INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

FPGA Implementation Of Highly Efficient Parallel Hardware Architecture For AES-GCM

Ms. Shilpashree M.C
Department of Electronics and Communication Engineering, B.G.S. Institute of Technology, B.G.Nagar, Mandya, India
Mrs. Prabhavathi K.
Department of Electronics and Communication Engineering, B.G.S. Institute of Technology, B.G.Nagar, Mandya, India

Abstract:

The Advanced Encryption Standard (AES) is a symmetric-key encryption algorithm and it provides standardized authentication by using Galois/Counter Mode (GCM). Hence it is utilized in various security-constrained applications. Many of the AES-GCM applications are power and resource constrained and requires efficient hardware implementations. In this paper, different Application-Specific Integrated Circuit (ASIC) architectures of building blocks of the AES-GCM algorithms are evaluated and optimized to identify the high-performance and low-power architectures for the AES-GCM. In AES, to obtain the least complexity S-box (Sub-Bytes), the formulations for the Galois Field (GF) subfield inversions in GF (2^4) are optimized. By conducting exhaustive simulations for the input transitions, the average and peak power consumptions of the AES S-boxes can be analysed by considering the switching activities, gate-level netlists, and parasitic information and the S-box realisation based on lookup tables (LUTs) could be area efficient when implemented utilizing the memory resources available on FPGAs. The proposed parallel method uses two GF (2^{128}) multipliers and it results high-throughput and low latency GCM hardware architectures which is suitable for high-performance applications.

Key words: *Advanced Encryption Standard, Galois/Counter mode, Galois Field, high performance, low power.*

1.Introduction

The Advanced Encryption Standard-Galois/Counter Mode (AES-GCM) provides authentication and confidentiality for sensitive data simultaneously. In AESGCM, data confidentiality is provided by the Advanced Encryption Standard (AES) [1]. The AES was accepted by the National Institute of Standards and Technology (NIST in 2001 as the replacement for the previous cryptographic standards. Since then, it has been included in wireless standards of Wi-Fi [2] and WiMAX [4] and many other

applications, ranging from the security of smart cards to the bitstream security mechanisms in FPGAs [4]. The authentication of the AES-GCM is provided by the Galois/Counter Mode (GCM) [6] using a universal hash function. The AES-GCM has been used for a number of applications such as the new LAN security standard WLAN 802.1ae (MACSec) [6] and Fibre Channel Security Protocols (FC-SP) [7]. Moreover, it has been utilized in a number of cores from industry, see, for example, [8], [9], and [10]. In addition, two AES-GCM software-based implementations have been presented in [11] and [12].

- The contributions of this project can be summerised as follows: In paper to achieve low cost, methods of integration and resource sharing are used in designing a very low-complexity architecture, especially in (inverse) byte substitution ((inv) SubBytes) modules and (inverse) mix column ((inv) MixColumn) modules.
- The paper proposed a method to integrate the AES encrypter and the AES decrypter into a full functional AES crypto-engine.
- This method can make it a very low-complexity architecture, especially in saving the hardware resource in implementing the AES (Inv) SubBytes module and (Inv)Mixcolumns module.
- Instead of implementing only one cryptographic algorithm (encryption or decryption) a novel on-the-fly key expansion design is also proposed for.128-, 192-, and 256-bit keys. This unified hardware can run both the original AES algorithm and the extended AES algorithm.
- The proposed architecture can provide up to 2 different AES block cipher schemes within a reasonable hardware cost. Data can the encrypted not only with secret keys and initial vectors, but also by different block ciphers during the communication.

