A Low Power AES-GCM Authenticated Encryption Core in 65nm SOTB CMOS Process

Van-Phuc Hoang, Van-Tinh Nguyen,
and Anh-Thai Nguyen
Le Quy Don Technical University
236 Hoang Quoc Viet Str., Hanoi, Vietnam
Email: phuchv@mta.edu.vn

Cong-Kha Pham
Graduate School of Informatics and Engineering,
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan
Email: phamck@uec.ac.jp

***Abstract*—This paper presents a low power AES-GCM authenticated encryption IP core which combines an improved four-parallel architecture, an advanced 65nm SOTB CMOS technology and a low complexity clock gating technique. As a result, the power consumption of the proposed AES-GCM core is only 8.9mW which is lower than other AES-GCM IP cores presented in literature. The detail implementation results are also presented and discussed.**

I. Introduction

The Advanced Encryption Standard with Galois/Counter Mode (AES-GCM) simultaneously provides authentication and confidentiality for security-constrained applications [1]. Its data confidentiality is provided by the Advanced Encryption Standard (AES) [1] which is employed in many applications such the wireless standard of Wi-Fi [2] and WiMAX [3]. The authentication of the AES-GCM is performed by the Galois/Counter Mode (GCM) which utilizes a universal HASH function [4] to provide both data encryption and authentication, called authenticated encryption (AE). Since AES-GCM can be fully pipelined or parallelized, it is a promising AE scheme for the broadband wireless networks such as WRANs [5] and Internet of Things (IoT). However, reducing area and power consumption is among the most emerging issues in designing IP cores in IoT era [6] which requires high speed computation platforms and high secure hardware implementations.

Therefore, many researches are focusing on design low-power, low area AE cores. The main contribution of this work is the design and implementation of an efficient architecture for an ultra-low power four-parallel AES-GCM core with new GHASH computation method, combined with advanced 65nm SOTB CMOS ASIC library [6] and a low complexity clock gating technique. A pipelined AES architecture [7] and high-performance GHASH function [8] are employed together to achieve the high speed and high throughput implementation. After that, clock gating technique is used for AES-GCM structure to reduce the power consumption. Different AES-GCM architectures are synthesized by using Synopsys Design Compiler tool with a 65nm CMOS standard library. The power consumption results of these hardware architectures are estimated and analyzed as well.

The rest of this paper is organized as follows. Section II provides the background about AES-GCM AE method. Section III presents the proposed low power, high speed AES-GCM implementation by applying a new architecture and simple clock gating technique. Section IV presents the implementation results and the comparison of our work with previous ones. Finally, section V concludes the paper.

II. AES-GCM for Authenticated Encryption

AE is employed for encryption of confidential data and providing the authentication tags. The data flow of the authenticated encryption is in Fig. 2 in [9]. It is clear that there are various ways of achieving the confidentiality of data with the block cipher in counter mode denoted as GCTR (Galois Counter with key K) [10]. In this AE algorithm, Galois HASH function in GCM mode is able to support both data authentication and confidentiality. It is conducted by $GF(2^{128})$ multiplications with parameters that are created by the HASH $subkey(H)$. The GHASH function calculates as shown in (1).

$$\sum_{j=1}^{n} X_j.H^{n-j+1} = X_1.H^n \oplus X_2.H^{n-1} \oplus \ldots \oplus X_n.H \qquad (1)$$

In addition, the block cipher counter mode is performed by GCTR function with an initial counter block (ICB), its increments $(CB_2 - CB_N)$ and the plaintext blocks $(P_1 - P_i)$ as the inputs.

Moreover, 128-bit key AES algorithm requires 10 rounds of transformation and each round contains four phases including SubBytes, ShiftRows, MixColumns and AddRoundKey [1]. Figure 1 shows a pipelined AES architecture conducted by placing 128-bit registers between each round to reach a high computation speed.

