The logic design of SM4 S-box in hardware based on asymmetric matrix expansion and sub-module series connection

First A. Author, *Fellow, IEEE,* Second B. Author, and Third C. Author Jr., *Member, IEEE*

***Abstract*—The design of S-box is not only the key step in the compact implementation of SM4 algorithm, but also the underlying structure that SM4 resists side channel attacks. We propose a hardware-based combinational logic design for the SM4 S-box, implement and evaluate this solution. Starting from the overall structure, this solution proposes a new SM4 S-box combinational logic technology to optimize area and depth. This technology is divided into three steps: First, an asymmetric transformation matrix is proposed for the SM4 S- box, which preserves the multiplicative inverse operation, which includes a custom affine matrix, an isomorphic mapping matrix, and a linear transformation extension matrix. Second, the asymmetric transformation matrix is solved by a compromise between an exhaustive algorithm and a heuristic algorithm, which is constrainted with a maximum depth, so guaranting the minimum depth while reducing the circuit area. Third, a low-latency sub-module series connection method based on a backtracking judgment mechanism is proposed, which can be used in the tower field inversion scheme for the SM4 S-box, which can reduce the module gap, and the circuit depth. We synthesized the design using the Synopsys Design Compiler tool. Compared with previous work, we achieved an optimized design with the least circuit elements and the smallest area in the SM4 S-box field of the ASIC platform, using only 102 circuit elements. The circuit area is 195.48GE, 192.8GE and 188GE, 186.5GE at SMIC130nm and SMIC65nm. At the same time, the depth of this area scheme has achieved the optimal design, with a depth of 19. In addition, we used the Xilinx Vivado tool to perform a comprehensive simulation of the SM4 algorithm containing this S-box scheme, comparing with the most commonly lookup table scheme on the FPGA platform, it has smaller resource usage, less energy consumption, and can better resist energy analysis attacks. Our design solution of SM4 S-box and the test of this design in SM4 are publicly available at the following open source address: https://github.com/mvpchenxin/paper_sbox.**

***Index Terms*—SM4, s-box, tower field, affine transformation, inverse operation**

I. Introduction

The International Organization for Standardization ISO/IEC officially announced that the SM4 block cipher algorithm became an international standard on June 25, 2021 [1]. Due to its high efficiency and strong security, it is widely used in finance, e-government, and commercial cryptographic applications. The SM4 algorithm uses a fixed key length of 128 bits and ensures the security of information transmission through a complex encryption process. Compared with the AES encryption algorithm, SM4 is simpler in structure and easier to implement in hardware, while maintaining high encryption efficiency and security, meeting the security requirements of a variety of business environments. SM4 is designed with a multi-round encryption structure, which makes it highly resistant to various attack methods. During the encryption process, SM4 enhances its encryption strength through key expansion and nonlinear transformation, thereby providing strong protection for data.

Based on the security of SM4 and the publication of international standards, the society urgently needs SM4 hardware solutions for FPGA and ASIC, because it is more secure and more power-efficient than software implementation. In some applications, such as credit cards, mobile phones, PDAs, etc., the complexity of hardware is a very important factor affecting cost and energy consumption. Therefore, it is very necessary to optimize the main operation parts of SM4 in both encryption and decryption . In the SM4 algorithm, the S- box is the only nonlinear unit. In encryption and decryption, especially byte substitution and inverse byte substitution operations, the S- box and the inverse S- box need to be executed separately . To build a 16×16 S- box , it is usually implemented in a lookup table to pursue simplicity and speed, but it takes up a lot of hardware resources , which is not conducive to resource-constrained scenarios and multi-functional deployment scenarios . Therefore, hardware optimization of the S-box is an important step to achieve efficient SM4 , and it is also an important link to promote SM4 from standards to the society .

This paragraph of the first footnote will contain the date on which you submitted your paper for review, which is populated by IEEE. It is IEEE style to display support information, including sponsor and financial support acknowledgment, here and not in an acknowledgment section at the end of the article. For example, “This work was supported in part by the U.S. Department of Commerce under Grant 123456.” The name of the corresponding author appears after the financial information, e.g. *(Corresponding author: Second B. Author).* Here you may also indicate if authors contributed equally or if there are co-first authors.

The next few paragraphs should contain the authors’ current affiliations, including current address and e-mail. For example, First A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@ boulder.nist.gov).

Second B. Author Jr. was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

Third C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba 305-0047, Japan (e-mail: author@nrim.go.jp)

There are many methods for constructing S-boxes, including random function screening, construction based on finite field multiplicative inverse, construction based on cryptographic structure, and construction based on cellular automata. Among them, the finite field $GF(2^8)$ multiplicative inverse composite affine transformation is one of the main methods for constructing S-boxes [2]. The construction process based on composite fields can be divided into linear transformation and nonlinear transformation.

Linear transformations only involve simple XOR operations, so the criteria for judging the implementation of linear components are the number of XOR gates and the delay of the circuit. The problem of finding an implementation with the least number of XORs for linear components is called the SLP problem. The algorithm for solving the SLP problem is called the SLP algorithm. In the literature [3] , the SLP problem was proved to be NP-hard. For smaller-scale linear transformations, such as 4×4 and 8×8 matrices on a binary field, an exhaustive search method can be used to find an implementation solution with the least number of XORs. However, for larger-scale linear transformations, relatively good implementations can only be found through heuristic algorithms. The Paar algorithm [4] and the BP algorithm [5] are the two most classic algorithms for solving the SLP problem. Most of the SLP algorithms currently used are derived from these two algorithms (referred to as the Paar algorithm and the BP algorithm, respectively). Since the advantages and disadvantages of the algorithms are not intuitive, some results need to be compared.

In 2017, Kranz et al. applied the BP algorithm to the implementation of some common MDS matrices and gave an implementation with only 97 gates [6]. In 2019, Maximov et al. selected different update strategies based on the BP algorithm and expanded the search range of the algorithm by establishing a search tree, thus giving an implementation that only requires 92 XOR gates. This scheme is also the scheme with the smallest number of gates currently implemented [7]. None of the above works restrict the depth of the circuit. For example, the circuit depth of Kranz's scheme is 8, and the circuit depth of Maximov's scheme is 6.

Nonlinear transformations are generally designed as nonlinear functions on finite fields, which have relatively complex algebraic structures. When using algorithms for optimization, the algorithm complexity is relatively high. Even for an 8-bit × 8-bit S-box, algorithmic search cannot be achieved. Therefore, the structure is generally analyzed by algebraic means, and then the implementation scheme is designed.

At CHES2015, Ueno et al. [8] used redundant basis and polynomial ring in a composite field structure $GF(2^4)^2$, providing an AES S-box implementation circuit with better comprehensive performance than previous ones. The implementation has a depth of only 15 and requires only 148 logic gates. Later, they refined the $GF(2^8)$ multiplication inverse module and provided the most efficient multiplication inverse implementation circuit known to date [9]. At CHES2018, Reyhani et al. [10] used regular basis in a composite field structure $GF(2^4)^2$and designed two types of AES S-box implementations with the help of CAD tools, this design is compact and efficient. Compared with the solutions of Ueno et al., the delay of these two types of implementations is slightly higher (the depth is 16 and 20 respectively), but the area is significantly reduced. At the same time, they also designed a bidirectional S-box and provided an implementation that is better than Canright's result [11].

Previous studies on composite field methods mostly focus on the construction of composite fields and the selection of bases, but there is little research on the overall structure of composite field methods. This paper's research on composite field methods no longer focuses only on the construction of fields, but also focuses on the improvement of the overall structure. In addition, from the perspective of application, the application background of previous research is relatively limited and cannot meet the security needs of more commercial environments. Therefore, in view of the limitations of the above research work, this paper makes the following contributions:

First, for the SM4 S-box, an asymmetric transformation matrix based on preserving the multiplication inverse is proposed, which includes a customize affine matrix, an isomorphic mapping matrix and a linear transformation extension matrix, and the input and output functions of the composite field transformation are constructed in the form of a matrix. By checking all the tower field representations under the normal basis caused by irreducible polynomials, we propose a maximize representation method of the linear matrix and that can reduce the optimization difficulty of the inversion process.

Second, a more adaptable linear component solution algorithm is proposed. The BP algorithm is the main framework for solving large-scale linear matrices. This paper improves the BP algorithm by adopting a new update strategy of k-step bundling based on depth constraints. The asymmetric transformation matrix is solved by a compromise between the exhaustive algorithm and the heuristic algorithm with the maximum depth constraint. The algorithm is adjusted according to the matrix scale to enhance the adaptability, so that the minimum depth is guaranteed while reducing the circuit area.

Third, for nonlinear components, a low-latency sub-module series connection method based on the backtracking judgment mechanism is proposed. In the process of nonlinear component logic operation, the implementation of adjacent modules is considered as a whole to avoid delay problems caused by independent optimization of each module. By reducing the module gap, the circuit depth is reduced.

II. Preliminaries

A. Basic principles of the tower field method

The S-box constructed based on the multiplicative inverse of finite fields is widely used in AES, SM4, Camellia, etc. The key designing the combinational logic circuit of the S-box is an efficient $GF(2^8)$multiplicative inverse circuit. At present, the composite field method is the basic method to realize the multiplicative inverse of finite fields $GF(2^8)$. The basic principle of the composite field method is shown in Fig. 1:

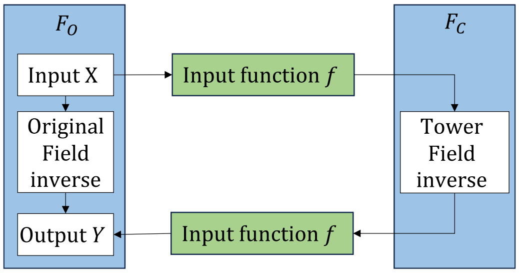


Fig.1. Basic principle diagram of composite field construction S-box

First, use the input function f to map the elements in the original field X to the composite field, then perform the inverse operation in the composite field, and finally use the output function g to map back to the original field, obtain the output Y. For the convenience of description, the finite field where the multiplicative inverse operation is to be found is called the original field, denoted by F_O; The newly constructed finite field is called the composite field, denoted by F_C; The input and output function pair is called the (f,g)conversion function pair between $F_O$and F_C.

