International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Efficient and High-Performance Parallel Hardware Architectures for the AES-GCM Encryption and Decryption

P. Ajay Kumar
M. Tech .Student
CVSR College of Engineering
pantham.ajay@gmail.com

A. Krishna Kumari
Professor, ECE department
CVSR College of Engineering
akk_gnec@rediffmail.com

***Abstract* — The Advanced Encryption Standard (AES) is a symmetric key algorithm and its recently standardized authentication Galois/Counter Mode (GCM) have been utilized in various security-constrained applications. Many of the AES-GCM applications are power and resources constrained and require 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. The implementation of GHASH operation requires 11-clock cycles for 128-block, and is pipelined with AES 12/14/16 clock cycles for 128/192/256-bits. It has high throughput and less hardware architecture.**

***Keywords* — Advanced Encryption Standard, Galois/ Counter Mode, High Performance, Low Power.**

I. INTRODUCTION

The Advanced Encryption Standard-Galois/Counter Mode (AES-GCM) provides authentication and confidentiality for sensitive data simultaneously. In the AES-GCM, data confidentiality is provided by the Advanced Encryption Standard (AES) . 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 and Wi MAX and many other applications, ranging from the security of smart cards to the bit stream security mechanisms in FPGAs . The authentication of the AES-GCM is provided by the Galois/Counter Mode (GCM) 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. Moreover, it has been utilized in a number of cores from industry, see, for example and. In addition, two AES-GCM software-based implementations have been presented in and. Among the transformations in the AES encryption, the *Sub Bytes* (S-boxes) is the only non-linear one, requiring the highest area and consuming much of the AES power. Therefore, the performance metrics of the S-boxes affect those for the entire AES encryption significantly. For low-complexity implementations, the S-box can be realized using logic gates in composite fields. These Sboxes can also be pipelined for achieving high performance.. In some previous works such as and one specific S-box and in, three reported S-boxes have been synthesized on ASIC. However, exhaustive search has not been performed for all suitable composite fields to evaluate their performance metrics using the same technology.

II. THE AES-GCM

In this section, we present preliminaries for the AES-GCM algorithm. 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.

The Advanced Encryption Standard

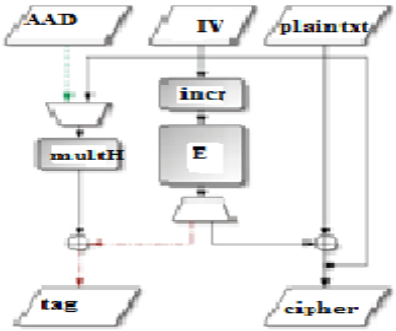
In the AES-GCM, only the AES encryption is utilized with the input and the output blocks of 128 bits. However, based on the security requirements, the key size could be determined as AES-128 (with 10 rounds), AES- 192 (with 12 rounds), or AES-256 (with 14 rounds) [1]. In the AES encryption, all the rounds 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. The transformation *SubBytes* (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 *ShiftRows*, 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 . For realizing the S-box, the irreducible polynomial of $P(x) = x8 + x4 + x3 + x + 1$ is used to construct the binary field $GF(28)$. Let I $GF(28)$ and O $GF(28)$ be the input and the output of the S-box. Then, the S-box consists of finding the multiplicative inversion, i.e., $I-1$ $GF(28)$ with the exception of mapping the zero input to the zero output, followed by the affine transformation in $GF(28)$. In what follows, we present the preliminaries regarding the hardware implementations of the S-boxes within the AES using LUTs and composite fields.

The Galois/Counter Mode

Authenticated encryption and decryption are the two functions within the GCM. The authenticated encryption performs two tasks; encrypting the confidential data and computing an authentication tag. The authenticated decryption function decrypts the confidential data and verifies the tag . The data flow of the authenticated encryption is shown in Fig. 2. As seen in this figure, the mechanism for the confidentiality of data is a variation of the block cipher counter mode of operation, denoted by *GCTRK* (Galois Counter with the key *K*). For the AES-GCM, the block cipher encryption with the specific key *K*

