SCIS 2011 The 2011 Symposium on Cryptography and Information Security
Kokura, Japan, Jan. 25-28, 2011
The Institute of Electronics, Information and Communication Engineers

128ビットブロック暗号CLEFIAの小型ハードウェア実装評価
Compact Hardware Implementations of the 128-bit Blockcipher CLEFIA

秋下 徹 * 樋渡 玄良 *
Toru Akishita Harunaga Hiwatari

あらまし 128 ビットブロック暗号 CLEFIA は高いハードウェア実装効率を持ったブロック暗号であることが知られている．本稿では, CLEFIA のハードウェア実装の更なる小型化を目指して, 128 ビット鍵の CLEFIA に対して 8 ビット・シリアルアーキテクチャに基づいた実装を行ない, 0.13 μm CMOS 標準セルライブラリを用いてその実装性能を評価した．その結果, 暗号化実装において 2.9 kGE 以下, 暗復号実装において 3.0 kGE 以下というゲート規模の評価が得られた．これらの数値は AES の最小実装を下回る数値であり, CLEFIA のハードウェア実装における小型実装性能を示している．

キーワード ブロック暗号, CLEFIA, ハードウェア実装

1 はじめに

CLEFIA [4, 5] はブロック長 128 ビット, 鍵長 128, 192, 256 ビットに対応した, AES [1] と入出力仕様の互換性を持つブロック暗号である．CLEFIA には最新の研究成果や設計手法が盛り込まれており, 既存の攻撃法に対して十分な安全性を有しつつ, ハードウェア・ソフトウェアを問わず効率的な実装が可能である．特に, ハードウェア実装については顕著な性能を発揮していることが, 数値データにより示されている [6, 7, 8].

一方, RFID のような実装上の制約条件の高い環境下では暗号プリミティブに使用可能なゲート規模は 250 – 4,000 GE 程度であるとの報告もあり [9], より小さなゲート規模で実装可能であることも重要となる．実際に AES は 2005 年に Feldhofer らによって RAM を用いた実装アーキテクチャが提案されており, 暗復号のサポートで 3.4 kGE で実装可能であることが報告されている [2]. また, 2006 年には Hämäläinen らによりシフトレジスタを用いたシリアルアーキテクチャが提案されている [3]. 暗号化のみのサポートではあるが, 入出力を除くと 160 サイクルと比較的高速で処理を行ないながらも 3.1 kGE で実装可能であることが示されている．

そこで, 本稿ではハードウェア実装の更なる小型化を目指して, 128 ビット鍵の CLEFIA に対してシフトレジスタ による 8 ビット・シリアルアーキテクチャの設計を行なった．CLEFIA では, AES で行列演算を行う際にブロック長分のレジスタとは別に追加される 24 ビット分のレジスタが不要となることを示す．また, 復号処理の場合に偶数ラウンド目の F 関数 F_0 と F_1 の処理順を入れ替えることにより, 暗号化実装のデータパスをほとんど変えずに暗復号とも実装可能となることを示す．これらの実装手法を適用することにより 0.13 μm CMOS 標準セルライブラリを用いた評価において 2,893 GE で暗号化実装が, 2,996 GE で暗復号実装が可能となった．これらの数値は 128 ビット鍵の CLEFIA に対する既存のハードウェア実装の中で最小の数値である．また, Feldhofer らや Hämäläinen らの実装を下回る数値であり, CLEFIA のハードウェア実装における小型実装性能を示している．

以下, まず第 2 節にて 128 ビット鍵 CLEFIA のアルゴリズム概要, 第 3 節にて AES の Hämäläinen らの小型実装について述べる．次に, 第 4 節にて 8 ビット・シリアルアーキテクチャを用いた CLEFIA 小型暗号化実装について, 第 5 節にて CLEFIA 小型暗復号実装について述べる．その後, 第 6 節にて ASIC による実装性能評価について述べ, 最後に第 7 節にてまとめを行なう．

2 128 ビット鍵 CLEFIA

128 ビット鍵 CLEFIA のアルゴリズムの概要を述べる．詳細については, 文献 [4, 5] を参照されたい．

* ソニー株式会社, 〒 141-0001, 東京都品川区北品川 5-1-12 御殿山 Tec, Sony Corporation, Gotenyama Tec. 5-1-12 Kitashinagawa Shinagawa-ku, Tokyo, 141-0001.

