On the Usage of Isomorphic Fields in Hardware AES Modules for Optimizing the Efficiency

Luca Crocetti[1] and Sergio Saponara[1]

Department of Information Engineering, University of Pisa, Via G. Caruso 16, Pisa, 56122, Italy
{luca.crocetti,sergio.saponara}@unipi.it

Abstract. The Advanced Encryption Standard (AES) is widely accepted as the de-facto standard for symmetric-key encryption, and it is going to be used in the coming decades because of its resistance against Post-Quantum Cryptography. For this reason, it is the subject of many research works, and almost all converge on the usage of composite/tower fields for the hardware implementation of the S-box, the most expensive circuit in terms of both area and critical delay. Anyway, the debate is still open on applying isomorphic fields also to the other AES algorithm operations. In the attempt to give an answer, it is analyzed the application of the two approaches to the most recent and performing solutions from the state-of-the-art with the synthesis of the corresponding circuits on a 7nm standard-cell technology. In addition, the presented work constitutes also a guideline for implementing hardware AES modules that execute all operations over composite/tower fields.

Keywords: AES, Round, S-box, Composite, Tower, Field, Galois, Encryption, High Performance, Hardware accelerator

1 Introduction

The Advanced Encryption Standard (AES) [1] was released by the National Institute of Standards and Technology (NIST) and represents the de-facto standard for symmetric-key encryption, also because of its efficiency and performance [2]. Indeed, it is employed in several application fields such as High-Performance Computing [3, 4] and Automotive Security [5], and it is going to be used in the coming decades because of its resistance against Post-Quantum Cryptography [6]. For this reason, a high volume of works focusing on its optimization can be found in the literature. Concerning hardware implementations, almost all of the works converge on the usage of composite (or tower) fields to reduce the complexity of the S-box circuit [7–13], which consists in the calculation of the multiplicative inverse of a byte and an affine transformation. This approach consists in mapping the AES native field $GF(2^8)$ to an isomorphic field $GF((2^4)^2)$ (*composite* field) or $GF(((2^2)^2)^2)$ (*tower* field) [11], reducing the problem of the multiplicative inversion on a 4-bit vector, $GF(2^4)$, or on a 2-bit vector, $GF(2^2)$. In both cases, the basis used to represent the bytes on the isomorphic field can

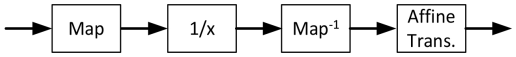

Fig. 1. Outline of the composite/tower field S-box circuit.

be a *Normal Basis* (NB), a *Polynomial Basis* (PB) [12, 13], a *Redundant Representation Basis* (RRB) [10, 11], or a *Mixed Basis* (MB) [7–9]. Whatever the composite/tower field and the basis used, the S-box circuit has always the same structure that is illustrated in Fig. 1.

In Fig. 1, the blocks Map and Map^{-1} represent respectively the isomorphic mapping (M) and the inverse isomorphic mapping (M^{-1}): this last to map back the multiplicative inverse on $GF(2^8)$ before the affine function (block Affine Trans.). Because both the inverse mapping and the affine transformation correspond to a matrix-vector multiplication, respectively, with the matrix M^{-1} and the matrix A, these two operations are merged into a unique matrix-vector multiplication ($A \cdot M^{-1}$), reducing both the gate count and the gate delay. Anyway, some works analyzed also the effects of extending the isomorphism to other AES operations, and the researchers have not found a common conclusion on this aspect. Some works propose architectures that extend the isomorphism to the other AES transformations [7], some others explicitly declare that this approach has no advantages, rather it worse the efficiency both in terms of area and maximum frequency [8, 9], or they do not consider this aspect at all, making the implicit assumption that the isomorphism should involve only the multiplicative inverse [10–13]. To investigate the effects of isomorphism on the other AES operations, we selected the most recent and performing works from the literature that use different composite/tower fields, and both approaches were implemented for each of them using SystemVerilog. The circuits were synthesized on a 7nm standard-cell technology and characterized by maximum frequency, area, and efficiency (expressed as frequency per area). The analysis and the results that follow refer to the encryption algorithm of AES.

