RESEARCH ARTICLE OPEN ACCESS

High Speed VLSI Architecture for AES-Galois/Counter Mode

Navjeet Singh*, Aman Dahiya**
**Department of Electrical and Electronics Engineering, Maharaja Surajmal Institute of Technology, New Delhi*
***Department of Electronics and Communication Engineering, Maharaja Surajmal Institute of Technology, New Delhi*

ABSTRACT

Galois/Counter Mode of Operation (GCM) is a block cipher mode operation used to provide encryption and authentication using universal Hashing based on multiplication over binary Galois/Finite Field.GCM can be implemented on both hardware and software effectively and efficiently. GCM supports pipelined and parallelized implementations to have minimal computational latency in order to be useful at high data rates. However need for continual performance improvement is still presented due to continuous increase in network bandwidth and inefficiency of existing parallelization methods. This paper presents use of modified parallel GHASH module and modified key Expansion module to improve overall efficiency. GCM architecture is modeled in Verilog HDL and Simulated in Xilinx ISE. ASIC implementation is done on 130 nm CMOS technology. Test case 4 of NIST submission for Galois/Counter Mode (GCM) is also verified.

Keywords: Galois Fields; Galois/Counter Mode; Verilog HDL, high performance, efficiency, Encryption, Authentication.

I. INTRODUCTION

The Galois/Counter Mode was designed by David A. McGrew and John Viega to improve Carter-Wegman Counter mode (CWC). GCM uses AES in Counter Mode of operation(CTR) for encryption, because it can be efficiently pipelined for high speed applications and message authentication code (MAC) for authentication that uses universal hashing. It uses polynomial hashing in Galois/Finite Field GF(2^w),core of this is multiplication by a fixed finite element [1]. The operation of GCM was designed to meet the need for authentication and encryption at a rate of 10 Gb/sec or higher in hardware. GCM also has low computational latency and supports parallelization and pipelining which makes it suitable for many applications such as Ethernet Security (WiGig), Fiber Channel Security Protocols (FC-SP) etc.

Bo Yang, Sambit Mishra, Ramesh Karri [7] have proposed an implementation of GCM by analyzing architectural level data dependencies and have achieved a throughput of 34 Gb/sec using 0.18μm CMOS standard cell library. Akashi Satoh [8] has proposed GCM hardware and GHASH module in parallel to achieve a throughput of 100 Gb/sec or more. However architecture presented was not suitable for many high speed applications as in order to encrypt correctly, number of input data blocks must be known and it is not possible in many applications.

In this paper we have presented and implemented architecture of GCM to improve overall hardware efficiency. Parallel modified GHASH module along with modified key expansion module is used implement the overall hardware and the hardware Efficiency of 0.176 is obtained using approx. 584000 gates. Rest of the paper is organized as follows. Introduction to GCM algorithm is given in Section 2.AES core is briefly discussed in Section 3.Modified architecture is discussed in Section 4.Implementation Results are presented in Section 5. Discussion and Acknowledgement is in Section 6 and Section 7 respectively.

II. GCM ALGORITHM

Galois/Counter Mode of Operation (GCM) is a block cipher mode operation used to provide encryption and authentication using universal Hashing based on multiplication over binary Galois/Finite Field. Galois/ Counter mode supports both authenticated encryption and authenticated decryption[1].

2.1 GCM Encryption

Authenticated Encryption in GCM has 4 Input bit-strings-:

- A plaintext *P* which can have upto ~2^{39} bits
- An additional authenticated data *A* up to 2^{64} bits. It is not encrypted
- A secret key K of 128 bits for AES block cipher.
- An initialization vector IV A 96-bit IV is recommended for efficiency.

Two output bit-strings:-

- A Ciphertext *C* with length is identical to that of the plaintext P.

An authentication tag T which can have up to 128 bits. *P* consists of a sequence of n bit-strings denoted as (P_1,P_2……P_{n-1},P^*_n) in which the bit length of the last bit-string is *u*, and the bit length of the other bit-strings is 128 bits [1]. Similarly, Ciphertext is denoted as ($\mathbf{C_1}$,$\mathbf{C_2}$……$\mathbf{C_{n-1}}$,$\mathbf{C^*_n}$),where the number of bits in the final block C^* is u. The additional authenticated data A is denoted as (A_1,A_2……$\mathbf{A_{m-1}}$,$\mathbf{A^*_m}$), where the last bit string A^* may be a partial block of length v, and m and v denote the unique pair of positive integers such that the total number of bits in A is (m−1)128+v and1≤v≤128.Encryption in GCM is defined as:

$$\begin{cases} H = E\,(K, 0^{128}) \\ Y0 \begin{cases} IV \parallel 0^{31}1 & \text{if } len(IV) = 96 \\ GHASH\,(H,\{\ \}, IV) & \text{otherwise} \end{cases} \\ Y_i = incr\,(Y_{i-1}) \text{ for } i=1, 2,\ldots ,n \\ C_i = P_i \oplus E\,(K, Y_i) \text{ for } i=1, 2,\ldots n-1 \\ C^*_n = P^*_n \oplus MSB_u\,(E\,(K, Y_n)) \\ T = MSB_t\,(GHASH\,(H , A, C) \oplus E(\,K, Y_n)) \end{cases}$$

With the input as A and C, the function GHASH is defined by GHASH (H, A, C) = X_{m+n+1}.And the variables X_i for i = 0,..., m + n +1 are defined as follows:

$$Xi= \begin{cases} 0 & \text{for } i = 0 \\ (X_{i-1} \oplus A_i)\cdot H & \text{for } i = 1,\ldots, m-1 \\ (X_{m-1}\oplus(A^*_m \| 0^{128-v}))\cdot H & \text{for } i = m \\ (X_{i-1} \oplus C_{i-m})\cdot\ H & \text{for } i = m+1,\ldots,m+n-1 \\ (X_{m+n-1} \oplus (C^*_n \| 0_{128-u}))\cdot H & \text{for } i = m+n \\ (X_{m+n}\oplus(len(A)\|len(C)))\cdot H & \text{for } i =m+n+1 \end{cases}$$

Hence the authenticated encryption operation [1] can be represented as Figure 2. For simplicity, a case with only a single block of AAD (labeled Auth Data 1) and two blocks of plaintext is shown. Here E_K denotes the block cipher encryption using the key K, $mult_H$ denotes multiplication in GF(2^{128})by the hash key H, and incr denotes the counter increment function.

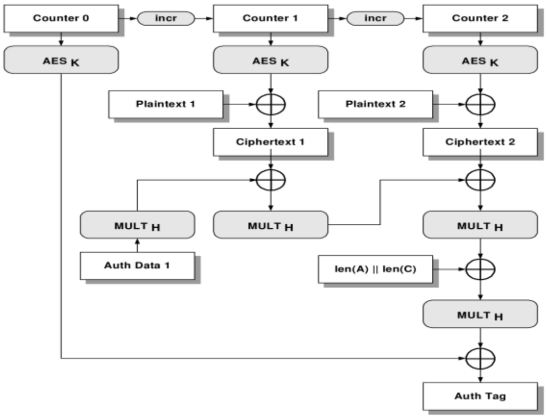

Figure1: Authenticated Encryption in GCM

2.2 GCM Decryption

The authenticated decryption operation is similar to the encrypt operation, however the order of the hash step and encrypt step is reversed. Hence it can be defined by the following equations:

$$\begin{cases} H = E\,(K, 0^{128}) \\ Y0 \begin{cases} IV \parallel 0^{31}1 & \text{if } len(IV) = 96 \\ GHASH\,(H,\{\ \}, IV) & \text{otherwise} \end{cases} \\ T' = MSB_t(GHASH\,(H , A, C) \oplus E(K, Y_0)) \\ Y_i = incr\,(Y_{i-1}) \quad \text{for } i=1, 2,\ldots ,n \\ P_i = C_i \oplus E\,(K, Y_i) \quad \text{for } i=1, 2,\ldots n \\ P^*_n = C^*_n \oplus MSB_u\,(E\,(K, Y_n)) \end{cases}$$

The tag T’ which is computed by the decryption operation is compared to the tag T of ciphertext C and If the two tags match (in both length and value), then the ciphertext is returned. Otherwise, the special symbol **FAIL** is returned [1].Figure of Authenticated Decryption is similar to authenticated encryption.