The key to the composite field approach lies in two points:

1) The composite field has an obvious algebraic structure, so the multiplication inverse operation in the large field can be easily converted into a series of operations on its sub-fields. Then, by optimizing the operations on the smaller field, the operations on the larger field can be realized.
2) There is a pair of transformation functions between the original field and the composite field that preserves the multiplicative inverse operation, satisfying (1):

$$inverse_{F_O}(X) = g\left(inverse_{F_C}\big(f(X)\big)\right) \qquad (1)$$

Among them, $\forall X \in F_O$, $inverse_{F_O}$and $inverse_{F_C}$represent the inverse operations in F_O and F_C respectively, and the inverse element of 0 is agreed to be 0.

The construction of composite fields essentially relies on the expansion theory of finite fields. Here we introduce a special method for constructing composite fields - "tower fields".

Let $r(x)$ be the quadratic irreducible polynomial on $GF(2)$, that is $r(x) \in GF(2)[x]$. According to the theorem, the remainder class $GF(2)[x]/r(x)$ constitutes a finite field. Since the frequency of $r(x)$ is 2, $GF(2)[x]/r(x)$ is a finite extension of the finite field with $GF(2)$ that algebraic frequency is 2. Thus we get a 4-element field, denoted by $GF(2)^2$. Similarly, select the quadratic irreducible polynomial $s(x)$ on $GF(2)^2$, get the remainder class $GF(2)^2[x]/s(x)$, and form a 1 6-element field, denoted by $GF((2)^2)^2$. Finally, select the quadratic irreducible polynomial $t(x)$ on $GF((2)^2)^2$, get the remainder class $GF((2)^2)^2[x]/t(x)$, and form a 256-element field, denoted by $GF(((2)^2)^2)^2$, which is a 256-element composite field.

From the above construction process, we can see that the newly constructed 256-element composite field $GF(((2)^2)^2)^2$ has an obvious hierarchical structure, so the composite field constructed by this method is also called a "tower field". According to the finite field expansion theory, $GF(((2)^2)^2)^2$ can be regarded as two-dimensional linear space on its subfield $GF((2)^2)^2$, so by choosing an appropriate basis, the elements on $GF(((2)^2)^2)^2$ can be represented by elements on the subfield $GF((2)^2)^2$, and then the operations on $GF(((2)^2)^2)^2$ can be converted to operations on its subfield $GF((2)^2)^2$. Similarly, by choosing an appropriate basis, the operations on $GF((2)^2)^2$ can be converted to its subfield $GF(2)^2$. The operations on $GF(2)^2$ can be converted into a series of operations on $GF(2)$. Then, by analyzing the operations on the small field, the operations on the large field can be realized step by step.

B. SM4 S-Box Tower Field Representation

SM4 S-box expression based on polynomial basis is (2):

$$S(b) = M_2 \cdot I_{PB}^q(M_1 \cdot b \oplus C_1) \oplus C_2\,, \quad b \in F_{2^8} \qquad (2)$$

I_{PB}^q is based on irreducible polynomials $q(x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1 \in F_2[x]$and polynomial basis. Where: $M_1 = M_2$, $C_1 = C_2$is the initial affine transformation matrix, the specific matrix is as follows:

$$M_1 = M_2 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$C_1 = C_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Since it is difficult to implement directly circuits I_{PB}^q, we consider using tower field to implement the inverse operation, converting the inverse operation based on polynomial basis into the inverse operation based on tower field basis, as follows (3):

$$I_{PB}^q(A) = M_t \cdot I_{TB}(M_t^{-1} \cdot A) \qquad (3)$$

where M_t and M_t^{-1} are the transformation matrices between the polynomial basis and the tower field basis.

S- box expression based on the tower field basis is (4):

$$S(b) = M_2 M_t \cdot I_{TB}(M_t^{-1} M_1 \cdot b \oplus M_t^{-1} C_1) \oplus C_2 \qquad (4)$$

I_{TB} is an inverse operation based on the tower field basis. The calculation process is affected by $r(y)$、$s(z)$、$t(w)$ irreducible polynomials. Among them, $r(y)$、$s(z)$、$t(w)$the irreducible polynomials are selected as table 1:

TABLE I
SM4 S-BOX POLYNOMIAL COEFFICIENTS AND ROOTS

Polynomial	Coefficient	Root
$\boldsymbol{t(w) = w^2 + w + 1}$		$W = 0x5C$
$\boldsymbol{s(z) = z^2 + Tz + N}$	$T = 0x5C, N = 0x01$	$Z = 0x7A$
$\boldsymbol{r(y) = y^2 + \tau y + \upsilon}$	$\tau = T^2 \cdot Z^4 + 1 \cdot Z$ $= 0x77$ $\upsilon = T \cdot Z^4 + T^2 \cdot Z$ $= 0x27$	$Y = 0x66$

C. S-box field inversion operation

According to the above tower field representation, let $F_2 = \{0,1\}$, which F_{2^8} is expand from F_2. Conversely, low-order inverse operations can also be achieved by downgrading. For any element $b \in F_{2^8}$, there is the following formula:

First, F_{2^8} downgrade to F_{2^4}. Here $\mathrm{b} \in F_{2^8}$, $\gamma \in F_{2^4}$, $\mathrm{b} = \gamma_1 Y^{16} + \gamma_0 Y$, then:

$$b^{-1} = (bb^{16})^{-1} b^{16}$$
$$= [(\gamma_1 Y^{16} + \gamma_0 Y)(\gamma_0 Y^{16} + \gamma_1 Y)]^{-1}(\gamma_0 Y^{16} + \gamma_1 Y)$$
$$= [\gamma_1 \gamma_0 \tau^2 + (\gamma_1 + \gamma_0)^2 \upsilon]^{-1} \gamma_0 Y^{16} + [\gamma_1 \gamma_0 \tau^2 + (\gamma_1 + \gamma_0)^2 \upsilon]^{-1} \gamma_1 Y \tag{5}$$

At this point, according to the (5),the inverse operation on F_{2^8} is converted into a multiplication operation and an inverse operation on F_{2^4}.

Then, F_{2^4} downgrade to F_{2^2}. Here $\gamma \in F_{2^4}$, $\lambda \in F_{2^4}$, $\Gamma \in F_{2^2}$, $\Lambda \in F_{2^2}$, we have (6) and (7):

$$\gamma = \Gamma_1 Z^4 + \Gamma_0 Z \tag{6}$$
$$\lambda = \Lambda_1 Z^4 + \Lambda_0 Z \tag{7}$$

Then, the multiplication operation on F_{2^4} can be expressed as follows (8):

$$\gamma\lambda = (\Gamma_1 Z^4 + \Gamma_0 Z)(\Lambda_1 Z^4 + \Lambda_0 Z)$$
$$= [\Gamma_1 \Lambda_1 T + (\Gamma_1 + \Gamma_0)(\Lambda_1 + \Lambda_0) N T^2] Z^4 + [\Gamma_0 \Lambda_0 T + (\Gamma_1 + \Gamma_0)(\Lambda_1 + \Lambda_0) N T^2] Z \tag{8}$$

Finally, F_{2^2} downgrade to F_2 : $\Gamma \in F_{2^2}$, $\Delta \in F_{2^2}$, $u \in F_2$, $v \in F_2$, then we have (9) and (10):

$$\Gamma = u_1 W^2 + u_0 W \tag{9}$$
$$\Delta = v_1 W^2 + v_0 W \tag{10}$$

Then, the multiplication operation on F_{2^2} is represented as follows (11):

$$\Gamma\Delta = (u_1 W^2 + u_0 W)(v_1 W^2 + v_0 W)$$
$$= [u_1 v_1 \oplus (u_1 \oplus u_0)(v_1 \oplus v_0)] W^2 + [u_0 v_0 \oplus (u_1 \oplus u_0)(v_1 \oplus v_0)] W \tag{11}$$

At this point, we have completed the transformation of inverse operation based on polynomial basis to inverse operation based on tower basis.

III. SOLUTION DESIGN

A. Overall Structural Design

Based on the tower field construction principle and the SM4 S-box tower field inversion principle, combined with the previous research foundation and the security requirements in different business scenarios, the research focus of this paper should not only on the construction of the tower field, but also on the improvement of the overall structure of the tower field method. Based on the above point, we propose a new SM4 S-box combinational logic technology. This technology is divided into two steps. Different optimization measures are used in the design process of each step to achieve area and depth optimization.

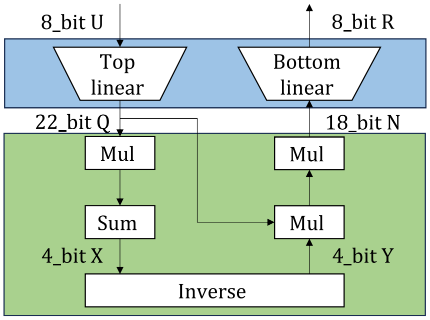


Fig.2. Architecture of the AES SBox according to [12] and [13]

Fig. 2 is our starting point. We will provide a set of methods to develop the new linear matrices at the top and bottom of the SM4 S-box from both design and implementation; improve the nonlinear inversion process, and then achieve a comprehensive optimization of the area and depth of the SM4 S-box .

