ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization   Vol. 5, Special Issue 4, March 2016

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

A High Throughput Low Power AES-GCM for FPGAs

Albert Jose

Asst. Professor, College of Engineering, Munnar, Kerala, India

ABSTRACT: This paper presents a combinational logic based S-Box implementation for the SubByte transformation in the Advanced Encryption Standard (AES) algorithm for Field Programmable Gate Arrays (FPGAs). The AES implementation have shown that the combinational logic based S-Box is proven for its small area occupancy and high throughput, given the fact that pipelining can be applied to this S-Box implementation as compared to the typical ROM based lookup table implementation in which access time is fixed and unbreakable. This compact and high speed architecture allows the S-Box to be used in both area limited and demanding throughput AES chips for various applications, ranging from small smart cards to high speed servers. The proposed architecture for AES-GCM is providing high performance because of its pipelined operation. In this pipelined architecture 10 rounds will be completed in 10 computation cycles. Therefore this proposed architecture provides 4 times more throughput which is a significant advantage in many existing AES application which is time critical.

KEYWORDS: AES, CTR, Encryption, Galois, GCM.

I. INTRODUCTION

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. AES was published by National Institute of Standards and Technology (NIST) in 2001 as the replacement for the previous cryptographic standards. The AES algorithm is a symmetric block cipher that can encrypt(encipher) and decrypt(decipher) information. Encryption converts data to an unintelligible form called cipher text; decrypting the cipher text converts the data back into its original form, called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. One of the main features of AES is simplicity that is achieved by repeatedly combining substitution and permutation computations at different rounds. That is, AES encrypts/decrypts a 128-bit plaintext/cipher text by repeatedly applying the same round transformation a number of times depending on the key size. In this paper, the AES forward cipher operation (i.e. AES encryption) with 128-bit key is mainly discussed since it is invoked in the GCTR module of the AES-GCM standard to provide confidentiality.Authentication and confidentiality for sensitive data is provided simultaneously in Advanced Encryption Standard-Galois/Counter Mode. The AES-GCM has been used for a number of applications such as the new LAN security standard WLAN 802.1ae (MACSec) and Fiber Channel Security Protocols.

II. AES ENCRYPTION

Here the preliminaries for the AES Algorithm is presented. In the AES-GCM, only the AES encryption is utilized with the input and the output blocks of 128 bits. In the AES encryption, all the rounds except for the last round have four transformations of SubBytes, ShiftRows, MixColumns, and AddRoundKey. For the last round, MixColumns is eliminated and only three transformations of SubBytes, ShiftRows, and AddRoundKey are used.

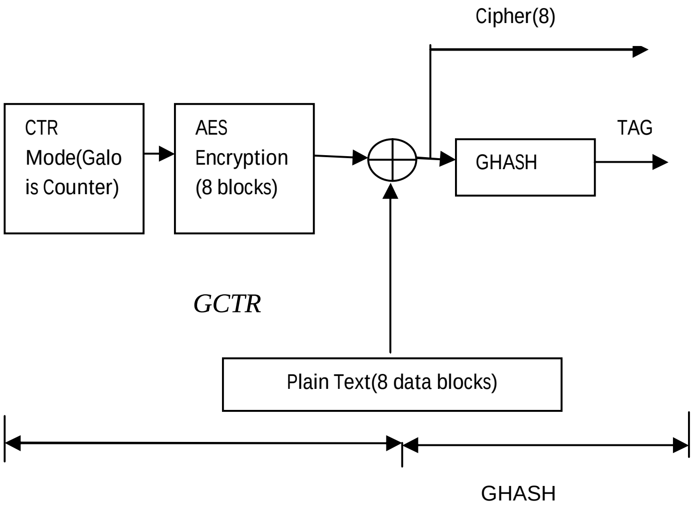

Fig.1: Simple AES_GCM Block Diagram

In AES encryption, CTR mode encryption method is used. Here I am using Galois Counter Mode. IV is the initial vector of size 128 bit given to the counter block. INC 8 block increment the initial vector by 8. 8 blocks of AES encryption units are used .Each unit consist of 10 rounds. Each round transformation consist of 4 phases: **Sub bytes, Shift rows, Mix columns, Add round key.**

a) Sub Bytes:

The function SubBytes used to substitute each bytes of the input using S-box. In this project S Box is implemented using combinational logic. This section illustrates the steps involved in constructing the multiplicative inverse module for the S-Box using composite field arithmetic. The multiplicative inverse computation will first be covered and the affine transformation will then follow to complete the methodology involved for constructing the S-Box for the SubByte operation. For the InvSubByte operation, the reader can reuse multiplicative inversion module and combine it with the Inverse Affine Transformation, as shown above in Figure 2.

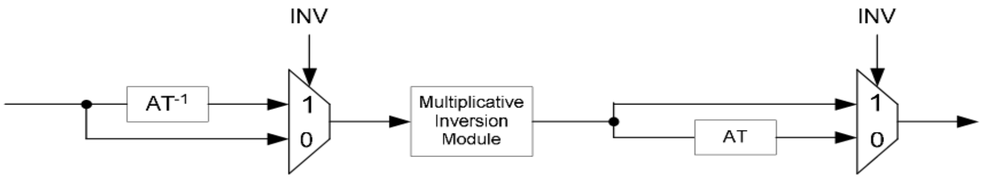

Fig.
2: The InvSubByte operation

Multiply, addition, squaring and multiplication inversion in GF(2^4) operations in Galois Field. Each of these operators can be transformed into individual blocks when constructing the circuit for computing the multiplicative inverse. The multiplicative inverse circuit GF(2^8) can be produced as shown in fig. 3.

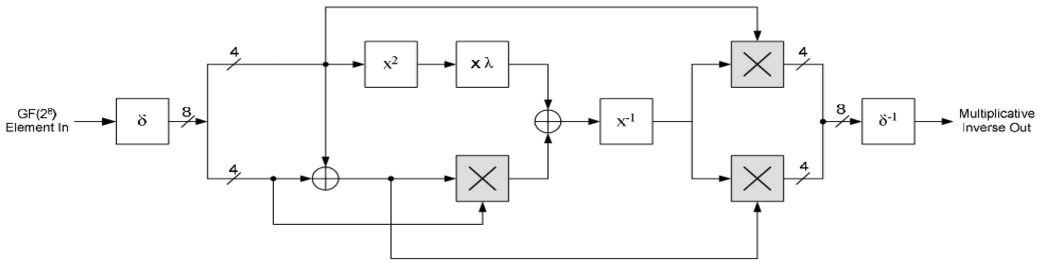

Fig.3: Multiplicative inversion module for the S-Box

b) Shift Rows

In shift rows,a circular byte shift in each row is performed.The 1st row is unchanged.2nd row does 1 byte circular shift to left.3rd row does 2 byte circular shift to left.4th row does 3 byte circular shift to left.

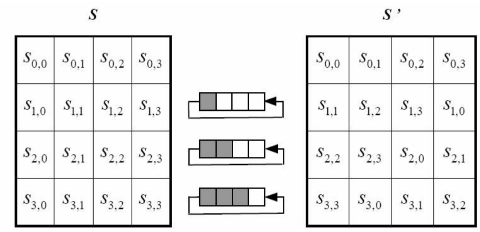
s
s'
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Fig.4: Shift Rows

c) Mixcolumns

The MixColumns stage is a substitution that makes use of arithmetic over GF(2^8).Each byte of a column is mapped into a new value that is a function of all four bytes in that column. It is designed as a matrix multiplication where each byte is treated as a polynomial in GF(2^8).

d) AddRoundKey

AddRoundKey operation is designed as a stream cipher.All the 128 bits of state are XORed with 4 32-bit words of key. AddRoundKey is the only operation that involves using the key to ensure security.

