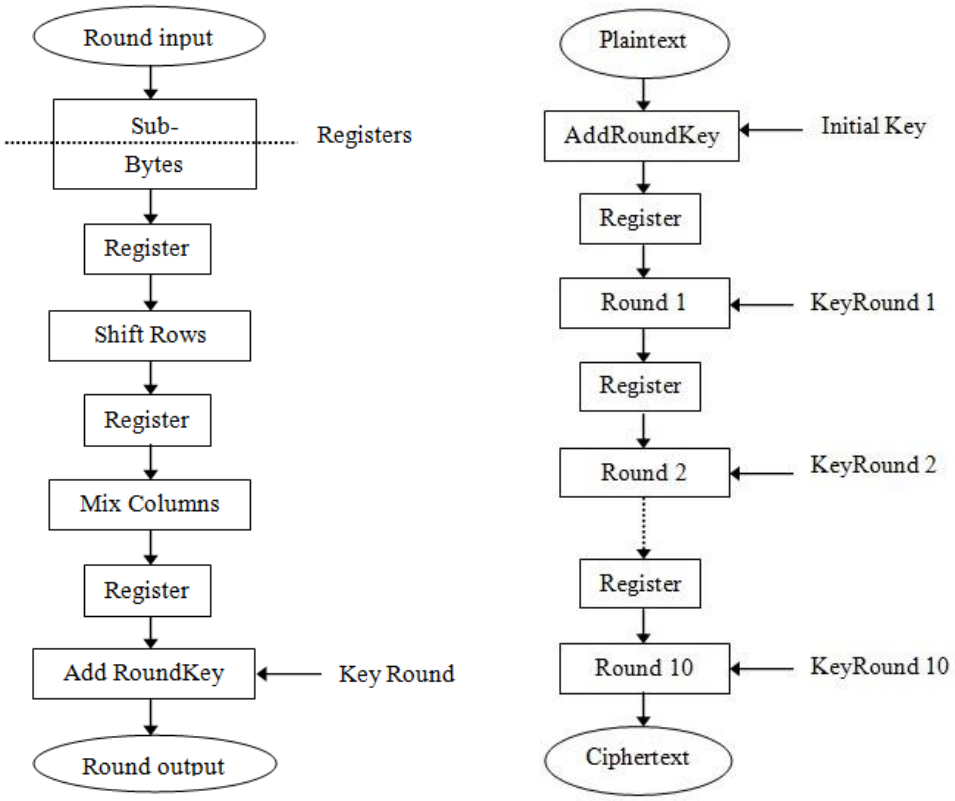

Figure 3. (a) General Block of Full-Pipelining AES (b) Full-Pipelining Round of AES

S-box substitution is the only non-linear transformation in the four transforms of AES algorithm and is the key point to improve the performance of the architecture. The traditionally implementation of the S-box is to have the all pre-computed 256 8-bit values stored in a ROM based LUT or in block of RAM memories. Using pre-computed S-box in high speed applications requires a high volume of gates and prohibit the architecture from being divided into more than stage in order to break the critical path delay and achieve any further speedup. Therefore, S-box using composite field is adopted according to Satoh et al [37], where the field operation is implemented in lower order fields and by lower cost subfield operations. This S-box structure has the advantage of being able to be pipelined in order to achieve high throughput and it occupies small area. To design efficient high speed pipelined S-box, we first place pipeline registers in all possible placements. Then, we remove the pipeline stage which contains the

lowest reduction of frequency until we achieve a high speed and low area architecture. Proposed 7-stage pipelining S-box using composite field is shown in Figure 4. The proposed AES round is divided into 10 stages. Thus, the proposed AES algorithm is divided into 100 stages.