First, an asymmetric transformation matrix based on preserving the multiplicative inverse is proposed for the SM4 S-box, which includes a customize affine matrix, an isomorphic mapping matrix, and a linear transformation extension matrix. Second, the asymmetric transformation matrix modules are solved by a compromise with a maximum depth constraint between an exhaustive algorithm and a heuristic algorithm, so that the minimum depth is guaranteed while reducing the circuit area. Third, a low-latency sub-module series connection method based on a backtracking judgment mechanism is proposed to implement a tower field inversion solution for the SM4 S -box, that can reduce the module gap and the circuit depth. The overall design process is shown in Fig. 3:

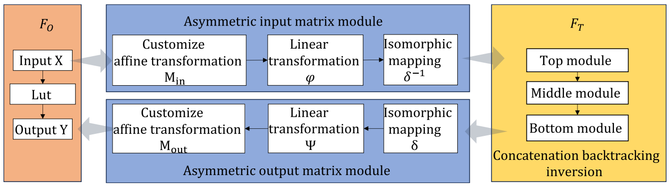

F_O
Input X
Lut
Output Y
Asymmetric input matrix module
Customize affine transformation $\mathrm{M_{in}}$
Linear transformation φ
Isomorphic mapping δ^{-1}
Customize affine transformation $\mathrm{M_{out}}$
Linear transformation Ψ
Isomorphic mapping δ
Asymmetric output matrix module
F_T
Top module
Middle module
Bottom module
Concatenation backtracking inversion

Fig. 3. Basic process of implementing SM4 S-box in tower field

1) **Customize Affine Transformation Matrix**

In this section, we will give a method for constructing a customize affine transformation matrix, first giving the SM4 S-box implementation structure. As shown in Fig. 4,

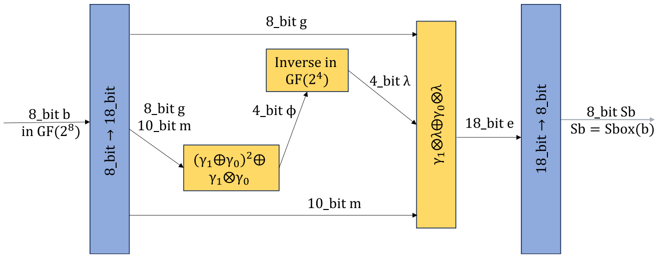


Fig. 4. SM4 S-box structure

Compared with the traditional architectures of [12] and [13], we changed the input and output matrix modules and integrated the multiplication and addition operations in the inversion operation. According to our SM4 S-box structure, we can see that the bit sequence b is converted into a bit sequence g through affine transformation, and the bit sequence m is generated by linear transformation. Therefore, the sequence g and m both have the linear expression of b. The input module can be regarded as consisting of a matrix $M_{in} \in F_2^{18\times 8}$ and a column vector $C_{in} \in F_2^{18}$; similarly, the output sequence e of the bottom module is combined into the inverse operation result, and then transformed into the output sequence Sb of the S-box through affine transformation. Therefore , Sb is a linear combination of e, and the output module is consisted of a matrix $M_{out} \in F_2^{8\times 18}$and a column vector $C_{out} \in F_2^{18}$. $M_{in} \in F_2^{18\times 8}$ represents 18 linear forms on 8 different variables, $M_{out} \in F_2^{8\times 18}$represents 8 linear forms on 18 different variables. That is as follows (12) and (13), the original symmetric affine transformation is expanded to the following asymmetric affine transformation:

$$\begin{bmatrix} g \\ m \end{bmatrix} = M_{in} \cdot b \qquad (12)$$

$$[Sb] = M_{out} \cdot e \qquad (13)$$

Then, based on the way J.Boyar et al. [14] determined the linear matrix of the AES S-box, we define the initial affine transformation matrix of the proposed structure as follows:

$$M_{in} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M_{out} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

As the number of matrix representations increases, the probability of finding a better matrix for circuit implementation also increases. Therefore, in the next work, we focus on expanding the search range of input and output matrices that maintain multiplicative inverses so that the matrix implementation is optimized.

2) Isomorphic Mapping Matrix

As can be seen from Section 2.3, the isomorphic mapping transforms the inversion of the finite field into the composite field, realizing the downgrade mapping transformation from $GF(2^8)$ to $GF(2^4)$or $GF(2^2)$, which is crucial to the SM4 S-box implementation.

The calculation of the isomorphic mapping matrix is described as follows: Let the element α of the field $GF((2^n)^m)$ be the root of $P(x)$, and the element ω on the field $GF(2^n)$ be the root of $Q(y)$. The SM4 S-box is constructed based on the following operation on the finite field $GF(2^8)$ as follows (14).

$$q(x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1 \qquad (14)$$

Let β be the root of $q(x)$, B be the standard basis on the field $GF(2^8)$. In order to construct an isomorphic mapping, use the primitives k in B to represent $GF((2^n)^m)$, then there is a one-to-one correspondence β with α^t, a one-to-one correspondence β^2 with α^{2t}, and so on, then we have (15):

$$M_t \cdot \beta^i = \alpha^{it}, i = 1,1, \dots k-1 \qquad (15)$$

To ensure that the multiplication operations in the two fields are also one-to-one mapped, the (16) must also be satisfied:

$$q(\alpha^t) = 0(modQ(y), P(x)) \qquad (16)$$

According to the finite field theorem, there are k elements that satisfie the above M_t conditions, namely t^{2^j} the operation module polynomial $2^k - 1$. Based on the above theory, we propose an isomorphic mapping matrix algorithm 1:

Algorithm 1 The algorithm of tower field isomorphism mapping matrix M_t

```
Input: $q(x), P(x), Q(y)$
Output: $M_t$
a)  Initialize, let the primitive element $\alpha$ of the field
    $GF((2^n)^m)$ to be the root of $P(x)$; let $t = 1$, the first
    column on the right side of the matrix $M_t$ be a vector
    $(0, 0, \dots, 0, 1)$, thus completing a mapping;
b)  Calculate $q(\alpha)(modQ(y), P(x))$. If the result is 0, the
    element is found, otherwise go to g;
c)  $\alpha^{t^{2^j}}$ $(j = 0, 1, \dots, k-1)$ are not the mapping elements of
    $\beta$;
```

d) Let $\boldsymbol{t} = \boldsymbol{t} + \mathbf{1}$, repeat this step until $\boldsymbol{t}^{\mathbf{2}^{j}}$ has not been calculated;
e) Calculate the $\boldsymbol{GCD}(\boldsymbol{t}, \mathbf{2}^{\boldsymbol{k}} - \mathbf{1})$and determine whether $\boldsymbol{\alpha}^{\boldsymbol{t}}$ is a prime element. If not, go to d;
f) Go to b;
g) Place $\boldsymbol{\alpha}^{\boldsymbol{t}}, \boldsymbol{\alpha}^{\mathbf{2}\boldsymbol{t}}, \dots, \boldsymbol{\alpha}^{\mathbf{7}\boldsymbol{t}}$ into $\boldsymbol{M}_{\boldsymbol{t}}$ as the binary vectors, from the second column of the matrix up to the leftmost column.

Let $P(x) = x^2 + \tau x + \upsilon$, $Q(y) = y^4 + y + 1$, and use the method to get the isomorphic mapping of $GF(2^8)$ and $GF((2^4)^2)$ as follows:

$$M_t = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$M_t^{-1} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

In this way, all elements on the field $GF(2^8)$ are mapped to the $GF(2^4)$ through operation $M_t \cdot M_{out}$ and $M_{in} \cdot M_t^{-1}$, preparing for the inverse operation of the tower field.

3) Linear Transformation Structure Preserving Multiplicative Inverse

Next, we will solve the problem of constructing the structure of all linear transformations that preserve multiplicative inverse operations, which can be from a finite field into itself, using some finite field theory.

Theorem 1 : Let $\varphi, \psi \in L^*(F_{p^n}/F_p)$, be called (φ, ψ) is a linear transformation structure that preserves the multiplicative inverse operation on F_{p^n}, if $\forall x \in F_{p^n}$, $\psi(\varphi(x)^{-1}) = x^{-1}$, and the inverse of element 0 is agreed to be 0.

Theorem 2 : A polynomial f of p on finite field F_{p^n} has the form: $f(x) = \sum_{i=0}^{i=n-1} a_i x^{p^i}$, where $a_i \in F_{p^n}$. If $p$is fixed with respect to the following, the p-polynomial is also called a linearized polynomial over F_{p^n}.

Theorem 3: There is a one-to-one correspondence between polynomials f of p on finite fields F_{p^n} and linear transformations in $L(F_{p^n}/F_p)$.

From the above theorem, it can be deduced that for any linear transformation φ in $L(F_{p^n}/F_p)$, there is a unique p-polynomial f, such that , $\forall x \in F_{p^n}, \varphi(x) = f(x)$. Called the polynomial f is the representation of the linear transformation φ.

Theorem 4: Let $a_i, b_j \in F_{p^n}$, and $\left(\sum_{i=0}^{i=n-1} a_i x^{p^i}\right) \cdot \left(\sum_{j=0}^{j=n-1} b_j x^{-p^j}\right) = 1$, hold for all $x \in F_{p^n}$, then there exists $0 \le i \le n-1$, such that $a_i b_i = 1$, and when $j \neq i$, we have $a_j = b_j = 0$.

Theorem 5: (φ, ψ) is a linear transformation structure on F_{p^n} that preserves the multiplicative inverse operation, then there exists $a \in F_{p^n}\{0\}, 0 \le k \le n-1$, such that $\varphi(x) = a x^{p^k}, \psi(x) = (ax)^{p^{n-k}}$.

From Theorems 4 and 5, we know that the linear transformation structure (φ, ψ) that preserves the multiplicative inverse operation has a specific polynomial form. And it can be uniquely determined by $(a, k), a \in F_{p^n}, 0 \le k \le n-1$, so it is also called the linear transformation structure $(\varphi_{a,k}, \psi_{a,k})$.

