

High Performance AES Design Using OFB Mode

M.Anandababu¹, Mrs.D.Devapriya², C.Vijayalakshmi³

¹PG Student, Department of ECE, Sree Sakthi Engineering College, Coimbatore, India;

²Faculty, Department of ECE, Sree Sakthi Engineering College, Coimbatore, India;

³PG Student, Department of ECE, Sree Sakthi Engineering College, Coimbatore, India

ABSTRACT

The Advanced Encryption Standard (AES) is an encryption standard chosen by the National Institute of Standards and Technology (NIST) in 2001, which has its origin in the Rijndael block cipher. In this work, we derive three novel composite field arithmetic (CFA) Advanced Encryption Standard (AES) S-boxes of the field GF. The best construction is selected after a sequence of algorithmic and architectural optimization processes. Furthermore, for each composite field constructions, there exists eight possible isomorphic mappings. Therefore, after the exploitation of a new common sub expression elimination algorithm, the isomorphic mapping that results in the minimal implementation area cost is chosen. High throughput hardware implementations of our proposed CFA AES S-boxes are reported towards the end of this paper. Through the exploitation of both algebraic normal form and seven stages fine-grained pipelining, our best case achieves a high throughput in field-programmable gate array. At the same time while encryption process immunity of encryption is taken into the account. Five modes of AES have been used to perform security on satellite data. The different modes are ECB, CBC, OFB, CFB and CTR. All the above techniques except OFB lead to fault during data transmission to ground because of noisy channels. It is due to Single Event Upsets (SEU). In order to avoid data corruption due to SEU's a novel fault tolerant model of AES is presented. This reduces

the data corruption and increases the performance. As a result we can identify the error and also we can encrypt the image as color. Thus the data corruption due to Single Event Upset can be avoided and the performance was increased.

1.INTRODUCTION

To address the reliability issues of AES algorithm and to overcome the SEU. Five modes are used in AES. They are Cipher block chaining mode (CBC), Electronic code Book mode (ECB), Cipher Feedback Mode (CFM), Counter mode (CTR) and Output Feedback mode (OFB). Cipher Block Chaining is not suitable for satellite images. Because data is corrupted due to fault propagations. In Electronic Code Book if a single bit is corrupted the entire block is corrupted. In cipher Feedback mode the fault is propagated to next blocks. No fault is propagated in counter mode. And also satellite image communications are not suitable for counter mode. So to rectify the faults while transmission of data from satellites in noise an On-Board AESOFB based encryption is used. The faults are rectified by using Hamming Error Correction code Algorithm. The proposed approach reduces the SEU while transmission of data from satellites with noise.

In the OFB mode the output of the encryption is fed back into the input to generate a key stream, which is then XOR ed with the plain data to

generate the cipher data. If an SEU occurs during encryption in the OFB mode then all the subsequent blocks will be corrupted starting from the point where the fault has occurred. This is because the key stream required for encryption and decryption is independent of the plain and cipher data and hence the feedback propagates the faults from one block to another until the end of the encryption process.

2. ADVANCED ENCRYPTION STANDARD

AES is a symmetric encryption algorithm, and it takes a 128-bit data block as input and performs several rounds of transformations to generate output ciphertext. Each 128-bit data block is processed in a 4-by-4 array of bytes, called the state. The round key size can be 128, 192 or 256 bits. The number of rounds repeated in the AES, N_r , is defined by the length of the round key, which is 10, 12 or 14 for key lengths of 128, 192 or 256 bits, respectively. Fig.1 shows the AES encryption steps with the key expansion process.

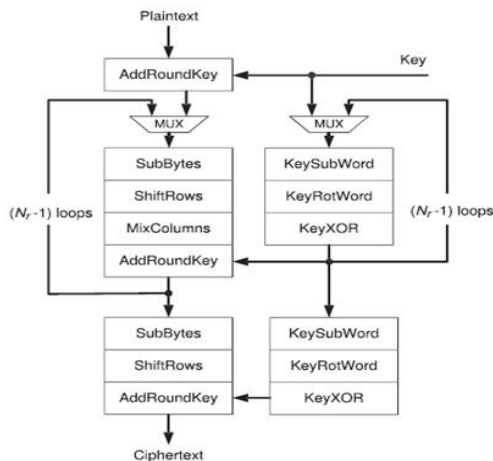


Fig1. Block diagram of AES encryption.

For encryption, there are four basic transformations applied as follows:

2.1 SubBytes

The Sub Bytes operation is a nonlinear byte substitution. Each byte from the input state is

replaced by another byte according to the substitution box (called the S-box). The S-box is computed based on a multiplicative inverse in the finite field $GF(2^8)$ and a bitwise affine transformation.

2.2 Shiftrows

In the ShiftRows transformation, the first row of the state array remains unchanged. The bytes in the second, third, and fourth rows are cyclically shifted by one, two, and three bytes to the left, respectively.