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

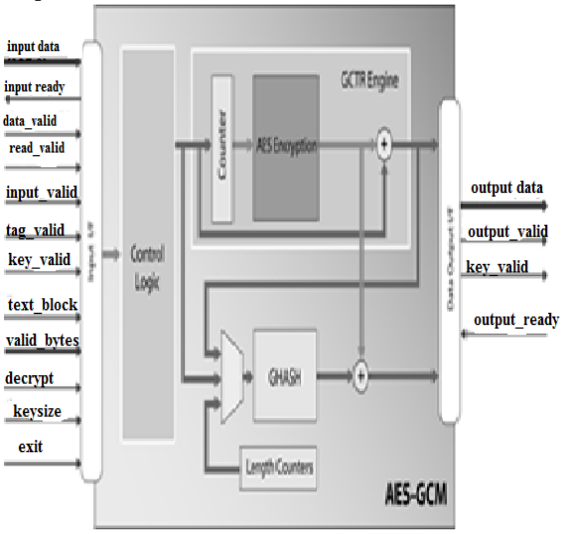
is shown by *AESK* in Fig. 2. Then, the function *GCTRK* performs t he block cipher counter mode with the *Initial Counter Block (ICB)* and its increments.



(*CB*2 − *CBi*) and the plaintext blocks (*P*1 − *Pi*) as the inputs. As shown in Fig. 2, the Galois Hash (*GHASHH*) function within the GCM provides the authentication for the confidential data. This function is constructed by *GF*(2128) multiplications with a fixed parameter, called the hash subkey (*H*). The *GHASHH* function calculates_*n* *j*=1

$$XjHn-j+1 = X1 \cdot Hn \quad X2 \cdot Hn-1 \quad \ldots \quad Xn \cdot H, \tag{1}$$

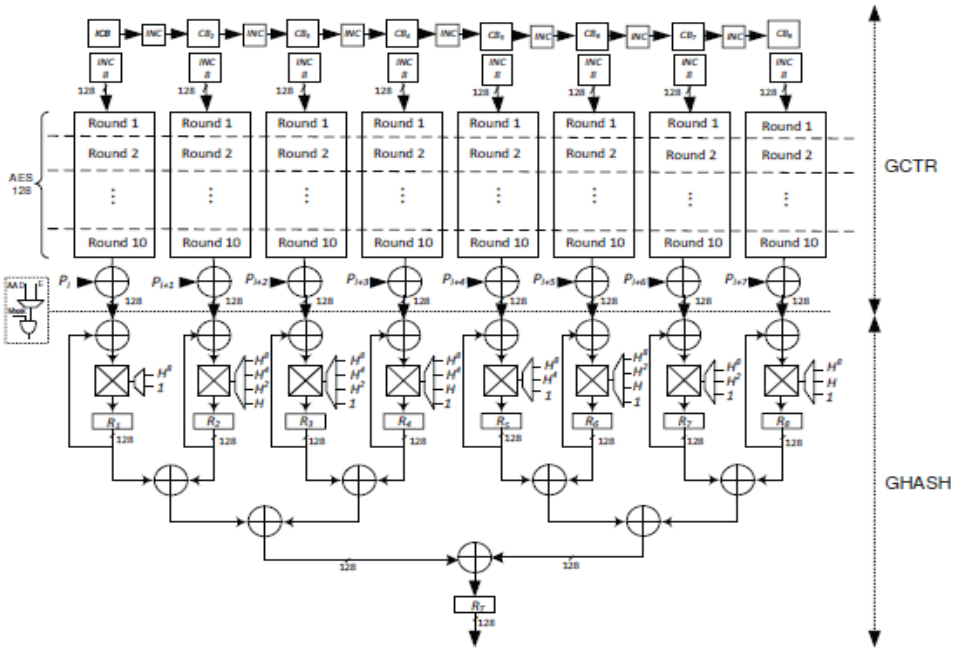
where *X*1 to *Xn* are the *n*, 128-bit blocks of the input [5]. It is noted that the hash subkey is generated by applying the AES to the *zero* block, i.e., 0 = (0, 0, ..., 0) *GF*(2128). Then, the *GHASHH* function calculates (1). All the arithmetic operations in (1), i.e., additions, GF multiplications, and exponentiations are performed over *GF* (2128) constructed by the irreducible polynomial $P(x) = x128 + x7 + x2 + x + 1$. As seen in Fig. 2, the total number of input blocks to *GHASHH* is $n = m + i + 1$, where *m* and *i* are the number of blocks for the additional authenticated data (*A*1 − *Am*) and the output of *GCTRK*, respectively. Eventually, the authentication tag *T* with length of *t* bits is derived. In the authenticated decryption, the same *GHASHH* procedure is performed on the authenticated data and cipher text blocks to verify the tag. For the entire description of the GCM, one can refer to.