According to the above theorem, we can infer that different (a, k) corresponds to different linear transformation structures (φ, ψ), so there are $n(p^n - 1)$ linear transformation structures that preserve the multiplication inverse operation on F_{p^n}. In addition, if (φ, ψ) is a linear transformation structure that preserves the multiplication inverse operation on F_{p^n}, then (ψ, φ) is also a linear transformation structure that preserves the multiplication inverse operation on F_{p^n}.

According to the above finite field theory, we construct a linear transformation function pair. The specific implementation is shown in Fig. 5:

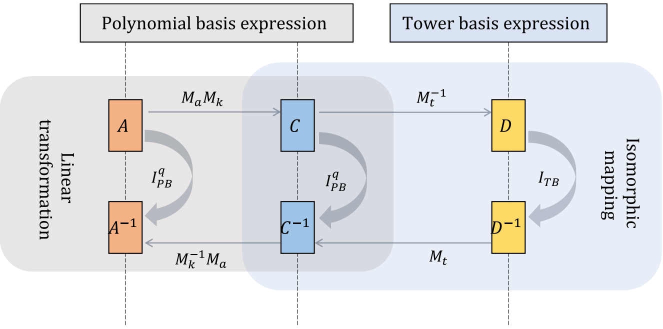


Fig. 5. Linear transformation and isomorphic mapping conversion mechanism

After linear transformation, the conversion formula between polynomial basis and tower field basis is as follows (17):

$$I_{PB}^q(A) = M_k^{-1} \cdot M_a \cdot I_{PB}^q(M_a \cdot M_k \cdot A) = M_k^{-1} \cdot M_a \cdot M_t \cdot I_{TB}(M_t^{-1} \cdot M_a \cdot M_k \cdot A) \tag{17}$$

Therefore, the improved S-box expression can be obtained as (18):

$$\mathrm{S}(b) = M_{\mathrm{t}} M_k^{-1} M_a M_{out} \cdot I_{TB}(M_{in} M_a M_k M_t^{-1} \cdot b \oplus M_a M_k M_t^{-1} C_1) \oplus C_2 \tag{18}$$

here:

$$a \cdot A \leftrightarrow M_a \cdot A$$
$$A^{2^k} \leftrightarrow M_k \cdot A$$
$$\sqrt[2^k]{A} \leftrightarrow M_k^{-1} \cdot A$$

Among them, α has 256 values in the range of 0-255 and β has 8 values in the range of 0-7, so there are 2040 combinations of $(\alpha,\ \beta)$.

According to the matrix design in the above two sections, we get the final optimization matrix as follows (19) and (20):

$$M_{in^*} = M_{in}M_aM_kM_t^{-1} \tag{19}$$

$$M_{out^*} = M_{\mathrm{t}}M_k^{-1}M_aM_{out} \tag{20}$$

M_{in^*} and M_{out^*} are the final asymmetric input and output matrix of the SM4 S-box, M_t, M_t^{-1} are the isomorphic mapping matrix, and M_a, M_k are the linear expand transformation matrix.

B. Matrix Optimization Solution

In hardware implementation, the Boolean circuit of a binary matrix can be given according to the logical operations. Such as, XOR operation corresponds to an XOR logic gate. The Boolean circuit can be viewed as a directed graph, where each logic gate corresponds to a node, and the input relationship of the logic gate reflects the edge of the directed graph. The depth of each node in the Boolean circuit is determined by the depth of its parent node, and the depth of the node is represented by $depth(Node)$. In a Boolean circuit, if the node $Node_1 = Node_2\ op\ Node_3$, where op is a logic operation, then there is (21),

$$depth(Node_1) = \max\big(depth(Node_2), depth(Node_3)\big) + 1 \tag{21}$$

Given the depth of the input node, the depth of each node in the Boolean circuit can be obtained by the above formula.

Based on our asymmetric input and output matrix, the SLP algorithm [3] is optimized with depth as a constraint. The goal description is summarized as follows:

Given a binary matrix $M_{m\times n}$ and a maximum allowed depth is $maxD$, given n bit inputs $X = (x_0 \dots x_{n-1})$, the circuit should compute m linear combinations $Y = (y_0 \dots y_{m-1})$, and $Y = M \cdot X$. Any circuit that implements the linear expression becomes a solution.

Given a set of input points x_i, we want to find the XOR sequence that results in m desired target point y_i, with maximum delay $maxD$.

At present, the methods for solving the above problems are mainly divided into two categories: Paar algorithm [4] and BP algorithm [5]. Although the Paar algorithm based on matrix representation has a faster running speed, the update process is not accurate. The Paar algorithm does not support cancellation operations (cancellation-free). This also points out the limitations of the Paar algorithm.

Boyar and Peralta circumvented the inaccuracy of the selection strategy brought by the matrix representation, proposed a BP algorithm that supports elimination operations. The BP algorithm uses a simple heuristic strategy to update the node set, so it is easy to fall into extremes. The node set update process is essentially a multi-stage decision problem.

This paper proposes a new k-step bundling compromise selection strategy based on the basic framework of the BP algorithm , which is described below :

1) In the update phase of the node set S, all combinations of consecutive k new nodes are traversed so that the updated distance vector Dist has the smallest sum of components .
2) If there are multiple combinations that all have the same sum of components, then use the backtracking method to traverse .
3) The entire algorithm terminates if and only if every component of Dist is 0 .

Among this,there is (22),

$$Dist(i) = distance(y_i, S), i = 1,2, \dots, n \tag{22}$$

The distance from the node set to the target node is defined as (23),

$$distance(y_i, S) = \min\{\#W\colon W \subset S, \oplus_{f\in W} f = y_i\} - 1 \tag{23}$$

The BP algorithm only considers the combination of the choices with the greatest advantage, while the k-step bundle compromise selection strategy considers the combination of all choices of consecutive k nodes. When k=1, the compromise selection strategy degenerates into a simple heuristic strategy, that is, the strategy of the BP algorithm. When k is large enough, it is obvious that the algorithm is equivalent to an exhaustive search, and the result must be optimal. Therefore, the compromise selection strategy is essentially a balance of the advantages of simple heuristics and exhaustive searches .

The K -step bundle compromise selection strategy provides a state update method with variable parameters, which can be set according to the scale of the problem to be optimized. Under the condition that computing power allows, the larger the k setting, the better the result. Therefore, for small-scale linear components, a larger k value can be appropriately selected. Since efficient combinational circuits not only require the number of logic gates, but also require the circuit to have as low delay as possible, we also use the formula $depth(Node)$ to track each node in the set S. Therefore, when the multiple combinations with the same advantages, a node with a smaller depth can be selected to ensure that the depth of the added node of the entire matrix is less than $maxD$.

According to the k-step bundle compromise selection strategy with depth as constraint, for searching the minimum solution, the Input matrix has only 8 inputs, the k value can be large enough, and exhaustive search is used to solve; the Output matrix has 18 inputs, and a large k value can easily cause exhaustive burden, so the k value is as small as possible, and a heuristic algorithm can be used. When the Hamming weight of the remaining rows becomes small enough, a complete exhaustive search is used for the end part. The final implementation result of the Input matrix is as follows table 2:

TABLE II

INPUT MATRIX IMPLEMENTATION

$\mathrm{t_1 = XOR(b_7, b_5)}$	$\mathrm{t_2 = XNOR(b_5, b_1)}$	$\mathrm{t_3 = XNOR(b_0, t_2)}$
$\mathrm{t_4 = XOR(b_6, b_2)}$	$\mathrm{t_5 = XOR(b_3, t_3)}$	$\mathrm{t_6 = XOR(b_4, t_1)}$
$\mathrm{t_7 = XOR(b_1, t_5)}$	$\mathrm{t_8 = XOR(b_1, t_4)}$	$\mathrm{t_9 = XOR(t_6, t_8)}$
$\mathrm{t_{10} = XOR(t_6, t_7)}$	$\mathrm{t_{11} = XNOR(b_3, t_1)}$	$\mathrm{t_{12} = XNOR(b_6, t_9)}$
$\mathrm{t_{13} = XOR(t_4, t_{10})}$	$\mathrm{t_{14} = XOR(t_2, t_{11})}$	$\mathrm{t_{15} = XOR(t_{12}, t_{14})}$
$\mathrm{t_{16} = XOR(t_3, t_{12})}$	$\mathrm{t_{17} = XOR(t_{11}, t_{16})}$	

$$g = (t_{15}, t_{14}, NOTb_0, t_2, t_5, t_{13}, t_7, t_{10})$$
$$m = (t_{12}, t_9, t_{17}, b_1, t_{11}, t_4, t_{16}, t_8, t_3, t_6)$$

The final result of the Output matrix is as follows table 3:

TABLE III
OUTPUT MATRIX IMPLEMENTATION

$\mathrm{E}_{11} = \mathrm{XOR}(\mathrm{e}_{17}, \mathrm{e}_{16})$	$\mathrm{E}_{10} = \mathrm{XOR}(\mathrm{e}_{15}, \mathrm{e}_{16})$	$\mathrm{E}_9 = \mathrm{XOR}(\mathrm{e}_{14}, \mathrm{e}_{13})$
$\mathrm{E}_8 = \mathrm{XOR}(\mathrm{e}_{12}, \mathrm{e}_{13})$	$\mathrm{E}_7 = \mathrm{XOR}(\mathrm{e}_{11}, \mathrm{e}_{10})$	$\mathrm{E}_6 = \mathrm{XOR}(\mathrm{e}_9, \mathrm{e}_{10})$
$\mathrm{E}_5 = \mathrm{XOR}(\mathrm{e}_8, \mathrm{e}_7)$	$\mathrm{E}_4 = \mathrm{XOR}(\mathrm{e}_6, \mathrm{e}_7)$	$\mathrm{E}_3 = \mathrm{XOR}(\mathrm{e}_5, \mathrm{e}_4)$
$\mathrm{E}_2 = \mathrm{XOR}(\mathrm{e}_3, \mathrm{e}_4)$	$\mathrm{E}_1 = \mathrm{XOR}(\mathrm{e}_2, \mathrm{e}_1)$	$\mathrm{E}_0 = \mathrm{XOR}(\mathrm{e}_0, \mathrm{e}_1)$
$\mathrm{t}_1 = \mathrm{XOR}(\mathrm{E}_9, \mathrm{E}_7)$	$\mathrm{t}_2 = \mathrm{XOR}(\mathrm{E}_1, \mathrm{t}_1)$	$\mathrm{t}_3 = \mathrm{XOR}(\mathrm{E}_2, \mathrm{t}_2)$
$\mathrm{t}_4 = \mathrm{XOR}(\mathrm{E}_5, \mathrm{E}_3)$	$\mathrm{t}_5 = \mathrm{XOR}(\mathrm{E}_4, \mathrm{t}_4)$	$\mathrm{t}_6 = \mathrm{XOR}(\mathrm{E}_4, \mathrm{E}_0)$
$\mathrm{t}_7 = \mathrm{XOR}(\mathrm{E}_{11}, \mathrm{E}_7)$	$\mathrm{t}_8 = \mathrm{XOR}(\mathrm{t}_1, \mathrm{t}_4)$	$\mathrm{t}_9 = \mathrm{XOR}(\mathrm{t}_1, \mathrm{t}_6)$
$\mathrm{t}_{10} = \mathrm{XOR}(\mathrm{E}_2, \mathrm{t}_5)$	$\mathrm{t}_{11} = \mathrm{XOR}(\mathrm{E}_{10}, \mathrm{E}_8)$	$\mathrm{t}_{12} = \mathrm{XNOR}(\mathrm{t}_3, \mathrm{t}_{11})$
$\mathrm{t}_{13} = \mathrm{XOR}(\mathrm{t}_{10}, \mathrm{t}_{12})$	$\mathrm{t}_{14} = \mathrm{XNOR}(\mathrm{t}_3, \mathrm{t}_7)$	$\mathrm{t}_{15} = \mathrm{XNOR}(\mathrm{E}_{10}, \mathrm{E}_6)$
$\mathrm{t}_{16} = \mathrm{XOR}(\mathrm{t}_6, \mathrm{t}_{14})$		

$$Sb = (t_{15}, t_{13}, t_8, t_{14}, t_{11}, t_9, t_{12}, t_{16})$$

C. Tower Field Inversion Optimization Scheme Based On Modular Series Method

According to the different parameters of the constructed tower field, there are many different combinations of operation modules. At the same time, the diversity of the tower field structure also provides a basis for designing efficient multiplication inverse circuits. The following will use the logical relationship between the sub-modules in the multiplication inverse and deep optimization techniques to give a method that compact module series connection.

According to the above SM4 S-box multiplication inverse scheme, the circuit design process can be divided into three steps:

1) The circuit is divided into several sub-modules through the constructed tower field.
2) Optimize the design of each sub-module based on algebraic or algorithms.
3) Cascade the optimized sub-modules according to the logical relationship.

In previous works [12],[13], the focus is generally on the study of step 1) and 2), and a simple cascade method is used for step 3). This paper proposes a new method to improve the gap between submodules, thereby reducing the overall depth of the multiplication inverse circuit. First, give our theorem of the proposed method.

Lemma: There are n signals $\{x_1, x_2, x_3 \dots x_n\}$, and $depth(x_i) = d_i (1 \le i \le n)$, then the minimum depth of the implemented Boolean circuit of $y = x_1 \oplus x_2 \oplus \dots \oplus x_n$ is $\left\lceil \log_2 \sum_{i=1}^{n} 2^{d^i} \right\rceil$, denoted by $mindepth(y)$.

The lemma gives the minimum depth to calculate a Boolean function, and this minimum depth is always reachable, that is, there exists a circuit with the minimum depth to calculate the Boolean function.

Theorem: There are n signals $\{x_1, x_2, x_3 \dots x_n\}$, and $depth(x_i) = d_i (1 \le i \le n)$, $op \in \{XOR, AND, OR \dots\}$, the minimum depth of the Boolean circuit implemented $y = x_1\ op\ x_2\ op \dots op\ x_n$ is $\left\lceil \log_2 \sum_{i=1}^{n} 2^{d^i} \right\rceil$.

According to the above theorem, this paper construct a signal connection model between modules. Assume that there are two modules A and B, and there is a cascade relationship between A and B: the output of A is the input of B. Module A have n output signals, denoted as $y_1, y_2, y_3 \dots y_n$, and they also enter module B as input. The algorithm model is shown in Fig. 6.

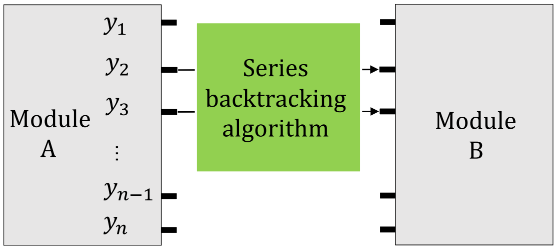


Fig.6. Signal cascade backtracking algorithm model

The signal series backtracking algorithm is described as follows, as shown in Fig. 7:

1) For the signal $y_i (1 \le i \le n)$, the signal is generated by its parent node through op_A. Determine whether it is involved in the operation op_A in module B. If not, skip the signal and judge the remaining signals; if yes, perform operation 2).
2) Initialize the set N_{y_i} in module A (including the parent node of y_i); perform backtracking operation on each node in N_{y_i}: if the node is obtained by its parent node through op_A, replace the node with its parent node; otherwise, backtracking terminates. Until any node in N_{y_i} does not meet the backtracking condition, the iteration terminates.
3) Replace all signals y_i in module B that have been traced back in step 2) with $N_1\ op_A\ N_2\ op_A \dots op_A\ N_m$, $N_j \in N_{y_i}$, $(1 \le j \le m)$.

Now $y_i (1 \le i \le n)$ in module B are obtained by a series of signals through the same operation, so the theorem is used in this paper give an implementation plan with the minimum depth.

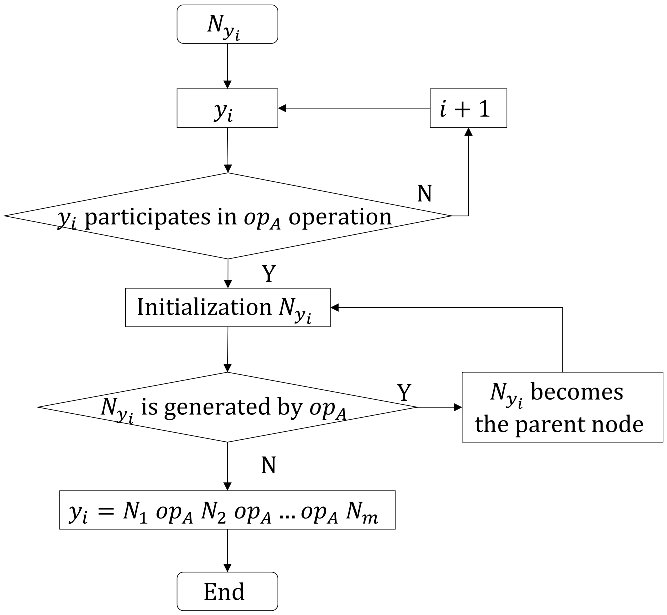

N_{y_i}
y_i
$i+1$
y_i participates in op_A operation
N
Y
Initialization N_{y_i}
N_{y_i} is generated by op_A
Y
N_{y_i} becomes the parent node
N
$y_i = N_1\ op_A\ N_2\ op_A \dots op_A\ N_m$
End

Fig. 7. Flowchart of signal series backtracking algorithm

According to the above signal cascade backtracking algorithm, we integrate the addition and multiplication operations in the inversion operation.

First, the SM4 S-box inverse operation Top module is designed with low depth.

From 3.1.2, we know that the finite field $GF(2^8)$elements are downgrade to $GF(2^4)$, and according to the tower field inversion expression, Let $\gamma_0, \gamma_1 \in GF((2)^2)^2$, in the tower field structure, be expressed as $\gamma_1 = [g_7, g_6, g_5, g_4]$, $\gamma_0 = [g_3, g_2, g_1, g_0]$.

Assume that,

$$m_9 = g_7 \oplus g_6, m_8 = g_3 \oplus g_2, m_7 = g_7 \oplus g_5, m_6 = g_3 \oplus g_1,$$
$$m_5 = g_6 \oplus g_4, m_4 = g_2 \oplus g_0,\ m_3 = g_7 \oplus g_6 \oplus g_5 \oplus g_4, m_2 = g_3 \oplus g_2 \oplus g_1 \oplus g_0, m_1 = g_5 \oplus g_4, m_0 = g_1 \oplus g_0$$

Since the calculation $\gamma_1\gamma_0\tau^2$ is as follows (24):

$$\gamma_1\gamma_0\tau^2 = (\Gamma_3 Z^4 + \Gamma_2 Z)(\Gamma_1 Z^4 + \Gamma_0 Z)(T^2 \cdot Z^4 + 1 \cdot Z)^2 \quad (24)$$

The (24) is converted to bit level and simplified as (25):

$$\begin{aligned}\gamma_1\gamma_0\tau^2 &= [s_3, s_2, s_1, s_0] \\ &= g_6g_2 \oplus g_7g_3 \oplus g_5g_1 \oplus m_1m_0 \\ &+ g_7g_3 \oplus m_9m_8 \oplus g_4g_0 \oplus m_1m_0 \\ &+ m_9m_8 \oplus g_7g_3 \oplus g_5g_1 \oplus m_1m_0 \\ &+ g_6g_2 \oplus m_9m_8 \oplus g_4g_0 \oplus m_1m_0\end{aligned} \quad (25)$$

Take the calculation of the top module $\gamma_1\gamma_0\tau^2 + (\gamma_1+\gamma_0)^2\upsilon$ as an example: this section describes in detail how to connect different modules in series to make the overall depth smaller.