2 Application of the isomorphism only to the AES S-box

The AES encryption algorithm iteratively processes 16-byte data blocks arranged in a 4×4 matrix according to a certain number of rounds. In the case of 128-bit keys, the encryption process performs 10 (main) rounds composed by the operations *SubBytes* (the substitution of each byte by means of the S-box, Fig. 1), *ShiftRows* (a byte re-ordering), *MixColumns* (a linear transformation), and *AddRoundKey* (a 128 bitwise XOR between the data block and a round-key), as shown in Fig. 2. The illustrated architecture includes also an additional XOR between the input data block (data in) and the input key (key in) for the preliminary round, and a multiplexer after the *MixColumns* block to skip this operation in the last round. The round-keys are derived from the input key using operations similar to the ones of the AES round, such as the *SubWord*, the substitution through the S-box of a 32-bit word (4 bytes of the key). The only different oper-

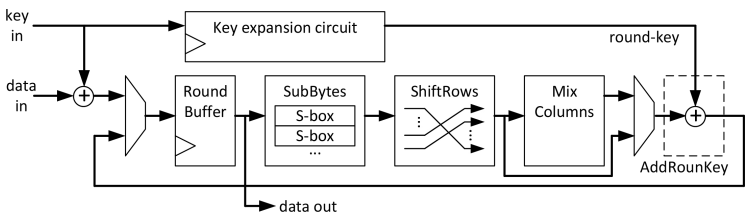

Fig. 2. Architecture of the AES encryption round using the composite/tower field for the S-box (SubBytes) and the native Galois field for the remaining (linear) operations. The internal architecture of each S-box corresponds to the one shown in Fig. 1.

ation is the XOR with the *Rcon* constant, anyway, the overall area and timing complexity of the key expansion circuit (Fig. 2) is lower than the one of the AES round circuit [3,6], which contains the critical, and it is:

$$t_R = \underbrace{t_{Map} + t_{MultInv} + t_{InvMap||Aff}}_{t_{S\text{-}box}} + t_{MixCol} + 2 \cdot t_{MUX} + t_{XOR} \qquad (1)$$

In Equation 1, t_{Map}, $t_{MultInv}$, $t_{InvMap||Aff}$, t_{MixCol}, t_{MUX}, t_{XOR} are, respectively, the propagation delays of isomorphic mapping, multiplicative inverse, inverse isomorphic mapping merged with the affine transformation, the *MixColumns*, a multiplexer, and an XOR gate (*AddRoundKey*).

3 Application of the isomorphism also to other AES encryption operations

To apply the isomorphism of the composite/tower field to the other encryption round operations, the *MixColumns* transformation can be expressed as:

$$b_o = 2 \cdot (b_{i_0} \oplus b_{i_1}) \oplus 1 \cdot (b_{i_1} \oplus b_{i_2} \oplus b_{i_3}) \qquad (2)$$

In Equation 2, the output byte b_o is generated by multiplying $(\cdot)$ the inputs bytes b_{i_j} by the coefficients 1 and 2 that corresponds, respectively, to the identity function, and the shift on the left by one position plus the XOR with *00011011* if the most significant bit of the multiplied byte is *1*. Both coefficients can be expressed in the matrix form, respectively, as:

$$C_1 = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \quad C_2 = \begin{bmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 1&0&0&0&1&0&0&0 \\ 1&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 1&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \end{bmatrix} \qquad (3)$$

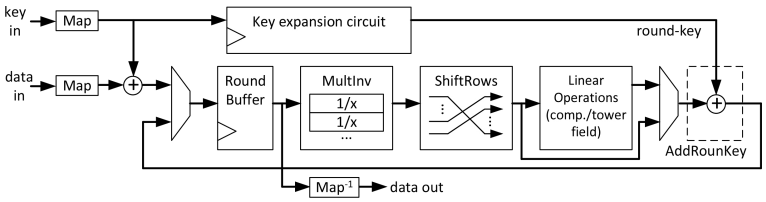

Fig. 3. AES round using composite/tower field for all the operations. The S-box in Fig. 2 is substituted by the only multiplicative inverse (block 1/x in Fig. 1), while the affine transformation (block Affine Trans. in Fig. 1) is merged with the MixColumns inside the block Linear Operations. The isomorphic mapping (Map) and inverse mapping (Map^{-1}) blocks are moved, respectively, to the beginning of the input data paths (key in, data in) and to the end of the output data path (data out).

Exploiting Equation 3, Equation 2 can be reformulated as:

$$b_o = C_2 \cdot (b_{i_0} \oplus b_{i_1}) \oplus C_1 \cdot (b_{i_1} \oplus b_{i_2} \oplus b_{i_3}) \tag{4}$$

From a mathematical point of view, the bytes b_{i_j} in Equation 2 and Equation 4 can be expressed as $b_{i_j} = (A \cdot M^{-1}) \cdot b'_{i_j}$, where b'_{i_j} is the isomorphic multiplicative inverse. Hence, the isomorphic byte at the output of the *MixColumns* can be computed as:

$$\begin{aligned} b'_o &= M \cdot b_o = M \cdot C_2 \cdot (b_{i_0} \oplus b_{i_1}) \oplus M \cdot C_1 \cdot (b_{i_1} \oplus b_{i_2} \oplus b_{i_3}) \\ &= \underbrace{M \cdot C_2 \cdot A \cdot M^{-1}}_{C'_2} \cdot (b'_{i_0} \oplus b'_{i_1}) \oplus \underbrace{M \cdot C_1 \cdot A \cdot M^{-1}}_{C'_1} \cdot (b'_{i_1} \oplus b'_{i_2} \oplus b'_{i_3}) \end{aligned} \tag{5}$$

