S-Box Absorption Design for Key-specific AES circuits

Shunsuke Matsuoka, Naoki Fujieda, and Shuichi Ichikawa
Department of Electrical and Electronic Information Engineering,
Toyohashi University of Technology
E-mail: {matsuoka,fujieda}@ee.tut.ac.jp, ichikawa@ieee.org

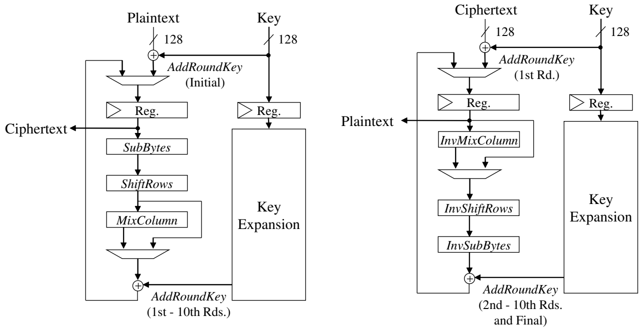
***Abstract*—A key-specific AES (Advanced Encryption Standard) has been studied to reduce the amount of hardware and the power consumption, where the key expansion logic and the *AddRoundKey* function are replaced with ROMs. This paper presents a new design named S-Box absorption, which integrates another function *SubBytes* into the ROMs for *AddRoundKey*. According to our evaluation results on Xilinx Virtex-5 FPGA technology, the proposed encryption circuit achieved 59% reduction of the logic scale with 27% reduction of the power consumption over the existing key-specific AES.**

I. Introduction

In 2000, Rijndael algorithm was selected as the Advanced Encryption Standard (AES) by the NIST (National Institute of Standards and Technology) [1]. Since then, various designs of AES hardware have been studied [2] for various different goals: e.g., higher throughput, smaller hardware, and lower power consumption. However, the embedded system has strong constraints in hardware resources and computational power. AES circuits in embedded systems are thus implemented to balance the hardware cost, throughput, and power consumption.

Many preceding studies targeted ASIC (standard cell) technology, and evaluated their circuits by computer simulation [2] [3] [4]. According to their simulation results, the power consumption is mainly determined by the propagation of dynamic hazards. Meanwhile, FPGAs have been widely adopted in recent embedded systems. Since the internal structure differs between ASIC and FPGA in many aspects, the circuits suited for ASIC are not always suited for FPGAs [5]. In the researches targeted for FPGAs, the operational frequency and the amount of hardware were usually evaluated [6] [7], while few studies presented the measurement results of the power consumption for practical platforms.

Generally, the logic circuit can be reduced, if any input of the circuit is given as a constant. It is the same as *partial evaluation* in software [8]. In case of AES circuit, the circuit can be reduced if the key is fixed to a specific value. The derived AES circuit becomes specific to a given key, and thus called as *key-specific* AES circuit. A key-specific AES circuit naturally suits with reconfigurable devices (e.g. FPGA), because the circuit has to be reconfigured to change the key. Recent FPGA devices provide partial reconfiguration, which makes key-specific AES circuit applicable for practical systems. Our previous study [9] presented various designs of key-specific AES encryption circuit, and 41–54% reduction of logic scale with 24–74% increase of throughput. The following paper [10] reported the power consumption of the circuits, while the power reduction was little against the original AES circuit.



(a) encryption (b) decryption

Fig. 1. Loop architecture of the AES cryptographic circuit [11].

This study presents a new key-specific AES encryption circuit, which integrates two functions of the AES algorithm: *AddRoundKey* and *SubBytes*. This new technique is referred to *S-Box absorption* in the following discussion. We implemented our circuit for two FPGA platforms, and evaluated them with various existing AES circuits to show the advantages of S-Box absorption in the key-specific AES circuit.

II. AES Cryptographic Circuit

The AES encryption algorithm repeatedly applies four primitive functions: *SubBytes*, *ShiftRows*, *MixColumn*, and *AddRoundKey*. A set of the four functions is called a round. The key lengths defined in AES standard are 128, 192, and 256 bits. The number of rounds is 10 for a 128-bit key. An initial *AddRoundKey* is performed before the first round, and *MixColumn* is skipped in the last (10th) round. The key expansion logic permutes the initial key and generates the round keys. The *AddRoundKey* function performs bitwise XOR (exclusive or) operation on the round key. The *SubBytes* function is nonlinear transformation that uses byte substitution referred to as S-Box.