2.3 Mix columns

During the Mix Columns process, each column of the state array is considered as a polynomial over $GF(2^8)$. After multiplying modulo x^4+1 with a fixed polynomial $a(x)$, given by

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

the result is the corresponding column of the output state.

2.4 Addroundkey

A round key is added to the state array using a bitwise exclusive-or (XOR) operation. Round keys are calculated in the key expansion process. If Round keys are calculated on the fly for each data block, it is called AES with online key expansion. On the other hand, for most applications, the encryption keys do not change as frequently as data. As a result, round keys can be calculated before the encryption process, and kept constant for a period of time in local memory or registers. This is called AES with offline key expansion. In this paper, both the online and offline key expansion AES algorithms are examined. Similarly, there are three steps in each key expansion round.

2.5 Keysubword

The KeySubWord operation takes a fourbyte input word and produce an output word by

substituting each byte in the input to another byte according to the S-box.

2.6 Keyrotword

The function Key Rot Word takes a word [a3; a2; a1; a0], performs a cyclic permutation, and returns the word [a2; a1; a0; a3] as output.

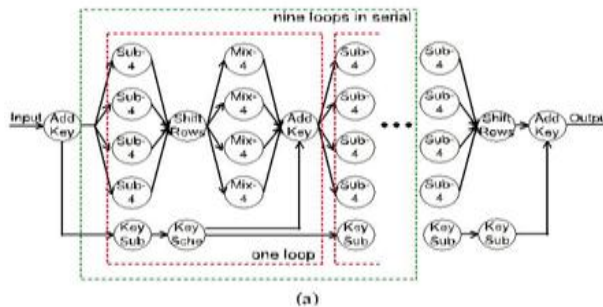
2.7 Key xor

Every word $w[i]$ is equal to the XOR of the previous word, $w[i - 1]$, and the word N_k positions earlier, $w[i - N_k]$. N_k equals 4, 6 or 8 for the key lengths of 128, 192 or 256 bits, respectively. The decryption algorithm applies the inverse transformations in the same manner as the encipherment. As a result, we only consider the encryption algorithm in this work for simplicity, since the decipherment yields very similar results.

3. AES IMPLEMENTATIONS ON ASAP

3.1 One-task one-processor

The most straight forward implementation of an AES cipher is to apply each step in the algorithm as a task in the dataflow diagram as shown in figure 2(a). Then each task in the dataflow diagram can be mapped on one processor on the targeted many-core platform. We call this implementation as One-task One-processor.



For simplicity, all of the execution delay input rates, and output rates.

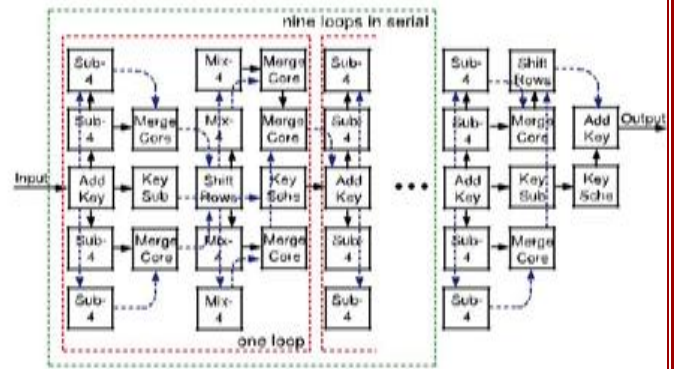
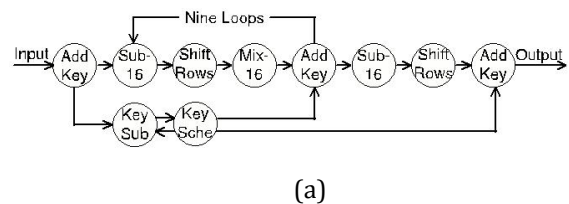


Fig2. One-task One-processor (a) dataflow diagram and (b) 10 cores ASAPs mapping.

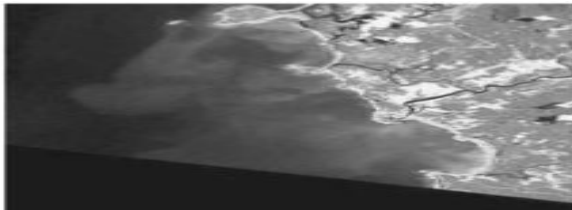
In the following dataflow diagrams are omitted. Since the key expansion is processing in parallel with the main algorithm, the throughput of the OTOP implementation is determined by the nine ($N_r-1=9$) loops in the algorithm. The OTOP implementation requires 10 cores on ASAP as shown in figure 2(b). The throughput of the OTOP implementation is 3,582 cycles per data block, equaling 223,875 clock cycles per byte.