III. Proposed Low-Power AES-GCM Core Architecture

A. High performance GHASH implementation

According to the reference [7], GHASH function is presented as in (2).

$$X_i = (A_i \oplus X_{i-1}).H \qquad (2)$$

To achieve a high throughput for AES-GCM core, authors in [7] proposed the parallel architecture in which every multiplier uses a fixed operand. For instance, for 4-parallel

978-1-5090-6389-5/17/$31.00 ©2017 IEEE

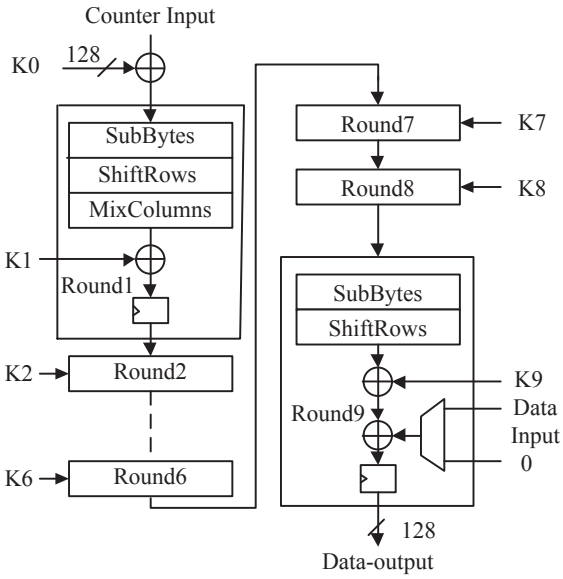

Fig. 1. The pipelined AES architecture.

architecture HASH function is conducted by using HASH subkeys denoted as H, H^2, H^3, H^4 in (3).

$$\begin{aligned} &X_i = (A_i \oplus X_{i-1}).H \\ &= (A_i.H) \oplus (A_{i-1}.H)(A_i.H) \oplus [(A_{i-1} \oplus X_{i-1}).H^2] \\ &= (A_i.H) \oplus (A_{i-1}.H^2) \oplus [(A_{i-2} \oplus A_{i-3}).H^3] \\ &= (A_i.H) \oplus (A_{i-1}.H^2) \oplus (A_{i-2}.H^3) \oplus [(A_{i-3} \oplus X_{i-4}).H^4] \end{aligned} \quad (3)$$

The implementation of H^3 requires a block multiplier. However, the computation of H^2 and H^4 can by implemented by block squaring. It is highly potential that the performance and area efficiency of squaring could outperform the multiplier. Therefore, to reduce the latency and power consumption of HASH subkey generator, we propose a method to produce HASH subkeys by using squaring operation as presented in the following part.

Reference [11] recommended using 10 data blocks for 4-parallel architecture as shown in (4).

$$\begin{aligned} &X_{10} = A_1.H^{10} \oplus A_2.H^9 \oplus A_3.H^8 \oplus A_4.H^7 \oplus A_5.H^6 \\ &\oplus A_6.H^5 \oplus A_7.H^4 \oplus A_8.H^3 \oplus A_9.H^2 \oplus A_{10}.H \end{aligned} \quad (4)$$

To further improve this subkey computation, we propose a method of using classical squaring as shown in (5).

$$\begin{aligned} &X_{10} = \left(\left(A_1.H^4 \oplus A_5\right).H^4 \oplus A_9\right).H^2 \\ &\oplus \left(\left(A_2.H^4 \oplus A_6\right).H^4 \oplus A_{10}\right).H \\ &\oplus \left(\left(A_3.H^4 \oplus A_7\right).1 \oplus 0\right)H^4 \\ &\oplus \left(\left(A_4.H^4 \oplus A_8\right).H^2 \oplus 0\right)H \end{aligned} \quad (5)$$

In the first block multiplier, three data blocks A_1, A_5, A_9 are multiplied by H^4, H^4 and H^2, respectively. The next three data blocks A_2, A_6, A_{10} are multiplied by H^4, H^4, H in the second block multiplier. However, in the third block, only two data blocks A_3 and A_7 are received and multiplied by H^4 and 1, respectively. The intermediate result will be stored in a register, then, in the next cycle, it will be multiplied by H^4. The data blocks A_4 and A_8 are multiplied by H^4, H^2, respectively and their product will be multiplied by H.

Therefore, the proposed architecture for 4-parallel AES-GCM as presented in Fig. 2, the AES-GCM core contains two parts which are a low complexity GHASH function and a low area 128-bit multiplier using Karatsuba-Ofman method [10]. Moreover, the proposed AES-GCM employs a clock gating technique based on a demultiplexer to reduce the power consumption. The pipelined 128-bit AES architecture is shown by dashed lines. The AES-128 structure in GCTR block generates the ciphertext which is combined with plaintext to produce the inputs for GHASH block. The function GCTR consists of the pipelined AES block combined with the initial counter block. Moreover, the initial vector (IV) in the GCM is a 96-bit vector, as recommended for the high-throughput implementations [10].

