Construction of a Low Multiplicative Complexity GF (2^4) Inversion Circuit for Compact AES S-Box

Construction of a Low Multiplicative Complexity $GF(2^4)$ Inversion Circuit for Compact AES S-Box

Jia Jun Tay
Faculty of Engineering, Computing and Science
Swinburne University of Technology Sarawak Campus
Sarawak, Malaysia
e-mail: jtay@swinburne.edu.my

M. L. Dennis Wong
Heriot-Watt University Malaysia
Putrajaya, Malaysia
e-mail: D.Wong@hw.ac.uk

Ming Ming Wong
School of Computer Science and Engineering
Nanyang Technological University
Nanyang Avenue, Singapore

Cishen Zhang, Ismat Hijazin
Faculty of Science, Engineering and Technology
Swinburne University of Technology Hawthorn Campus
Victoria, Australia

***Abstract*—In this work, we construct a compact composite AES S-Box by deriving a new low multiplicative complexity $GF(2^4)$ inversion circuit. A deterministic tree search algorithm is applied to search for constructions that are optimum in terms of multiplicative complexity. From the results, the circuit with the smallest gate count is selected for $GF(2^4)$ inversion. To the best of our knowledge, the proposed AES S-Box requires the smallest gate count to date with the size of 112 gates and depth of 25 gates.**

***Index Terms*—Advanced Encryption Standard (AES), S-Box, composite field arithmetic (CFA), low multiplicative complexity.**

I. Introduction

In this age of ubiquitous computing, security and privacy concerns surrounding digital communication and sensitive data storage are legitimate challenges. The Advanced Encryption Standard (AES) was announced in 2001 by the National Institute of Standards and Technology (NIST) [1]. To this day, AES remains the most popular block cipher in many security systems. To enable applications in heavily constrained environments, various optimizations had been made to the AES circuit since its introduction to minimize circuit area.

Several studies on the block cipher had revealed the substitution process to be the major bottleneck in circuit optimization. To minimize circuit complexity for the transformation, many works have experimented with new composite field for the AES S-Box [2]–[7]. Recently, works in [8]–[11] attempted to optimize the combinational circuit in the composite AES S-Box using a low multiplicative complexity heuristic. The results reported showed significant improvement in either size or depth compared to previous works. However, due to the randomness involved in the optimization algorithm used, it is possible that better solutions can be derived using the low multiplicative complexity heuristic.

In this work, we explore the same tower field architecture used in [8] and use a deterministic tree search algorithm on the $GF(2^4)$ inversion circuit. The purpose of the algorithm is to identify a large set of solutions that are optimum in terms of multiplicative complexity to perform the $GF(2^4)$ inversion. From the solutions, we select the circuit with the smallest gate count as our proposed AES S-Box.

The rest of the paper is organized as follows. The chosen tower field architecture for our AES S-Box is described in Section II. Section III explains the methodology used to optimize the $GF(2^4)$ inversion circuit for our proposed AES S-Box. The complete circuit of our proposed AES S-Box is described in Section IV along with comparison against selected previous works. Finally, concluding remarks are drawn in Section V.

II. Derivation of the Composite Field

The substitution process in AES encryption (a.k.a. S-Box function) is widely known to be the most demanding operation in the block cipher. The operation involves finding the multiplicative inverse of the 8-bit input over $GF(2^8)$ followed by an affine transformation.

Optimizing the multiplicative inverse circuit over $GF(2^8)$ is a difficult problem due to the high order of the finite field. Therefore, a common approach is to map the element in $GF(2^8)$ to a subfield of lower order with the goal of reducing complexity in the calculation of its multiplicative inverse. This mapping process is facilitated through the use of composite field arithmetic (CFA). CFA allows a composite field such as the $GF(2^8)$ field of the AES S-Box to be built iteratively from fields of lower order. This means that the actual mathematical manipulation on the data can be done in the subfields instead of the original field. As a result, this approach allows the multiplicative inverse process in the AES S-Box to be constructed in the less complex subfields.

