Jurnal
Teknologi

Full Paper

NEW ARCHITECTURE OF LOW AREA AES S-BOX/ INV S-BOX USING VLSI IMPLEMENTATION

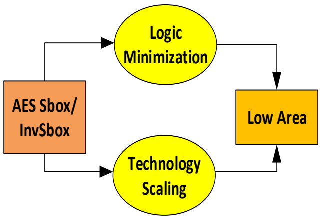
Nabihah Ahmad*

Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia

Article history
Received
11 June 2015
Received in revised form
13 November 2015
Accepted
3 January 2016

*Corresponding author
nabihah@uthm.edu.my

Graphical abstract



Abstract

The Substitution box (S-box) is one of the core of Advanced Encryption System (AES) implementation and the only non-linear transformation. It is consumed most of the power in AES hardware. This paper present a low-complexity design methodology for the S-box/ InvS-box which includes minimising the comprehensive circuit size and critical path delay, scaling down the transistor size, along with selecting an advanced technology for an optimised CMOS full custom design. The area of the circuit is about 39.44 μm^2, while the hardware cost of the S-box/InvS-box is about 147 logic gates, with a critical path propagation delay of 3.235ns.

Keywords: AES; S-box/ InvS-box; VLSI; low area

1.0 INTRODUCTION

Hardware implementation of the AES consist of the S-box which occupied the most expansive building block and the multiplicative inversion is the most complicated steps of the S-box transformation. Implementation of an S-box in a SubBytes transformation operates on individual bytes using a substitution table (S-box) containing a permutation of all 256 possible 8-bit values. It have two transformations, firstly byte replacement, with its multiplicative inverse in the finite field GF(2^8), using the irreducible polynomial p(x) =$x^8+x^4+x^3+x+1$ and secondly, an affine transformation over GF(2^8). For the decryption operation, an inverse S-box is obtained by applying an inverse affine transformation, followed by a multiplicative inversion in GF(2^8).

This proposed design focuses on area-efficient full-custom CMOS implementation using the composite field technique for the AES S-box. Several existing different construction schemes using composite field are found in [1], [2] and [3]. Daemen and Rijmen [4] propose the first efficient hardware implementation of the multiplication inversion in GF(2^8), by decomposing the finite field GF(2^8) to its sub-field GF(2^4), which leads to low complexity in hardware. Furthermore, Rudra et al. [3] and Wolkerstofer, Oswald and Lamberger [5] decompose the elements of GF (2^8) into GF(2^4)^2 to implement the multiplicative inverse in SubBytes and [5] implement it in ASIC implementation. The transformation matrix from GF(2^8) to GF((2^2)^2)^2 is proposed by Satoh et al. [1] and is claimed to be the most minimised hardware implementation to date, with a gate complexity of 5400 gates. Mentens et al. [6] also use the same approach as Satoh et al., [1] but with different polynomial coefficients, and achieve slightly better optimised hardware than that by Marioko and Satoh [7]. Other efforts towards the efficient implementation of the S-box include those by Canright [8], Burns et al. [9] and Liu and Parhi [10], which further improve the performance in area, power and delay.

This paper also take the advantage of a full custom design using state of the art CMOS processes make it possible to scale all the transistors down with process scaling, without deteriorating the overall performance, while increasing speed in most cases. This leads to a smaller chip area. Another design method is through using advanced process technology that leads to a reduction in the die area.

2.0 NEW AES S-BOX/ INVS-BOX ARCHITECTURE

New S-box/ InvS-box architecture employs combinational logic using composite field arithmetic based on Satoh et al.'s work [2] and polynomial coefficients and the implementation of constant multiplication with λ optimised by Mentens et al. [6]. This architecture is implemented using XOR circuits, multiplexers, and basic logic gates.