The decryption is the application of inverse of the primitive functions in the reverse order, and thus a round for the decryption is composed of *AddRoundKey*, *InvMixColumn*, *InvShiftRows*, and *InvSubBytes*. Note that the inverse of *AddRoundKey* is *AddRoundKey* itself. The round keys are also used in the reverse order: the initial key for encryption is the final key for decryption; the first round key of encryption is used in the tenth round of decryption; and so on.

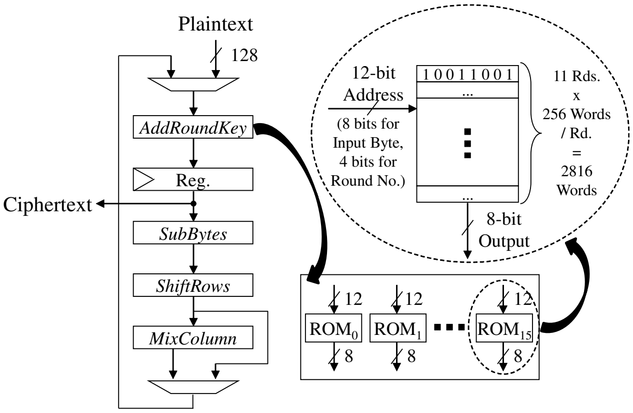


Fig. 2. *XOR_by_ROM* circuit: a key-specific AES circuit that implements XOR operations in *AddRoundKey* with ROMs [9].

Figure 1 illustrates the loop architecture of the AES encryption [11]. The loop architecture has a single set of round components, which is repeatedly applied to the data to generate the ciphertext. Temporal data are stored in the internal register before the next round.

This loop might be unrolled for higher throughput. The pipeline architecture has several sets of round components to handle multiple messages or rounds in parallel. The pipeline architecture is not included in the scope of this paper, because it naturally involves more resources and more power consumption.

The implementations of the loop architecture, which are available from Aoki laboratory [11], are adopted as the basis of evaluation in this study. Four types of S-Box implementation are provided in [11]: *AES_TBL*, *AES_Comp*, *AES_PPRM1*, and *AES_PPRM3*, whose respective S-boxes are based on a look-up table, a composite field inverter [12], a single-stage Positive Polarity Reed Muller (PPRM) logic, and a 3-stage PPRM logic [2].

III. Key-specific AES Cryptographic Circuit

As with *partial evaluation* in software [8], constant input to a logic circuit can reduce its logic scale and power consumption. In key-specific AES circuits where the initial key is fixed to a specific value, the derived round keys are also constant. The key expansion logic is thus reduced to a small ROM that is referred to with the round number. Replacement of the key can be done by modifying the contents of the ROM.

The architecture of the existing *XOR_by_ROM* circuit [9] [10] is shown in Fig. 2. The small ROM for the round keys and XOR operations in *AddRoundKey* are integrated and replaced by larger ROMs for the further reduction of the logic scale and power consumption. The input address and the output data are 12 bit and 8 bit wide, respectively. The higher 4 bits of the address correspond to the round number, and the lower 8 bits correspond to the input byte. The 128-bit XOR operation is divided into 16 ROMs. Since *AddRoundKey* is applied 11 times including the initial one (which corresponds to the round number of zero), the round number ranges from 0 to 10 and the depth of the ROMs is $11 \times 2^8 = 2816$ words, which is suitable size to implement with BRAM elements. The contents of the ROM for the i-th byte are calculated by the following formula:

$$\mathrm{ROM}_i[r * 256 + b] = b \oplus \mathrm{RKey}[r][i],$$

where $\mathrm{RKey}[r][i]$ stands for the i-th byte of the r-th round key.

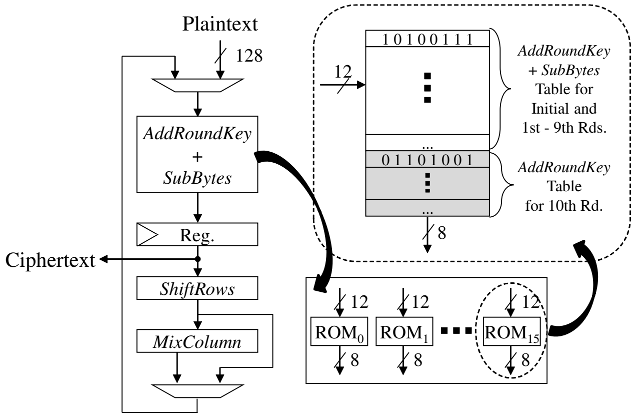


Fig. 3. *XOR&S-Box_by_ROM* circuit where S-Boxes in *SubBytes* are absorbed by ROMs.