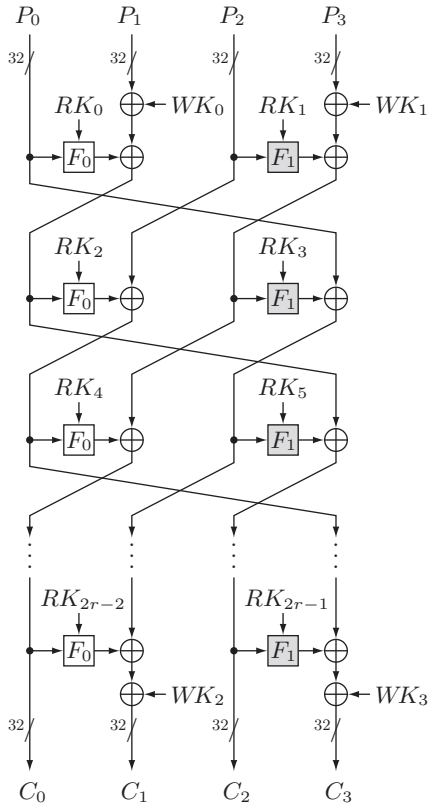


図 1: CLEFIA のデータ処理部

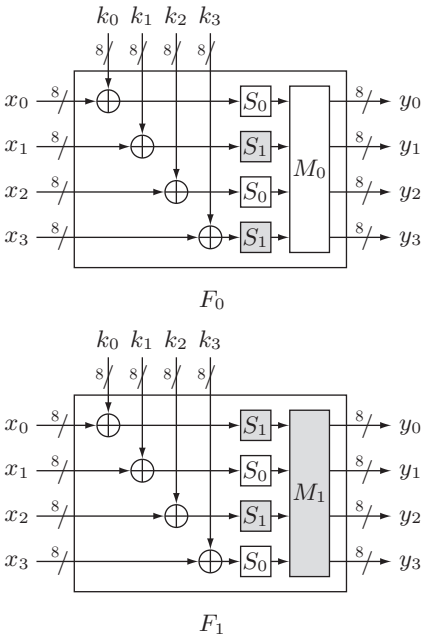


図 2: F 関数 F_0, F_1

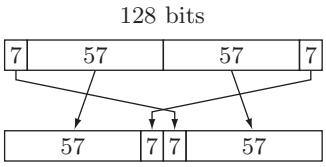


図 3: *DoubleSwap* 関数 Σ

2.1 データ処理部

CLEFIA のデータ処理部は 4 系列の Type-2 一般化 Feistel 構造 [10] を採用しており, 1 ラウンドで 2 つの異なる 32 ビット入出力 F 関数 F_0, F_1 を使用する．ラウンド数 r は使用する鍵長に応じて決まっており, 128 ビット鍵の場合は 18 となっている．図 1 に CLEFIA のデータ処理部を示す.

F 関数 F_0, F_1 は, それぞれが 4 つの 8 ビット入出力 S-box と 1 つの拡散行列から構成されている．これらの F 関数は S-box としても 2 つの異なる S-box を使用しており, 1 つはランダムに選択された 4 つの 4 ビット入出力 S-box をベースにした S-box S_0 であり, もう 1 つは GF(2^8) 上の逆元演算をベースにした S-box S_1 である．F 関数 F_0 ではこれらの S-box を S_0, S_1, S_0, S_1 の順に配置し, 拡散行列として M_0 を使用する．もう 1 つの F 関数 F_1 では S-box を S_1, S_0, S_1, S_0 と F_0 とは異なる順に配置し, 拡散行列も F_0 で使用されているものとは異なる M_1 を使用する．図 2 に F 関数 F_0, F_1 をそれぞれ示す．拡散行列 M_0, M_1 は以下のように定義される．なお, 行列内の要素は 16 進数表現である．

$$M_0 = \begin{pmatrix} 01 & 02 & 04 & 06 \\ 02 & 01 & 06 & 04 \\ 04 & 06 & 01 & 02 \\ 06 & 04 & 02 & 01 \end{pmatrix}, \quad M_1 = \begin{pmatrix} 01 & 08 & 02 & 0A \\ 08 & 01 & 0A & 02 \\ 02 & 0A & 01 & 08 \\ 0A & 02 & 08 & 01 \end{pmatrix}.$$

これらは各要素 h_{ij} $(0 \le i, j \le 3)$ が 8 ビットの 4×4 Hadamard-type 行列であり, その要素は 4 つの 8 ビット値 a_0, a_1, a_2, a_3 を用いて $h_{ij} = a_{i \oplus j}$ として表現できる．これらの行列, ベクトル間の乗算は原始多項式 $z^8 + z^4 + z^3 + z^2 + 1$ で定義される GF(2^8) 上の演算として定義される.

2.2 鍵スケジュール部

CLEFIA の鍵スケジュール部は, 秘密鍵を入力とし, ホワイトニング鍵 WK_i $(0 \le i < 4)$, およびラウンド鍵 RK_j $(0 \le j < 2r)$ を出力する．この鍵スケジュール部は (1) 秘密鍵 K より中間鍵 L を生成, (2) K と L より WK_i, RK_j を生成, という 2 つの処理より構成される.

(1) では, データ処理部と同様の関数を用い, 入力として秘密鍵 K, ラウンド鍵として定数 CON を用い, 出力として中間鍵 L を得る．この変換では 128 ビット鍵の場合は, データ処理部と同じ 4 系列の Type-2 一般化 Feistel 構造を 12 ラウンド実行する．(2) では, 中間鍵 L を *DoubleSwap* 関数 Σ と呼ばれる置換関数を用いて順次変換し, 秘密鍵 K, 定数 CON と混ぜ合わせることにより, ホワイトニング鍵 WK_i, ラウンド鍵 RK_j を生

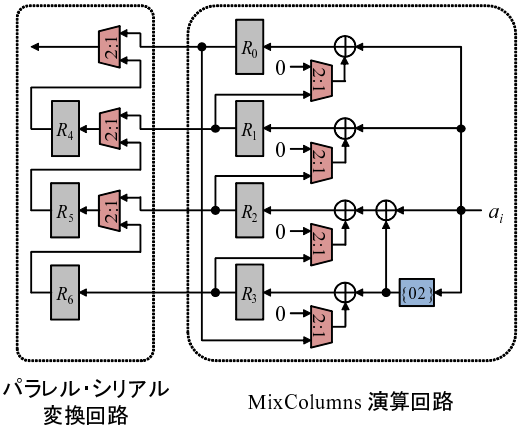


図 4: Hämäläinen 実装 MixColumns 演算

成する．*DoubleSwap* 関数 Σ は, 128 ビット入出力の置換関数であり, 高い攪拌性能と効率的な実装が可能という特長を併せ持つ．この *DoubleSwap* 関数 Σ を図 3 に示す.