The optimization techniques for the low-voltage and low-area composite field S-box implementation has been further enhanced by using six-transistor XOR gate in [12]. In comparison to [2] and [6], the inverse S-box for decryption is also implemented by the same chip. The composite field inversion by extending GF(2^8) over GF(($(2^2)^2)^2$) has been used to create compact AES implementations [2] and [3]. This approach was chosen to achieve small area design.

First, the column vectors of the *State* matrix go into the isomorphic transformation from GF(2^8) into the composite field GF(($(2^2)^2)^2$), followed by inversion in the composite field and inverse isomorphic transformation. Then an affine transformation is carried out to create the cipher data. This can be represented by the elementary transformations

$$b \xrightarrow{ISO} q \xrightarrow{MINV} q' \xrightarrow{INVISO} b' \xrightarrow{AFFINE} b''$$

where, *b* are byte elements from the *State* matrix, *q* are byte elements of the isomorphic mapping transformation, *q*' are multiplicative inverse elements of the isomorphic state, *b*' are the elements after inverse isomorphic mapping, and finally *b*" are elements after affine transformation. The InvSubBytes involves an isomorphic transformation followed by an inverse affine transformation.

The inversion in composite field is carried out followed by inverse isomorphic mapping. This can be represented by the elementary transformations

$$b'' \xrightarrow{ISO} q'' \xrightarrow{INVAFFINE} q' \xrightarrow{MINV} q \xrightarrow{INVISO} b$$

The affine transformation, AT operates on the GF(2^8) multiplicative inverse of bytes, b of the state matrix, while the inverse affine transformation AT^{-1} operates on the isomorphic affine transformed GF (2^8) multiplicative inverse of the same bytes, b represented by q". By performing the inversion operation in a composite Galois field of GF($(2^4)^2$) or GF(($(2^2)^2)^2$) the S-box is optimised obtained from GF(2^8) via isomorphic mapping. Figure 1 illustrated the multiplicative inverse in GF(2^8) as extension of degree 2 over GF($(2^2)^2$)

The constants are chosen based on Mentens et al. [6] for the optimal hardware solution, so that $\varphi = \{10\}_2$ and $\lambda = \{1000\}_2$, which leads to a total number of '*1*' entries of the transformation matrices equal to 54, reduced by five from Satoh's [2] constant multiplication, but this requires one extra XOR gate. It also has the lowest gate count and the shortest critical path.

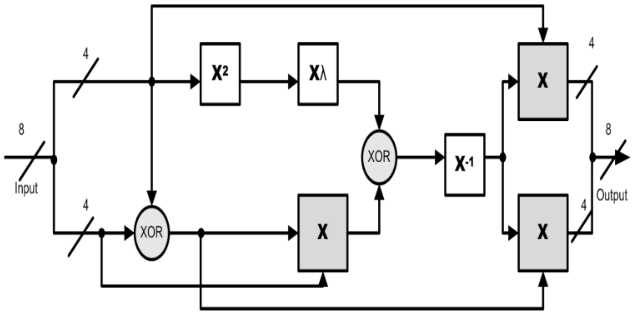


Figure 1 Multiplicative Inverse in GF(2^8) as extension of degree 2 over GF($(2^2)^2$)

A new proposed SubByte and InvSubByte merged the sub-component of the typical multiplicative inverse by reducing the hardware complexity of the circuit using a circuit optimization and minimisation technique consists of Stage 1, the inversion and the combination of multiplication in GF(2^4).

Stage 1 block includes a logic optimisation of multiplication in GF(2^4), multiplication with constant, squaring in GF(2^4), and addition included in one circuit. CombineXAXB block is minimized for multiplication in GF(2^4) after multiplicative inversion in GF(2^4).

