High-performance ASIC Implementations of the 128-bit Block Cipher CLEFIA

Takeshi Sugawara, Naofumi Homma, Takafumi Aoki
Graduate School of Information Sciences,
Tohoku University,
Sendai, Miyagi, Japan
{sugawara, homma}@aoki.ecei.tohoku.ac.jp
aoki@ecei.tohoku.ac.jp

Akashi Satoh
National Institute of
Advanced Industrial Science and Technology,
Research Center for Information Security,
1-18-13, Sotokanda, Chiyoda, Tokyo, 101-0021, Japan
Akashi.satoh@aist.go.jp

Abstract— **In the present paper, we introduce high-performance hardware architectures for the 128-bit block cipher CLEFIA and evaluate their ASIC performances in comparison with the ISO/IEC 18033-3 standard block ciphers (AES, Camellia, SEED, CAST-128, MISTY1, and TDEA). We designed five types of hardware architectures for CLEFIA, combining two loop structures and three F-functions. These designs were synthesized with a 90-nm CMOS standard cell library, and size and speed performances were evaluated. The highest hardware efficiency (defined as throughput/gates) obtained was 400.96 Kbps/gates, which is 1.5 times higher than previously achieved efficiencies.**

I. INTRODUCTION

CLEFIA is a 128-bit block cipher developed by SONY Corporation [1]. Previous studies [2, 3] reported that CLEFIA has the advantages of speed and size in hardware implementation, as compared to conventional algorithms, and a throughput of over 1.6 Gbps with less than 6 Kgates was achieved in a 90-nm CMOS standard cell library. However, in these studies, other block ciphers were not implemented, and the 90-nm library used was more advanced than libraries used in other studies on hardware implementation of block ciphers. Therefore, the advantage of CLEFIA hardware remains unclear.

In the present paper, we present high-speed and compact datapath architectures for CLEFIA. In addition, we applied various optimization techniques to each basic function block. Five architectures combining these methods were then synthesized using a 90-nm CMOS standard cell library with speed and area optimization options. We also designed and synthesized all of the ISO18033-3 standard block ciphers (AES, Camellia, SEED, CAST-128, MISTY1, and TDEA) [4] in order to fairly compare their performances with CLEFIA under the same conditions.

II. CLEFIA ALGORITHM

Figure 1 shows a data randomization block of CLEFIA using the Generalize Feistel Network (GFN) structure, which is an *n*-branch expansion of the Feistel Network. CLEFIA uses a four-branch GFN dividing a 128-bit block into four 32-bit sub-blocks. Two types of 32-bit input/output F-functions, F_0 and F_1, are used for the data randomization. The F-functions consist of XORs, 8-bit input/output S-boxes, and multiplications by diffusion matrices M_0 and M_1. The S-box S_1 is defined as a multiplicative inverse over a Galois field $GF(2^8)$ and an affine transformation similar to AES and Camellia. The S-box S_0 is defined as a combination of four 4-bit input/output S-boxes (SS_0 ~ SS_0) and constant multiplications over $GF(2^4)$.

The key scheduler block of CLEFIA is shown in Figure 2. Round keys RK_i are generated by XORing a secret key K, an intermediate key L, and constant values CON_i. The intermediate key L is generated by mixing K and CON_i using the GFN. The intermediate key L is updated by a *DoubleSwap* function Σ, which is defined as a bit-wise permutation. The constant values CON_i are defined as a combination of smaller constants T, P, and Q with constant multiplications over $GF(2^{16})$.

III. OPTIMIZATION OF CIRCUIT COMPONENTS

A. *Compact implementations of S-boxes S_0 and S_1*

A straightforward implementation of S-boxes is based on a lookup table that is automatically generated by synthesis tools.