3 Hämäläinen らの AES 小型実装

Hämäläinen らは低コスト, 低消費電力のデバイスに適した 128 ビット鍵 AES の小型暗号化コアを目指して, シフトレジスタを用いた 8 ビット・シリアルアーキテクチャを提案した [3]．Feldhofer らの実装 [2] が RAM を用いた実装アーキテクチャであるのに対して, シフトレジスタを用いることにより, 入出力を除くと 160 サイクルと比較的高速で処理を行ないながらも 3.1 kGE で実装可能であることが示されている．

データ処理回路には S-box 演算回路が 1 つのみ配置されており, AES 1 ラウンド分の処理を 16 サイクルかけて実行する．MixColumns 演算を行なうために 4 サイクル連続で S-box の出力を MixColumns 演算回路に入力する．図 4 に MixColumns 演算回路を示す．図内のデータ幅は 8 ビットとなっている．S-box の出力 a_i $(0 \le i \le 3)$ は順に MixColumns 演算回路に入力され, 4 サイクル後に MixColumns 演算の出力がレジスタ R_0, R_1, R_2, R_3 に格納される．しかしながら, 次のサイクルでは別の MixColumns 演算用の S-box の出力が入力されるため R_1, R_2, R_3 に格納されているデータはパラレル・シリアル変換回路内にあるレジスタ R_4, R_5, R_6 に一旦移され, 順に出力されることになる．従って, ブロック長 128 ビット以外に 8 ビットのレジスタが 3 本の計 24 ビット分のデータレジスタが必要となる．

4 CLEFIA 小型暗号化実装

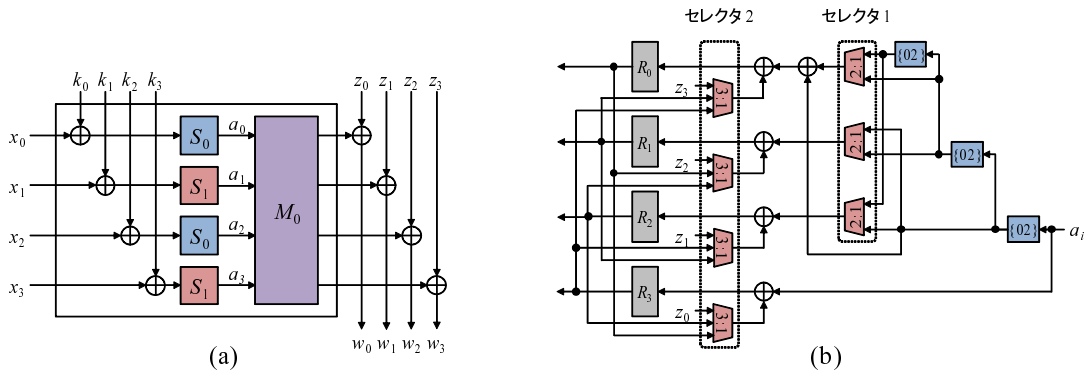
本節では, 8 ビット・シリアルアーキテクチャを用いた 128 ビット鍵 CLEFIA の小型暗号化実装について述べる．CLEFIA は第 2 節で述べたように, 鍵スケジュール部において *DoubleSwap* 関数と呼ばれるビット単位の置換関数を含んでいる．従って, Feldhofer らの実装のような RAM を用いたアーキテクチャでは, *DoubleSwap* 関数を用いてラウンド鍵 RK_j を生成するために必要なサイクル数または回路規模が増大すると考えられる．そこで, 我々は Hämäläinen らの実装と同様に, シフトレジスタを用いた 8 ビット・シリアルアーキテクチャの設計を行なった.

4.1 行列演算の実装手法

CLEFIA は第 2 節で述べたように 2 種類の拡散行列 M_0, M_1 を用いている．M_0, M_1 は, AES の MixColumns が 4×4 の巡回型行列であるのに対し, 4×4 の Hadamard-type 行列となっている．また, CLEFIA は SP 型の一般化 Feistel 構造であるため, 図 5 (a) に示すように F 関数 F_0 内の行列 M_0 の出力は隣の系列のデータと XOR されることになる．まず, この XOR と行列演算回路の XOR を共有することにより回路規模の削減を図る．