Among the transformations in the AES encryption, the SubBytes (S-boxes) is the only nonlinear one, requiring the highest area and consuming much of the AES power [13]. Therefore, the performance metrics of the S-boxes affect those for the entire AES encryption significantly. For lowcomplexity implementations, the S-box can be realized using logic gates in composite fields. These S-boxes can also be pipelined for achieving high performance. On the other hand, the S-boxes based on lookup tables (LUTs) could be area efficient when implemented utilizing the memory resources available on FPGAs. In some previous works such as [14], [15], [16], one specific S-box and in[15], three reported S-boxes have been synthesized on application-specific integrated circuit (ASIC). However, exhaustive search has not been performed for all suitable composite fields to evaluate their performance metrics using the same technology. The hardware and timing complexities of different composite field S-boxes have been evaluated in terms of logic gates (software implementations have been performed). However, benchmarking the performance (including power consumptions through simulation-based approaches) of the S-boxes implementations on hardware platforms has not been performed in these works.

2.The AES-GCM

In this section, we present preliminaries for the AES-GCMlgorithm. In what follows, the AES (used for confidentiality)and the GCM (used for authentication) algorithms in the AES-GCM and their hardware architectures are presented.

2.1 The Advanced Encryption Standard

In the AES-GCM, only the AES encryption is utilized with the input and the output blocks of 128 bits. However, based on the security requirements, the key size could be determined

as AES-128 (with 10 rounds), AES-192 (with 12 rounds), or AES-256 (with 14 rounds) [2]. In the AES encryption, all therounds except for the last round have four transformations of Sub Bytes, Shift Rows, Mix Columns, and AddRoundKey. For the last round, Mix Columns is eliminated and only three transformations of Sub Bytes, Shift Rows, and AddRoundKey are used.

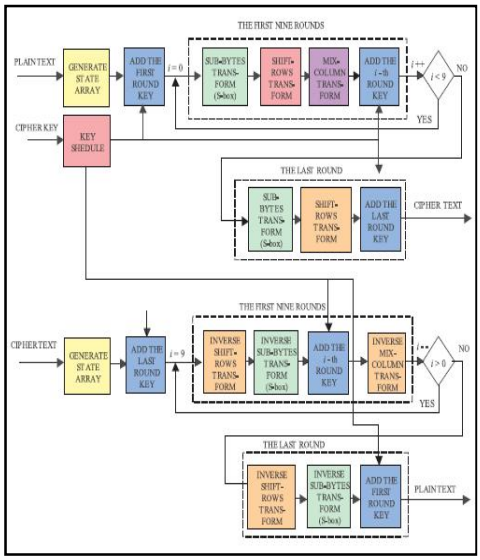

Figure 1: Structure of AES algorithm

The transformation Sub Bytes (S-boxes) is implemented by 16 S-boxes. In the S-box, each byte of the input state is substituted by a new byte. In Shift Rows, the first row of the state remains intact and the four bytes of the last three rows of the input state are cyclically shifted. In the Mix Columns transformation, each column is modified individually and in the final transformation, AddRoundKey, modulo-2 addition of the input state and the key of the corresponding round is performed [1].

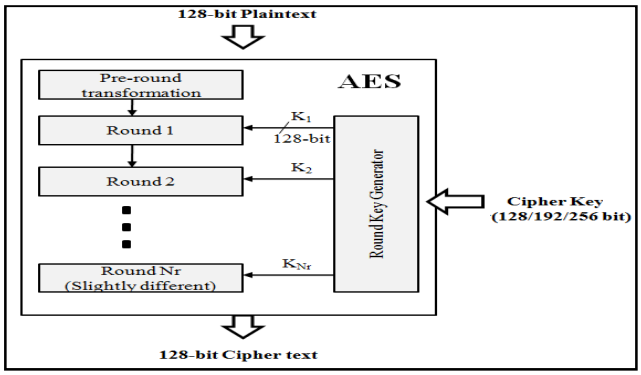

Figure 2: Block diagram of AES algorithm

3. High-Performance Gcm Parallel Architecture

In this section, we propose high-performance parallel architectures for the GCM. These architectures improve the throughput and the latency of the structures for $GHASH_H$. They also remove the need for consecutive $GF(2^{128})$ multiplications with for deriving [2].