3.2 Full-parallelism

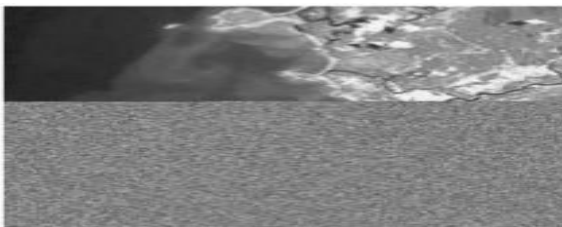
The Full-parallelism AES implementation combines the Parallel-Sub Bytes-Mix Columns model and loop unrolling. The dataflow diagram and the mapping of the Full-parallelism model are shown in Figs. 3(a) and 3(b). As expected, the throughput of this design is the highest among all of the models introduced in this paper since it employs most data and task parallelism.



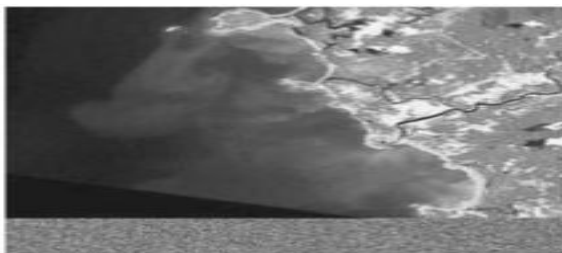
byte in the 6th round. In contrast, if a bit is corrupted during transmission, only a single bit in the plain data is affected and the error does not propagate to other parts of the message again for the same reason that the keystream does not depend on the plain or cipher data. So the transmission fault is not propagated



(a)



(b)



(c)

Fig 5. (a) Plain data multispectral satellite image.

(b) Decrypted OFB mode image with SEU occurring at 20,000th block. (c) Decrypted OFB mode image with SEU occurring at 40000th block.

This property is very useful to applications such as satellites where the transmission channels are very noisy. Hence the OFB mode has an advantage over the CBC and CFB modes in that any bit errors that might occur inside cipher data are not propagated to affect the decryption of subsequent blocks.

5. CONCLUSION

We have presented 16 different AES cipher implementations with both online and offline key expansion on a fine-grained many-core system. Each implementation exploits different levels of data and task parallelism. The smallest design requires only six processors, equaling 1:02 mm² in a 65 nm fine-grained many-core system. The fastest design achieves a throughput of 4.375 cycles per byte, which is 2.21 Gbps when the processors are running at a frequency of 1.2 GHz. We also optimize the area of each implementation by examining the workload of each processor, which reduces the number of cores used as much as 18 percent. The design on the fine-grained manycore system achieves energy efficiencies approximately 2.9-18.1 times higher than other software platforms, and performance per area on the order of 3.3-15.6 times higher. Overall, the fine-grained many-core system has been demonstrated to be a very promising platform for software AES implementations.

6. REFERENCE

- [1] Bin Liu , Bevan and M. Baas , 2013
Parallel AES Encryption Engines for Many-Core Processor Array .
- [2] roohi banu, and vladimirova, january 2009, Fault-tolerant encryption for space applications.
- [3] NIST, Nov. 2001. "Advanced Encryption Standard (AES),".
- [4] I. Verbauwhede, P. Schaumont, and H. Kuo, Mar. 2003. Design and Performance Testing of a 2.29 gb/s Rijndael Processor.
- [5] D. Mukhopadhyay and D. Roy Chowdhury, Jan. 2005. An Efficient end to End Design of Rijndael Cryptosystem in 0:18_m CMOS .
- [6] J.L. Hennessy and D.A. Patterson, Morgan Kaufmann, 2007. Computer Architecture: A Quantitative Approach.
- [7] S. Morioka and A. Sato, July 2004, A 10-gbps full-AES Crypto Design with a Twisted BDD s-Box Architecture.
- [8] J. Daemen and V. Rijmen, 2002, The Design of Rijndael.
- [9] A. Hodjat and I. Verbauwhede, Apr. 2006.
Area-Throughput Trade-Offs for Fully Pipelined 30 to 70 Gbits/s AES Processor .
- [10] S.K. Mathew, F. Sheikh, M. Kounavis, S. Guerin, A. Agarwal, S.K. Hsu, H. Kaul, M.A. Anders, and R.K. Krishnamurthy Apr. 2006, 53 gbps Native GF(2²⁴) Composite-Field AES-Encrypt/Decrypt Accelerator for Content-Protection in 45 nm High-Performance Microprocessors.
- [11] A. Hodjat and I. Verbauwhede, Apr. 2004.
A 21.54 gbits/s Fully Pipelined AES Processor on FPGA.
- [12] C.J. Chang, C.-W. Huang, K.-H. Chang, Y.-C. Chen, and C.-C. Hsieh, , Nov. 2008 High Throughput 32-Bit AES Implementation in FPGA.
- [13] J. Granado-Criado, M. Vega-Rodriguez, J. Sanchez-Perez and J. Gomez-Pulido, Nov. 2008. A New Methodology to Implement the AES Algorithm Using Partial and Dynamic Reconfiguration.