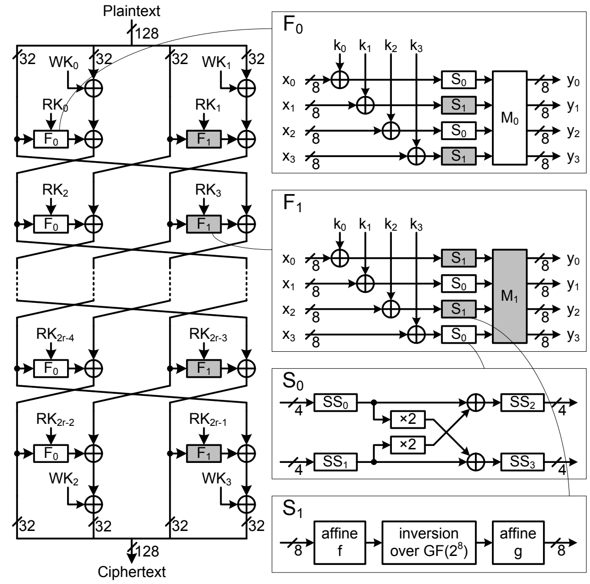

Figure 1 Data randomization block

A different type of implementation technique is also available for the S-box S_1 using a Galois field arithmetic [5], which considerably reduces the circuit area [2, 3]. The compact implementation of S_1 is derived from an inversion over the composite field $GF((2^4)^2)$ transformed from the original field $GF(2^8)$. The size of S-box S_0 can also be reduced by using four smaller S-boxes SS_0 ~ SS_0 and constant multipliers over $GF(2^4)$.

B. *Sharing the diffusion matrices M_0 and M_1*

Permutation functions in F-functions are defined by constant matrices M_0 and M_1 with elements of {02, 04, 06, 08, 0A} $\in GF(2^8)$. The compact implementations, which only require constant multipliers of {02, 04, 08} $\in GF(2^8)$ and XORs, are proposed in References [2, 3]. In addition, we introduce a further reduction technique to share resources between M_0 and M_1. Figure 3 shows the linear transformation circuit sharing constant {02} multipliers and XOR gates.

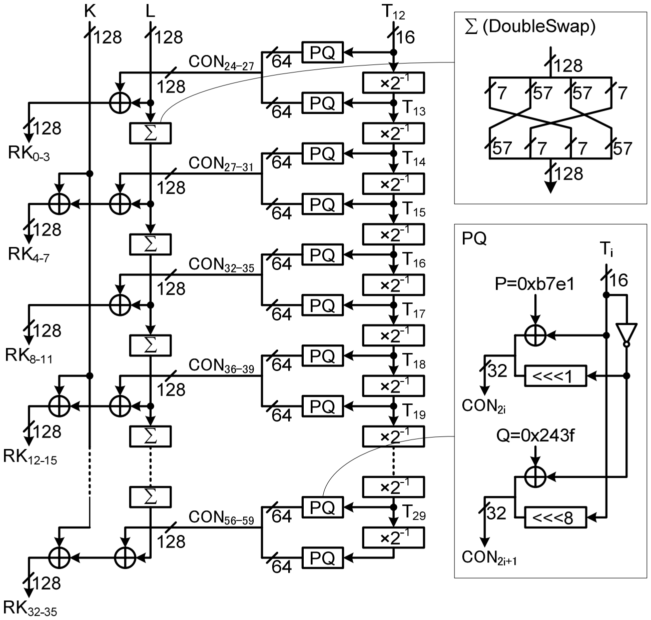

Figure 2 Key-scheduling block

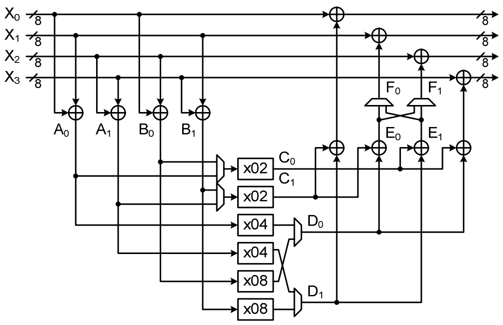

Figure 3 Linear transformation circuit for multiplications by M_0 and M_1

C. *Fast implementation using T-boxes*

T-box is a technique to merge an S-box and following linear transformation layers to shorten the critical path, although it requires large hardware resources. The T-box technique can also be applied to CLEFIA by merging S-boxes S_0 and S_1 and multiplications by the diffusion matrices M_0 and M_1. Reference [2] proposed four types of T-boxes considering a trade-off between throughput and hardware cost. For example, the Type-1 T-box consists of 8-bit input 32-bit output tables TB_{00} ~ TB_{13} defined by the following equations:

$$\begin{aligned}
TB_{00}(x) &= (\quad S_0(x), 02\times S_0(x), 04\times S_0(x), 06\times S_0(x))\\
TB_{01}(x) &= (02\times S_1(x),\quad S_1(x), 06\times S_1(x), 04\times S_1(x))\\
TB_{02}(x) &= (04\times S_0(x), 06\times S_0(x),\quad S_0(x), 02\times S_0(x))\\
TB_{03}(x) &= (06\times S_1(x), 04\times S_1(x), 02\times S_1(x),\quad S_1(x))\\
TB_{10}(x) &= (\quad S_1(x), 08\times S_1(x), 02\times S_1(x), 0\text{A}\times S_1(x))\\
TB_{11}(x) &= (08\times S_0(x),\quad S_0(x), 0\text{A}\times S_0(x), 02\times S_0(x))\\
TB_{12}(x) &= (02\times S_1(x), 0\text{A}\times S_1(x),\quad S_1(x), 08\times S_1(x))\\
TB_{13}(x) &= (0\text{A}\times S_0(x), 02\times S_0(x), 08\times S_0(x),\quad S_0(x))
\end{aligned}$$

The S-boxes and the linear transformations in CLEFIA can be executed using TB_{00} ~ TB_{13} and XORs.