We also derive the hardware implementations of the exponentiations of the hash sub key to the powers of two, i.e., in the form of H^{2j}, needing only XOR gates. Because of the low complexity of the implementations of these exponents, we take advantage of these low-cost hash sub key powers in

the proposed high-performance architectures. The powers in the form of H^{2j} are utilized to obtain the other powers of the hash sub key with the least number of GF multiplications over GF(2^{128}) for proposed architectures

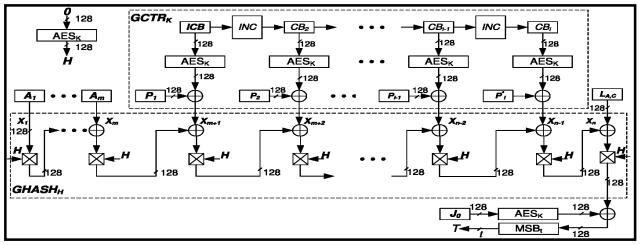

Figure 3: GCM architecture

3.1 High-Speed Structures For Hash Sub Key Powers

In the following, using squaring operations, we present three methods for implementing the hash subkey exponentiations. Using a complexity reduction algorithm, we also derive their hardware-optimized architectures. According to [6], it is less likely that the GCM is invoked with the same key on distinct sets of input data. Thus, a new hash subkey and its powers need to be obtained in each invocation. It is known that the squaring operation in binary extension fields leads to a linear structure. In other words, implementing squaring in hardware is less costly than GF($2^{128)}$ multiplications. For the GCM, the critical path delay of squaring is obtained as 3TX, where TX is the XOR gate delay. Moreover, it requires 202 XOR gates. To implement H^{2j}. one can cascade j squaring architectures or use a feedback for deriving them. We refrain using the feedback structure because of its low throughput and high latency. According to the hardware and timing complexities of squaring derived in this section, for H^{2j} , the cascade structure yields to the hardware and timing complexities of 202j XOR gates and3jTX, respectively. This leads to low-speed implementations which are not desirable in applications requiring high performance. It is possible to reduce the delay of the implementations of these exponentiations for the high performance hardware implementations.

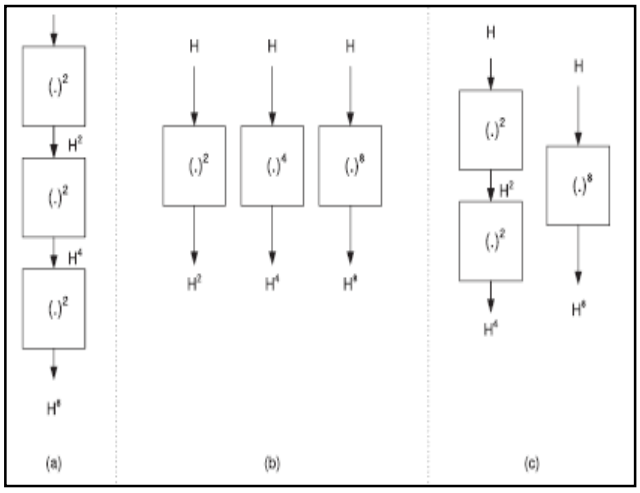

Figure 4:(a) Cascade, (b) parallel and
(c) hybrid realization methods for hash sub key powers

3.2 GF (2^{128}) Multiplier s For The GCM

In order to get high-speed AES-GCM core. It is noted that the considered GF (2^{128}) multipliers in these works include the Mastrovito multiplier with quadratic space complexity, the Karatsuba-Ofman multiplier, and the GF(2^{128}) multiplier . Based on proposed technology hardware and timing specifications, the performance of the GF(2^{128}) multipliers in six different steps for the Karatsuba-Ofman multipliers are considered. We denote these realizations by KO1 (for the case that only one step is performed) to KO6 (for which the 128-bit GF(2^{128}) multiplier is broken all the way to 2-bit multiplications using Karatsuba- Ofman method). Applying the Karatsuba-Ofman method recursively to obtain KOi, $2 < i < 6$ for the GCM would result in low-area implementations with higher timing complexities. As seen from this table, although the sub quadratic multiplier KO5 is the most compact implementation, the sub quadratic multiplier KO4 reaches the best efficiency.

