# 冗長表現基底による$\mathbb{F}_{(2^4)^2}$上の逆元計算を用いた AES の SubBytes 変換

根角 健太†　　野上 保之†　　森岡 恵理†

†岡山大学大学院自然科学研究科（岡山大学工学部通信ネットワーク工学科）
700–8530　岡山県岡山市北区津島中 3–1–1　工学部セキュア無線方式学研究室
{nekado,nogami,morioka}@trans.cne.okayama-u.ac.jp

**あらまし**　AES 暗号の SubBytes 変換では，線形解読法の対策として非線形処理である有限体$\mathbb{F}_{2^8}$上の逆元計算を組み込んでいる．この逆元計算を回路実装する場合，逆元計算の処理時間を短くし，かつ使用する論理ゲートを少数で済ませるためには，$\mathbb{F}_{2^8}$上の逆元計算の替わりに，それと同型な逐次拡大体上の逆元計算を適用することが望ましい．そこで本稿では，まず同型な逐次拡大体$\mathbb{F}_{(2^4)^2}$を構成するために必要な既約多項式および基底の中で最適なものを模索する．さらに，逆元計算を高速化するため，$\mathbb{F}_{2^4}$上の基底を冗長に表現する方法を提案する．その冗長表現された基底を用いることで，逆元計算内部における処理の並列化を促し，論理ゲート数を増加させることなく処理時間をより短くできることを示す．

# SubBytes Transform for AES Adopting Inversion in $\mathbb{F}_{(2^4)^2}$ with Redundantly Represented Basis

Kenta Nekado†　　Yasuyuki Nogami†　　Eri Morioka†

†Graduate School of Natural Science and Technology, Okayama University
(Communication Network Engineering, Okayama University)
3–1–1 Tsushima–naka, Kita–ward, Okayama–city, Okayama, JAPAN
{nekado,nogami,morioka}@trans.cne.okayama-u.ac.jp

**Abstract**　A lot of improvements and optimizations for the hardware implementation of SubBytes transform for AES, in detail *inversion* in $\mathbb{F}_{2^8}$, have been reported. Instead of the AES original $\mathbb{F}_{2^8}$, it is known that not only its isomorphic tower field $\mathbb{F}_{((2^2)^2)^2}$ but also $\mathbb{F}_{(2^4)^2}$ have more efficient inversions. Thus, this paper first considers efficient inversion in $\mathbb{F}_{(2^4)^2}$ with conventional techniques. Moreover, in order to reduce the critical path delay of inversion in $\mathbb{F}_{(2^4)^2}$, this paper proposes *Redundantly Represented Basis* (RRB).

## 1　序論

AES [1] が NIST より公表されて丸 10 年経った現在でも，そのハードウェア実装に関する研究が数多く報告されている．AES アルゴリズムでは，SubBytes，ShiftRows，MixedColumns および AddRoundKey の 4 つの処理を適当な回数繰り返し行うことで暗号化・復号処理が行われる．4 つの処理の 1 つ SubBytes 変換では，ソフトウェア実装する際，8–bit 入出力のルックアップテーブルによって実装される．しかし，ハードウェア実装する際は一般的に，上記のテーブル参照方式よりも 8 次拡大された 2 元体（ガロア体）$\mathbb{F}_{2^8}$上の演算回路による実装方式の方が処理の効率が良いとされている．この演算回路

による実装方式では，線形解読法 [2] 対策として取り入れられている $\mathbb{F}_{2^8}$ 上の逆元計算処理が最も多くの時間を要する．

当初の論理ゲートによる SubBytes 変換実装では，$\mathbb{F}_{2^8}$ 上の元を表現するために $t^8+t^4+t^3+t+1$ を法多項式とする多項式基底が用いられ，その多項式基底による $\mathbb{F}_{2^8}$ 上の逆元計算を行っていた [1]．しかしその後の研究で，上記の逆元計算の替りに，$\mathbb{F}_{2^8}$ と同型な逐次拡大体（合成体）上の逆元計算を採用した方が処理時間が短く，かつ論理ゲート総数の少ない SubBytes 変換回路を実装可能なことが森岡ら [3] を筆頭に複数報告されている [4, 5, 6]．$\mathbb{F}_{2^8}$ と同型な逐次拡大体には $\mathbb{F}_{((2^2)^2)^2}$ と $\mathbb{F}_{(2^4)^2}$ が存在し，$\mathbb{F}_{((2^2)^2)^2}$ による実装 [3, 4, 5] よりも $\mathbb{F}_{(2^4)^2}$ による実装 [6] の方が処理時間の短い逆元計算回路を実現できる傾向にある．