B. Karatsuba-Ofman algorithm

Karatsuba-Ofman (KO) algorithm is an efficient method for polynomial multiplication [10]. It is a recursive method which decreases the above multiplicative and additive complexities [12]-[13]. Let $a(x)$ and $b(x)$ be two elements in $GF(2^m)$. We need to find the product $d(x) = a(x).b(x) \bmod g(x)$ Both elements could be represented in the polynomial basis as [3]:

$$\begin{aligned} &a(x) = x^{m/2}(x^{m/2-1}.a_{m-1} + ... + a_{m/2}) \\ &+(x^{m/2-1}.a_{m/2-1} + ... + a_0) = x^{m/2}Ah + Al \\ &b(x) = x^{m/2}(x^{m/2-1}.b_{m-1} + ... + b_{m/2}) \\ &+(x^{m/2-1}.b_{m/2-1} + ... + b_0) = x^{m/2}Bh + Bl \end{aligned} \quad (6)$$

Using (6), the polynomial product could be written as:

$$d(x) = x^m.Ah.Bh + x^{m/2}(Ah.Bl + Al.Bh) + Al.Bl \quad (7)$$

In addition, the 128-bit $GF(2^{128})$ multiplier could be broken into 2-bit multiplications using KO method recursively to obtain KO_i, $(2 \leq i \leq 6)$ for the AES-GCM that would lead to a decrease of hardware area. Figure 3 shows the 128-bit multiplier architecture using the KO method. In this work, we use KO with $i = 2$ to balance between complexity and speed.

C. Classic squaring

The squaring operation in (8) can be implemented by algorithm 1 [16] in which $poly\_multiplication$ is polynomial multiplication and $reduction\_matrix\_R(f)$ is polynomial reduction matrix. Moreover, $m2xor$ and $m2and$ are modulo-2 XOR and AND operations, respectively. The irreducible function $f(x)$ is presented in (9).

$$H^2 = H.H \bmod f(x) \quad (8)$$

$$f(x) = x^{128} + x^7 + x^2 + x + 1 \quad (9)$$

D. AES-GCM core with a low complexity clock gating technique

In this paper, an advanced 65nm SOTB CMOS technology is used to provide an ultra-low power AES-GCM implementation. Among various CMOS processes, SOTB CMOS is a good candidate for low power electronics since it provides the back-bias control applied through thin BOX layer to reduce the leakage as well as the standby power consumption as