D. *On-the-fly CON_i generation*

The constants CON_i require several memory element (e.g., registers or memory modules) if they are computed and stored in advance. For more details, CON_1 ~ CON_{60} requires 1,920 bits (= 32 bits × 60). Even if we store the smaller constants T_1 ~ T_{30} instead of CON_i, 480 bits (16 bits × 30) are required. Therefore, on-the-fly CON_i generation is quite effective for reducing the hardware resources. Figure 4 shows a schematic diagram of the on-the-fly CON_i generator, which consists of a PQ operator, which generates CON_i from T_i, and two constant multipliers where the constants are 2, $2^{-1} \in GF(2^{16})$. Register T stores the value of T_i updated by the constant multipliers on the fly. 16-bit constants IV, T_{12}, and T_{29} are used to reset the value in register T.

IV. CIRCUIT ARCHITECTURES

Figures 5 and 6 show the proposed datapath architectures with the above-described component optimizations. The Type-A architecture in Figure 5 executes one round operation in one clock cycle, while the Type-B architecture in Figure 6 performs one round operation in two clock cycles. These architectures support both encryption and decryption in the ECB mode with a 128-bit key. Therefore, both Type-A and Type-B architectures require 18 and 36 clock cycles,

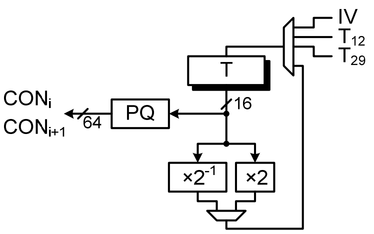

Figure 4 On-the-fly CON_i generator

respectively, for one-block encryption (or decryption). The intermediate key L is generated by the data randomization block. The encryption (or decryption) processes can be executed continuously without any interval cycles. In both architectures, we implemented two types of S-boxes: the lookup table and composite field versions described in Section III-A. In addition, the T-box described in Section III-C was also applied to the Type-A architecture. Round keys RK_i are computed and stored in an internal register one clock prior to use, which enables the reduction of the critical path. We also reduce the number of selectors using the functions Ω, Ψ, and Φ, rather than the *DoubleSwap* function Σ, as described in [4].

In the Type-B architecture, we designed an F_0/F_1 component combining F-functions F_0 and F_1, where we shared the S-boxes as well as the constant matrices M_0 and M_1, as described in Section III-B. In addition, the circuit area for the component PQ is reduced by sharing XOR gates.

V. Performance Evaluation in ASIC

Table I shows the synthesis results of the Type-A and Type-B architectures using the STMicroelectronics 90-nm CMOS standard cell library (1.2V-volt version) [6]. The designs are synthesized by a Synopsys Design Compiler with two optimization options: size and speed. Hardware sizes were estimated based on a two-way NAND equivalent gate, and the speeds were evaluated under the worst-case conditions. The efficiency is defined as the throughput per gate, and thus higher efficiency indicates better implementation. For the comprehensive performance comparison of block ciphers, we also designed all of the ISO standard algorithms AES, Camellia, SEED, CAST-128, MISTY1, and TDEA [4] and evaluated their sizes and speeds under the same condition. A compact AES implementation [7] with a 0.13 μm library is also shown in Table I.