III. GALOIS COUNTER MODE

Authenticated encryption and decryption are the two functions within the GCM. The authenticated encryption performs two tasks: ***encrypting confidential data and computing an authentication tag.*** The authenticated decryption performs two tasks: ***decrypts the confidential data*** and ***verifies the tag.*** IV – Initialization vector is fed to AES Encryption Block. There are total 8 parallel AES blocks Incremented IVs are provided to each parallel AES blocks simultaneously for maximum throughput. Encrypted text is generated by applying AES on Incremented IVs. Cipher is generated by XOR-ing Plain text with Encrypted text.

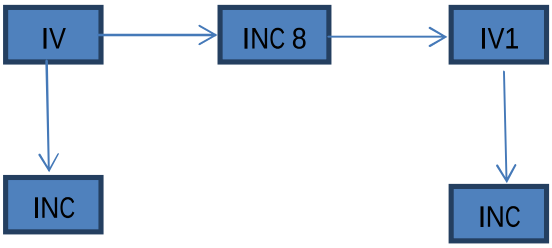

Fig. 5: Galois Counter Mode

IV. GHASH FUNCTION

The authentication mechanism within GCM is based on the hash function, GHASH, that features multiplication by a fixed hash subkey, over a binary Galois field GF(2^128). The hash subkey, denoted as H, is generated by applying the block cipher to the 128-bit “0” string. GHASH is a keyed hash function. The authentication mechanism within GCM is based on the hash function, GHASH, that features multiplication by a fixed hash subkey, over a binary Galois field GF(2^{128}). The hash subkey, denoted as H, is generated by applying the block cipher to the 128-bit “0” string. GHASH is a keyed hash function. A 128-bit multiplier over GF(2^{128}) is the core of the GHASH architecture. The GHASH architecture is shown in Figure 6 .One operand of the GF multiplier is the hash subkey H which can be treated as a fixed 128- bit constant for it will not change if the 128-bit key does not change.