We have considered the bit-parallel GF (2^{128}) multiplier which has quadratic hardware complexity. It is noted that this GF (2^{128}) multiplier has lower timing complexity compared to the sub quadratic hardware complexity GF (2^{128}) multipliers. However, we note that according to the latency of the proposed architectures i.e. increasing the number of parallel structures (q) results in having higher throughputs. On the other hand, having higher values for q increases the hardware complexities of $GHASH_H$. Therefore, for reducing the hardware complexity, using sub quadratic hardware complexity GF (2^{128}) multipliers is beneficial when high values of q are utilized.

For reducing the hardware complexity of the AES-GCM, efficient realization will be achieved by using of the Karatsuba- Ofman multiplier as the sub quadratic hardware complexity GF(2^{128}) multiplier.

4.AES-GCM Performance Comparisons

In this section, first, different AES architectures are presented and then we present and compare the ASIC synthesis results of the proposed and the previously presented architectures for the AES-GCM function.

We have presented different AES-128 architectures in Fig. 4. As seen in the AES simple loop structure (Fig. 4a), the AES rounds are executed serially (in the last round, Mix Columns is bypassed). This architecture is the most compact AES architecture and has been used in the literature, see, for instance, [15]. However, it suffers from low throughput. In Fig. 4(b), the AES unrolled pipelined structure is shown in which the pipeline stages are shown by dotted lines .As seen in this figure, 10 AES rounds are duplicated, with the last round without the Mix Columns transformation. Although this architecture needs 10 AES rounds to be implemented, it allows the designers to use pipelining and hence process multiple inputs sequentially for achieving high throughput. For further increasing the throughput, sub pipelining of the AES transformations can be used as depicted in Fig. 4(c). Sub pipelining is useful in increasing the frequency of the AES at the expense of more area used for the pipeline block cipher for the GCM proposed Algorithm for the GCM and utilize the parallel method in Fig. 3(b) for hash sub key exponentiations (hardware optimized through complexity reduction methods in the previous section).

Finally however, it increases the latency. For instance, the latency of a three-stage sub pipelined AES is three times more than that of the unrolled pipelined. We also note that if the critical path delay is determined by the multipliers in the GCM architecture, sub pipelining of the AES transformations cannot increase the frequency. Although both pipelined and sub pipelined AES architectures can be utilized, in this paper, for the syntheses and comparisons, we use pipelined AES architecture presented in Fig. 4(b).

Moreover, for analyzing the effect of sub pipelining, we have used sub pipelined AES for two AES-GCM architectures.

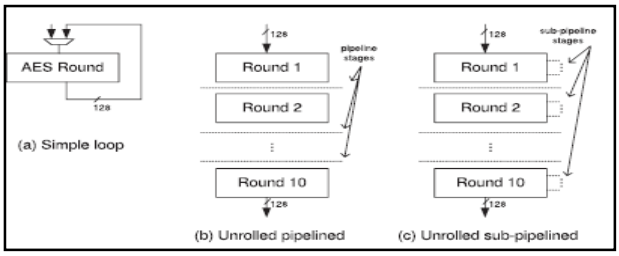

Figure 5: The AES-128 structure for (a) simple loop (b) unrolled pipe lined and (c) unrolled sub pipelined architectures (Mix columns is bypassed in last round)

5. The Proposed Architecture For The AES-GCM

Fig. 5 presents the proposed architecture for the AESGCM for q =8 parallel structures. The AES-128 pipeline registers are shown by dashed lines in Fig. 5. As seen in this fig. ten clock cycles are needed for obtaining the cipher text. After these first 10 clock cycles, the results are obtained after each clock cycle.

According to Fig. 5, eight parallel AES-128 structures are implemented as part of $GCTR_K$ to provide inputs to $GHASH_H$. As seen in this figure, the function $GCTR_K$ performs the AES counter mode with the Initial Counter Block and its one-increments (CBi). Moreover, q ¼ 8 increments (using INC 8 module) and the plaintext blocks (Pi) are used as the inputs. It is assumed that the data are encrypted and the authenticated in the GCM is 96 bits which is recommended for high-throughput implementations [6].