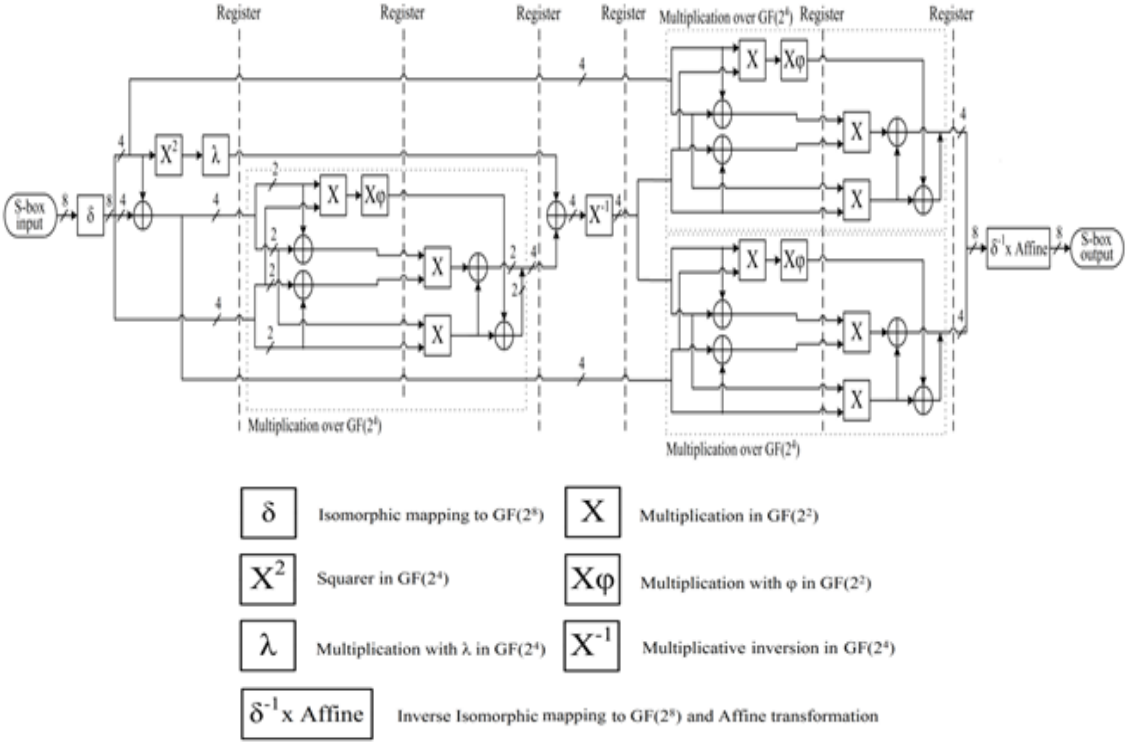

Figure 4. Proposed 7-Stage S-box Architecture Using Composite Field

4.2. Proposed Key Expansion Module

Key expansion module is responsible of key rounds generation and is one of the crucial steps in the realization of AES and decides the throughput of the cipher process. Round keys can be generated on the fly or generated beforehand and stored in memories. In order to preserve the advantage of high throughput of our pipelined encryption process, we propose pipelined key expansion module, where key rounds are generated as at the same time as the encryption process. It must be considered that the key expansion architecture must be divided the same as the number of existent stages in encryption round unit or less, to provide round key to the corresponding encryption round.

Our proposed key expansion module is divided into 9 sub-stages. Knowing that the encryption round unit contains 10. The S-box design used is the one previously proposed, 7-stage based on composite field. Figure 5 presents the proposed key expansion round i.

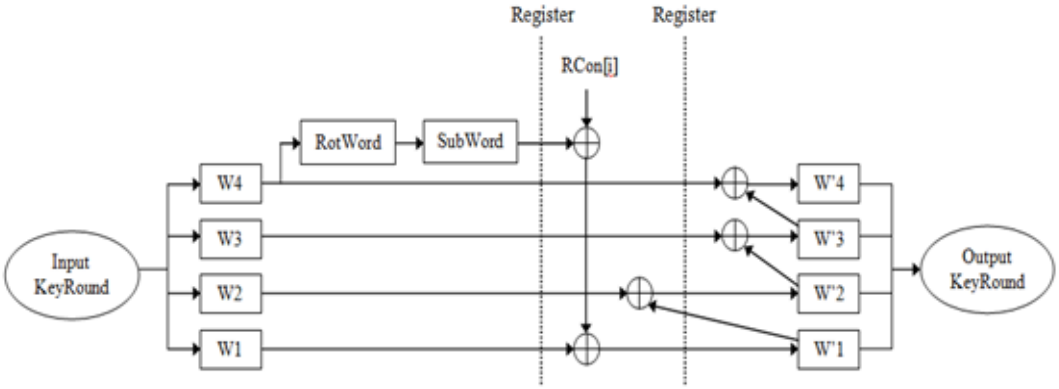

Figure 5. Proposed Key Expansion Round i

5. Results and Comparisons

Our proposed AES architecture was implemented on Virtex-5 XC5VLX85 and Virtex-6 XC6VLX200 FPGAs, which are widely used in recent related works. We employed Xilinx ISE Design Suite v14.7 for synthesis and simulation. The design was coded using VHDL language and it takes 100 clock pulse cycles to cipher the first data block, then the cipher blocks are recovered at each clock cycle. The implementation on virtex-5 occupied 7385 (56%) slices and achieved a frequency and throughput of 638.162 MHz, 81.68 Gbps. On virtex-6, it occupied 6361 (16%) slices and achieved a frequency and throughput of 849.185 MHz, 108.69 Gbps. We employ well- known equations (1) and (2), to calculate the throughput and the efficiency, respectively.