III. AES CORE

The Advanced Encryption Standard (AES) is the most widely used symmetric cipher today. Rijndael with a block length of 128 bits is known as the AES algorithm [2] .AES encrypts 128-bit data blocks under the control of a 128-bit user key. AES encryption or decryption uses 10 rounds, with each round using one round key. An additional key is used during pre-processing.AES consists of different layers and 128 bit data is manipulated in each layers. Each round except first round essentially consists of three layers. These layers can be defined as:

- **Key addition layer:** In this layer a 128-bit round key/subkey, derived from the main key in the key schedule, is XORed to the state.
- **Byte Substitution layer (S-Box):** In this layer each element of the state is nonlinearly transformed. For this Lookup tables having special mathematical properties are used. This introduces confusion to the data, i.e., it assures that changes in individual state bits propagate quickly across the data path.
- **Diffusion layer** This layer provides diffusion over all state bits and consists of two sublayers, which perform linear operations:
- TThe ShiftRows layer permutes the data on a byte level.
- The MixColumn layer is a matrix operation which combines (mixes) blocks of four bytes.

Also, the last round in AES does not make use of the MixColumn transformation, which makes the encryption and decryption scheme symmetric and the hardware implementations of AES can be either iterative [3][4][6] or pipelined [4][6][9] as shown in Figure 2.

However Pipelined AES implementations can only be used for Electronic Code Book mode (ECB) and Counter Mode (CTR).ECB mode use only input plaintext to determine output ciphertext. To achieve the high throughput pipelined AES implementations, GCM encryption and GCM decryption run AES in counter mode (CTR).

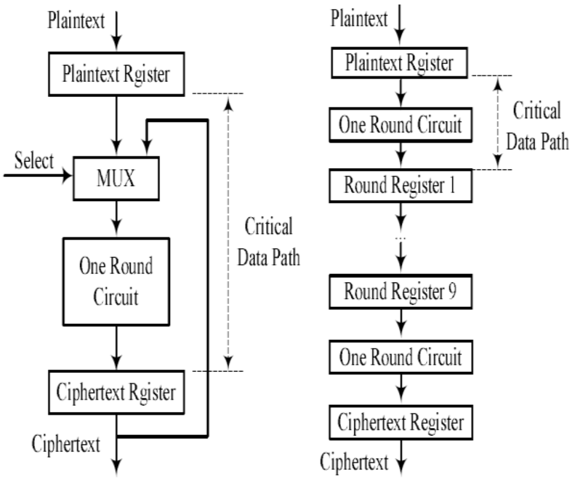

Figure 2: (a) AES iterative data path (b) AES Pipelined data path

IV. MODIFIED ARCHITECTURE

Several techniques have been used to achieve high throughput for various practical applications such as pipelined GCM hardware and GHASH module in parallel. In [8] Akashi Satoh proposed a generalized q-parallel equation with pq as the number of data blocks:

$$Xi = (\ldots.\ ((A_1H^q \oplus A_{q+1})H^q \oplus \cdots\cdots\oplus A_{(p-1)q+1})H^q$$
$$\oplus(\ldots.\ ((A_2H^q \oplus A_{q+2})H^q \oplus\cdots\cdots\oplus A_{(p-1)q+2})H^{q-1}$$
$$\oplus(\ldots.\ ((A_qH^q \oplus A_{2q})H^q \oplus A^{3q})H^q \oplus\cdots\cdots\oplus A_{pq})H$$

However in many practical applications such as sequential data transmission, the number of blocks of data is not available till the end of input data sequence. Hence Encryption process pattern to encrypt data cannot be determined in [8].To solve this problem modified parallel circuit suitable for sequential data transmission is introduced.

Let total number of data blocks is (ab+ N), and 1≤ N ≤b. So the modified general equation can be written as:

$$X_{ab+N} = (\ldots.\ ((A_1H^b \oplus A_2H^{b-1} \oplus\cdots\cdots\oplus A_bH) \oplus$$
$$A_{b+1})\ H^b \oplus A_{b+2})H^{b-1} \oplus\cdots\cdots\oplus A_{2b}H) \oplus$$
$$\ldots\ldots..$$
$$A_{ab+1})H^N \oplus A_{ab+2})H^{N-1} \oplus\ldots\ldots..\ A_{ab+N}H$$

4-parallel multiply add circuit for above equation can be obtained. Circuit can be represented as shown in Figure 4.