IV. S-Box Absorption

This section describes the S-Box absorption, the integration of *AddRoundKey* and *SubBytes*, for key-specific AES cryptographic circuits. It is based on the fact that both the ROMs in *AddRoundKey* and the S-Boxes in *SubBytes* are bytewise operation in the *XOR_by_ROM* circuit. the S-Box absorption merges these two into a single set of ROMs. Since the S-Box is the most complex part of the AES and it is usually considered as the target of the optimization [2] [3] [4] [12], a significant impact on the reduction of logic scale is expected by its removal.

Figure 3 illustrates the architecture of the proposed circuit *XOR&S-Box_by_ROM*, where the S-Box absorption is applied. The *AddRoundKey* and *SubBytes* functions are now considered as a single function. It should be noted that *SubBytes* is absorbed into *AddRoundKey* of the previous round. *SubBytes* of the first round is merged with the initial *AddRoundKey*, and *SubBytes* of the second round is integrated with *AddRoundKey* of the first round. Since the last *AddRoundKey* has no *SubBytes* to be paired, the results of *AddRoundKey* in the last round are separately stored in the ROMs. The contents of the ROM for the encryption are thus calculated as follows:

$$\mathrm{ROM\_E}_i[r * 256 + b] = \begin{cases} \text{S-Box}[b \oplus \mathrm{RKey}[r][i]] & (r \neq 10) \\ b \oplus \mathrm{RKey}[r][i] & (r = 10). \end{cases}$$

In the decryption, *InvSubBytes* is absorbed into *AddRoundKey* of the next round (or the final *AddRoundKey*) and *AddRoundKey* of the first round is not paired with *InvSubBytes*. The contents of the ROM for the decryption are given as follows:

$$\mathrm{ROM\_D}_i[r * 256 + b] = \begin{cases} \text{InvS-Box}[b] \oplus \mathrm{RKey}[10 - r][i]] \\ \qquad\qquad\qquad\qquad (r \neq 0) \\ b \oplus \mathrm{RKey}[10 - r][i] \\ \qquad\qquad\qquad\qquad (r = 0). \end{cases}$$

TABLE I. FPGA IMPLEMENTATION RESULTS OF AES ENCRYPTION CIRCUITS WITH VIRTEX-5 XC5VLX30.

Design	ROM Style	Logic Scale [Slices]	36kb BRAM	Max. Freq. [MHz]	AT Product [Slices/MHz]
AES_TBL [11]	BRAM	522	6	220	2.4
	Dist. RAM	660	2	257	2.6
AES_Comp [11]	-	923	2	134	6.9
AES_PPRM1 [11]	-	924	2	191	4.8
AES_PPRM3 [11]	-	1013	2	133	7.6
XOR_by_ROM [9]	BRAM	432	18	156	2.8
	Dist. RAM	453	2	198	2.3
XOR&S-Box_by_ROM	BRAM	214	18	215	1.0
	Dist. RAM	2656	2	165	16.1

As seen above, the contents of the ROM differ between for the encryption and the decryption. This causes a difference between *XOR_by_ROM* and *XOR&S-Box_by_ROM* designs when both the encryption and the decryption circuits are implemented on a single FPGA. Since a BRAM element has two read/write ports, ROMs can be shared without multiplexers between the encryption and the decryption circuits in *XOR_by_ROM*. This sharing is not applicable to *XOR&S-Box_by_ROM*, which thus requires another set of ROMs. However, when either the encryption or the decryption circuit is needed, the number of ROMs required is the same between the two designs.

V. EVALUATION OF FPGA IMPLEMENTATION

A. Encryption Circuit

In this section, AES encryption circuits, based on the loop architecture [11], are implemented and evaluated with a Xilinx Virtex-5 XC5VLX30 FPGA. Virtex-5 was selected here, because we use SASEBO board [13] with Virtex-5 FPGA to measure the power consumption in the following section.

As the usual AES encryption circuits that are not key-specific, *AES_TBL*, *AES_Comp*, *AES_PPRM1*, and *AES_PPRM3* circuits are examined. The *XOR_by_ROM* and the *XOR&S-Box_by_ROM* circuits are evaluated as key-specific AES circuits. These designs are synthesized and implemented with Xilinx ISE 14.7. The default options are used in both synthesis and implementation. Since ROMs are included in *AES_TBL*, *XOR_by_ROM* and *XOR&S-Box_by_ROM*, BRAM implementation (`-rom_style block`) and distributed RAM implementation (`-rom_style distributed`) are examined for each of them.