To achieve the $\gamma_1\gamma_0\tau^2 + (\gamma_1+\gamma_0)^2\upsilon$, independent calculations between modules need to be calculated $(\gamma_1+\gamma_0)^2\upsilon = [z_3, z_2, z_1, z_0]$ and $\gamma_1\gamma_0\tau^2 = [s_3, s_2, s_1, s_0]$. Then the two modules are XORed, that is (26) ,

$$\begin{aligned}\gamma_1\gamma_0\tau^2 + (\gamma_1+\gamma_0)^2\upsilon &= [z_3 \oplus s_3, z_2 \oplus s_2, z_1 \oplus s_1, z_0 \oplus s_0] \\ &= [r_3, r_2, r_1, r_0]\end{aligned} \quad (26)$$

Then there is (27),

$$depth(r_i) = \max\big(depth(z_i), depth(s_i)\big) + 1 \quad (27)$$

But the signal between modules is connected in series by using the module compact series connection technology. The calculation scheme after the series connection is as follows (28):

$$\begin{aligned}r_3 &= z_3 \oplus s_3 = z_3 \oplus g_6g_2 \oplus g_7g_3 \oplus g_5g_1 \oplus m_1m_0 \\ r_2 &= z_2 \oplus s_2 = z_2 \oplus g_7g_3 \oplus m_9m_8 \oplus g_4g_0 \oplus m_1m_0 \\ r_1 &= z_1 \oplus s_1 = z_1 \oplus m_9m_8 \oplus g_7g_3 \oplus g_5g_1 \oplus m_1m_0 \\ r_0 &= z_0 \oplus s_0 = z_0 \oplus g_6g_2 \oplus m_9m_8 \oplus g_4g_0 \oplus m_1m_0\end{aligned} \quad (28)$$

Therefore, $mindepth(r_3)$ is determined by the depth of the 5 signals, that is, $z_3, g_6g_2, g_7g_3, g_5g_1, m_1m_0$the depth of .

From the theorem, it is easy to see (29),

$$mindetph(z_3, g_6g_2, g_7g_3, g_5g_1, m_1m_0) \leq \max(depth(z_3), mindepth(z_3, g_6g_2, g_7g_3, g_5g_1, m_1m_0) + 1 \quad (29)$$

Therefore, the module series connection solution we proposed can make the overall depth of the circuit smaller.

Using the above-mentioned backtracking and series connection mechanism, the input signals are integrated for calculation, and the top module is designed to implement the overall operation $\gamma_1\gamma_0\tau^2 + (\gamma_1+\gamma_0)^2\upsilon$, which is recorded as ϕ.

In summary, the optimized Boolean logic expression of the top module is as follows (30):

$$\begin{aligned}\phi &= g_6g_2 \oplus g_7g_3 \oplus g_5g_1 \oplus m_1m_0 \oplus m_7m_6 \oplus m_3m_2 \\ &+ g_7g_3 \oplus m_9m_8 \oplus g_4g_0 \oplus m_1m_0 \oplus m_5m_4 \oplus m_3m_2 \\ &+ m_9m_8 \oplus g_7g_3 \oplus g_5g_1 \oplus m_1m_0 \\ &+ g_6g_2 \oplus m_9m_8 \oplus g_4g_0 \oplus m_1m_0\end{aligned} \quad (30)$$

Based on the above Boolean logic expression, using the MiniSat solver [15], we find the final logic gate implementation result of $\gamma_1\gamma_0\tau^2 + (\gamma_1+\gamma_0)^2\upsilon$ as (31):

$$\begin{aligned}\phi = &\{\overline{g_6 \mid g_2} \oplus [\overline{g_7 \cdot g_3} \oplus (\overline{g_5 \cdot g_1} \oplus \overline{m_1 \cdot m_0})]\} \\ &\oplus [\overline{m_7 \mid m_6} \oplus \overline{m_3 \cdot m_2}] \\ &+ \{\overline{g_7 \mid g_3} \oplus [\overline{m_9 \cdot m_8} \oplus (\overline{g_4 \cdot g_0} \oplus \overline{m_1 \cdot m_0})]\} \\ &\oplus [\overline{m_5 \mid m_4} \oplus \overline{m_3 \cdot m_2}] \\ &+ \overline{m_9 \mid m_8} \oplus [\overline{g_7 \cdot g_3} \oplus (\overline{g_5 \cdot g_1} \oplus \overline{m_1 \cdot m_0})] \\ &+ \overline{g_6 \mid g_2} \oplus [\overline{m_9 \cdot m_8} \oplus (\overline{g_4 \cdot g_0} \oplus \overline{m_1 \cdot m_0})]\end{aligned} \quad (31)$$

We can see that implementing this expression requires 7 NANDs , 5 NORs , and 12 XORs .

Then, the S-box inverse operation Middle module is designed.

We design the Middle module to complete the inverse operation on F_{2^4}, denoted as $\lambda = \phi^{-1}$.

Assuming that the calculation result of the Top module is $\Phi = p_3 + p_2 + p_1 + p_0$, then there is (32),

$$\begin{aligned}\lambda &= p_2p_1p_0 \oplus p_2p_0 \oplus p_3p_0 \oplus p_3p_1 \oplus p_1 \\ &+ p_3p_1p_0 \oplus p_2p_0 \oplus p_3p_0 \oplus p_0 \oplus p_1 \\ &+ p_3p_2p_0 \oplus p_2p_0 \oplus p_2p_1 \oplus p_3p_1 \oplus p_3 \\ &+ p_3p_2p_1 \oplus p_2p_0 \oplus p_2p_1 \oplus p_2 \oplus p_3\end{aligned} \quad (32)$$

Because $MUX(s, a, b) = s \cdot a \oplus s \cdot b \oplus b$, we propose a MUX solution to replace the original logic gate as follows table IV:

TABLE IV
INVERSE OPERATION IMPLEMENTATION WITH MUX

$\mathrm{t}_9 = \mathrm{MUX}(\mathrm{t}_3, \mathrm{p}_1, \mathrm{t}_5)$	$\mathrm{t}_6 = \mathrm{MUX}(\mathrm{p}_2, \mathrm{t}_3, \mathrm{p}_3)$	$\mathrm{t}_3 = \mathrm{XNOR}(\mathrm{t}_1, \mathrm{t}_2)$
$\mathrm{t}_8 = \mathrm{MUX}(\mathrm{p}_0, \mathrm{t}_3, \mathrm{p}_1)$	$\mathrm{t}_5 = \mathrm{MUX}(\mathrm{p}_3, \mathrm{p}_0, 1)$	$\mathrm{t}_2 = \mathrm{NOR}(\mathrm{p}_3, \mathrm{p}_1)$
$\mathrm{t}_7 = \mathrm{MUX}(\mathrm{t}_3, \mathrm{p}_3, \mathrm{t}_4)$	$\mathrm{t}_4 = \mathrm{MUX}(\mathrm{p}_1, \mathrm{p}_2, 1)$	$\mathrm{t}_1 = \mathrm{NAND}(\mathrm{p}_2, \mathrm{p}_0)$

$$\lambda = (t_9, t_8, t_7, t_6)$$

To implement this inverse multiplication operation, 1 NAND , 1 NOR , 1 XNOR, and 6 MUXs are required.

Because there is (33),

$$MUX(s, a, 1) = s \cdot a \oplus s \oplus 1 = NANDN(a, s) \qquad (33)$$

So we have (34),

$$t_4 = NANDN(p_2, p_1), t_5 = NANDN(p_0, p_3) \qquad (34)$$

Finally, the low-depth design of the S-box inverse operation Bottom module is performed.

We design the Bottom module to implement $\lambda\gamma_1$and $\lambda\gamma_0$.

Assume $\lambda = l_3 + l_2 + l_1 + l_0$, first, the signal integration is performed using the compact module series connection method, the definition is as follows:

$$k_4 = l_3 \oplus l_2 , k_3 = l_3 \oplus l_1 , k_2 = l_2 \oplus l_0 ,$$
$$k_1 = l_3 \oplus l_2 \oplus l_1 \oplus l_0 , k_0 = l_1 \oplus l_0$$

Combining the above γ_1 and γ_2 , the calculation Bottom module has the following (35) and (36):

$$\begin{aligned}\lambda\gamma_1 = {} & \overline{g_6 \cdot l_2} \oplus \overline{g_7 \cdot l_3} \oplus \overline{m_3 \cdot k_1} \oplus \overline{m_5 \cdot k_2} \\ & + \overline{m_9 \cdot k_4} \oplus \overline{g_7 \cdot l_3} \oplus \overline{m_7 \cdot k_3} \oplus \overline{m_5 \cdot k_2} \\ & + \overline{g_4 \cdot l_0} \oplus \overline{g_5 \cdot l_1} \oplus \overline{m_3 \cdot k_1} \oplus \overline{m_5 \cdot k_2} \\ & + \overline{m_1 \cdot k_0} \oplus \overline{g_5 \cdot l_1} \oplus \overline{m_7 \cdot k_3} \oplus \overline{m_5 \cdot k_2} \qquad (35)\end{aligned}$$

$$\begin{aligned}\lambda\gamma_0 = {} & \overline{g_2 \cdot l_2} \oplus \overline{g_3 \cdot l_3} \oplus \overline{m_2 \cdot k_1} \oplus \overline{m_4 \cdot k_2} \\ & + \overline{m_8 \cdot k_4} \oplus \overline{g_3 \cdot l_3} \oplus \overline{m_6 \cdot k_3} \oplus \overline{m_4 \cdot k_2} \\ & + \overline{g_0 \cdot l_0} \oplus \overline{g_1 \cdot l_1} \oplus \overline{m_2 \cdot k_1} \oplus \overline{m_4 \cdot k_2} \\ & + \overline{m_0 \cdot k_0} \oplus \overline{g_1 \cdot l_1} \oplus \overline{m_6 \cdot k_3} \oplus \overline{m_4 \cdot k_2} \qquad (36)\end{aligned}$$

To achieve the 18-bit output signal, 18 NANDs are required. In addition, to achieve the signal integration based on the backtracking concatenation algorithm, 5 XORs are required.