図 5 (b) に行列演算回路のデータパス, (c) に F_0 を演算する際にレジスタ R_i $(0 \le i \le 3)$ に格納されるデータを示す．1 サイクル目には S-box S_0 の出力 a_0 をそれぞれ {01}, {02}, {04}, {06} 倍したデータと, データ z_0, z_1, z_2, z_3 が XOR され, それぞれレジスタ R_3, R_2, R_1, R_0 に格納される．$2-4$ サイクル目には S-box の出力 a_i $(i = 1, 2, 3)$ をそれぞれ {01}, {02}, {04}, {06} 倍したデータとセレクタ 2 を通して選択されたデータが XOR され, それぞれレジスタ R_3, R_2, R_1, R_0 に格納される．図 5 (c) に示すように, 4 サイクル後には R_i にはそれぞれ w_i が格納される．また, 行列演算回路のセレクタ 1 を切り替えることにより F 関数 F_1 内の行列 M_1 の処理を行なうことも可能となる．

図 4 に示した AES の MixColumns 回路と比較すると, セレクタ 1 が追加され, セレクタ 2 が 2 入力セレクタから 3 入力セレクタに変更されている．セレクタ 1 の追加は CLEFIA が 2 種類の行列 M_0, M_1 を使用しているためで, セレクタ 2 の変更は行列 M_0, M_1 が Hadamard-type 行列であることによる．また, AES の MixColumns 回路と異なり, パラレル・シリアル変換回路が不要となる．レジスタ R_i に w_i が格納された後, 次の F 関数の処理を行なうために新たに z_i および a_0 が行列演算回路に入力されるが, z_i は入力された後不要となり, z_i が格納されていたレジスタに w_i を格納することが可能となるからである．従って, 8 ビット・シリアルアーキテクチャで CLEFIA のデータ処理部を実装した場合には, AES と異なりブロック長 128 ビット以外のデータレジスタは不要となり, 更なる回路規模の削減が可能となる.



cycle	1	2	3	4
R_0	$z_3 \oplus \{06\}a_0$	$z_2 \oplus \{04\}a_0 \oplus \{06\}a_1$	$z_1 \oplus \{02\}a_0 \oplus \{01\}a_1 \oplus \{06\}a_2$	$z_0 \oplus \{01\}a_0 \oplus \{02\}a_1 \oplus \{04\}a_2 \oplus \{06\}a_3\ (= w_0)$
R_1	$z_2 \oplus \{04\}a_0$	$z_3 \oplus \{06\}a_0 \oplus \{04\}a_1$	$z_0 \oplus \{01\}a_0 \oplus \{02\}a_1 \oplus \{04\}a_2$	$z_1 \oplus \{02\}a_0 \oplus \{01\}a_1 \oplus \{06\}a_2 \oplus \{04\}a_3\ (= w_1)$
R_2	$z_1 \oplus \{02\}a_0$	$z_0 \oplus \{01\}a_0 \oplus \{02\}a_1$	$z_3 \oplus \{06\}a_0 \oplus \{04\}a_1 \oplus \{02\}a_2$	$z_2 \oplus \{04\}a_0 \oplus \{06\}a_1 \oplus \{01\}a_2 \oplus \{02\}a_3\ (= w_2)$
R_3	$z_0 \oplus \{01\}a_0$	$z_1 \oplus \{02\}a_0 \oplus \{01\}a_1$	$z_2 \oplus \{04\}a_0 \oplus \{06\}a_1 \oplus \{01\}a_2$	$z_3 \oplus \{06\}a_0 \oplus \{04\}a_1 \oplus \{02\}a_2 \oplus \{01\}a_3\ (= w_3)$

(c)

図 5: CLEFIA 小型実装の行列演算回路: (a) F 関数 F_0 データ構造, (b) データパス, (c) レジスタ格納データ

4.2 回路アーキテクチャ

図 6 に 8 ビット・シリアルアーキテクチャを用いた CLEFIA の小型暗号化実装のデータパスを示す．なお, 図中で指定されていないデータパス幅は 8 ビットとなっている．データパスは, 大きくデータ処理回路と鍵スケジュール回路に分けることができる．CLEFIA は 1 ラウンド分のラウンド関数に S-box が 8 個配置されていることから, 1 ラウンドを 8 サイクルで処理する．1 ラウンドの処理でデータ処理回路内のレジスタ R_{ij} $(0 \le i, j \le 3)$ に格納されるデータの詳細な流れは付録の図 8 に示す．

暗号化処理時には, 8 ビットのデータ入力 data_in から供給された平文データが 16 サイクルかけて R_{ij} に配置された後, 暗号化を開始し, 18 ラウンドのラウンド関数に 144 サイクルを要する．最終ラウンドにはワード巡回処理がないことから, 暗号文データはレジスタ R_{30} を通して 16 サイクルかけて出力される．鍵セットアップ時には key_in から入力された秘密鍵 K が 16 サイクルかけてレジスタ R_{ij} に配置された後, 鍵セットアップを開始し, 12 ラウンド分のラウンド関数に 96 サイクルを要する．生成された中間鍵 L はレジスタ R_{30} を通して 16 サイクルかけて中間鍵レジスタ L_{ij} $(0 \le i, j \le 3)$ に格納される．