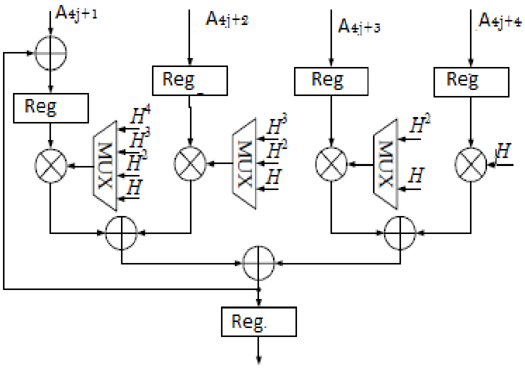

Figure 4: 4-parallel multiply add circuit

To fulfill the requirement of high throughput, Key Expansion module is also modified to generate round Keys on the fly with Sub Pipelined architecture. It generates 11 round keys hence 11 key expansion units are included. Architecture of key expansion unit (1-9) is same but the architecture of Unit 0 and Unit 10 are different. Unit 0 will simply perform Exclusive-OR operation. Architecture of round 1-9 is shown in Figure 5.

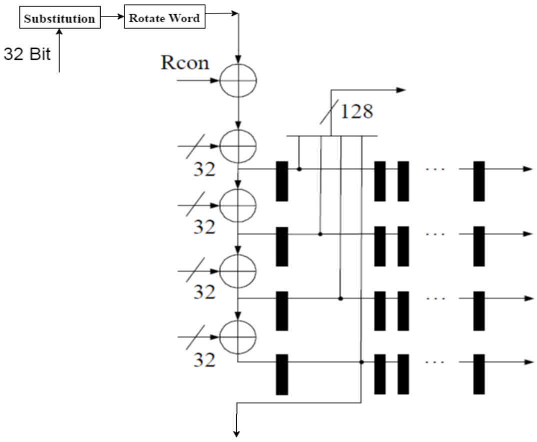

Figure 5: Key Expansion architecture Unit (1-9)

Substitution Box applies the S-box value used in SubBytes to each of the four bytes in one word, and the Rotate Word transformation performs a 1byte circular left shift on a word.

Hence Overall Block diagram of GCM can shown in **Figure 6**. . Each round except RU0, which represent simple Exclusive OR operation, essentially consists of three layers. Three layers are

- Key addition Layer
- Byte Substitution Layer
- Diffusion Layer which is subdivided into

(i) Shift Row
(ii) Mixcolumn

Also last round does not make use of the MixColumn. To achieve throughput both pipelining and Sub pipelining are used.

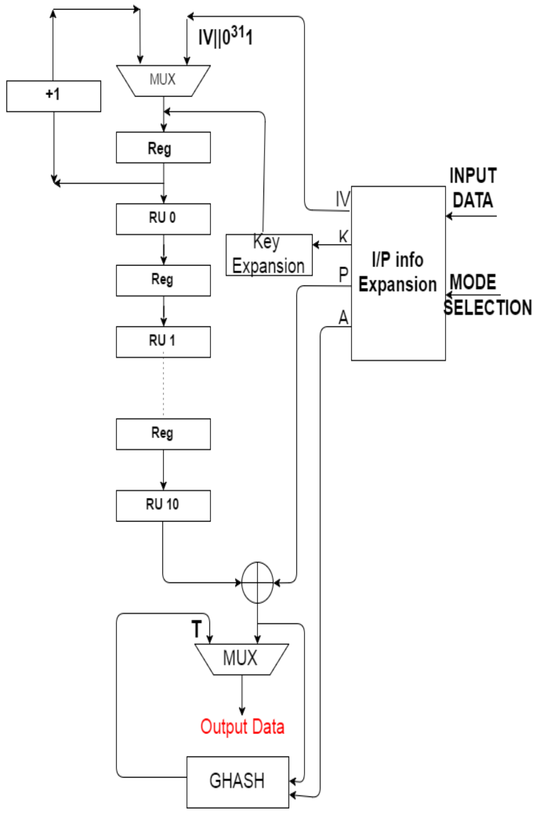

Figure 6: GCM Architecture.Here RU denotes Round Unit.

V. RESULTS AND COMPARISON

The modified GCM authenticated encryption architecture was modeled in Verilog HDL and simulated using Xilinx ISE and Test Vector 4 of [1] are verified. For ASIC implementation standard EDA tools have been used. Synopsys Design Compiler and Synopsys Prime have been used for synthesis and timing analysis while Cadence SOC encounter is used for placement and route. GCM architecture implementation is done on 130 nm 1.2V CMOS Technology. For Simulation test vector 4 are reproduced. Simulation results are shown in Figure 7 .Results of ASIC Implementation is compared with [7][8].