The highest efficiency of 400.96 Kbps/gate was obtained by the Type-A architecture with compact S-boxes. This value was more than twice as high as that of AES (175.72 Kbps/gates), which is the highest value among the ISO standard ciphers. The highest throughput of 5.31 Gbps was achieved in the Type-A architecture with the T-box, but the efficiency remained low due to the large T-box circuit. The throughput was lower than the value of 7.31 Gbps achieved by AES because the Feistel-type CLEFIA requires 18 rounds while the SPN-type AES takes 10 rounds. The Type-B architecture achieved a circuit area of 5.49 Kgates, which is the smallest among all of our CLEFIA implementations. AES in [7] has a smaller circuit area of 3.10 Kgates, but it does not support decryption and its efficiency is much lower than that of our Type-B even if the difference in circuit technology is considered. On the other hand, the circuit area of 5.49 Kgates for Type-B with compact S-box is only 9% smaller than that of 6.05 Kgates for Type-A with the compact S-box. This result indicates that the F_0/F_1 merged technique has a limited effect on area reduction because the size of additional selectors is not negligible. Compared with References [2] and [3], the present circuit has a higher throughput but a larger circuit area. This is due to the design strategy of the present study of prioritizing the shortening of the critical path compared to the reduction of the circuit area. As a result, we improved the efficiency and the throughput by 50% and 75%, respectively, as compared to the designs reported in References [2] and [3].

VI. Conclusion

In the present paper, we proposed high-performance hardware architectures for the 128-bit block cipher CLEFIA,

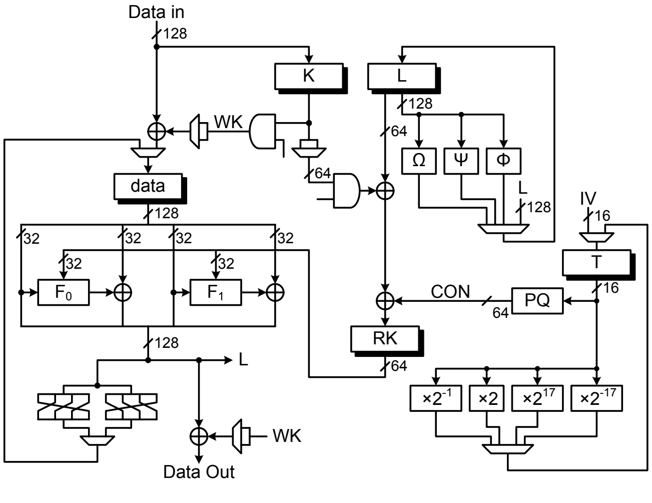

Figure 5 Type-A architecture

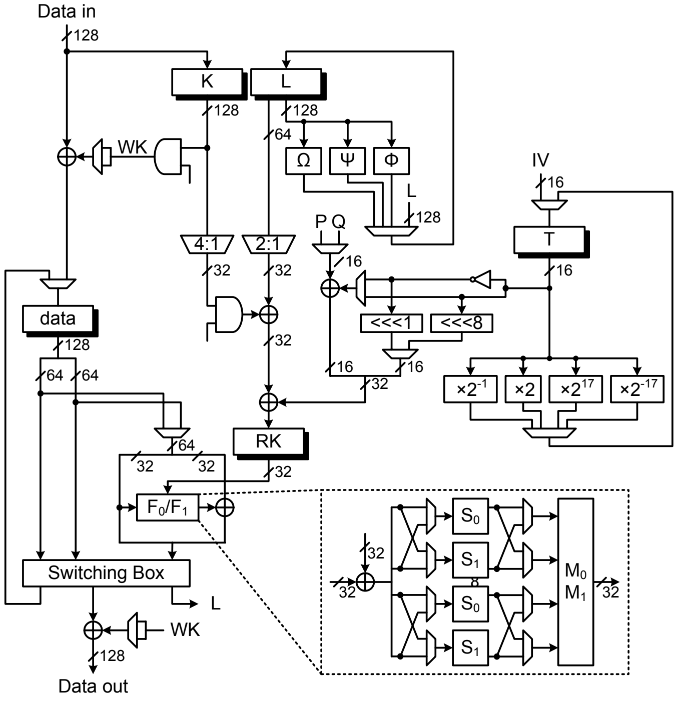

Figure 6 Type-B architecture

and evaluated their performances using a 90-nm CMOS standard cell library. The highest efficiency of 400.96 Kbps/gates was achieved, which is the highest among the ISO standard block ciphers and is 50% higher than value of 268.63 Kbps/gates of References [2] and [3], in which a different 90-nm library was used. The highest throughput of 5.31 Gbps is also higher than the value of 3.00 Gbps of References [2] and [3]. These results clarified that CLEFIA is suitable for hardware implementation, and the proposed architectures have significant advantages over previous architectures.

REFERENCES

[1] Sony Corporation, “The 128-bit Block Cipher CLEFIA Algorithm Specification,” Jun. 2007, http://www.sony.co.jp/Products/clefia /technical/data/clefia-spec-1.0.pdf.

[2] Sony Corporation, “The 128-bit Block Cipher CLEFIA Security and Performance Evaluations,” Jun. 2007, http://www.sony.co.jp /Products/clefia/technical/data/clefia-eval-1.0.pdf.