データ処理回路には 2 種類の S-box S_0, S_1 が配置され, それらの出力がセレクタにより選択されて行列演算回路に入力される．また, CLEFIA のハードウェア実装では, 文献 [6] で示されているように暗号化処理時のラウンド鍵の秘密鍵 K に関する部分を等価変形し, ホワイトニング鍵の処理と共有することにより, 回路規模が削減できることが知られている．データ処理回路では, この秘密鍵 K に関する処理を行列演算回路内に配置することにより更なる回路規模の削減を図っている．鍵入力 key_in から入力された秘密鍵 K は $K = K_0|K_1|K_2|K_3$ と 32 ビットに分割された後 4 入力の 32 ビットセレクタで選択され, 更に選択された 32 ビットデータが 8 ビットずつに分割された後 4 入力の 8 ビットセレクタで選択され, 行列演算回路に供給される．

鍵スケジュール回路には中間鍵レジスタ L_{ij} が配置され, 暗号化処理時にはレジスタ L_{00} と, CON_i を 8 ビットに分割し 4 入力の 8 ビットセレクタで選択したデータとが XOR され, データ処理回路に供給される．中間鍵レジスタ L_{ij} に格納された L は 8 ビットずつシフトする形でデータ処理部に供給されるが, 2 ラウンド毎に $DoubleSwap$ 関数 Σ でアップデートする必要があるため, 偶数ラウンドの 8 サイクル目には 8-bit shift + Σ の出力が選択され L_{ij} に入力される．また, 暗号化処理の最後には, 次の入力ブロックに対する処理のために 8-bit shift + Σ^{-8} の出力が選択され, L_{ij} を元の中間鍵 L に戻す．

5 CLEFIA 小型暗復号実装

本節では 8 ビット・シリアルアーキテクチャを用いた 128 ビット鍵 CLEFIA の小型暗復号実装について述べる．

CLEFIA の復号処理に対して, 文献 [6] で提案されたラウンド鍵の一部の等価変形を施すと, 図 7 (a) のように表すことができる．図中の RK_i $(0 \le i \le 35)$ はラウンド鍵の中間鍵 L に関する部分を示している．CLEFIA のデータ処理部は 4 系列の Type-2 一般化 Feistel 構

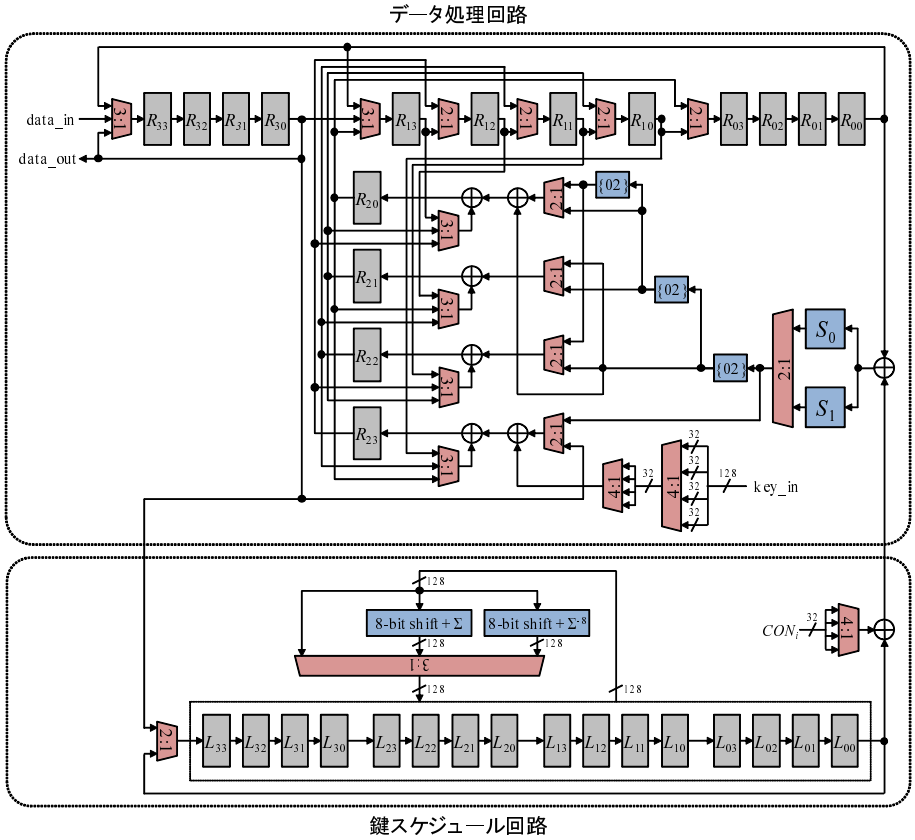


図 6: CLEFIA 小型暗号化実装のデータパス

造を採用しているため, 暗号化処理と復号処理ではラウンド関数のワード巡回処理が逆方向となる．そのため, 第 4.2 節で述べた回路アーキテクチャに復号処理を追加した場合にはセレクタ数が大幅に増加すると考えられる．そこで, 我々は偶数ラウンド目の F 関数 F_0 と F_1 の配置を入れ替えて復号処理を再構成した．すると, 図 7 (b) で示すようにラウンド関数のワード巡回方向が暗号化処理と同じ方向になる．従って, 回路アーキテクチャ上は図 6 のデータパスに大きな変更を加える必要はなく, 復号処理の偶数ラウンド目では F 関数 F_1 の処理を先に実行し, その後に F_0 の処理を実行すればよい．ただし, F 関数へのラウンド鍵入力の順序も入れ替わるため, 上記の F_1 の処理では中間鍵レジスタ L_{10} から, F_0 の処理では中間鍵レジスタ L_{30} からラウンド鍵を供給しなければならず, L_{00} を含めて 8 ビット 3 入力のセレクタを追加する必要がある．また, 復号処理終了時にはワード巡回処理の関係で先頭データが R_{30} ではなく R_{10} に格納されるため, 出力 data_out の選択を行なう 8 ビット 2 入力のセレクタが必要となる．