$$Throughput = \frac{Number\ of\ outputted\ bits}{Delay\ of\ the\ critical\ path} \quad (1)$$

$$Efficiency = \frac{Throughput}{Area\ (slices)} \quad (2)$$

Numerous hardware implementations for the AES algorithm have been reported in literature, which aim to achieve the most efficient architecture, by improving high throughput and area efficiency. Table 1 shows performances for recent FPGA-based AES designs up to our knowledge. It provides values of hardware utilization, throughput, maximum frequency and efficiency.

Table 1. Maximum Frequency, Throughput and Hardware Utilization Results

Design	Device	Slices	Critical delay (ns)	Max-freq. (MHz)	Throughput (Gbps)	Efficiency (Mbps/slice)
[15]	Virtex-5 XC5VLX50	1745	-	242.15	3.09	-
[16]	Virtex-5 XC5VLX50	573+8 BRAMs	-	-	5.25	9.16
[17]	Spartan-3 XC3S500E	326+3 BRAMs	-	50	6.4	-
[18]	Virtex-2 Pro XC2VP30	6211+1 BRAM	-	142.5	18.2	0.2347
[19]	Virtex-4 XC4VLX200	1209	-	165	21.2	-
[20]	Virtex-2 XC2V6000	3576+80 BRAMs	5.1	194.7	24.92	6.97
[21]	Virtex-2 XC2V3000	139357	4.5	222.2	28.40	0.20
[22]	Virtex-4 VLX60FF668	18855+200 BRAMs	-	-	28.510	1.512
[23]	Virtex-5 XC5VLX110T	-	4	250	31.25	-
[24]	Virtex-2 XC2VP30	1127	4	247.365	31.66	-
[25]	Virtex-2XC2V6000	10662	-	305.1	39.05	3.6
[26]	Virtex-2 Pro XC2VP20	7865	2.9	341.53	43.71	5.55
[27]	Virtex-2 Pro XC2VP30	12556+100 BRAMs	2.6	373	47.74	3.8
[28]	Virtex-2 Pro XC2VP30	1835+40 BRAMs	2.4	405.227	51.87	28.27
[29]	Virtex-5 XC5VLX50T	1656	1.7	557	70	-
[30]	Virtex-4 XC4VLX200	3425	1.73	576.037	73.73	21.53
[31]	Virtex-5 XC5VLX85	22994	1.7	576.07	73.73	3.21
Our design	Virtex-5 XC5VLX85	7385	1.56	638.162	81.68	11.06

From Table 1, the highest throughput and the highest frequency reported to our knowledge by virtex5 FPGA are 73.737 Gbps and 576.07 MHz respectively, using 22994 slices [31]. By comparing these results with our implementation on virtex5, we notice that ours achieves 1.107 times more throughput and only 0.321 times slices used. Therefore, it improves data throughput by 10.78%. Furthermore, it is 3.44 times more efficient.

In Table 2, we give the synthesis results of AES implementations on Virtex-6 FPGA. These research works achieved the highest throughput among the others. We had implemented our proposed AES design on Virtex-6 FPGA too in order to hold a fair comparison. The implementation achieves 1.056 times more throughput, 4.74 times more efficiency and only 0.223 times slices used than the implementation in Abolfazl el al [36], which achieves the highest throughput reported. Hence, we improve data throughput by 5.6%.

Table 2. AES Implementations on Virtex-6

Design	Device	Slices	Critical delay (ns)	Max-freq. (MHz)	Throughput (Gbps)	Efficiency (Mbps/slice)
[32]	Virtex-6 XC6VLX75T	2056+48 BRAMs	1.9	505.5	37.1	15.56
[33]	Virtex-6 XC6VLX240T	9071+400 BRAMs	3.1	319.29	40.86	4.51
[34]	Virtex-6 ML605	2547+204 BRAMs	1.8	553	58	-
[35]	Virtex-6 XC6VLX75T	2597	2.2	450.045	59.59	22.94
[36]	Virtex-6 XC6VLX240T	28520	1.2	803.988	102.91	3.6
Our design	Virtex-6 XC6VLX240T	6361	1.17	849.185	108.69	17.08

By considering all reported results in this section, we notice that our proposed AES implementations improve performances in terms of throughput, efficiency and surface area compared to all reported ones, known to date. This is due to the following facts:

a. Using full-pipelining technique in order to achieve high throughput implementations.
b. Using pipelined S-box based on composite field.
c. Inserting pipeline registers in appropriate placements with optimum stage numbers.