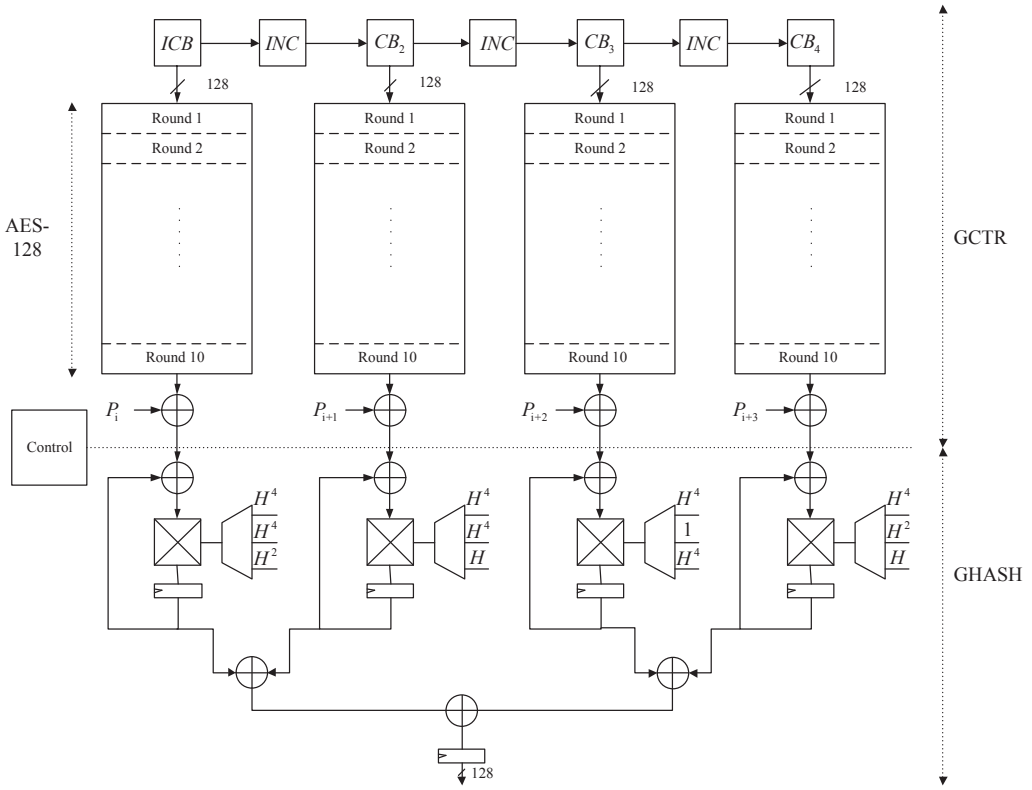

Fig. 2. The proposed improved 4-parallel AES-GCM architecture.

Algorithm 1 Classic squaring [15]

1: $d := poly\_multiplication(a, a)$;
2: $R := reduction\_matrix\_R(f)$;
3: **for** j in $0\,..\,m-1$ **loop** $c(j) := d(j)$;
4: **end loop;**
5: **for** j in $0\,..\,m-1$ **loop**
6: **for** i in $0\,..\,m-2$ **loop**
7: $c(j) := m2xor(c(j), m2and(R(j,i), d(m+i)))$;
8: **end loop;**
9: **end loop;**

presented in [14]. Especially, the ultra-low supply voltage of 0.4V is a key point to reduce the power consumption. Also, a SOTB CMOS device has an insulator layer between source drain and substrate to prevent CMOS device from the latch-up phenomenon. Moreover, the clock gating technique is applied to minimize the power consumption of the proposed AES-GCM IP core. In this technique, the clock signals are selected by the control signal to be distributed for registers in the system. A clock signal is provided to a component only when this component need to change its output values. Therefore, this could reduce the power consumption of the circuit [15]. In this paper, we propose a low complexity clock gating structure using a demultiplexer for the proposed AES-GCM core as shown in Fig. 4. The clock signal is switched by the signal sel which is generated from a finite state machine based control block.

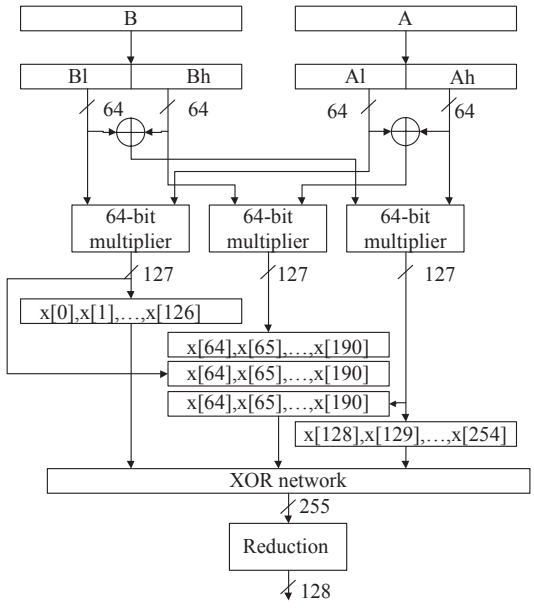

Fig. 3. The 128-bit multiplier architecture using KO ($i = 2$).

IV. IMPLEMENTATION RESULTS

The ASIC implementation of the proposed AES-GCM core and its components was performed by using Synopsys

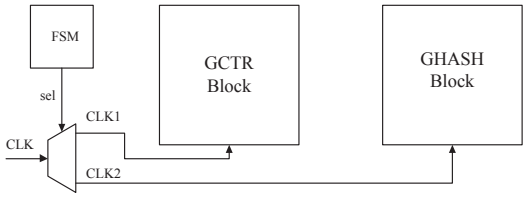

Fig. 4. Clock gating structure for the proposed AES-GCM core.

TABLE I. THE IMPLEMENTATION RESULTS COMPARISON BETWEEN KO MULTIPLIER AND SQUARING SCHEMES.

Design	Area (kgates)	Power (mW)	Max. Fre. (MHz)
KO Multiplier (128 bit)	1.32	0.31	901
Squaring (128 bit)	0.4	0.02	1923

Design Compiler tool with an advanced 65nm SOTB CMOS standard library. Firstly, Table I presents the comparison of implementation results between KO multiplier and squaring for HASH Subkey in this 65nm SOTB CMOS technology. It can be seen that the squaring implementation leads to a high improvement over KO multiplier in area, power consumption and maximum operation frequency.

