Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 696717, 5 pages
http://dx.doi.org/10.1155/2013/696717

Research Article

An 8-Bit ROM-Free AES Design for Low-Cost Applications

Ming-Chih Chen and Wei-Ting Li

Department of Electronic Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung City 811, Taiwan

Correspondence should be addressed to Ming-Chih Chen; mjchen@nkfust.edu.tw

Received 13 September 2013; Accepted 14 October 2013

Academic Editor: Teen-Hang Meen

We have presented a memory-less design of the advanced encryption standard (AES) with 8-bit data path for applications of wireless communications. The design uses the minimal 160 clock cycles to process a 128-bit data block. For achieving the requirements of low area cost and high performance, new design methods are used to optimize the MixColumns (MC) and Inverse MixColumns (IMC) and ShiftRows (SR) and Inverse ShiftRows (ISR) transformations. Our methods can efficiently reduce the required clock cycles, critical path delays, and area costs of these transformations compared with previous designs. In chip realization, our design with both encryption and decryption abilities has a 29% area increase but achieves 4.85 times improvement in throughput/area compared with the best 8-bit AES design reported before. For encryption only, our AES occupies 3.5 k gates with the critical delay of 12.5 ns and achieves a throughput of 64 Mbps which is the best design compared with previous encryption-only designs.

1. Introduction

The AES algorithm has been widely used in data transmission in wireless communications [1–3] and RFID applications [4, 5]. The AES design with ASIC chip(s) can achieve the requirements of low cost and high performance. The design with low area cost usually also results in low power consumption. The area reduction of designing the AES can be achieved by optimizing the architectures of its subfunctions [4, 6–11], sharing the same operations of subfunctions [6, 9, 10, 12], and reducing the data path of overall architectures [1, 2, 4–7, 10, 12–15]. The feature of inherently iterative AES algorithm can be exploited to reduce the data path of overall architecture. The data path design of AES can be shrunk to 8-bit versions [1, 2, 4–7, 10, 12, 13] for reducing the area cost. The ASIC design of 8-bit AES reported in [5] has the smallest area cost compared with other versions but also leads to the lowest performance since more clock cycles are needed in encryption and decryption. For the objective of reducing the area cost but still keeping the acceptable performance, the proposed AES uses 8-bit data path and minimum clock cycles to perform the encryption/decryption processes.

For the portability of AES in different platforms and CMOS technologies, our AES uses pure combination logic to design the overall circuit without any memory blocks. The new proposed design methods in major transformations led to the reduction of area cost in AES but still keep the high throughput that meets the requirements of wireless communications. The experiment results show that our AES design has better performance/area ratio compared with previous designs. The remainder of this paper is organized as follows. Section 2 briefly describes the AES algorithm and its transformations. The new designs of transformations and overall AES architecture are proposed in Section 3. Section 4 describes experimental results and comparisons with other previous designs. Finally, conclusions are given in Section 5.

2. AES Algorithm

2.1. AES Algorithm. The AES algorithm for 8-bit data path that processes a 128-bit data block will take at least 160 rounds. The encryption processes perform ShiftRows (SR), SubBytes (SB), MixColumns (MC), and AddRoundKey (ARK) transformations. A separate KeyExpansion (KE) unit is required to generate the Kth round key for each ARK. The decryption process has three reversed transformations, InvShiftRows (ISR), InvSubBytes (ISB), and InvMixColumns (IMC), and one ARK. The normal rounds perform the four inversed transformations. The round keys operated in the decipher process are the reverse of the round keys generated in each round in the cipher process.

2.2. AES Transformations. Four kinds of transformations and one key generation unit in the AES algorithm are described as follows.

(a) SB/ISB Transformations. The transformations are nonlinear substitution operations where each byte of the input state is computed with multiplicative inverse (MI) in GF(2^8) and followed by an affine transformation (AF) over the same field. Similarly, the ISB transformation performs the inverse affine transformation (IAF) followed by the operation of MI in GF(2^8).

(b) MC/IMC Transformations. The transformations operate column-by-column on the 4×4 byte array and treat each column as four-term polynomial with coefficients over GF(2^8). The MC transforms each column to a new one by multiplying it with a constant polynomial $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$ modulo $x^4 + 1$. The IMC operation is a multiplication of each column with $b(x) = a^{-1}(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}$ modulo $x^4 + 1$.