According to Equation 5, the isomorphic output bytes of *MixColumns*, b'_o, can be directly obtained from b'_{i_j} by implementing the matrix-vector multiplications with C'_1 and C'_2. The other round operations do not require modification because the *ShiftRows* is just a byte re-ordering, and the *AddRoundKey* (i.e. an XOR) is invariant with respect to the isomorphism. This approach consents to eliminate the timing cost of the isomorphic mappings in Equation 1, according to the architecture shown in Fig. 3. For this purpose, also the key derivation process has to be modified accordingly by using the isomorphic values of the *Rcon* constant, i.e. $Rcon(r)' = M \cdot Rcon(r)$, for $r = 1, 2, \ldots, 10$. Hence, the critical delay of the fully isomorphic AES round is:

$$t'_R = t_{MultInv} + t_{LinOp} + 2 \cdot t_{MUX} + t_{XOR} \tag{6}$$

If t_{LinOp}, the delay of merged linear operations (Equation 5), is such that $t_{LinOp} < t_{Map} + t_{InvMap||Aff} + t_{MixCol}$, then $t'_R < t_R$ (i.e. the frequency increases).

4 Results

The two architectural approaches described in Section 2 and Section 3 were implemented in SystemVerilog reproducing the state-of-the-art works that use

composite/tower fields, i.e. [7], [10], [12], and [13]. It is to be noted that [7] already proposes the usage of isomorphism for linear operations, hence the counterpart with isomorphic mapping applied only to the S-box has been derived, and vice versa for [10], [12], and [13]. The hardware circuits were synthesized on a 7nm standard-cell technology included in the logic product kit SCH300MCPP64 from TSMC process CLN07FF41001, in the PVT corner slow process, 0.90 V and 125 °C. Table 1 reports the synthesis results.

Table 1. Approaches comparison for [7], [10], [12], and [13]. The second area value in column Area (*) is the area for the same maximum synthesis frequency of the corresponding architecture that uses the AES native field for linear operations.

Ref.	Field for linear operations	Maximum frequency	Area (Gate Equivalent, GE)	Area Efficiency
[7]	Native AES field	2.83 GHz	14.07 kGE	0.201 GHz/kGE
	Isomorphic field	3.06 GHz	14.78 (13.27*) kGE	0.207 GHz/kGE
[10]	Native AES field	2.83 GHz	13.79 kGE	0.206 GHz/kGE
	Isomorphic field	3.10 GHz	14.75 (12.88*) kGE	0.210 GHz/kGE
[12]	Native AES field	2.69 GHz	16.61 kGE	0.162 GHz/kGE
	Isomorphic field	3.08 GHz	18.12 (15.28*) kGE	0.170 GHz/kGE
[13]	Native AES field	2.80 GHz	17.67 kGE	0.158 GHz/kGE
	Isomorphic field	2.94 GHz	18.01 (16.87*) kGE	0.163 GHz/kGE

Referring to results in Table 1, the usage of isomorphic mapping also for the linear operations of the AES encryption round is advantageous. In each case, this approach allows reducing the area for the same synthesis frequency and increasing the maximum supported frequency with a low area cost, which however leads to an overall improvement of the efficiency. For instance, focusing on the experiment based on the work proposed in [10], the usage of the isomorphic mapping only in the composite/tower field S-box (case Native AES field) achieves a maximum frequency of 2.83 GHz at the cost of 13.79 kGE, i.e. an efficiency in terms of frequency per area of 0.206 GHz/kGE. If extending the same isomorphic mapping also to the remaining (linear) operations of the AES algorithm (case Isomorphic field), one effect is that for the same synthesis frequency of 2.83 GHz the area consumption is reduced to 12.88 kGE (about the 6.6%), i.e. the area value indicated between the round brackets and the symbol *. This would correspond to an improved area efficiency of $\frac{2.83 \text{ GHz}}{12.88 \text{ kGE}} \approx 0.220$ GHz/kGE. On the other hand, another effect is the reduction of the critical path of the round according to Fig. 3 and Equation 6, therefore the possibility to increase the maximum synthesis frequency which rises to 3.10 GHz at the cost of 14.75 kGE. This gives again an improved efficiency of 0.210 GHz/kGE.

The same effects can be found in every experiment conducted, hence the overall results suggest that the implementation of all the AES operations on isomorphic fields leads to more efficient hardware solutions, improving both the area consumption and the maximum supported frequency.