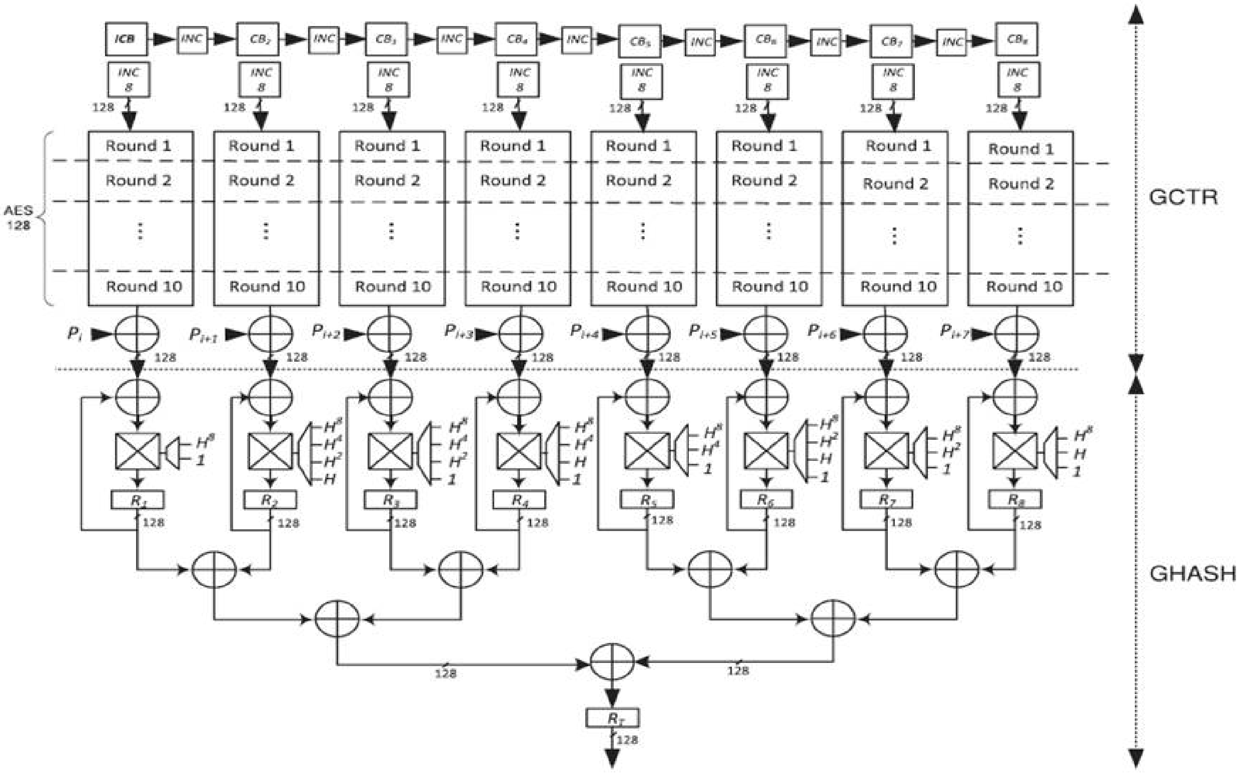

Fig. 6: AES GCM High Performance Architecture

Fig. 6 presents the proposed architecture for the AES-GCM for q = 8 parallel structures. The AES-128 pipeline registers are shown by dashed lines in Fig. 6. As seen in this figure, 10 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. 6, 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 = 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 IV in the GCM is 96 bits which is recommended for high-throughput implementations .The architecture shown in Fig. 6 assumes that the number of blocks n is a multiple of the number of parallel structures q and there is no additional authenticated data.

V. AES-GCM Implementation Results

Table 1: AES-GCM Implementation based on simulation results.

AES-GCM encryption/decryption designs	Technology	Areas	Clock Rates	Throughputs
Elliptic Semiconductor Inc.: CLP-15: [10]	ASIC	97k gates	300 MHz	7 Gbps
MehranMozaffariKermani August (2012) [6]*	FPGA	21409 slices	76 MHz	9.7 Gbps
Work in this thesis*	FPGA	18403 slices	63 MHz	40 Gbps

VI. CONCLUSION

This project presents high performance hardware architecture for encryption and decryption. Parallelism is achieved in AES operation and GCM. As compared to the typical ROM based lookup table, the combinational logic based S box implementation is capable of higher speeds since it can be pipelined. Also reduced area occupancy and increased throughput for subbyte operation is achieved. Overall reduction in will also boost more areas of implementation which otherwise used weaker encryption methods due to the increased latency of AES.

REFERENCES

[1] Canright.D and Osvik D.A, Sep (2009)"A More Compact AES," Selected Areas in Cryptography, vol 60, pp. 157-169, Springer-Verlag,

[2] Mehran Mozaffari-Kermani August (2012) "Efficient And High-Performance ParallelHardware Architectures For The AES-GCM ", Member, Ieee, And Arash Reyhani-Masoleh,Member.

[3] Good T and M. Benaissa, Dec. (2010)"692-nW Advanced Encryption Standard (AES) on a 0:13 CMOS," IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 18, no. 12, pp. 1753-1757, Dec. 2010.

[4] Jankowski.K and P. Laurent, Jan. (2011) "Packed AES-GCM AlgorithmSuitable for AES/PCLMULQDQ Instructions," IEEE Trans. Computers,vol. 60, no. 1, pp. 135-138.

[5] Lin S.Y and Huang C.T , (2007) "A High-Throughput Low-Power AES Cipher for Network Applications," Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC '07), pp. 595- 600,

[6] Edwin NC Mui "Practical Implementation of Rijndael S-Box Using Combinational Logic

[7] Pate P l,Januvary (2007) "Parallel Multiplier Designs for the Galois/Counter Mode of Operation," Master of Applied Science thesis, The Univ. of Waterloo.

[8] Zhang X and K.K. Parhi, Sept. (2004). "High-Speed VLSI Architectures for the AES Algorithm," IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 12, no. 9,pp. 957-967

[9] D.A. McGrew and J. Viega, "The Galois/Counter Mode of Operation (GCM),"

[10] Elliptic Semiconductor Inc.: CLP-15: Ultra-High Throughput AES-GCM Core-40 Gbps, 2008.