(c) SR/ISR Transformations. The SR transformation rotates the last three rows of the state to the left by one, two, or three bytes depending on the row numbers. The ISR rotates them in the inverse direction of the SR.

(d) ARK Transformation. In each round, the ARK transformation performs an addition of the state with the round key using a bitwise XOR operation.

(e) KE Unit. In each round, the KE unit generates a new 128-bit round key for the XOR operation with the state in the ARK transformation.

3. Design of Our AES Architecture

3.1. Designs of Major Transformations. The optimization of separate transformations focuses on two major transformations, SR and MC, and their inverses, ISR and IMC. The designs of these transformations are described as follows.

(a) The Design of SR/ISR Unit. In this paper, we propose a combined SR/ISR design as shown in Figure 1. It uses twelve 8-bit registers for receiving and storing data from MC or ARK units. The output sequences are generated after performing the SR rotations. Equation (1) shows the original 4 by 4 state matrix and the output state matrix after the SR rotations. The original states in the first row after performing the SR are unchanged. The states in the second, third, and forth rows are rotated by right shifting one, two, and three positions, respectively,

$$\underbrace{\begin{pmatrix} S_{12} & S_8 & S_4 & S_0 \\ S_{13} & S_9 & S_5 & S_1 \\ S_{14} & S_{10} & S_6 & S_2 \\ S_{15} & S_{11} & S_7 & S_3 \end{pmatrix}}_{\text{Original Input States}} \Longrightarrow \underbrace{\begin{pmatrix} S_{12} & S_8 & S_4 & S_0 \\ S_1 & S_{13} & S_9 & S_5 \\ S_6 & S_2 & S_{14} & S_{10} \\ S_{11} & S_7 & S_3 & S_{15} \end{pmatrix}}_{\text{Output States After SR}}. \quad (1)$$

For completing the rotation sequences, several multiplexers are added in Figure 1. The states are inputted to the SR unit by the sequences of their state number. Therefore, the state S_0 is the first one that is inputted to the SR in the first clock cycle, and the state S_{15} is the last input to the SR in the sixteenth clock cycle. The output sequences of the states in the first row are unchanged after performing the SR rotations. The input state S_0, S_4, and S_8 are stored in register $R8$, $R4$, and $R0$, respectively, after several clock cycles. The state S_{12} is stored in register $R0$ after outputting the state S_0 from register $R8$. In the second row, the first state S_1 is delayed to the last one and other states S_5, S_9, and S_{13} bypass the state S_1 using the multiplexer. These three states are outputted before the state S_1.

The states S_2 and S_6 in the third row are delayed behind states S_{10} and S_{14} after the rotation of the third row. In the fourth row of original state matrix, the state S_{15} is the last one but becomes the first output state of that row after performing the SR. Similarly, the ISR performs the rotations in the inverse direction of the SR by using the multiplexers to bypass some states for outputting the correct sequences. The design in [13] is the best method to solve the SR and ISR rotations reported so far, but our design can reduce four 8-bit registers and shorten the critical path delay of the unit.

(b) The Design of MC/IMC Unit. In our AES design, the MC, and IMC units are separated to reduce the complexity of the data paths. Equation (2) shows four input states S_0, S_1, S_2, and S_3 that multiply constant values {03}, {02}, {01}, and {01}, respectively, in Galois Field GF(2^8) for generating output states S'_0 to S'_3. The equation also shows that the constant values in the second, third, and fourth rows are rotated to left by one, two, and three positions corresponding to the first row

$$\underbrace{\begin{bmatrix} S'_0 \\ S'_1 \\ S'_2 \\ S'_3 \end{bmatrix}}_{\substack{\text{Output States} \\ \text{After MC}}} = \underbrace{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}}_{\text{Constant Matrix}} \times \underbrace{\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix}}_{\text{Original States}}. \quad (2)$$