6. Conclusion

This paper presents high throughput efficient 128-bit key AES implementation. Full-pipelining technique was introduced in order to increase the throughput than the basic AES structure. In addition, we had employed 7-stage pipelined S-box using composite field to achieve further speedup. Furthermore, efficient key expansion architecture adapted to the full-pipelined AES rounds is introduced. The proposed AES architecture was implemented on virtex-5 and virtex-6 FPGAs. It takes 100 clock cycles to load the first cipher block. Then, it will appear on consecutive clock cycles. Our AES implementation on virtex-6 improves throughput compared to the previousely reported ones, with consuming low area.

References

[1] Stalling W. *Editor*. Cryptography and Network Security Principles and Practices. 4th ed. New Jersey: Prentice Hall. 2005.
[2] Kahate A. *Editor*. Cryptography and Network Security. 2nd ed. New York: Tata McGraw Hill. 2007.
[3] Federal Information Processing Standards. Publication 46-3. *Data Encryption Standard*. Maryland. National Institute of Standards and Technology. 1999.
[4] Federal Information Processing Standards. Publication 197. *Advanced Encryption Standard.* Maryland. National Institute of Standards and Technology. 2001.
[5] Daemen J, Rijmen V. *AES Proposal: Rijndael*. National Institute of Standards and Technology. 1999. available at: http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf
[6] Yoo SM, Kotturi D, Pan DW, Blizzard J. An AES crypto chip using a high-speed parallel pipelined architecture. *Microprocessors and Microsystems*. 2005; 29(7): 317-326.