At this point, we have completed the tower field inversion operation.

IV. RESULTS AND COMPARISON

A. Synopsys Design Compiler Synthesis Results

We performed synthesis used Synopsys Design Compiler and comparied with other SM4 S-box implementations. The synthesis results of this implementation are shown in table 5:

TABLE V
NUMBER OF CIRCUIT COMPONENTS USED IN EACH MODULE

Module	Circuit components						
	XOR	NOR	XNOR	NAND	NOT	MUX	NAND
Input Module	13		4		1		
Top module	12	5		7			
Middle module		1	1	1		6	
		1	1	1		4	2
Bottom module	5			18			
Output Module	25		3				

As shown in table 6, the number of circuit components in this design has been greatly improved compared with previous research work.

TABLE VI
NUMBER OF CIRCUIT COMPONENTS USED IN DIFFERENT DESIGN SCHEMS

Literature	Circuit components							
	XOR/XNOR	NOR	AND	NAND	NOT	MUX	NAND	No.
Mar.-Her.【16】	99		58		11			168
Abbasi and Afza【17】	134		36		10			180
Bai et al.【18】	157		63		10			230
Liu.【19】	85		34		14			133
	80		34		9			123
Saarinen.M.O【20】	95		34					129
This paper	63	6		26	1	6		102
						4	2	102

Due to the difference in process technology, the same circuit element has a large difference in different process libraries. Generally, we use the unit of gate equalization (GE) to measure the area. 1GE represents the area of a NAND gate with 2 input signal and a 1 driving signal in the same process library.

The comprehensive results of SMIC130nm and SMIC60nm processes are shown in table 7 and Fig. 8. Although the depth is greater than [19], it has achieved a balance of minimizing the area while minimizing the depth.

TABLE VII
COMPREHENSIVE RESULTS UNDER DIFFERENT DESIGN SCHEEMES

Implementation	Performance		
	Area GE/SMIC 130nm	Area GE/SMIC 65nm	Depth
MartínezHerrera 【 16 】	315.18	318	32
Abbasi and Afzal 【17】	366.8	363	37
Bai et al. 【18】	456.3	455.25	46
Liu. 【19】	252.65	252.75	17
	237.65	237.75	24
Saarinen.M.O 【20】	266.57	264.75	25
This paper	195.48	188	19
	192.8	186.5	19

Compared with previous research results, the SM4 S box implemented using our solution has been greatly improved and can reach the best level in the same field.

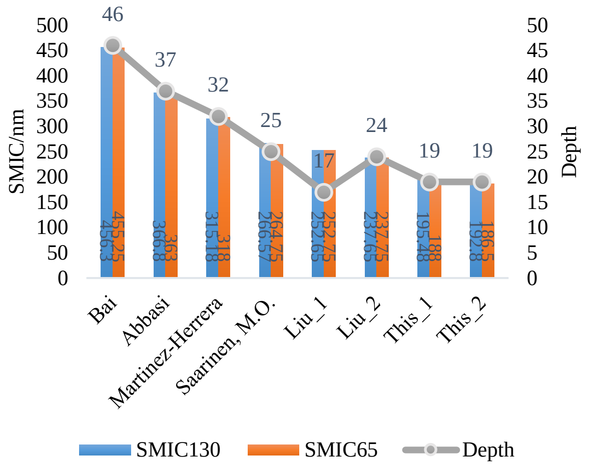
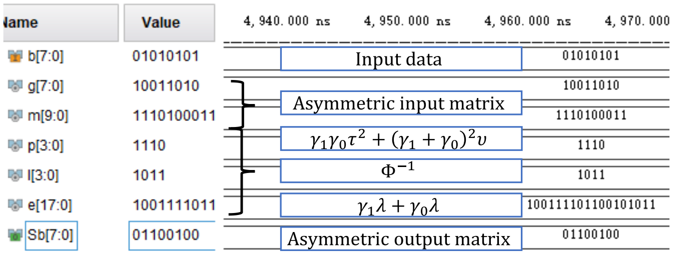


Fig. 8. Comprehensive results under different design schemes

B. Xilinx Vivado Tool Synthesis Result

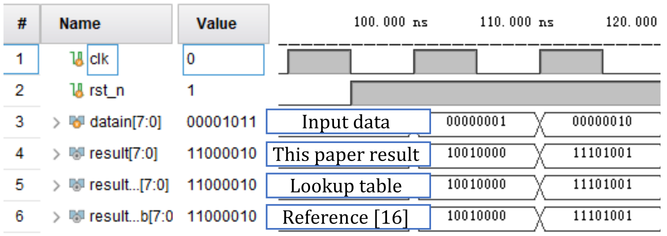
We simulated and synthesized the design through Xilinx Vivado tool to verify the correctness and efficiency of the scheme in this paper. First, we conducted SM4 S-box simulation test. The simulation of S-box optimization idea in this paper as shown in Fig. 9, including input and output matrix modules and inverse operation modules.



Name	Value
b[7:0]	01010101
g[7:0]	10011010
m[9:0]	1110100011
p[3:0]	1110
l[3:0]	1011
e[17:0]	1001111011
Sb[7:0]	01100100

Fig. 9. SM4 S-box input and output module and inverse operation module simulation

We observed the SM4 S-box output of different schemes by customizing the input . The simulation results are shown in Fig. 10. We can see that our results are the same as those in the reference, indicating that our solution is correct.



#	Name	Value
1	clk	0
2	rst_n	1
3	datain[7:0]	00001011
4	result[7:0]	11000010
5	result...[7:0]	11000010
6	result...b[7:0	11000010

Fig. 10. SM4 S-box output of different designs

In order to verify the functional correctness and performance advantages of our design, we designed a complete SM4 architecture. Overview of internal modules as the Fig. 11 shows, keyexp module implements key expansion algorithm; decenc module implements nonlinear iterative algorithm for encryption and decryption.

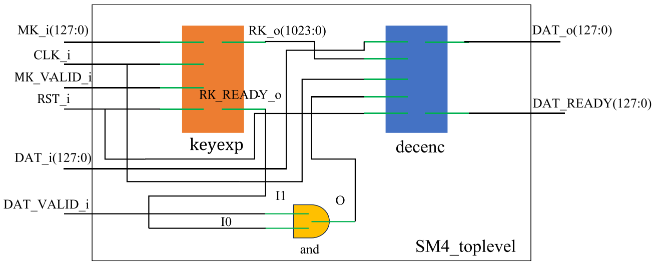
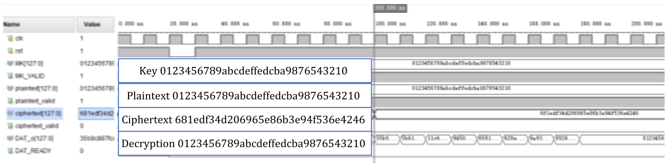


Fig. 11. SM4 top level design

The overall SM4 test was conducted, and the simulation results are shown in Fig. 12. During the test, the plaintext and key are both 0123456789abcdeffedcba9876543210, the ciphertext is 681edf34d206965e86b3e94f536e4246, and the decrypted data is 0123456789abcdeffedcba9876543210.



Key 0123456789abcdeffedcba9876543210

Plaintext 0123456789abcdeffedcba9876543210

Ciphertext 681edf34d206965e86b3e94f536e4246

Decryption 0123456789abcdeffedcba9876543210

Name	Value
clk	1
rst	1
MK[127:0]	0123456789
MK_VALID	1
plaintext[127:0]	0123456789
plaintext_valid	1
ciphertext[127:0]	681edf34d2
ciphertext_valid	0
DAT_o[127:0]	35b9c887fc
DAT_READY	0

Fig. 12. SM4 simulation test results

Finally, we obtained the resource occupancy of the S-box in this paper and the traditional scheme based on the Xilinx Vivado tool, as shown in Fig. 13. The scheme in this paper achieves zero BRAM resource occupancy, greatly saving resource occupancy.

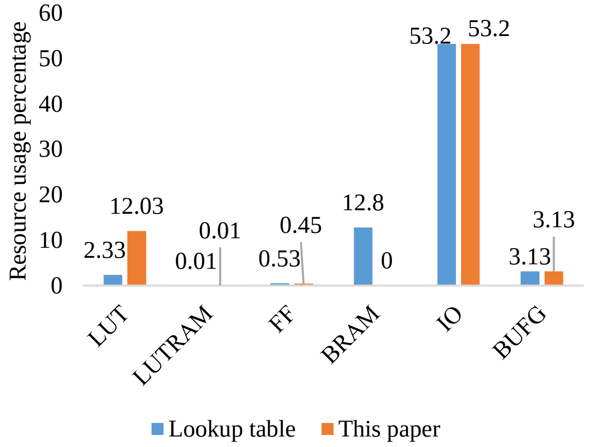


Fig. 13. Comparison of resource between the proposed S-box and the lookup table S-box

In the same scenario, we analyze the energy consumption of two different implementation schemes, as shown in Fig. 14. It is obvious that the S-box implemented in this paper consumes less energy and can effectively resist energy analysis attacks.