As shown in Figure 2, our MC design uses eight 8-bit registers to store the states and uses two multiplication units ($\{02\} \times X$) and ($\{03\} \times X$) for performing the multiplication in (2). These two multiplication units are realized by simple bit-level XOR operations. The MC design uses two levels of registers. The four registers in the first level receive data from the MC or the ARK units. The second-level registers prepare calculation operations for the outputs. For example, the result of output state S'_0 is calculated as ($\{02\}\times S_0 + \{03\}\times S_1 + \{01\}\times S_2 + \{01\}\times S_3$). The states S_1, S_2, S_3, and S_4 are inputted to registers $R4$, $R5$, $R6$, and $R7$, respectively, after four clock cycles. In the next cycle, the four states are stored in registers $R0$–$R3$, respectively, and perform the multiplications ($\{02\} \times R0$) and ($\{03\} \times R1$). The MC unit outputs the states S'_0–S'_3 in the subsequent four clock cycles. At the same time, the next four states are inputted to registers $R4$–$R7$ and wait for performing the multiplication with the constant matrix. The MC unit needs sixteen clock cycles to complete the calculation of a 128-bit state. The design in [1, 4–7, 14, 15] is the best method to solve the MC and IMC operations reported before, but our design can further reduce twenty-four 8-bit registers and shorten the critical path delay of the MC unit. The similar

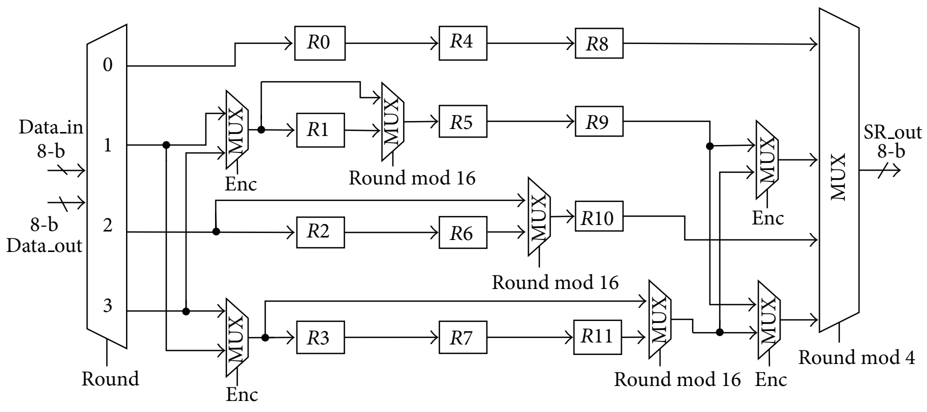

Figure 1: Proposed combined SR/ISR architecture.

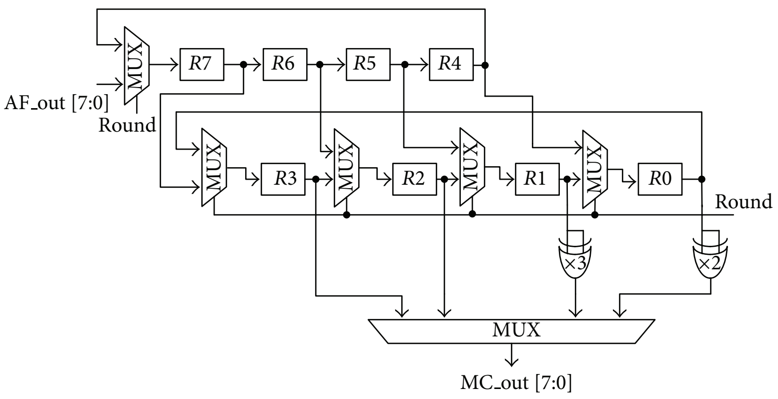

Figure 2: Proposed MC architecture.

optimization results are also obtained in the design of IMC unit.

3.2. The Design of Overall Architecture. We realized iterative AES architecture designs using TSMC 0.18 μm cell library. Figure 3 shows the 8-bit AES processor architecture. A plaintext block and the encryption key are loaded to the AES through the 8-bit input ports *data_in* and *key_in*. The *enc* signal is used to select the encryption or decryption processes. The SB can be realized by the calculation of Multiplicative Inverse (MI) in GF$((2^4)^2)$ and Affine Transformation (AF) units. The ISB can be realized by the same MI calculation with SB and inversed affine transformation (IAF) units. For reducing the area cost of the combined implementations of SB/ISB units, the MI logic is usually shared. The key expansion unit is used to generate and output the required 8-bit round key to the ARK. Since the round keys are in reverse order in decryption, the inverse cipher process can start only after generating the last round key. Afterward, the key expansion with the same round keys can be executed concurrently with the decryption process.