そこで，本稿では $\mathbb{F}_{(2^4)^2}$ に注目し，論理ゲートによる $\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路について考える．さらに，その逆元計算を高速化するために，冗長表現基底（Redundantly Represented Basis: RRB）を提案する．この RRB を用いれば，$\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路内部における $\mathbb{F}_{2^4}$ 上の演算処理において並列化を促し，その結果，$\mathbb{F}_{(2^4)^2}$ 上の逆元計算が $4T_{\mathrm{AND}}+7T_{\mathrm{XOR}}$ で計算できることを示す．ただし，$T_{\mathrm{AND}}$ および $T_{\mathrm{XOR}}$ は AND と XOR ゲートにおけるクリティカルパスの遅延時間を示している．

## 2 $\mathbb{F}_{(2^4)^2}$ の構成法および四則演算

$\mathbb{F}_{(2^4)^2}$ の構成は，まず拡大体 $\mathbb{F}_{2^4}$ を構成し，それを 2 次逐次拡大することで成しえる．本稿では，$\mathbb{F}_{2^4}$ の構成および 2 次逐次拡大に多項式基底および正規基底を用いる．$\mathbb{F}_{2^4}$ を構成するための多項式基底および正規基底は複数存在し，$\mathbb{F}_2$ 上の 4 次既約多項式の根により識別できる．一方で，$\mathbb{F}_{2^4}$ を 2 次逐次拡大するための多項式基底および正規基底も複数存在し，$\mathbb{F}_{2^4}$ 上の 2 次既約多項式の根により識別できる．$\mathbb{F}_{2^4}$ および $\mathbb{F}_{(2^4)^2}$ 上の四則演算は，それを構成する基底によって演算効率が変化する．そこで本節では，$\mathbb{F}_{2^4}$ の構成およびその 2 次逐次拡大するための多項式基底および正規基底のうち，演算効率の良い基底を紹介する．さらに，その基底を採用した際の四則演算の式および演算回路を考える．

### 2.1 4 次拡大体 $\mathbb{F}_{2^4}$

**既約多項式と基底**： $\mathbb{F}_2$ 上の 4 次既約多項式は次式に示す 3 種のみ存在する．

$$g_1(t)=t^4+t+1,\quad g_2(t)=t^4+t^3+1,$$
$$g_3(t)=t^4+t^3+t^2+t+1. \tag{1}$$

$g_1(t)$ の根を $\beta$ とするとき，$\{\beta,\beta^2,\beta^{2^2},\beta^{2^3}\}$ は正規基底を成さないが，$\{1,\beta,\beta^2,\beta^3\}$ は多項式基底を成す．この多項式基底は Jeon らによって $\mathbb{F}_{2^4}$ 上の四則演算，特に逆元計算が効率の良く行えることが示されている [6]．一方で，$g_3(t)$ の根を $\beta$ とするとき，$\{1,\beta,\beta^2,\beta^3\}$ は多項式基底，$\{\beta,\beta^2,\beta^{2^2},\beta^{2^3}\}$ は正規基底をそれぞれ成す．特に，正規基底の方は $\mathbb{F}_{2^4}$ 上の type–I optimal normal basis (ONB) [7] を成しており，効率良く四則演算が行える [8]．Type–I ONB $\{\beta,\beta^2,\beta^{2^2},\beta^{2^3}\}$ は $g_3(\beta)=0$ より，次式に示す性質を持つ．

$$g_3(\beta)+1=\beta^4+\beta^3+\beta^2+\beta=1,$$
$$\Leftrightarrow\ g_3(\beta)+1=\beta+\beta^2+\beta^{2^2}+\beta^{2^3}=1, \tag{2a}$$
$$\because\ (\beta+1)g_3(\beta)+1=\beta^5=1. \tag{2b}$$

$\mathbb{F}_{(2^4)^2}$ 上の逆元計算は $\mathbb{F}_{2^4}$ 上の四則演算により構成される．そこで以降では，type–I ONB で構成された $\mathbb{F}_{2^4}$ 上の四則演算の式および演算回路について紹介する．