また, 復号処理の開始時には中間鍵データ L に文献 [6] で提案された $FinalSwap$ 関数 Φ を適用する必要がある．そこで，鍵スケジュール回路のセレクタ増加を防ぐため，暗号化処理の終了後，暗号文出力中に 8 サイクルかけて 64 ビット巡回シフトを行なった後に $FinalSwap$ 関数 Φ の出力を選択し, L_{ij} を元の中間鍵 L に戻すことになる．

以上より, データパス上は図 6 に 8 ビット 2 入力および 8 ビット 3 入力のセレクタを追加し, 8-bit shift + Σ^{-8} を $FinalSwap$ 関数 Φ に変更することで暗復号処理が可能となる．

6 ASIC ライブラリによる実装性能評価

本節では, 第 4 節で述べた CLEFIA 小型暗号化実装と第 5 節で述べた CLEFIA 小型暗復号実装の ASIC ライブラリによる実装性能評価を行なう．なお, ハードウェア設計および評価環境は以下の通りである．

記述言語	Verilog-VHDL
設計ライブラリ	0.13 μm CMOS 標準セルライブラリ
シミュレータ	VCS version 2006.06
論理合成ツール	Design Compiler version 2007.03-SP3

1 GE は 2 入力 NAND ゲートに相当し, 遅延時間は最悪条件での評価を行なっている．

表 1 に評価結果を示す．2 種類の実装に対して, 規模優先で回路を合成した．なお, cycle には入出力に必要なサイクル数を含めている．また, 比較として, CLEFIA の既存の結果としては最小となる文献 [6] の小型版実装

表 1: ASIC による実装性能評価

Algorithm	Mode	Cycle	Area (GE)	Freq. (MHz)	Throughput (Mbps)	Technology (μm)
CLEFIA	enc	176	2,893	67	49	0.13
	enc/dec	176	2,996	61	44	0.13
CLEFIA [6]	enc/dec	38	4,950	201	677	0.09
AES [3]	enc	177	3,100	152	110	0.13
AES [2]	enc/dec	1,032	3,400	80	10	0.35

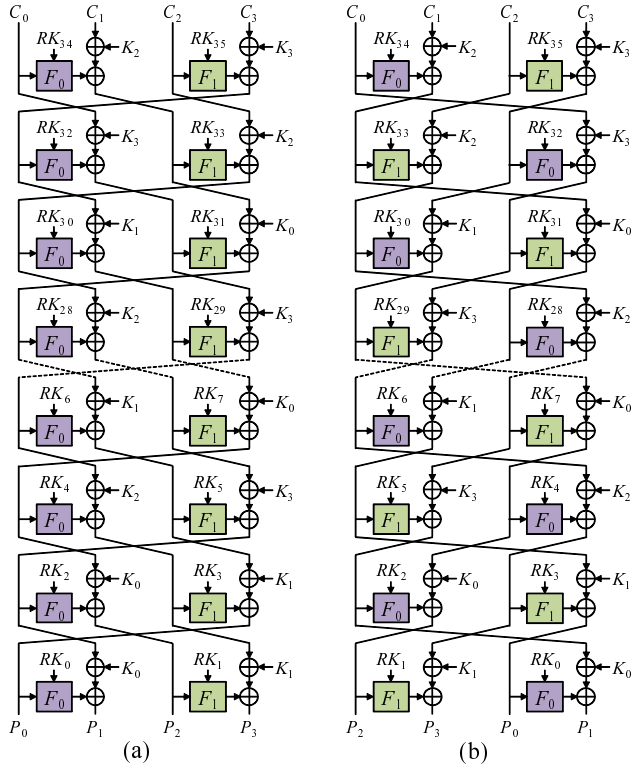


図 7: (a) CLEFIA 復号処理, (b) 偶数ラウンド目の F 関数を入れ替えた復号処理

の結果と, AES の Hämäläinen らの実装と Feldhofer らの実装の結果を掲載した．今回得られた結果は CLEFIA の既存の最小ゲート規模と比較して, 暗復号実装で約 39 % の小型化を実現している．また, 暗号化実装, 暗復号実装ともに AES の最小実装を下回る数値であり, CLEFIA のハードウェア実装における小型実装性能を示している．

7 まとめ

本稿では, 8 ビット・シリアルアーキテクチャを用いた CLEFIA の小型ハードウェア実装について報告を行なった．その結果, CLEFIA が高いハードウェア実装効率だけではなく, 小型実装性能を持ったブロック暗号であることを示した．

参考文献

[1] J. Daemen and V. Rijmen, *The Design of Rijndael: AES – The Advanced Encryption Standard (Information Security and Cryptography)*, Springer, 2002.

[2] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, “AES Implementation on a Grain of Sand”, *IEE Proceedings Information Security*, vol. 152, pp. 13–20, 2005.