4. Experimental Results

In Table 1, various 8-bit AES designs in different technologies are listed for comparison. The designs in [1, 4, 7, 14] only have the encryption ability. The design in [4] is the encryption-only version of the previous design [5], for application in radio frequency identification (RFID). The design in [5] adopts the clock gating method to reduce the power consumption. One pipeline stage is used to reduce the critical path delay in the SB/ISB design. The SR/ISR units are implemented by random access memory (RAM). In [7], the encryption-only design merges the ARK and SR operations by using four pipeline stages to generate the correct output order of the computed state. The design in [1] provides a low power AES design for the RFID application. It uses gated clock design to reduce unwanted switching activity, the same approach as in [5]. Good and Benaissa [14] proposed a low power/area AES chip design that provides a series of finite-field doubling, tripling, and XOR operations to perform the MC transformation. It also adopts separate data and key memories, the same approach as in [1], for parallel processing the state and round key.

In Table 1, we provide two kinds of implementation information of our AES design including AES with encryption ability and AES with encryption/decryption abilities in the chip level. The chip design is used to compare with other chip results that have their circuits fabricated.

We observe that most realizations are encryption only due to the fast verification of their designs. Most realizations of 8-bit data path AES require more clock cycles to compute

Table 1: Performance comparison of different 8-bit AES designs.

Design	Tech. (um)	Mode	Max. clock freq. (MHz)	Clock cycles	Area (k-gates)	Max. throughput (Mbps)	Max. throughput/area (Mbps/k-gates)	Power consumption
Feldhofer et al. [4] (Syn.)	0.35	Enc only	0.1	992	3.628	0.013	0.0036	26.9 uW at 100 KHz
Kaps and Sunar [7] (Syn.)	0.13	Enc only	0.5	534	4.07	0.12	0.0295	23.85 uW at 500 KHz
Kim et al. [1] (Syn.)	0.25	Enc only	0.1	870	3.9	0.015	0.0038	4.85 uW at 100 KHz
Feldhofer and Wolkerstorfer [5] (chip)	0.35	Both	80	Enc: 1,032 Dec: 1,165	3.4	9.9	2.91	4.5 uW at 100 KHz 1.5 V
Good and Benaissa [14] (chip)	0.13	Enc only	12	356	5.5	4.31	0.78	99 uW at 12 MHz 0.8 V
Ours (chip)	0.18	Enc only	80	160	3.5	64	18.3	65 uW at 80 MHz 1.8 V
Ours (chip)	0.18	Both	60	160	4.4	48	10.9	93 uW at 60 MHz 1.8 V

*Tech.: technology; Syn.: synthesis; Freq.: frequency.

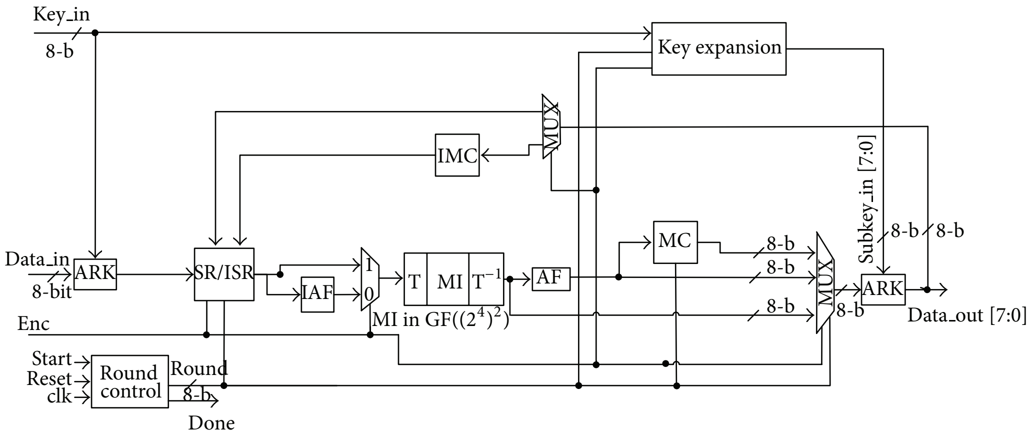

Figure 3: Proposed AES architecture.

a 128-bit data block, resulting in smaller throughput rate. Therefore, most realizations are suitable for those applications with low frequency and throughput rate requirement, such as RFID. On the other hand, our design with higher throughput can be used in applications such as 802.11 series wireless network. Our AES design with only encryption ability occupies 3.5 k gates with the critical delay of 12.5 ns. The major improvement of our AES in this version is to minimize the required number of clock cycles and critical path delay for processing MC and SR operations by our architecture designs.