In general, the process to implement multiplicative inverse in lower fields using CFA can be summarized in three steps:

1) Establish an isomorphism function to map all elements of the original field to a subfield.
2) Compute the multiplicative inverse over the subfield.
3) Establish an inverse isomorphism function to map the results to the original field.

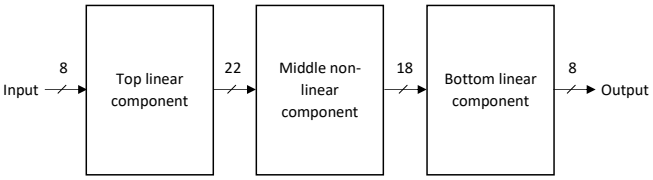

Fig. 1. The AES S-Box as a three-part circuit.

The mapping of $GF(2^8)$ to $GF(((2^2)^2)^2)$ necessitates three stages of isomorphism and the associated field polynomials. With reference to [7], these can be stated in general form as follow:

- Isomorphism for $GF(2^8)/GF(2^4)$: $r(y) = y^2 + \tau y + v$
- Isomorphism for $GF(2^4)/GF(2^2)$: $s(z) = z^2 + Tz + N$
- Isomorphism for $GF(2^2)/GF(2)$: $t(w) = w^2 + w + 1$

In this work, we use the tower field construction in [8] as the foundation for optimization. The same tower field construction was also used in [9], [11]. The composite field AES S-Box architectures are commonly presented in block diagrams detailing the isomorphism and inverse isomorphism circuits, squarers and scalers, as well as adders and multipliers. However, it is easier to view the tower field architecture as a three-part circuit composing of: (a) a top linear component, (b) a middle non-linear component (which includes the $GF(2^4)$ inversion), and (c) a bottom linear component as illustrated in Figure 1.

The top linear component and bottom linear component are 8-to-22-bit and 18-to-8-bit transformations respectively. They are "linear" in the sense that they only involve additions (XOR) but not multiplications. The transformations can be viewed as multiplying the respective input with the matrices U in (1) and B in (2) respectively.

Boyar and Peralta [8] utilize a short linear program (SLP) heuristic that allows for XOR-cancellation to optimize the top linear and bottom linear components. The resultant linear circuits are fairly compact and we do not observe potential for further improvement without significant alteration to the tower field construction. However, the $GF(2^4)$ inversion circuit that is part of the middle non-linear component can be further optimized for lower gate count.

III. Low Multiplicative Complexity $GF(2^4)$ Inversion Circuit

Composite field arithmetic allows multiplicative inversion of the AES S-Box to be implemented in a subfield of lower order. Using the tower field construction specified in [8], the resultant $GF(2^4)$ inversion can be represented by the following polynomials, where $x_1, x_2, ..., x_4$ represent the 4-bit input and $y_1, y_2, ..., y_4$ represent the inverted output (note that additions are synonymous to XOR and multiplications are synonymous to AND in the expressions):

- $y_1 = x_2x_3x_4 + x_1x_3 + x_2x_3 + x_3 + x_4$
- $y_2 = x_1x_3x_4 + x_1x_3 + x_2x_3 + x_2x_4 + x_4$
- $y_3 = x_1x_2x_4 + x_1x_3 + x_1x_4 + x_1 + x_2$
- $y_4 = x_1x_2x_3 + x_1x_3 + x_1x_4 + x_2x_4 + x_2$

The next task is to construct a compact circuit to perform the functions described above. For this purpose, Boyar and Peralta [8] proposed a low multiplicative complexity heuristic to solve for low gate count $GF(2^4)$ inversion circuit. The low multiplicative complexity heuristic implies that an implementation of a function using the minimal number of AND gates often results in close to optimal gate count for the function. In the original approach, a two-step algorithm [12] is used to solve for low multiplicative complexity implementations for the $GF(2^4)$ inversion circuit. However, due to the randomized selection process involved in the non-linear step (first step) of the algorithm, there are uncertainties regarding the optimality of the results.

In this work, we construct the $GF(2^4)$ inversion circuit based on the same low multiplicative complexity heuristic but with a deterministic tree search algorithm [13] in place of the original non-linear step.