III. 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.

Method	Hardware Complexity	Hardware Complexity after complexity reduction	Complexity reduction (%)	Timing Complexity
Cascade (Fig. 5a)	606 XORs	**594 XORs**	≈ 2%	$9T_X$
Parallel (Fig. 5b)	1986 XORs	1099 XORs	≈ **45%**	$5T_X$
Hybrid (Fig. 5c)	1627 XORs	1062 XORs	≈ 35%	$6T_X$



Sub-pipelining is useful in increasing the frequency of the AES at the expense of more area used for the pipeline registers, however, it increases the latency. For instance, the latency of a 3-stage sub-pipelined AES is 3 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. Moreover, for analyzing the effect of sub-pipelining, we have used sub-pipelined AES for two AES-GCM architectures. The details of our implementations are presented in this section. According to Table , we use the most efficient Sbox presented in Table , i.e., the one using polynomial basis (PB) based on (2), to reach the AES-GCM with the highest performance. The AES-128 encryption is considered as the block cipher for the GCM the 10 rounds of the AES-128 are unrolled and pipelined. Moreover, as seen in Table , we use the proposed Algorithm 1 for the GCM and utilize the parallel method in Fig for hash subkey exponentiations (hardware optimized through complexity reduction methods in the previous section).Finally, As seen in this table, we use both quadratic and sub-quadratic multipliers presented in Table.

Multiplier		GE[a]	Delay	Efficiency[b] $(10^3 \times \frac{Throughput}{GE})$
Complexity	Type			
Quad. [52]	-	56,957	$T_A + 10T_X$	$0.21/T_X$
Sub-quad. [53]	KO_1	44,338	$T_A + 12T_X$	$0.23/T_X$
	KO_2	34,660	$T_A + 14T_X$	$0.25/T_X$
	KO_3	28,195	$T_A + 16T_X$	$0.27/T_X$
	KO_4	24,517	$T_A + 18T_X$	$0.28/T_X$
	KO_5	23,443	$T_A + 20T_X$	$0.27/T_X$
	KO_6	24,961	$T_A + 21T_X$	$0.24/T_X$

IV. 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*(24), 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.

References

[1] DAEMEN, J.—RIJMEN, V.(1999): AES Proposal: Rijndael, TheRijndael Block Cipher, AES Proposal.

[2] Marko Mali, Franc Novak, Anton Biasizzo,(2005) "Hardware Implementation Of AES Algorithm", Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia, Volume: 56, Issue: 9, Pages: 265-269.

[3] Yibo Fan, Takeshi Ikenaga, YukiyasuTsunoo, and Satoshi Goto, (2008). "A Lowcost Reconfigurable Architecture for AES Algorithm" proceedings of world academy of science, engineering and technology volume 31 july 2008 ISSN 2070-3740.

[4] William Stallings, "Cryptography and Network Security", Third Edition, www.williamstallings.com/Crypto3e.html.

[5] P. Chodowiec, P. Khuon and K. Gaj, (2001) "Fast Implementations of Secret-Key Block Ciphers Using Mixed Inner- and Outer-Round Pipelining," Proc. ACM/SIGDA Int. Symposium on Field Programmable Gate Arrays, FPGA'01, Monterey, CA..

[6] Satoh, "High-speed Parallel Hardware Architecture for Galois Counter Mode," *In Proc. of ISCAS*, pp. 1863-1866, 2007.

[7] Satoh, T. Sugawara, and T. Aoki, "High-Performance Hardware Architectures for Galois Counter Mode," *IEEE Trans. Computers*, vol. 58, no. 7, pp. 917 930, 2009.