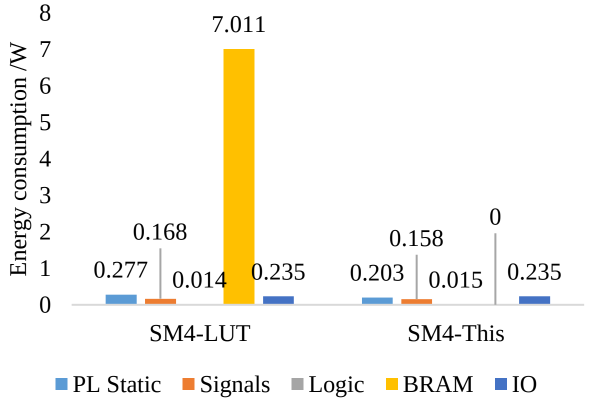


Fig. 14. Comparison of energy consumption between the S-box in this paper and the S-box with a lookup table.

In order to verify the correctness and effectiveness of the SM4 S-box method on commercial hardware, we built a real SM4 test platform and implemented our design on it. The test platform consists of encryption and decryption equipment and network analysis tools. The encryption and decryption equipment is MLK-ZYNQ-MZ7035FA, the laptop model is Dell G3 15 3590, and the network analysis tool is Wireshark4.0.8. The network is built according to the topology shown in the figure 14. The frame generator sends the original frame, performs encryption and decryption processes respectively, and finally sends it to the network analysis tool.

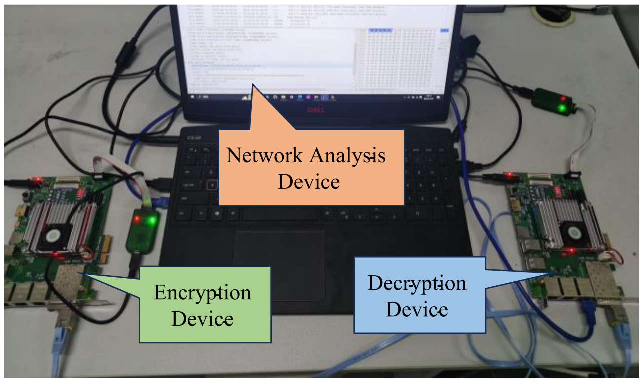


Fig. 14. Comparison of resource between the proposed S-box and the lookup table S-box

After the encryption module completes the encryption, it is sent to the Ethernet function module. The FPGA is connected to the computer through the hardware network port using a network cable. Wireshark is run on the computer to capture the data frames on the network port within 10 seconds, and the data content captured by the network port is observed to verify the encryption result. The Ethernet rate in the actual test bench is 1Gbps. After wireshark analysis, the throughput of the encryption board device is 890Mbps and the throughput of the decryption device is 860Mbps. As shown in Fig. 15 and Fig. 16:

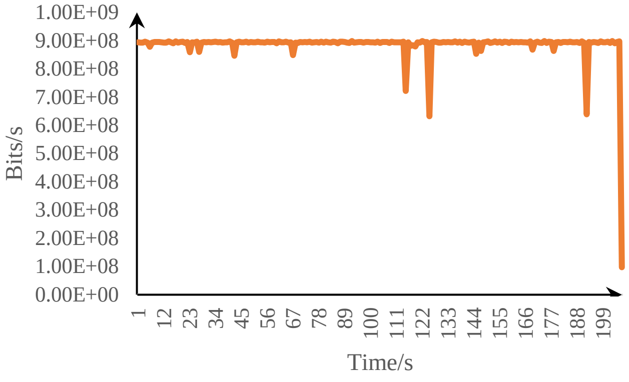


Fig. 15. Encryption throughput statistics from wireshark

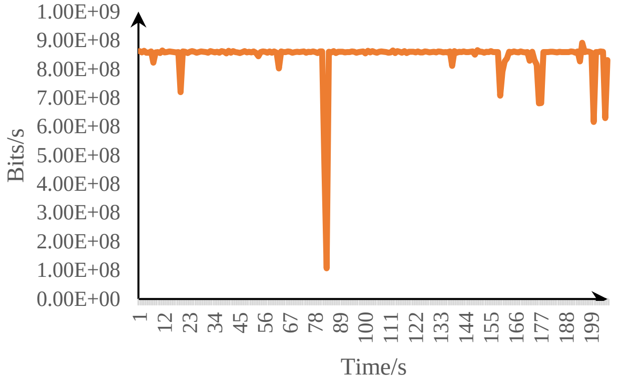


Fig. 16. Decryption throughput statistics from wireshark

V. Conclusion

We proposed a new combinational logic design for SM4 S-boxe, implement and evaluate the scheme. This scheme proposes a new combinational logic technology to optimize area and depth. This technology includes: proposing an asymmetric transformation matrix based on preserving the multiplicative inverse for SM4 S-box; solving the asymmetric transformation matrix by a compromise between exhaustive algorithm and heuristic algorithm with a maximum depth constraint; proposing a low-latency sub-module series connection method based on a backtracking judgment mechanism to implement a tower field inversion for SM4 S-boxe. We evaluate the design scheme. Compared with previous work, we have achieved a design with the least circuit elements, and the best area and depth in the ASIC platform. In addition, we simulated the SM4 algorithm containing this S-box scheme and compared it with the most commonly used lookup table scheme on the FPGA platform. It occupies less resources, consumes less energy, and can better resist energy analysis attacks.

Acknowledgment

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” In most cases, sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page, not here.

References

[1] https://www.iso.org/standard/81564.html

[2] Perrin L. Cryptanalysis, Reverse-Engineering and Design of Symmetric Cryptographic Algorithms [D]. University of Luxembourg, 2017.

[3] Perrin L. Cryptanalysis, Reverse-Engineering and Design of Symmetric Cryptographic Algorithms [D]. University of Luxembourg, 2017.

[4] Boyar J, Matthews P, Peralta R. On the Shortest Linear Straight-Line Program for Computing Linear Forms [C]. Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008, Torun, Poland, August 25-29, 2008, Proceedings 168-179.

[5] Christof Paar. Optimized arithmetic for Reed-Solomon encoders. 04 1997.

[6] Boyar J, Peralta R. A New Combinational Logic Minimization Technique with Applications to Cryptology [C]. Experimental Algorithms, 9th International Symposium, SEA 2010, Ischia Island, Naples, Italy, May 20-22, 2010. Proceedings, 2010, 178-189.

[7] Maximov A. AES MixColumn with 92 XOR gates [J] . IACR Cryptology ePrint Archive, 2019, 2019:833.

[8] Ueno R, Homma N, Sugawara Y, et al. Highly Efficient GF(2) Inversion Circuit Based on Redundant GF Arithmetic and Its Application to AES Design [C]. Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings, 2015:63-80.

[9] Ueno R, Homma N, Nogami Y, et al. Highly efficient GF(2) inversion circuit based on hybrid GF representations [J] . J. Cryptographic Engineering, 2019, 9(2):101-113.

[10] Reyhani-Masoleh A, Taha M M I, Ashmawy D. Smashing the Implementation Records of AES S-box [J] . IACR Trans. Cryptogr. Hardw. Embed. Syst, 2018, 2018(2):298-336.

[11] Reyhani-Masoleh A, Taha M M I, Ashmawy D. New Area Record for the AES Combined S-Box/Inverse S-Box [C]. 25th IEEE Symposium on Computer Arithmetic, ARITH 2018, Amherst, MA, USA, June 25-27, 2018. 2018b:145-152.

[12] D. Canright. A very compact S-Box for AES. In Josyula R. Rao and Berk Sunar, editors, Cryptographic Hardware and Embedded Systems – CHES2005, pages 441–455, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[13] Joan Boyar and René Peralta. A small depth-16 circuit for the AES S-Box. In Dimitris Gritzalis, Steven Furnell, and Marianthi Theoharidou, editors, SEC,volume 376 of IFIP Advances in Information and Communication Technology,pages 287–298. Springer, 2012.

[14] Boyar, J., Matthews, P. & Peralta, R. Logic Minimization Techniques with Applications to Cryptology. *J Cryptol* **26**, 280–312 (2013).

[15] http://minisat.se/

[16] Mart´ınez-Herrera, A.F., Mex-Perera, J.C. and Nolazco-Flores, J.A. (2013) ‘Merging the Camellia, SMS4 and AES S-Boxes in a single S-Box with composite bases’, Information Security, 16th International Conference, ISC 2013, Proceedings, 13–15 November, Dallas, Texas, USA, pp.209–217.

[17] Abbasi, I. and Afzal, M. (2011) 'A compact S-Box design for SMS4 block cipher', IT Convergence and Services, Vol. 107, pp.641–658.

[18] Bai, X., Xu, Y. and Guo, L. (2009) 'Securing SMS4 cipher against differential power analysis and its VLSI implementation', IEEE Singapore International Conference on Communication Systems, pp.167–172.

[19] Liu Jian. Research on the Implementation Optimization Method of Two Types of Cryptographic Components[D]. Information Engineering University of Strategic Support Force, 2020.

[20] Saarinen, MO (2020). A Lightweight ISA Extension for AES and SM4. ArXiv, abs/2002.07041.

First A. Author (Fellow, IEEE) and all authors may include biographies if the publication allows. Biographies are often not included in conference-related papers. Please check the Information for Authors to confirm. Author photos should be current, professional images of the head and shoulders. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author's educational background is listed. The degrees should be listed with the type of degree in what field, which institution, city, state, and country, and year the degree was earned. The author's major field of study should be lowercase.

The second paragraph uses the preferred third person pronoun (he, she, they, etc.) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. The format for listing publishers of a book within the biography is: *Title of Book* (publisher name, year) similar to a reference. Current and previous research interests end the paragraph.

Second B. Author, photograph and biography not available at the time of publication.

Third C. Author Jr. (Member, IEEE), photograph and biography not available at the time of publication.