[3] P. Hämäläinen, T. Alho, M. Hännikäinen, and T. Hämäläinen, “Design and Implementaion of Low-area and Low-power AES Encryption Hardware Core”, *DSD 2006*, pp. 577–583, IEEE Computer Society, 2006.

[4] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, “The 128-bit Blockcipher CLEFIA (Extended Abstract)”, *FSE 2007*, LNCS 4593, pp. 181–195, Springer-Verlag, 2007.

[5] 白井 太三, 渋谷 香士, 秋下 徹, 盛合 志帆, 岩田 哲, “128 ビットブロック暗号 CLEFIA”, IEICE Technical Report, ISEC 2007-1, 2007.

[6] 白井 太三, 渋谷 香士, 秋下 徹, 盛合 志帆, 岩田 哲, “128 ビットブロック暗号 CLEFIA のハードウェア実装評価”, IEICE Technical Report, ISEC 2007-49, 2007.

[7] 菅原 健, 本間 尚文, 青木 孝文, 佐藤 証, “128 ビットブロック暗号 CLEFIA の ASIC 実装”, *CSS 2007*, pp. 175–180, 2007.

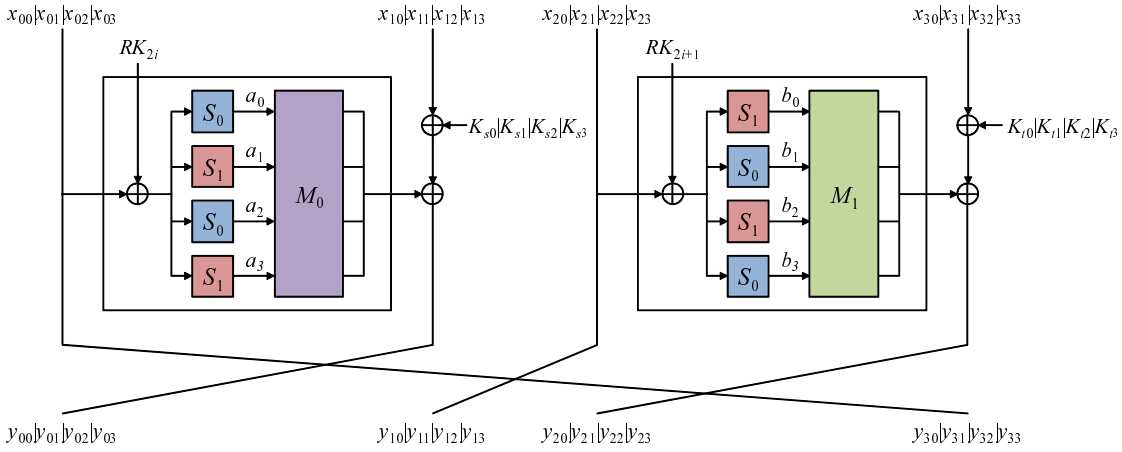
[8] T. Sugawara, N. Homma, T. Aoki, and A. Satoh, “ASIC Performance Comparison for the ISO Standard Block Ciphers”, *JWIS 2007*, pp. 485–498, 2007.

[9] Y. Zhang and P. Kitsos, *Security in RFID and Sensor Networks*, CRC Press, 2009.

[10] Y. Zheng, T. Matsumoto, and H. Imai, “On the Construction of Block Ciphers Provably Secure and not Relying on Any Unproved Hypotheses.” *Crypto’89*, LNCS 435, pp. 461–480, Springer-Verlag, 1989.

付録

CLEFIA のラウンド関数のデータ構造を図 8 (a) に示す．このとき, 図 6 で示された CLEFIA 小型暗号化実装のデータパスに基づいてラウンド関数 1 ラウンド分を実行する際に, データ処理部内のレジスタ R_{ij} $(0 \leq i, j \leq 3)$ に格納されるデータの詳細な流れを図 8 (b) に示す．



(a)