2.1 Stage 1

This architecture was developed by merging the transformation of multiplication in GF(2^4), multiplication with lambda, squaring in the GF(2^4) and modulo-2 addition process based on composite field arithmetic. The logic expression is present in the low-complexity formulation, and consists of 24 XOR gates, and 16 AND gates, with critical path delay of four gates. Table 1 shows a gate count comparison between typical composite field architecture and new proposed Stage 1. The input of Stage 1 be

$w = \{w_3\ w_2\ w_1\ w_0\}_2$ and $q = \{q_3\ q_2\ q_1\ q_0\}_2$. The output is $\gamma = \{\gamma_3\ \gamma_2\ \gamma_1\ \gamma_0\}_2$.

$$\gamma = \{\,squarer + multiplication\ with\ lambda + multiplication\ in\ \mathrm{GF}(2^4) + \mathrm{addition}\} \quad (1)$$

The formulation equation for output, γ of Stage 1 is as follows:

$$\left.\begin{aligned}
\gamma_3 &= q_3 + q_0 + m_0 w_3 + m_1 \partial + m_2 \chi + m_3 \varrho \\
\gamma_2 &= \kappa + q_1 + m_0 w_2 + m_1 w_3 + m_2 \upsilon + m_3 \chi \\
\gamma_1 &= \kappa + m_0 w_1 + m_1 x + m_2 \partial + m_3 w_2 \\
\gamma_0 &= q_2 + m_0 w_0 + m_1 w_1 + m_2 w_3 + m_3 \partial \\
&\text{where } \chi = w_3 + w_1,\ \upsilon = w_2 + w_0,\ \partial = w_3 + w_2, \\
&x = w_0 + w_1,\ \varrho = \chi + \upsilon \text{ and } \kappa = q_3 + q_2
\end{aligned}\right\} \quad (2)$$

Table 1 Gate count comparison between typical composite field architecture and proposed Stage 1

Architecture	Total num. of XOR gate	Total num. of AND gate	Critical path delay
multiplication in *GF(2⁴)* multiplication with lambda squaring in *GF(2⁴)* modulo-2 addition [2, 7, 11]	37	9	6
Stage 1	24	14	4

2.2 Simplification of multiplicative inverse of nibble in GF(2⁴)

Let the input of the inversion in GF(2⁴) be $\gamma = \{\gamma_3\ \gamma_2\ \gamma_1\ \gamma_0\}_2$. The output is $\theta = \{\theta_3\ \theta_2\ \theta_1\ \theta_0\}_2$. The formulation result obtained using polynomial basis with φ = {10}₂ is as follows:

$$\left.\begin{aligned}
\theta_3 &= \gamma_3\overline{\gamma_0} + \gamma_2(\overline{\gamma_3\gamma_1})\\
\theta_2 &= \gamma_3(\gamma_0 \cup \gamma_2) + \gamma_2\overline{\gamma_1}\\
\theta_1 &= \gamma_1\overline{\gamma_3\gamma_2} + \gamma_3\overline{\gamma_1\gamma_0} + \gamma_2\overline{\gamma_0}\\
\theta_0 &= (\gamma_0 + \gamma_1)\overline{\gamma_3\gamma_2} + \gamma_2(\gamma_0 \cup \overline{\gamma_1})
\end{aligned}\right\} \quad (3)$$

where ∪ and + are OR gate and XOR gate implementation, respectively. The gate count comparison between the new proposed inversion in GF(2⁴) and typical inversion in GF(2⁴) composite field architecture are shown in Table 2.

Table 2 Gate count comparison between typical inversion in GF(2⁴) composite field architecture and proposed inversion in GF(2⁴)

Inversion in *GF(2⁴)* Architecture	Total num. of XOR gate	Total num. of AND gate	Total num. of NAND gate	Total num. of OR gate	Critical path delay
[2, 7, 11]	14	8	-	-	5
Proposed	6	9	3	1	3

2.3 CombineXAXB

The architecture of CombineXAXB represents the merging of two multiplications in *GF (2⁴)*, after the multiplicative inverse of nibbles in GF(2⁴), using Boolean simplification to achieve a low gate count for this architecture.