In mathematics, decomposition (or factorization) is used to reduce the number of multiplications in an expression. Since the polynomial expressions for the $GF(2^4)$ inversion are essentially Exclusive-OR Sum-of-Products (ESOP), decomposition can also be performed to minimize the number of multiplications as demonstrated in [14].

The proposed tree search algorithm in [13] operates on the polynomial expression that describes a function to be optimized. The algorithm then attempts multiple stages of decomposition to arrive at an expression with the minimal number of multiplications. For example, the expression y_1 from Section II can be realized using only two multiplications after decomposition as shown below:

$$y_1 = x_3(x_2x_4 + x_1 + x_2) + x_3 + x_4$$

Since the $GF(2^4)$ inversion is a multiple-expression problem, the tree search algorithm would solve each expression sequentially while keeping a collective set of product terms from solved expression(s) to facilitate product sharing. The collected product terms can be added (XOR) to subsequent expression(s) to essentially change the target expression in hope of enabling solution(s) with less multiplications. For instance, the expression y_2 from Section II normally requires at least two multiplications to compute. However, if product sharing is enabled with y_1, the expression can then be realized with a single multiplication as shown below, where p_1 and p_2 are product terms that are shared with y_1:

$$y_2 = (x_3 + x_4)(\underbrace{x_3(x_2x_4 + x_1 + x_2)}_{p_1} + \underbrace{x_2x_4}_{p_2}) + x_4$$

After solving for all four expressions, the algorithm returns a set of optimum solutions. Note that the solutions are "optimum" in terms of multiplicative complexity but not gate count. As such, a last step is performed to select the "optimum" solution with the smallest number of gate count from the set. The overall approach to optimizing the $GF(2^4)$ inversion circuit can be summarized in Figure 2.

$$U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{1}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{2}$$

IV. Results and Discussions

The three components of the optimized AES S-Box circuit are described in Figures 3, 4 and 5 respectively.

Comparisons of circuit complexities between our optimized AES S-Box and selected previous works are summarized in Table I. The metrics of interest include *size* which represents the total gate count of the circuit and *depth* which indicates the number of gates that exist along the critical path of the circuit.

From Table I, we can observe that the proposed AES S-Box achieves the smallest total gate count compared to the other works. Specifically, against the results presented in [10], our proposed circuit managed to shave off three additional gates while reducing the circuit depth by the same amount. In particular, the improvement to the gate count is several-fold in a full AES circuit as multiple instances of the 8-bit S-Box are commonly used to encrypt the 128-bit data block as well as for key scheduling purposes [15].

We are aware that the search technique in [10] can potentially discover the same results presented in this work. However, due to the randomness involved in the selection process, it may be difficult for the search technique to arrive at a specific solution. On the contrary, the approach of our work explores all potential solutions achievable through polynomial decomposition that are optimal in terms of multiplicative

TABLE I
Comparison of circuit complexities between the proposed circuit and previous works

Work	Size				Depth
	AND	XOR	XNOR	Total	
[4]	36	126	0	162	29
[5]	36	91	0	127	27
Case I [7]	36	118	0	154	32
Case II [7]	36	106	0	142	26
Case III [7]	36	96	0	132	24
[9]	34	90	4	128	16
[10]	32	79	4	115	28
[11]*	-	-	-	125	16
This Work	32	76	4	112	25

* Information on the number of individual gates is not available.