5 Conclusions

This work presents the investigation of the usage of composite/tower fields in the AES algorithm. To the best of our knowledge, this is the first work that analyzes this aspect in a systematic fashion clearly pointing out how to implement the linear AES round operations on isomorphic fields. Indeed, the provided mathematical analysis and the highlighted correspondence with the hardware architectures constitute also a guideline for hardware designers of AES modules. In addition, the systematic approach used in this work allows us to easily extend the analysis (and the hardware implications) to the inverse transformations of the AES decryption algorithm for implementing decryption-only modules or encryption/decryption modules, and to AES modules supporting also (or only) 192-bit keys (12 rounds) and (or) 256-bit keys (14 rounds).
Future works will include the evaluation of the effects on the resistance to the Side-Channel attacks due to the application of the isomorphic mapping to the linear operations of AES, according to the methodology presented in [14].

Acknowledgments

This work was partially funded by the European Union's Horizon 2020 research and innovation programme "European Processor Initiative" (grant agreement No. 101036168, EPI SGA2) and partially supported by the Italian Ministry of University and Research (MUR) with the project CN4 - CN00000023 of Recovery and Resilience Plan (PNRR) program, grant agreement No. I53C22000720001, and in the framework of the FoReLab project (Departments of Excellence).

References

1. NIST: Advanced Encryption Standard (AES). Federal Information Processing Standards (FIPS) publication 197 (2001).
2. Singha, T. B., Palathinkal, R. P., Ahamed, S. R.: Securing AES designs against power analysis attacks: A survey. IEEE Internet of Things Journal (2023).
3. Nannipieri, P., Di Matteo, S., Baldanzi, L., Crocetti, L., Zulberti, L., Saponara, S., Fanucci, L.: VLSI Design of Advanced-Features AES Cryptoprocessor in the Framework of the European Processor Initiative. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30(2), pp. 177-186 (2021).
4. Nannipieri, P., Crocetti, L., Di Matteo, S., Fanucci, L., Saponara, S.: Hardware Design of an Advanced-Feature Cryptographic Tile within the European Processor Initiative. IEEE Transactions on Computers (2023).
5. Carnevale, B., Falaschi, F., Crocetti, L., Hunjan, H., Bisase, S., Fanucci, L.: An implementation of the 802.1AE MAC Security Standard for in-car networks. In: 2nd IEEE World Forum on Internet of Things (WF-IoT), pp. 24-28 (2015). IEEE.
6. Nannipieri, P., Baldanzi, L., Crocetti, L., Di Matteo, S., Falaschi, F., Fanucci, L., and Saponara, S.: CRFlex: A Flexible and Configurable Cryptographic Hardware Accelerator for AES Block Cipher Modes. In: Applications in Electronics Pervading Industry, Environment and Society (APPLEPIES), pp. 31-38 (2021). Springer.

7. Ueno, R., Morioka, S., Miura, N., Matsuda, K., Nagata, M., Bhasin, S., ... and Homma, N.: High throughput/gate AES hardware architectures based on datapath compression. IEEE Transactions on Computers, vol. 69(4), pp. 534-548 (2019).
8. Ueno, R., Morioka, S., Homma, N., and Aoki, T.: A High Throughput/Gate AES Hardware Architecture by Compressing Encryption and Decryption Datapaths – Toward Efficient CBC-Mode Implementation. In: 18th International Conference on Cryptographic Hardware and Embedded Systems (CHES), pp. 538-558 (2016). Springer.
9. Ueno, R., Homma, N., Sugawara, Y., Nogami, Y., and Aoki, T.: Highly efficient GF (2^8) inversion circuit based on redundant GF arithmetic and its application to AES design. In: 17th International Conference on Cryptographic Hardware and Embedded Systems (CHES), pp. 63-80 (2015). Springer.
10. Reyhani-Masoleh, A., Taha, M., and Ashmawy, D.: Smashing the implementation records of AES S-box. IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 298-336 (2018).
11. Reyhani-Masoleh, A., Taha, M., and Ashmawy, D.: New low-area designs for the AES forward, inverse and combined S-boxes. IEEE Transactions on Computers, vol. 69(12), pp. 1757-1773 (2019).
12. Gaded, S. V., and Deshpande, A.: Composite Field Arithematic Based S-Box For AES Algorithm. In: 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1209-1213 (2019). IEEE.
13. Kumar, T. M., Reddy, K. S., Rinaldi, S., Parameshachari, B. D., and Arunachalam, K.: A Low Area High Speed FPGA Implementation of AES Architecture for Cryptography Application. Electronics, vol. 10(16) (2023).
14. Crocetti, L., Baldanzi, L., Bertolucci, M., Sarti, L., Carnevale, B., and Fanucci, L.: A simulated approach to evaluate side-channel attack countermeasures for the Advanced Encryption Standard. Integration, vol. 68, pp. 80-86 (2019).