With the output is $\Lambda = \{\Lambda_7\ \Lambda_6\ \Lambda_5\ \Lambda_4\ \Lambda_3\ \Lambda_2\ \Lambda_1\ \Lambda_0\}_2$ and the input of $\theta = \{\theta_3\ \theta_2\ \theta_1\ \theta_0\}_2$, $m = \{m_3\ m_2\ m_1\ m_0\}_2$ and $q = \{q_3\ q_2\ q_1\ q_0\}_2$, the equations of the result are following:

$$\left.\begin{aligned}
\Lambda_7 &= q_0\theta_3 + q_1\varepsilon + q_2\alpha + q_3\zeta\\
\Lambda_6 &= q_0\theta_2 + q_1\theta_3 + q_2\beta + q_3\alpha\\
\Lambda_5 &= q_0\theta_1 + q_1\eta + q_2\varepsilon + q_3\ \theta_2\\
\Lambda_4 &= q_0\theta_0 + q_1\theta_1 + q_2\theta_3 + q_3\varepsilon\\
\Lambda_3 &= m_0\theta_3 + m_1\varepsilon + m_2\alpha + m_3\zeta\\
\\
\Lambda_2 &= m_0\theta_2 + m_1\theta_3 + m_2\beta + m_3\alpha\\
\Lambda_1 &= m_0\theta_1 + m_1\eta + m_2\varepsilon + m_3\ \theta_2\\
\Lambda_0 &= m_0\theta_0 + m_1\theta_1 + m_2\theta_3 + m_3\varepsilon
\end{aligned}\right\} \quad (4)$$

Based on Table 3, it is show that the logic expression present are consists of 28 XOR gates, and 15 AND gates, with critical path delay of four gates.

Table 3 Gate count comparison between typical multiplication in GF(2⁴) and proposed CombineXAXB

Architecture	Total num. of XOR gate	Total num. of AND gate	Critical path delay
two multiplication in *GF(2⁴)* [2, 7, 11]	42	18	5
CombineXAXB	28	15	4

3.0 RESULTS AND DISCUSSION

The new proposed S-box architecture illustrated in Figure 2 is implemented to perform both encryption and decryption, with the S-box and InvS-box sharing the same hardware simply by switching combinatorial logic blocks using multiplexers. It is an improved modification of the architecture using a composite field based on the polynomial basis. This modification enables the implementation of inverse SubBytes for decryption by reusing the same S-box resources.

In order to reduce the area complexity, an appropriate logic gate style should be used. Logic style can affect the size of a transistor, wiring load and power dissipation, especially in full-custom implementation. Full custom-design implementation offers an alternative to using a differential logic style. Rather than using the standard cell, the transistor gate can be scaled down without deteriorating performance. As most functionalities of the S-box are based on combinational logic dominated by an XOR gate, it is wise to use good performance and a low XOR gate count. Hence, by using the low-area XOR gate in [12], a circuit level optimised S-box/ InvS-box hardware, in terms of silicon area has been achieved.

The complete circuit simulation, optimisation, layout and parasitic extraction were carried out using Mentor Graphics tools. The mask layout of the S-box/ InvS-box illustrated in Figure 3 was customised (with manual *placement* and *routing*) in 130nm IBM CMOS, with copious instances of the XOR gate for the CMOS Galois field/ composite field arithmetic.

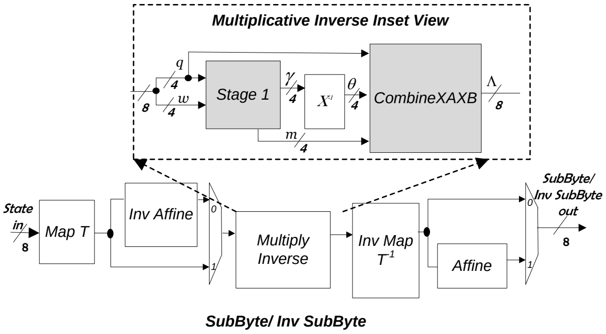


Figure 2 Proposed Multiplicative Inverse in GF(2^8) architecture

Minimum channel length was used for all devices, and optimum channel width was carefully chosen for each device to achieve verified functionality with the smallest possible propagation delay. Figure 4 shows the complete S-box/ InvS-box chip with the bonding I/O pads.