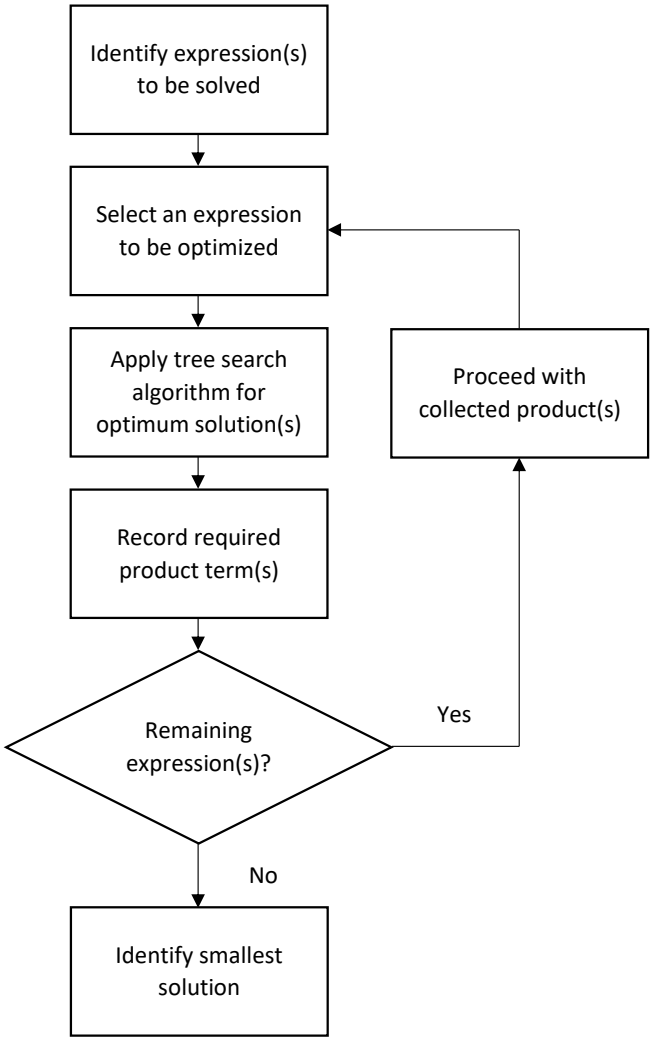

Fig. 2. Flow of methodology to optimize the AES $GF(2^4)$ inversion circuit.

complexity. This allows us to identify the proposed circuit as the smallest implementation among the alternatives.

On the other hand, results reported in [9] and [11] showed an advantage in terms of circuit depth at the cost of higher gate count. This is a classical example of trade-offs associated with VLSI designs. The low multiplicative complexity logic minimization algorithm used in this work is a heuristic aimed towards low gate count circuits. As such, circuit depth is not a priority when searching for the best solution. Consequently, the proposed design is more suitable for constrained applications where hardware limitations are more prevalent while low-depth variants are more suitable for high speed applications.

V. Conclusion

We presented a compact composite field construction for the AES S-Box with optimization on the $GF(2^4)$ inversion circuit using the low multiplicative complexity heuristic. A tree search algorithm [13] based on polynomial factorization and product sharing is used to solve for circuits with optimal multiplicative complexity. The best construction with the smallest gate count is then selected from the set of solutions. The proposed AES S-Box shows improvement in both size and depth compared to the smallest result reported previously in [10]. Our future work will involve experiments of the low multiplicative complexity heuristic on other tower field architectures of the AES S-Box using different isomorphic mapping.

Acknowledgment

This work has been supported in part by the Melbourne-Sarawak Research Collaboration Scheme.

References

[1] *Advanced Encryption Standard (AES)*, National Institute of Standards and Technology Federal Inf. Process. Stds. (NIST FIPS) - 197, 2001.

[2] V. Rijmen, "Efficient implementation of the rijndael s-box," 07 2000.

[3] A. Rudra, P. K. Dubey, C. S. Jutla, V. Kumar, J. R. Rao, and P. Rohatgi, "Efficient rijndael encryption implementation with composite field arithmetic," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, Ç. K. Koç, D. Naccache, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 171–184.

[4] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact rijndael hardware architecture with s-box optimization," in *Advances in Cryptology — ASIACRYPT 2001*, C. Boyd, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 239–254.

[5] D. Canright, "A very compact s-box for aes," in *Cryptographic Hardware and Embedded Systems – CHES 2005*, J. R. Rao and B. Sunar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 441–455.

[6] X. Zhang and K. K. Parhi, "On the optimum constructions of composite field for the aes algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 10, pp. 1153–1157, Oct 2006.