[7] Liu H, Zhou Y, Fan Y, Tsunoo Y, Goto S. Information Hiding for AES Core Based on Randomness. *Procedia Engineering*. 2011; 15: 2113-2117.
[8] Burns F, Murphy J, Koelmans A, Yakovlev A. Efficient advanced encryption standard implementation using lookup and normal basis. *IET Computers & Digital Techniques.* 2009; 3(3): 270-280.
[9] Sumathi M, Nirmala D, Rajkumar RI. Study of Data Security Algorithms using Verilog HDL. *International Journal of Electrical and Computer Engineering*. 2015; 5(5): 1092-1101.
[10] Wong MM, Wong MLD. *A High Throughput Low Power Compact AES S-box Implementation using Composite Field Arithmetic and Algebraic Normal Form Representation.* Proceedings of the 2nd Asia Symposium on Quality Electronic Design. Penang. 2010: 318-323.
[11] Rachh RR, AnandaMohan PV, Anami BS. *High Speed S-box architecture for Advanced Encryption Standard*. Proceedings of the IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application. Bangalore. 2011: 1-6.
[12] El Adib S, Raissouni N, Chahboun A, Azyat A, Lahraoua M, Ben Achhab N, Abbous A, Benarchid O. A Reconfigurable Cryptography Coprocessor RCC for Advanced Encryption Standard AES/Rijndael. *International Journal of Information & Network Security.* 2012; 1(2): 119-126.
[13] El Adib S, Raissouni N. AES Encryption Algorithm Hardware Implementation: Throughput and Area Comparison of 128, 192 and 256-bits Key. *International Journal of Reconfigurable and Embedded Systems.* 2012; 1(2): 67-74.
[14] Nalini C, Nagaraj, Anandmohan PV, Poornaiah DV, Kulkarni VD. *An FPGA Based Performance Analysis of Pipelining and Unrolling of AES Algorithm*. International Conference on Advanced Computing and Communications. Surathkal. 2006: 477-482.
[15] Rais MH, Qasim SM. *Fpga implementation of Rijndael algorithm using reduced residue of prime numbers*. Proceedings of the 4th International Design and Test Workshop. Riyadh. 2009: 1-4.
[16] Shakil A, Khairulmizam S, Abdul RR, Fakhrul ZR. An effective storage encryption solution. *Indian Journal of Science and Technology*. 2013; 6(4): 4384-4389.
[17] Sireesha K, Madhava SR. A novel approach of area optimized and pipelined FPGA implementation of AES encryption and decryption. *International Journal Scientific and Research Publications.* 2013; 3(9): 1-5.
[18] Jyrwa B, Paily R. *An Area-Throughput Efficient FPGA implementation of Block Cipher AES algorithm*, International Conference on Advances in Computing, Control, and Telecommunication Technologies. Kerala. 2009: 328-332.
[19] Gielata A, Russek P, Wiatr K. *AES hardware implementation in FPGA for algorithm acceleration purpose*. International Conference on Signals and Electronic Systems. Krakow. 2008: 137-140.
[20] José MGC, Miguel AVR, Juan MSP, Juan AGP. A new methodology to implement the AES algorithm using partial and dynamic reconfiguration. *Integration, the VLSI Journal.* 2010; 43(1): 72-80.
[21] Fan CP, Hwang JK. FPGA implementations of high throughput sequential and fully pipelined AES algorithm. *International Journal of Electrical Engineering*. 2008; 15(6): 447-455.
[22] Rizk MRM, Morsy M. *Optimized area and optimized speed hardware implementations of AES on FPGA*. Proceedings of the 2nd International Design and Test Workshop. Cairo. 2007: 207-217.
[23] Banu JS, Vanitha M, Vaideeswaran J, Subha S. *Loop parallelization and pipelining implementation of AES algorithm using OpenMP and FPGA*. International Conference on Emerging Trends in Computing, Communication and Nanotechnology. Tamilnadu. 2013: 481-485.
[24] Kaur A, Bhardwaj P, Kumar N. Fpga implementation of efficient hardware for the Advanced Encryption Standard. *International Journal of Innovative Technology and Exploring Engineering*. 2013; 2(3):186-189.
[25] Issam H, Kamal E, Ezz E. High-speed aes encryptor with efficient merging techniques. *IEEE Embedded Systems Letters*. 2010; 2(3): 67-71.
[26] Samiee H, Atani RE, Amindavar H. *A novel area-throughput optimized architecture for the AES algorithm*. International Conference on Electronic Devices, Systems and Applications. Kuala Lumpur. 2011: 29-32.
[27] Nalini I, Anandmohan PV, Poornaiah DV, Kulkarni VD. Efficient Hardware Architectures for AES on FPGA. In Das VV, Thankachan N. *Editors*. Computational Intelligence and Information Technology. New York: Springer-Verlag; 2011: 249-257.
[28] Naiem GF, Elramly S, Hasan BE, Shehata K. *An efficient implementation of CBC mode Rijndeal AES on an FPGA*. National Radio Science Conference. Tanta. 2008: 1-8.
[29] Mostafa IS, Ghada YA. Fpga implementation and performance evaluation of a high throughput crypto coprocessor. *Journal of Parallel and Distributed Computing*. 2011; 71(8): 1075-1084.
[30] Reza RF, Bahram R, Sayed MS. FPGA based fast and high-throughput 2-slow retiming 128-bit AES encryption algorithm. *Microelectronics Journal*. 2014; 45(8): 1014-1025.
[31] Shanxin Q, Guochu S, Yihong H, Zhigang G, Zongjue Q. *High Throughput, Pipelined Implementation of AES on FPGA*. International Symposium on Information Engineering and Electronic Commerce. Ternopil. 2009: 542-545.

[32] Rahimunnisa K, Karthigaikumar P, Rasheed S, Jayakumar J, SureshKumar S. FPGA implementation of AES algorithm for high throughput using folded parallel architecture. *Security and Communication Networks.* 2014; 7(11): 2225-2236.
[33] Wang Y, Ha Y. FPGA-based 40.9-gbits/s masked AES with area optimization for storage area network. *IEEE Transactions on Circuits and Systems- II: Express Briefs*. 2013; 60(1): 36-40.
[34] Anwar H, Daneshtalab M, Ebrahimi M, Plosila J, Tenhunen H. *FPGA implementation of AES-based crypto processor*. Proceedings of the 20th IEEE International Conference on Electronics, Circuits, and Systems. Abu Dhabi. 2013: 369-372.
[35] Rahimunnisa K, Karthigaikumar P, Christy NA, Kumar SS, Jayakumar J. Psp: parallel sub-pipelined architecture for high throughput AES on FPGA and ASIC. *Central European Journal of Computer Science*. 2013; 3(4): 173-186.
[36] Abolfazl S, Saeed S. An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA. *Microprocessors and Microsystems*. 2015; 39(7): 480-493.
[37] Satoh A, Morioka S, Takano K, Munetoh S. A Compact Rijndael Hardware Architecture with S-Box Optimization, In: Boyd C. *Editor*. Advances in Cryptology. New York: Springer-Verlag; 2001: 239-254.