The design simulations consisted of functional verification, power and timing analysis using the circuit simulator Eldo platform provided by Mentor Graphics, as well as design rule checking (DRC), and layout vs. schematic (LVS). Sets of NIST test vectors were used to verify the functionality. Simulation results verified the right functionality for every input SubByte combination with a supply voltage of only 0.8V. The worst case S-box input-to-output delay was around 3.235ns, thus allowing a throughput of around 309 Mega-SubBytes per second. The silicon-area of the S-box/InvS-box is only 39.44µm^{2,} using the 130nm CMOS process, and offers to-date, the smallest reported silicon-area of any implementation with shared S-box and inverse S-box.

This architecture achieves a small area using only 147 gates (105 XOR gates, 38 AND gates, 3 NAND gates and 1 OR gate, which is about 17% smaller than the typical composite field SubByte architecture recorded by Satoh et al. [2] with 172 gates (137 XOR gates and 35 AND gates). For multiplicative inverse, the proposed architecture has a critical path of 11 gate delay compared to a 17 gate delay for a typical composite field design which is cut off by more than 5%.

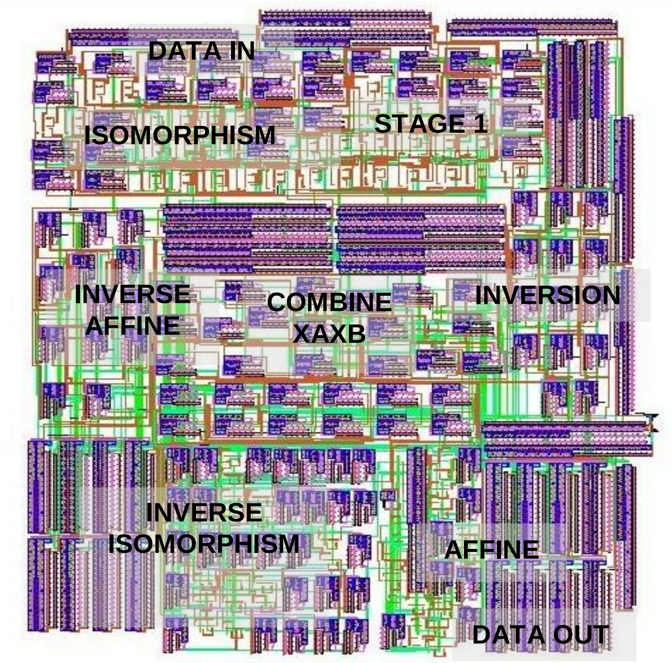


Figure 3 Complete layout of the S-box/ InvS-box.

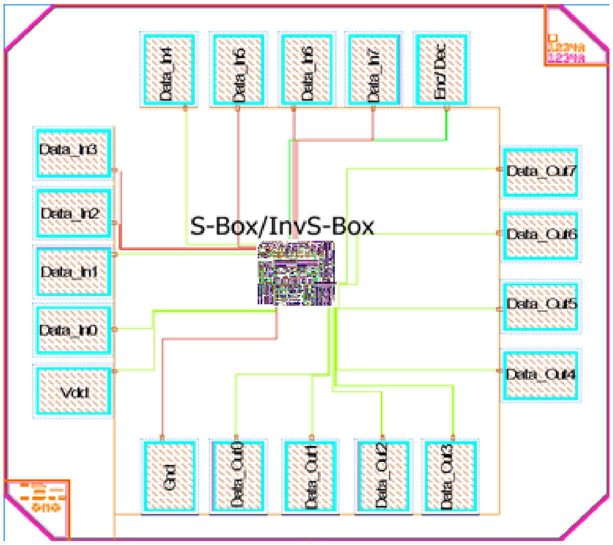


Figure 4 Complete chip die of the S-box/ InvS-box with bonding pads.