Results obtained are:-

K = feffe9928665731c6d6a8f9467308308
P = d9313225f88406e5a55909c5aff5269a
86a7a9531534f7da2e4c303d8a318a72
1c3c0c95956809532fcf0e2449a6b525
b16aedf5aa0de657ba637b39
AAD= feedfacedeadbeeffeedfacedeadbeefabaddad2
IV = cafebabefacedbaddecaf888

The corresponding cipher text and Tag is

CT = 42831ec2217774244b7221b784d0d49c
e3aa212f2c02a4e035c17e2329aca12e
21d514b25466931c7d8f6a5aac84aa05
1ba30b396a0aac973d58e091
Tag = 5bc94fbc3221a5db94fae95ae7121a47

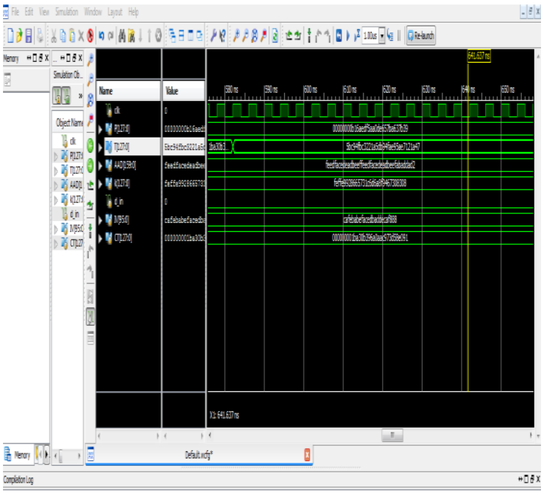

Figure7: Simulation Results of GCM Authentication and Encryption

ASIC implementation results of the modified design and comparisons with other references is done in Table (I)

TABLE (I): ASIC Result and Comparison

Parameter	**Modified GCM**	**[7]**	**[8]**
Data Length	128 bits	128 bits	128 bits
Gate Count	548000	499000	980000
Hardware Efficiency	0.176	0.070	0.166
Architecture of AES	Pipelined	Pipelined	Pipelined
GHASH Architecture	2-Parallel	Sequential	4-Parallel

VI. CONCLUSION

We expect the need for continual improvement of AES-GCM performance in hardware due to continued demand for increased network data rates and the desire for secure transmission. From the above results it can be seen that proposed modified GCM architecture has higher hardware efficiency as compared to [7] and [8].Also the design presented has high throughput and is suitable for high speed applications. Parallel and Sub Pipelined architecture for GHASH and Key Expansion Module is presented. Further speedup can be achieved using Parallelization strategy for AES-GCM.

ACKNOWLEDGEMENT

The Author greatly acknowledges 3st Technologies Pvt. Ltd. and Mr. Himashu Sharma, for project guidance and support.

REFERENCES

[1]. David A. McGrew; John Viega;" The Galois/Counter Mode of Operation (GCM)", IST [Online] **2005**, Available online: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-spec.pdf

[2]. J. Daemen; V. Rijmen, "AES proposal: Rijndael", 1991, Available online: http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf

[3]. S. Morioka;A. Satoh, "A 10 Gbps Full-AES Crypto Design with a Twisted-BDD S-Box Architecture," pp. 97-104, International Conference of Computer Design, 2002.

[4]. A. Hodjat; D.Hwang; B.C. Lai; K. Tiri; I. Verbauwhed, "A 3.84 Gbits/s AES Crypto Coprocessor with Modes of Operation in a 0.18um CMOS Technology," ACM Great Lake Symposium on VLSI, April 2005

[5]. A. Hodjat; I. Verbauwhede, "Minimum Area Cost for a 30 to 70 Gbits/s AES Processor," IEEE computer Society Annual Symposium on VLSI, pp. 83-88, Feb. 2004.

[6]. A. J. Elbirt; W. Yip; B. Chetwynd; and C. Paar; "An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 9(4), pp. 545-557, Aug. 2001.

[7]. Bo Yang, Sambit Mishra, and Ramesh Karri. (2005, June). High speed architecture for Galois/Counter mode of operation (GCM). Cryptology ePrint Archive.[Online], Available online: http://eprint.iacr.org/2005/146.pdf

[8]. Akashi Satoh, Research Center for Information Security, AIST."High-speed parallel hardware architecture for Galois counter mode," in Proc. IEEE Symposium Circuits and Systems (ISCAS), 2007.

[9]. X. Zhang; K. K. Parhi, "High-speed VLSI Architectures for the AESAlgorithm," IEEE Transanctions on Very Large Scale Integration (VLSI) Systems, vol. 12(9), pp. 957-967, 2004.