In the distributed RAM implementation of the key-specific circuits, the synthesis results vary widely due to the logic optimization, which is affected by the specific key value. Each circuit is thus measured ten times with different keys, and the average value is presented as the result. The evaluation items are logic scale in slices, the number of BRAM used, the estimated maximum frequency, and the area-time product (AT product). AT product is a measure of area/performance, defined by the number of slices divided by the frequency.[1] It should be also noted that the usage of BRAM is not included in AT product.

Table I summarize the evaluation results. Comparing the results among the four normal AES circuits, the *AES_TBL* circuits exhibited the best area-time product with the same number of BRAMs. Although the use of a composite field inverter and the PPRM excelled in ASIC implementation [2] [12], simple *AES_TBL* appears more suited for FPGA implementation.

When block RAM was specified for ROM, the *XOR_by_ROM* circuit reduced 17% of slices from *AES_TBL* in exchange for 12 more BRAMs and 29% degradation of performance. When distributed RAM was specified for ROM, 31% slices were reduced with the same number of BRAMs in exchange for 10% performance degradation. In short, *XOR_by_ROM* displays better logic scale and AT product than *AES_TBL* if distributed RAMs are used.

We now proceed to *XOR&S-Box_by_ROM* circuit. With the BRAM implementation, it achieved 59% reduction of the logic scale from *XOR_by_ROM*. The absorption of the S-Box in the ROM completely removed the look-up tables dedicated to the S-Box. *XOR&S-Box_by_ROM* also exhibited the sub-optimal performance among all implementations, which was only 2.3% lower than *AES_TBL*. However, when the proposed circuit was implemented with the distributed RAM, it required about ten times more slices on average than the BRAM implementation. Since the content of the ROM is relatively simple in *XOR_by_ROM*, it is likely that logic optimizer reduces the logic scale of ROMs. Meanwhile, the content of the ROM is much complicated in *XOR&S-Box_by_ROM*, where *AddRoundKey* and *SubBytes* are unified in a single set of ROMs. Thus, the reduction of distributed RAM is not expected in optimization. In summary, the proposed circuit takes better advantage of the feature of BRAMs than the existing circuits.

B. Encryption/Decryption Circuit

The AES cryptographic circuits that support both encryption and decryption are then implemented and evaluated. The target FPGA is changed to larger one, Virtex-5 XC5VLX50, due to the shortage of BRAM elements. Since the decryption circuits of *AES_PPRM1*, and *AES_PPRM3* are not provided, they are excluded from this evaluation. For *XOR_by_ROM*, *ROM sharing* version, where the ROMs are shared between the encryption and the decryption circuits as described in Section IV, is also evaluated. BRAM implementations are examined for all designs.

Table II summarize the evaluation results. The number of LUTs and Flip-Flops, primary components of slices, are also shown in the table for reference. The proposed circuit achieved 19% reduction of the logic scale and 31% improvement of performance over *XOR_by_ROM* without ROM sharing. The number of BRAMs required was the same between them. In comparison with the ROM sharing version, the reduction of the

[1] The number of clock cycles is the same for each design to encrypt a block.

TABLE II. FPGA IMPLEMENTATION RESULTS OF AES ENCRYPTION/DECRYPTION CIRCUITS WITH VIRTEX-5 XC5VLX50.

Design	Logic Scale			36kb	Max. Freq.	AT Product
	[Slices]	LUT	Flip-Flop	BRAM	[MHz]	[Slices/MHz]
AES_TBL [11]	886	2572	1036	2	236	3.8
AES_Comp [11]	1573	4262	1037	2	150	10.4
XOR_by_ROM [9]	784	2188	382	34	157	5.0
XOR_by_ROM (ROM sharing)	678	2187	382	18	157	4.3
XOR&S-Box_by_ROM	638	1180	382	34	205	3.1

TABLE III. POWER CONSUMPTION WITH SASEBO-GII BOARD.

Design	ROM Style	Power Consumption [nJ]
AES_TBL [11]	BRAM	306
	Dist. RAM	291
AES_Comp [11]	-	452
AES_PPRM1 [11]	-	431
AES_PPRM3 [11]	-	509
XOR_by_ROM [9]	BRAM	300
	Dist. RAM	301
XOR&S-Box_by_ROM	BRAM	218
	Dist. RAM	522

logic scale with the proposed circuit was decreased to 9.4% and the proposed circuit had 16 more BRAMs. Though the number of LUTs in use was almost halved from *XOR_by_ROM* regardless of whether ROM sharing is applied, it did not have much influence on the number of slices. The possible reason is the high utilization ratio of BRAMs, which makes the in-use logic elements dispersed. Nevertheless, the proposed circuit exhibited the best AT product among them.