4.0 CONCLUSION

In conclusion, this paper presents a new full custom hardware implementation of a low-area AES S-box/ InvS-box GF(2^8) Galois field inversion based on the polynomial basis, using composite field arithmetic architecture in the 130nm CMOS process, employing circuit level optimisation. Efficient design of S-box/ InvS-box was achieved by minimising the combinational logic and merging the sub-blocks of multiplicative inverse. The design demonstrated a new approach to minimise the silicon-area of an S-box by using a 2- input XOR gate in [12] for low-area composite field arithmetic. Based upon the performance of the implementation, the results indicate that this design is suitable for applications that require small-area and low-power consumption, such as RFID tags.

Acknowledgement

This research was conducted with financial support by Short Term Grant (STG) Vot Number 1328.

References

[1] Daemen J. and Rijmen V. 2002. The Design of Rijndael: AES - The Advanced Encryption Standard. *Springer-Verlag*.

[2] Satoh A., Morioka S., Takano K. and Munetoh, S. 2001. A Compact Rijndael Hardware Architecture with S-Box Optimization. Advances in Cryptology — *ASIACRYPT 2001*. 2248: 239-254.

[3] Liu Z., Zeng Y., Zou X., Han, Y. and Chen, Y. 2007. A High-Security and Low-Power AES S-Box Full-Custom Design for Wireless Sensor Network. *International Conference on Wireless Communications, Networking and Mobile Computing*. 2499-2502.

[4] Rudra A., Dubey P. K., Jutla C. S., Kumar, V., Rao, J. R. and Rohatgi, P. 2001. Efficient Rijndael Encryption Implementation with Composite Field Arithmetic. *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded System*. 2162: 171-184.

[5] Wolkerstorfer J., Oswald E. and Lamberger M. 2002. An ASIC Implementation of the AES SBoxes. *Proceedings of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology*. 67-78.

[6] Mentens N., Batina L., Preneel B. and Verbauwhede, I. 2005. A Systematic Evaluation Of Compact Hardware Implementations For The Rijndael S-Box. *Proceedings Of The 2005 International Conference on Topics in Cryptology*. 323-333.

[7] Morioka S. and Satoh, A. 2003. An Optimized S-Box Circuit Architecture for Low Power AES Design. *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*. 2523: 172-186.

[8] Canright D. 2005. A Very Compact S-Box For AES. *Proceedings Of The 7th International Conference On Cryptographic Hardware And Embedded Systems*. 441-455.

[9] Burns F., Murphy J., Koelmans, A. and Yakovlev, A. 2009. Efficient Advanced Encryption Standard Implementation Using Lookup And Normal Basis. *Computers & Digital Techniques, IET*. 3: 270-280.

[10] Hess N. J. E., Meyer B. and Schütze, T. 2000. Information Leakage Attacks Against Smart Card Implementations Of Cryptographic Algorithms And Countermeasures. *Proceedings of EUROSMART-Security-Conference 2000*. 53-64.

[11] Ahmad N. and Hasan, R. 2012. Low-power compact composite field AES S-Box/Inv S-Box design in 65nm CMOS using Novel XOR Gate. Integration, the VLSI Journal, Available online ISSN 0167-9260, http://dx.doi.org/10.1016/j.vlsi.2012.06.002.

[12] Ahmad N. and Hasan R. 2013. A 0.8 V 0.23 nW 1.5 ns Full Swing Pass-Transistor XOR gate in 130 nm CMOS. Active And Passive Electronic Components. 2013. *Hindawi Publishing Corporation*.