**四則演算**： Type–I ONB で構成された拡大体上で効率良く乗算を行えるアルゴリズムとして，type–I cyclic vector multiplication algorithm (CVMA) が提案されている [8]．ここで，$D$ と $E$ を $\mathbb{F}_{2^4}$ 上の元とする．これらを正規基底で表現すると，次式のようになる．

$$D=d_0\beta+d_1\beta^2+d_2\beta^{2^2}+d_3\beta^{2^3}\,(d_j\in\mathbb{F}_2), \tag{3a}$$
$$E=e_0\beta+e_1\beta^2+e_2\beta^{2^2}+e_3\beta^{2^3}\ (e_j\in\mathbb{F}_2). \tag{3b}$$

Type–I CVMA により，積 $M=D\times E$ は次式のように求まる（図 1(a)）．

$$\begin{aligned}M&=m_0\beta+m_1\beta^2+m_2\beta^{2^2}+m_3\beta^{2^3}\,(m_j\in\mathbb{F}_2)\\&=\check{m}_0\beta+\check{m}_1\beta^2+\check{m}_2\beta^{2^2}+\check{m}_3\beta^{2^3}+\check{m}_4\\&=(\check{m}_0+\check{m}_4)\beta+(\check{m}_1+\check{m}_4)\beta^2\\&\qquad+(\check{m}_2+\check{m}_4)\beta^{2^2}+(\check{m}_3+\check{m}_4)\beta^{2^3},\end{aligned} \tag{4}$$

$$\check{m}_0 = a_{1,2} + d_0 e_0, \quad \check{m}_1 = a_{2,3} + d_1 e_1,$$
$$\check{m}_2 = a_{0,3} + d_2 e_2, \quad \check{m}_3 = a_{0,1} + d_3 e_3,$$
$$\check{m}_4 = a_{0,2} + a_{1,3}, \quad a_{j,k} = (d_j + d_k)(e_j + e_k).$$

式(4)より，積 $N_0 = D\times\beta$，$N_1 = D\times(\beta+\beta^2)$，$N_2 = D\times(\beta+\beta^{2^2})$，$N_3 = D\times(\beta+\beta^2+\beta^3)$ は次式のように求まる．

$$\begin{aligned} N_0 &= n_{0,0}\beta + n_{0,1}\beta^2 + n_{0,2}\beta^{2^2} + n_{0,3}\beta^{2^3} \\ &= d_2\beta + b_{0,2}\beta^2 + b_{2,3}\beta^{2^2} + b_{1,2}\beta^{2^3}, \quad (5a) \\ N_1 &= n_{1,0}\beta + n_{1,1}\beta^2 + n_{1,2}\beta^{2^2} + n_{1,3}\beta^{2^3} \\ &= d_3\beta + (b_{0,3}+d_2)\beta^2 \\ &\qquad + b_{1,2}\beta^{2^2} + (b_{0,3}+b_{1,2})\beta^{2^3}, \quad (5b) \\ N_2 &= n_{2,0}\beta + n_{2,1}\beta^2 + n_{2,2}\beta^{2^2} + n_{2,3}\beta^{2^3} \\ &= (b_{0,1}+d_2)\beta + b_{2,3}\beta^2 \\ &\qquad + (b_{2,3}+d_0)\beta^{2^2} + b_{0,1}\beta^{2^3}, \quad (5c) \\ N_3 &= n_{3,0}\beta + n_{3,1}\beta^2 + n_{3,2}\beta^{2^2} + n_{3,3}\beta^{2^3} \\ &= (b_{1,3}+d_0)\beta + d_2\beta^2 \\ &\qquad + (b_{1,2}+d_0)\beta^{2^2} + b_{1,3}\beta^{2^3}, \quad (5d) \\ b_{j,k} &= d_j + d_k, \qquad (n_{j,k}\in\mathbb{F}_2). \quad (5e) \end{aligned}$$

また，$D$ に乗じる元が $\beta$ の共役元であれば式(5a)を，$\beta+\beta^2$ の共役元なら式(5b)を，$\beta+\beta^{2^2}$ の共役元なら式(5c)を，$\beta+\beta^2+\beta^{2^2}$ の共役元なら式(5d)を適当に変数変換した式より求まる．例えば積 $D\times\beta^2$ は，$\beta^2$ が $\beta$ の共役元であるため，式(5a)を $n_{0,0}\to n_{0,1}$, $n_{0,1}\to n_{0,2}$, $n_{0,2}\to n_{0,3}$, $n_{0,3}\to n_{0,0}$, $d_0\to d_1$, $d_1\to d_2$, $d_2\to d_3$, $d_3\to d_0$ と変数変換した式より求まる．それぞれの共役元については表 1 に示す．