The area cost and critical path delay of our AES are similar with the best design in [5]. But our design can achieve a throughput of 64 Mbps which is the best design compared with previous encryption-only designs. The area cost of our AES design with both encryption and decryption abilities increases about 29%, but the throughput improves 4.85 times compared with the best design in [5]. From the experimental results, we also observe that our AES design has the best normalized performance of throughput per gate compared with other previous designs.

5. Conclusions

In this paper, we have presented new design methods of AES transformations and their architecture. The major transformations, SR/ISR and MC/IMC, dominate the required clock cycles and path delays for processing the data encryption and decryption. We presented two design methods that can efficiently optimize these transformations, and the proposed architecture design can improve the throughput but keep low area cost compared with other previous designs. The design is suitable for area-limited applications that require high throughput, such as wireless communications. The implementation results demonstrate that the proposed design has the highest throughput with low area cost.

Acknowledgments

This work is supported in part by Taiwan's Ministry of Education under Project no. 101B-09-027. One of the authors would like to thank Professor Shen-Fu Hsiao for providing the comments.

References

[1] M. Kim, J. Ryou, Y. Choi, and S. Jun, "Low power AES hardware architecture for radio frequency identification," in *Advances in Information and Computer Security*, vol. 4266 of *Lecture Notes in Computer Science*, pp. 353–363, Springer, Berlin, Germany, 2006.

[2] O. Song and J. Kim, "An efficient design of security accelerator for IEEE 802.15.4 wireless senor networks," in *Proceedings of the 7th IEEE Consumer Communications and Networking Conference*, pp. 1–5, January 2010.

[3] Y. T. Tsou, C. S. Lu, and S. Y. Kuo, "MoteSec-aware: a practical secure mechanism for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2817–2829, 2013.

[4] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Cryptographic Hardware and Embedded Systems—CHES 2004*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 357–370, Springer, Berlin, Germany, 2004.

[5] M. Feldhofer and J. Wolkerstorfer, "Strong crypto for RFID tags—a comparison of low-power hardware implementations," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1839–1842, May 2007.

[6] T. Good and M. Benaissa, "AES on FPGA from the fastest to the smallest," in *Cryptographic Hardware and Embedded Systems—CHES 2005*, vol. 3659 of *Lecture Notes in Computer Science*, pp. 427–440, Springer, Berlin, Germany, 2005.

[7] J. P. Kaps and B. Sunar, "Energy comparison of AES and SHA-1 for ubiquitous computing," in *Emerging Directions in Embedded and Ubiquitous Computing*, vol. 4097 of *Lecture Notes in Computer Science*, pp. 372–381, Springer, Berlin, Germany, 2006.

[8] W.-T. Huang, C. H. Chang, C. W. Chiou, and S.-Y. Tan, "Non-XOR approach for low-cost bit-parallel polynomial basis multiplier over GF(2^m)," *IET Information Security*, vol. 5, no. 3, pp. 152–162, 2011.

[9] M. M. Wong, M. L. D. Wong, A. K. Nandi, and I. Hijazin, "Composite field GF($((2^2)^2)^2$) Advanced Encryption Standard (AES) S-box with algebraic normal form representation in the subfield inversion," *IET Circuits, Devices and Systems*, vol. 5, no. 6, pp. 471–476, 2011.

[10] T. A. Pham, M. S. Hasan, and H. Yu, "Area and power optimisation for AES encryption module implementation on FPGA," in *Proceedings of the 18th International Conference on Automation and Computing*, pp. 1–6, Septemper 2012.

[11] 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 Systems*, vol. 20, no. 6, pp. 1151–1155, 2012.

[12] J. Chu and M. Benaissa, "Low area memory-free FPGA implementation of the AES algorithm," in *Proceedings of the 22nd International Conference on Field Programmable Logic and Applications*, pp. 623–626, August 2012.

[13] H. Li and J. Li, "A new compact architecture for AES with optimized ShiftRows operation," in *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 1851–1854, May 2007.

[14] T. Good and M. Benaissa, "692-nW advanced encryption standard (AES) on a 0.13-μm CMOS," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 12, pp. 1753–1757, 2010.

[15] K. Thongkhome, C. Thanavijitpun, and S. Choomchuay, "A FPGA design of AES core architecture for portable hard disk," in *Proceedings of the 8th International Joint Conference on Computer Science and Software Engineering*, pp. 223–228, May 2011.