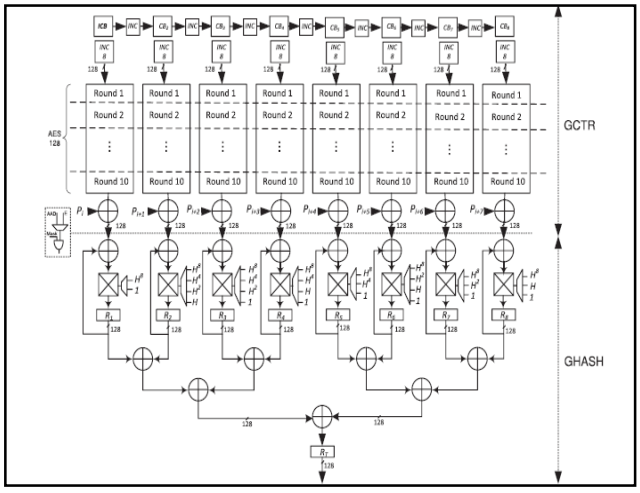

Figure 6: the proposed architecture for the AES-GCM

6.Implementation And Results

The architectures have been coded in VHDL as the design entry to the XILINX 12.2 with ISE simulator.. The proposed architectures can be implemented on SPARTAN 3 FPGA board, The syntheses are based on the case for q= 8 parallel addition-multiplications using the bit-parallel GF(2^8)multiplier. which has quadratic hardware complexity.

The simulation results for the AES is as shown in fig 6.The praposed parallel architecture for AES requirs 10 clock cycles for its operation ,after completion of 10 clock cycles the 'DONE' signal goes high.This shows that compare to sequential architecture this reuirs less number of clock cycles,hence this reduces the hardare delay.hence reduces the design complexity.while coding we used packages,this reduces the area 10 times lesser than the previous works.

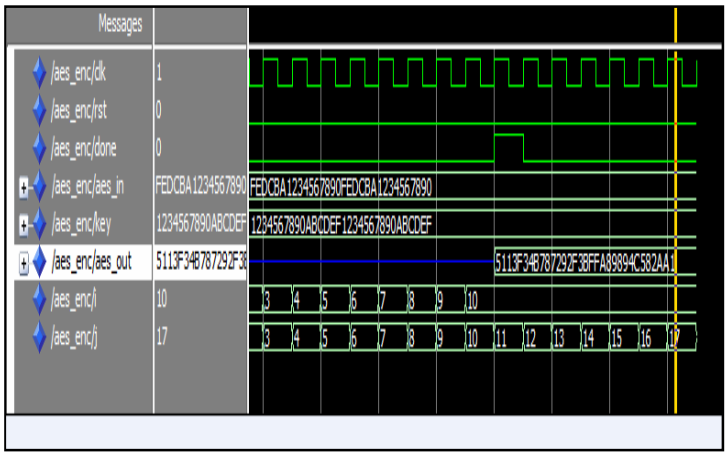

Figure 7: Simulation results for AES

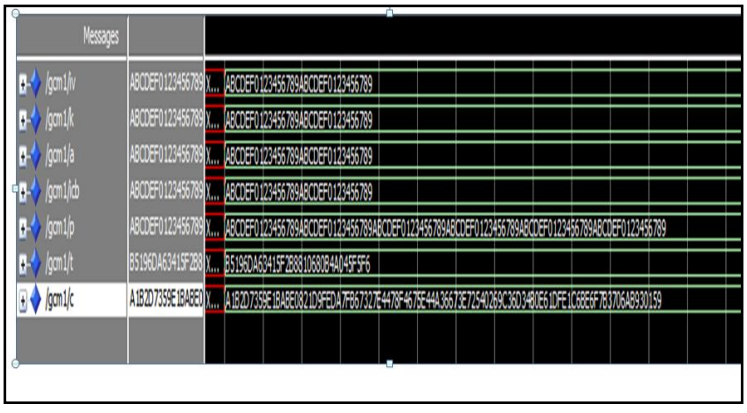

Figure 8: Simulation results for GCM.