[3] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, “Hardware Implementations of the 128-bit Blockcipher CLEFIA,” Technical report of IEICE, vol. 107, no. 141, ISEC2007-49, pp. 29-36, Jul. 2007 (in Japanese)

[4] T. Sugawara, N. Homma, T. Aoki, and A. Satoh, "ASIC Performance Comparison for the ISO Standard Block Ciphers," JWIS2007, pp. 485-498, Aug. 2007.

[5] A. Rudra, P.K. Dubey, C.S. Julta, V. Kumar, J.R. Rao, and P. Rohatgi, “Efficient Rijndael Encryption Implementation with Composite Field Arithmetic,” CHES 2001, LNCS 2162, pp. 171-184, Springer-Verlag, 2001.

[6] Circuits Multi-Projets (CMP), CMOS 90nm (CMOS090) from STMicroelectronics, http://cmp.imag.fr/products/ic/?p=STCMOS090

[7] P. Hämäläinen, T. Alho, M. Hännikäinen, and T. Hämäläinen, “ Design and Implementation of Low-area and Low-power AES Encryption Hardware Core,” DSD '06: Proceedings of the 9th EUROMICRO Conference on Digital System Design, pp. 577-583, 2006.

Table 1 Hardware Performance Comparison of Block Ciphers in a 90 nm-CMOS ASIC (gate = 2-way NAND)

Design	Algorithm	Mode	Circuit technology (nm)	Block Length (bits)	Key Length (bits)	Cycle	S-box	Opti-mize	Freq. (MHz)	Thr'put (Mbps)	Area (Kgates)	Efficiency (Kbps/ gates)
This work	CLEFIA Type-A	Enc./Dec.	90	128	128	18	Table	Area	362.32	2,576.49	10.38	248.18
								Speed	571.43	4,063.49	11.16	364.05
							Compact	Area	210.08	1,493.93	6.05	247.12
								Speed	526.32	3,742.69	9.33	400.96
							T-box	Area	564.97	4,017.58	13.83	290.52
								Speed	746.27	5,306.80	21.07	251.81
	CLEFIA Type-B	Enc./Dec.	90	128	128	36	Table	Area	361.01	1,283.59	8.10	158.55
								Speed	518.13	1,842.26	12.25	150.41
							Compact	Area	148.15	526.75	5.49	95.91
								Speed	361.01	1,283.59	6.90	186.00
[2, 3]	CLEFIA	Enc./Dec.	90	128	128	18		Area	225.83	1,605.94	5.98	268.63
								Speed	422.29	3,003.00	12.01	250.06
						36		Area	201.28	715.69	4.95	144.59
								Speed	389.55	1,385.10	9.38	147.71
[4]	AES	Enc./Dec.	90	128	128	10	Table	Area	267.38	3,422.46	27.77	123.26
								Speed	571.43	7,314.29	45.90	159.37
							$GF(((2^2)^2)^2)$	Area	174.22	2,229.97	15.14	147.29
								Speed	266.67	3,413.33	19.43	175.72
	Camellia	Enc./Dec.	90	128	128	23	Table	Area	267.38	1,488.03	11.44	130.11
								Speed	490.20	2,728.05	19.95	136.75
							$GF((2^4)^2)$	Area	174.52	971.24	7.79	124.75
								Speed	363.64	2,023.72	13.20	153.33
	SEED	Enc./Dec.	90	128	128	16	Table	Area	114.16	913.24	16.75	54.52
								Speed	194.55	1,556.42	25.14	61.90
						52	Table	Area	210.08	517.13	9.57	54.01
								Speed	364.96	898.37	12.33	72.88
	CAST-128	Enc./Dec.	90	64	128	17	Table	Area	102.56	386.12	20.11	19.20
								Speed	241.55	909.35	33.11	27.46
	MISTY1	Enc./Dec.	90	64	128	9	Table	Area	128.70	915.20	14.07	65.04
								Speed	209.21	1,487.68	17.22	86.37
						30	Table	Area	267.38	570.41	7.92	72.06
								Speed	362.32	772.95	10.12	76.41
	TDEA	Enc./Dec.	90	64	56, 112, 168	48	Table	Area	266.67	355.56	3.76	94.50
								Speed	574.71	766.28	5.28	145.10
[7]	AES	Enc.	130	128	128	160	$GF(((2^2)^2)^2)$	Area	152.00	121.00	3.10	39.03
								Speed	290.00	232.00	3.90	59.49