VI. EVALUATION OF POWER CONSUMPTION

This section examines the power consumption of the circuit with the SASEBO-GII [13] board, which has the Virtex-5 FPGA onboard. Each design was evaluated by ten rounds of encryption. A round is composed of 1000 blocks of plaintext. In order to minimize the influence of the temperature, one round of evaluation is made for all circuits in order, and then it is repeated ten times. The average power consumption per block is used as the result. The clock frequency is set to 2 MHz.

The results are summarized in Table III. Our *XOR&S-Box_by_ROM* circuit with BRAM consumed 218 nJ, which was 71% of *AES_TBL* or 73% of *XOR_by_ROM*, for the encryption of a block. Comparing the power consumption results shown in Table III with the implementation results shown in Table I, it is observed that the number of slices, rather than BRAMs, was strongly correlated to the power consumption. According to the power waveform, static power was dominant in these circuits. It is likely to be related to the number of logic elements in use. As *XOR&S-Box_by_ROM* was the smallest circuit in logic scale, it might have achieved the least power consumption.

VII. CONCLUSION

In this paper, we proposed the S-Box absorption where both the *AddRoundKey* and the *SubBytes* AES functions are integrated into ROMs. The proposed implementation reduced the logic scale by 59% and the power consumption by 27% in the Virtex-5 FPGA from the existing key-specific AES circuit. Particularly, it achieved the smallest power consumption in all designs examined.

The items left for future works include the application to other FPGA architecture and the quick method to replace AES key.

ACKNOWLEDGMENT

This study was partially supported by a Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1] National Institute of Standards and Technology (NIST), “ADVANCED ENCRYPTION STANDARD (AES),” FIPS Publication 197. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[2] S. Morioka and A. Satoh, “An Optimized S-Box Circuit Architecture for Low Power AES Design,” in *Proc. 4th International Workshop on Cryptographic Hardware and Embedded Systems*, 2003, pp. 172–186.

[3] N. Ahmad and S. R. Hasan, “Low-power compact composite field AES S-Box/Inv S-Box design in 65 nm CMOS using Novel XOR Gate,” *Integration, the VLSI Journal*, vol. 46, no. 4, pp. 333–344, 2013.

[4] S. Tillich, M. Feldhofer, and J. Großschädl, “Area, Delay, and Power Characteristics of Standard-cell Implementations of the AES S-box,” in *Proc. 6th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation*, 2006, pp. 457–466.

[5] T. Sugawara, N. Homma, T. Aoki, and A. Satoh, “Differential power analysis of AES ASIC implementations with various S-box circuits,” in *Proc. 19th European Conference on Circuit Theory and Design*, 2009, pp. 395–398.

[6] J. M. Granado-Criado, M. A. Vega-Rodráguez, J. M. Sánchez-P’erez, and J. A. Gómez-Pulido, “A new methodology to implement the AES algorithm using partial and dynamic reconfiguration,” *Integration, the VLSI Journal*, vol. 43, no. 1, pp. 72–80, 2010.

[7] F.-X. Standaert, G. Rouvroy, J.-J. Quisquater, and J.-D. Legat, “Efficient Implementation of Rijndael Encryption in Reconfigurable Hardware: Improvements and Design Tradeoffs,” in *Proc. 4th International Workshop on Cryptographic Hardware and Embedded Systems*, 2003, pp. 334–350.

[8] C. Consel and O. Danvy, “Tutorial Notes on Partial Evaluation,” in *Proc. 20th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL’93)*, 1993, pp. 493–501.

[9] R. Atono and S. Ichikawa, “Design and Evaluation of Data-dependent Hardware for AES Encryption Algorithm,” *IEICE Transactions on Information and Systems*, vol. E89-D, no. 7, pp. 2301–2305, 2006.

[10] S. Matsuoka and S. Ichikawa, “Reduction of power consumption in key-specific aes circuits,” in *Proc. 3rd International Conference on Networking and Computing*, 2012, pp. 323–325.

[11] Aoki Laboratory, “Cryptographic Hardware Project, Graduate School of Information Sciences, Tohoku University.” [Online]. Available: http://www.aoki.ecei.tohoku.ac.jp/crypto/web/cores.html

[12] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A Compact Rijndael Hardware Architecture with S-Box Optimization,” in *Proc. 7th International Conference on the Theory and Application of Cryptology and Information Security*, 2001, pp. 239–254.

[13] SASEBO Project, “Side-channel Attack Standard Evaluation Board (SASEBO) – SASEBO-GII.” [Online]. Available: http://satoh.cs.uec.ac.jp/SASEBO/en/board/sasebo-g2.html