[7] M. M. Wong, M. L. D. Wong, A. K. Nandi, and I. Hijazin, "Construction of optimum composite field architecture for compact high-throughput aes s-boxes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 6, pp. 1151–1155, June 2012.

[8] J. Boyar and R. Peralta, "A new combinational logic minimization technique with applications to cryptology," in *Experimental Algorithms*, P. Festa, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 178–189.

[9] ——, "A small depth-16 circuit for the aes s-box," in *Information Security and Privacy Research*, D. Gritzalis, S. Furnell, and M. Theoharidou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 287–298.

[10] J. Boyar, P. Matthews, and R. Peralta, "Logic minimization techniques with applications to cryptology," *Journal of Cryptology*, vol. 26, no. 2, pp. 280–312, Apr 2013. [Online]. Available: https://doi.org/10.1007/s00145-012-9124-7

[11] J. Boyar, M. G. Find, and R. Peralta, "Small low-depth circuits for cryptographic applications," *Cryptography and Communications*, Mar 2018. [Online]. Available: https://doi.org/10.1007/s12095-018-0296-3

[12] R. Peralta and J. Boyar, "Method of optimizing combinational circuits," Nov. 20 2012, uS Patent 8,316,338. [Online]. Available: https://www.google.com/patents/US8316338

[13] J. J. Tay, M. D. Wong, M. M. Wong, C. Zhang, and I. Hijazin, "A tree search algorithm for low multiplicative complexity logic design," *Future Generation Computer Systems*, vol. 83, pp. 132 – 143, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17320010

[14] J. J. Tay, M. M. Wong, M. L. D. Wong, C. Zhang, and I. Hijazin, "A novel approach to low multiplicative complexity logic design," in *2017 International Conference on Consumer Electronics and Devices (ICCED)*, July 2017, pp. 31–35.

[15] J. J. Tay, M. M. Wong, and I. Hijazin, "Compact and low power aes block cipher using lightweight key expansion mechanism and optimal number of s-boxes," in *2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Dec 2014, pp. 108–114.

$U_0 = x_4 + x_2$	$U_1 = x_7 + x_1$	$U_2 = x_7 + x_4$
$U_3 = x_7 + x_2$	$U_4 = x_6 + x_5$	$U_5 = U_4 + x_0$
$U_6 = U_5 + x_4$	$U_7 = U_1 + U_0$	$U_8 = U_5 + x_7$
$U_9 = U_5 + x_1$	$U_{10} = U_9 + U_3$	$U_{11} = x_3 + U_7$
$U_{12} = U_{11} + x_2$	$U_{13} = U_{11} + x_6$	$U_{14} = U_{12} + x_0$
$U_{15} = U_{12} + U_4$	$U_{16} = U_{13} + U_2$	$U_{17} = x_0 + U_{16}$
$U_{18} = U_{15} + U_{16}$	$U_{19} = U_{15} + U_3$	$U_{20} = U_4 + U_{16}$
$U_{21} = U_1 + U_{20}$	$U_{22} = x_7 + U_{20}$	

Fig. 3. Top linear component of the optimized AES S-Box. 8-bit inputs are $x_0, x_1, ..., x_7$. 22-bit outputs are $x_0, U_0, U_1, ..., U_{22}$ excluding U_4 and U_{11}.