7.Merits Of Proposed Architecture

- If it is implemented in the physical layer, the problem of hacking will be controlled efficiently
- AES is efficient than that of FIPS base paper
- H/W implementation on FPGA is easy.
- Because of LUT based a sub byte transformation is efficient when implemented using memory resources available on FPGA.
- Parallelism decreases the number of clock cycles, hence reduces the delay.
- The counter mode provides both H/W and S/W efficiency

8.Applications

The potential applications areas of proposed AES-GCM include the following

- In satellite communications for data transmission
- For secured remote access
- Fiber Channel Security
- For fast wireless LAN
- For ATM network security.

9.Conclusion

In this paper, we have obtained optimized building blocks for the AES-GCM to propose efficient and high-performance architectures. For the AES, through logic-gate minimizations for the inversion in GF (2^4), the areas of the S-boxes have been reduced. We have also evaluated and compared the performance of different S-boxes using an ASIC 65-nm CMOS technology. Furthermore, through exhaustive searches for the input patterns, we have performed simulation- based power derivations for different S-boxes to reach more accurate results compared to the statistical methods.

We have also proposed high-performance and efficient architectures for the GCM. For the case study of q = 8 parallel structures in $GHASH_H$, we have performed a hardware complexity reduction technique for the hash subkey exponentiations, having their timing complexities intact. The ASIC comparison results show that better efficiencies are achieved for the proposed architectures. Based on the available resources and performance goals to achieve, one can choose the proposed AES-GCM architectures to fulfill the constraints of different applications.

10.Reference

1. "MehranMozaffari-Kermeni,Member and Arash Reyhani-MASOLESH,Member,",IEEE Computer society publication,vol 61.no.8,August 2012.
2. Nat'l Inst. of Standards and Technologies "Announcing the Advanced Encryption Standard (AES)," Fed. Information Processing Standards Publication, no. 197, Nov. 2001.
3. Wi-Fi, http://standards.ieee.org/getieee802/download/802. 11-2007.pdf, 2011.
4. WiMAX, http://standards.ieee.org/getieee802/download/ 802.16e-2005.pdf, 2011.
5. S. Trimberger, "Security in SRAM FPGAs," IEEE Design and Test of Computers, vol. 24, no. 6, p. 581, Nov./Dec. 2007.
6. M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," NIST SP, 800-38D, 2007.
7. IEEE Standard for Local and Metropolitan Area Networks, Media Access Control (MAC) Security, 2006.
8. Fibre Channel Security Protocols (FC-SP), http://www.t10.org/ ftp/t11/document.06/06-157v0.pdf. 2006.
9. Algotronics Ltd.: GCM Extension for AES G3 Core, 2007.
10. Helion Technology: AES-GCM Cores, 200
11. Elliptic Semiconductor Inc.: CLP-15: Ultra-High Throughput AESGCM Core-40 Gbps, 2008.
12. E. Ka¨sper and P. Schwabe, "Faster and Timing-Attack Resistant AES-GCM," Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '09), pp. 1-17, 2009.
13. K. Jankowski and P. Laurent, "Packed AES-GCM Algorithm Suitable for AES/PCLMULQDQ Instructions," IEEE Trans. Computers, vol. 60, no. 1, pp. 135-138, Jan. 2011[13]
14. S. Morioka and A. Satoh, "An Optimized S-Box Circuit Architecture for Low Power AES Design," Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '02), pp. 172-186, Aug. 2002.
15. A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," Proc. Int'l Conf. Theory and

Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01), pp. 239-254, Dec. 2001.

16. J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC Implementation of the AES SBoxes," Proc. Cryptographers Track at the RSA Conf. (CT-RSA '02), pp. 67-78, Jan. 2002.