cycle	0	1	2	3	4
R_{00}	x_{00}	x_{01}	x_{02}	x_{03}	x_{20}
R_{01}	x_{01}	x_{02}	x_{03}	x_{20}	x_{21}
R_{02}	x_{02}	x_{03}	x_{20}	x_{21}	x_{22}
R_{03}	x_{03}	x_{20}	x_{21}	x_{22}	x_{23}
R_{10}	x_{10}	x_{21}	x_{22}	x_{23}	x_{30}
R_{11}	x_{11}	x_{22}	x_{23}	x_{30}	x_{31}
R_{12}	x_{12}	x_{23}	x_{30}	x_{31}	x_{32}
R_{13}	x_{13}	x_{30}	x_{31}	x_{32}	x_{33}
R_{20}	x_{20}	$x_{13} \oplus \{06\}a_0$	$x_{12} \oplus \{04\}a_0 \oplus \{06\}a_1$	$x_{11} \oplus \{02\}a_0 \oplus \{01\}a_1 \oplus \{06\}a_2 \oplus K_{s1}$	$x_{10} \oplus \{01\}a_0 \oplus \{02\}a_1 \oplus \{04\}a_2 \oplus \{06\}a_3 \oplus K_{s0}$ $(= y_{00})$
R_{21}	x_{21}	$x_{12} \oplus \{04\}a_0$	$x_{13} \oplus \{06\}a_0 \oplus \{04\}a_1$	$x_{10} \oplus \{01\}a_0 \oplus \{02\}a_1 \oplus \{04\}a_2 \oplus K_{s0}$	$x_{11} \oplus \{02\}a_0 \oplus \{01\}a_1 \oplus \{06\}a_2 \oplus \{04\}a_3 \oplus K_{s1}$ $(= y_{01})$
R_{22}	x_{22}	$x_{11} \oplus \{02\}a_0$	$x_{10} \oplus \{01\}a_0 \oplus \{02\}a_1 \oplus K_{s0}$	$x_{13} \oplus \{06\}a_0 \oplus \{04\}a_1 \oplus \{02\}a_2$	$x_{12} \oplus \{04\}a_0 \oplus \{06\}a_1 \oplus \{01\}a_2 \oplus \{02\}a_3 \oplus K_{s2}$ $(= y_{02})$
R_{23}	x_{23}	$x_{10} \oplus \{01\}a_0 \oplus K_{s0}$	$x_{11} \oplus \{02\}a_0 \oplus \{01\}a_1 \oplus K_{s1}$	$x_{12} \oplus \{04\}a_0 \oplus \{06\}a_1 \oplus \{01\}a_2 \oplus K_{s2}$	$x_{13} \oplus \{06\}a_0 \oplus \{04\}a_1 \oplus \{02\}a_2 \oplus \{01\}a_3 \oplus K_{s3}$ $(= y_{03})$
R_{30}	x_{30}	x_{31}	x_{32}	x_{33}	x_{00} $(= y_{30})$
R_{31}	x_{31}	x_{32}	x_{33}	x_{00}	x_{01} $(= y_{31})$
R_{32}	x_{32}	x_{33}	x_{00}	x_{01}	x_{02} $(= y_{32})$
R_{33}	x_{33}	x_{00}	x_{01}	x_{02}	x_{03} $(= y_{33})$

cycle	4	5	6	7	8
R_{00}	x_{20}	x_{21}	x_{22}	x_{23}	y_{00}
R_{01}	x_{21}	x_{22}	x_{23}	y_{00}	y_{01}
R_{02}	x_{22}	x_{23}	y_{00}	y_{01}	y_{02}
R_{03}	x_{23}	y_{00}	y_{01}	y_{02}	y_{03}
R_{10}	x_{30}	y_{01}	y_{02}	y_{03}	x_{20} $(= y_{10})$
R_{11}	x_{31}	y_{02}	y_{03}	x_{20}	x_{21} $(= y_{11})$
R_{12}	x_{32}	y_{03}	x_{20}	x_{21}	x_{22} $(= y_{12})$
R_{13}	x_{33}	x_{20}	x_{21}	x_{22}	x_{23} $(= y_{13})$
R_{20}	y_{00}	$x_{33} \oplus \{0A\}b_0$	$x_{32} \oplus \{02\}b_0 \oplus \{0A\}b_1$	$x_{31} \oplus \{08\}b_0 \oplus \{01\}b_1 \oplus \{0A\}b_2 \oplus K_{t1}$	$x_{30} \oplus \{01\}b_0 \oplus \{08\}b_1 \oplus \{02\}b_2 \oplus \{0A\}b_3 \oplus K_{t0}$ $(= y_{20})$
R_{21}	y_{01}	$x_{32} \oplus \{02\}b_0$	$x_{33} \oplus \{0A\}b_0 \oplus \{02\}b_1$	$x_{30} \oplus \{01\}b_0 \oplus \{08\}b_1 \oplus \{02\}b_2 \oplus K_{t0}$	$x_{31} \oplus \{08\}b_0 \oplus \{01\}b_1 \oplus \{0A\}b_2 \oplus \{02\}b_3 \oplus K_{t1}$ $(= y_{21})$
R_{22}	y_{02}	$x_{31} \oplus \{08\}b_0$	$x_{30} \oplus \{01\}b_0 \oplus \{08\}b_1 \oplus K_{t0}$	$x_{33} \oplus \{0A\}b_0 \oplus \{02\}b_1 \oplus \{08\}b_2$	$x_{32} \oplus \{02\}b_0 \oplus \{0A\}b_1 \oplus \{01\}b_2 \oplus \{08\}b_3 \oplus K_{t2}$ $(= y_{22})$
R_{23}	y_{03}	$x_{30} \oplus \{01\}b_0 \oplus K_{t0}$	$x_{31} \oplus \{08\}b_0 \oplus \{01\}b_1 \oplus K_{t1}$	$x_{32} \oplus \{02\}b_0 \oplus \{0A\}b_1 \oplus \{01\}b_2 \oplus K_{t2}$	$x_{33} \oplus \{0A\}b_0 \oplus \{02\}b_1 \oplus \{08\}b_2 \oplus \{01\}b_3 \oplus K_{t3}$ $(= y_{23})$
R_{30}	y_{30}	y_{31}	y_{32}	y_{33}	y_{30}
R_{31}	y_{31}	y_{32}	y_{33}	y_{30}	y_{31}
R_{32}	y_{32}	y_{33}	y_{30}	y_{31}	y_{32}
R_{33}	y_{33}	y_{30}	y_{31}	y_{32}	y_{33}

(b)

図 8: (a) CLEFIA ラウンド関数のデータ構造, (b) 小型暗号化実装におけるレジスタ格納データ