$M_0 = U_7 \times U_{12}$	$M_1 = U_{10} \times U_{14}$	$M_2 = M_1 + M_0$
$M_3 = U_6 \times x_0$	$M_4 = M_3 + M_0$	$M_5 = U_1 \times U_{20}$
$M_6 = U_9 \times U_5$	$M_7 = M_6 + M_5$	$M_8 = U_8 \times U_{17}$
$M_9 = M_8 + M_5$	$M_{10} = U_2 \times U_{16}$	$M_{11} = U_0 \times U_{18}$
$M_{12} = M_{11} + M_{10}$	$M_{13} = U_3 \times U_{15}$	$M_{14} = M_{13} + M_{10}$
$M_{15} = M_2 + U_{13}$	$M_{16} = M_4 + M_{14}$	$M_{17} = M_7 + M_{12}$
$M_{18} = M_9 + M_{14}$	$M_{19} = M_{15} + M_{12}$	$M_{20} = M_{16} + U_{19}$
$M_{21} = M_{17} + U_{21}$	$M_{22} = M_{18} + U_{22}$	
$M_{23} = M_{20} \times M_{22}$	$M_{24} = M_{19} + M_{20}$	$M_{25} = M_{23} + M_{24}$
$M_{26} = M_{21} \times M_{25}$	$M_{27} = M_{21} + M_{22}$	$M_{28} = M_{26} + M_{27}$
$M_{29} = M_{26} + M_{23}$	$M_{30} = M_{29} \times M_{27}$	$M_{31} = M_{22} + M_{30}$
$M_{32} = M_{27} + M_{23}$	$M_{33} = M_{19} \times M_{32}$	$M_{34} = M_{24} + M_{33}$
$M_{35} = M_{23} + M_{34}$	$M_{36} = M_{35} \times M_{24}$	$M_{37} = M_{19} + M_{36}$
$M_{38} = M_{34} + M_{28}$	$M_{39} = M_{37} + M_{31}$	$M_{40} = M_{37} + M_{34}$
$M_{41} = M_{31} + M_{28}$	$M_{42} = M_{39} + M_{38}$	$N_0 = M_{41} \times U_{12}$
$N_1 = M_{28} \times U_{14}$	$N_2 = M_{31} \times x_0$	$N_3 = M_{40} \times U_{20}$
$N_4 = M_{34} \times U_5$	$N_5 = M_{37} \times U_{17}$	$N_6 = M_{39} \times U_{16}$
$N_7 = M_{42} \times U_{18}$	$N_8 = M_{38} \times U_{15}$	$N_9 = M_{41} \times U_7$
$N_{10} = M_{28} \times U_{10}$	$N_{11} = M_{31} \times U_6$	$N_{12} = M_{40} \times U_1$
$N_{13} = M_{34} \times U_9$	$N_{14} = M_{37} \times U_8$	$N_{15} = M_{39} \times U_2$
$N_{16} = M_{42} \times U_0$	$N_{17} = M_{38} \times U_3$	

Fig. 4. Middle non-linear component of the optimized AES S-Box. 22-bit inputs are $x_0, U_0, U_1, ..., U_{22}$ excluding U_4 and U_{11}. 18-bit outputs are $N_0, N_1, ..., N_{17}$. Multiplicative inversion in $GF(2^4)$ is represented by M_{23} through M_{37}.

$B_0 = N_{15} + N_{16}$	$B_1 = N_{10} + B_0$	$B_2 = N_9 + B_1$
$B_3 = N_0 + N_2$	$B_4 = N_1 + N_0$	$B_5 = N_3 + N_4$
$B_6 = N_{12} + B_3$	$B_7 = N_7 + B_5$	$B_8 = N_8 + B_6$
$B_9 = B_7 + B_8$	$B_{10} = B_5 + B_4$	$B_{11} = N_3 + N_5$
$B_{12} = N_{13} + B_0$	$B_{13} = B_3 + B_{11}$	$y_4 = B_2 + B_{10}$
$B_{14} = N_6 + B_7$	$B_{15} = N_{14} + B_9$	$B_{16} = B_{12} + B_{13}$
$y_0 = (N_{12} + B_{16})'$	$B_{17} = N_{15} + B_{14}$	$B_{18} = B_1 + N_{11}$
$y_7 = B_2 + B_{14}$	$y_1 = (B_9 + B_{16})'$	$y_3 = B_{13} + y_4$
$y_6 = (y_4 + B_{14})'$	$B_{19} = B_{15} + B_{17}$	$y_5 = (B_{19} + N_{17})'$
$y_2 = B_{18} + B_{15}$		

Fig. 5. Bottom linear component of the optimized AES S-Box. 18-bit inputs are $N_0, N_1, ..., N_{17}$. 8-bit outputs are $y_0, y_1, ..., y_7$.