Moreover, the ASIC implementation results of the proposed AES-GCM core are illustrated in Table II, together with the comparison. The power consumption results are also obtained by using post-synthesis analysis in Synopsys tools. The implementation was carried out for the case of the 4-parallel architecture with addition and multiplications based on bit-parallel $GF(2^{128})$ operations. The throughput is calculated by using formula in [7] is 8.3Gbps. It can be seen that the power consumption of proposed AES-GCM is reduced significantly and it is 12.7 times smaller than that in [9]. With this high throughput result, the proposed AES-GCM core is promisingly applicable for WRAN standard as well as IoT systems and its merit of low power, low area could give a significant impact on the research topic.

V. CONCLUSIONS

A low power AES-GCM IP core with the new 4-parallel architecture in the advanced 65nm SOTB CMOS technology was presented in this paper. The ASIC implementation results have clarified the improvements of the proposed method. In the future, we aim to further optimize the area for the proposed AES-GCM core using different multiplier methods and apply it for high speed applications such as IEEE 802.22 WRANs and IoT systems.

TABLE II. ASIC IMPLEMENTATION RESULTS OF DIFFERENT AES-GCM CORES.

Design	Technology	Area (kgates)	Power (mW)	Throughput (Gbps)
This work	65nm SOTB CMOS	625	8.9	8.3
In [9]	65nm CMOS	702	113	16.4
In [11]*	65nm CMOS	894	144.3	8.07
In [14, 15, 16]	65nm CMOS	110	19.6	4.54

Note (*): This result was presented in Table 7 in [9].

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2015.20.

This work is supported by VLSI Design and Education Center (VDEC), The University of Tokyo in collaboration with Synopsys, Inc.

REFERENCES

[1] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," *FIPS Publication 197,* Nov. 2001

[2] http://standards.ieee.org/getieee802/download/802.11-2007.pdf, 2011.

[3] http://standards.ieee.org/getieee802/download/802.16e 2005.pdf, 2011.

[4] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," *NIST SP,800-38D,* 2007.

[5] IEEE Std 802.22-2011 "Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands," pp. 1-672, Jul. 2011.

[6] S. Koteshwara; A. Das, "Comparative study of Authenticated Encryption targeting lightweight IoT applications," in IEEE Design & Test , vol.PP, no.99, pp.1-1, Mar. 2017.

[7] K. M. Abdellatif, R. Chotin-Avot and H. Mehrez, "Improved method for parallel AES-GCM cores using FPGAs," *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig),* Cancun, 2013.

[8] K. M. Abdellatif, R. Chotin-Avot and H. Mehrez, "Efficient parallel-pipelined GHASH for message authentication," *2012 International Conference on Reconfigurable Computing and FPGAs,* pp. 1-6, 2012.

[9] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Efficient and High-Performance Parallel Hardware Architectures for the AES-GCM," *IEEE Transactions on Computers,* vol. 61, no. 8, pp. 1165-1178, Aug. 2012.

[10] L. Henzen and W. Fichtner, "FPGA parallel-pipelined AES-GCM core for 100G Ethernet applications," *2010 Proceedings of the ESSCIRC,* pp. 202-205, 2010.

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

[12] Yan Bai, G. Shou, Y. Hu and Z. Guo, "High performance pipelined architecture of Ghash," *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology,* pp. 716-720, 2010.

[13] G. Zhou et al., "Complexity Analysis and Efficient Implementations of Bit Parallel Finite Field Multipliers Based on Karatsuba-Ofman Algorithm on FPGAs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 18, no. 7, pp. 1057-1066, Jul. 2010.

[14] Shiro Kamohara et al., "Ultralow-Voltage Design and Technology of Silicon-on-Thin-Buried-Oxide (SOTB) CMOS for Highly Energy Efficient Electronics in IoT Era," *Proc. IEEE 2014 Symposium on VLSI Technology (VLSI-Technology),* pp. 9-12, Jun. 2014.

[15] T. Kitahara et al., "A clock-gating method for low-power LSI design," *Proceedings of 1998 Asia and South Pacific Design Automation Conference,* Yokohama, pp. 307-312, 1998.

[16] Jean-Pierre Deschamps, "Hardware Implementation of Finite-Field Arithmetic," McGraw-Hill Education; 1st ed., 2009 [Chaper 7].