Type–I ONB は正規基底であるため，自乗 $S = D^2$ は次式のように求まる．

$$\begin{aligned} S &= s_0\beta + s_1\beta^2 + s_2\beta^{2^2} + s_3\beta^{2^3} \ (s_j\in\mathbb{F}_2) \\ &= d_1\beta + d_2\beta^2 + d_3\beta^{2^2} + d_0\beta^{2^3}. \quad (6) \end{aligned}$$

ここで，$D$ を改めて $\mathbb{F}_{2^4}$ 上の非零元とする．その逆元 $I = D^{-1}$ は次式のように求まる（図2(a)）．ただし，$\overline{d}$ $(d\in\mathbb{F}_2)$ は“NOT $d$”を示す．

$$\begin{aligned} I &= i_0\beta + i_1\beta^2 + i_2\beta^{2^2} + i_3\beta^{2^3} \ (i_j\in\mathbb{F}_2) \\ &= \check{i}_0\beta + \check{i}_1\beta^2 + \check{i}_2\beta^{2^2} + \check{i}_3\beta^{2^3} + \check{i}_4 \\ &= (\check{i}_0+\check{i}_4)\beta + (\check{i}_1+\check{i}_4)\beta^2 \\ &\qquad + (\check{i}_2+\check{s}_4)\beta^{2^2} + (\check{i}_3+\check{i}_4)\beta^{2^3}, \quad (7) \end{aligned}$$

$$\check{i}_0 = d_2+d_0d_1b_{1,3}, \qquad \check{i}_1 = d_3+d_1d_2b_{0,2},$$
$$\check{i}_2 = d_0+d_2d_3b_{1,3}, \qquad \check{i}_3 = d_1+d_0d_3b_{0,2},$$
$$\check{i}_4 = d_0d_2\overline{b_{1,3}} + d_1d_3\overline{b_{0,2}}, \quad b_{j,k} = d_j+d_k.$$

以上で示した正規基底による各演算に必要な処理時間をまとめると，表 2 の通りになる．

### 2.2 2 次逐次拡大体 $\mathbb{F}_{(2^4)^2}$

**既約多項式と基底**： 2 次多項式を $h(t) = t^2 + Qt + R$ $(Q, R\in\mathbb{F}_{2^4})$ とし，その根を $\gamma$ とする．$h(t)$ が既約となり，かつ $\{1,\gamma\}$ および $\{\gamma,\gamma^{16}\}$ が基底（1 次独立）となる $Q$ と $R$ の組は限られる．しかし，本節では $Q$ と $R$ に具体的な $\mathbb{F}_{2^4}$ 上の非零元を当てはめず，多項式基底 $\{1,\gamma\}$ および正規基底 $\{\gamma,\gamma^{16}\}$ を採用した際の逆元計算の式および演算回路を一般化して考える．

**逆元計算**： $\mathbb{F}_{(2^4)^2}$ 上の非零元 $C$ を次式のように多項式基底 $\{1,\gamma\}$ で表すとき，

$$C = D + E\gamma \ (D, E\in\mathbb{F}_{2^4}), \quad (8)$$

伊東–辻井アルゴリズム (Itoh–Tujii inversion Algorithm: ITA) [9] より，その逆元 $X = C^{-1} = (CC^{16})^{-1}C^{16}$ は次式のように求まる（図 3(a)）．

$$\begin{aligned} X &= Y + Z\gamma \ (Y, Z\in\mathbb{F}_{2^4}) \\ &= \{D^2+DEQ+E^2R\}^{-1}\{(D+EQ)+E\gamma\}. \quad (9) \end{aligned}$$

一方で，$\mathbb{F}_{(2^4)^2}$ 上の非零元 $C$ を次式のように正規基底 $\{\gamma,\gamma^{16}\}$ で表すとき，

$$C = D\gamma + E\gamma^{16} \ (D, E\in\mathbb{F}_{2^4}), \quad (10)$$

ITA より，その逆元 $X = C^{-1} = (CC^{16})^{-1}C^{16}$ は次式のように求まる（図 4(a)）．

$$\begin{aligned} X &= Y\gamma + Z\gamma^{16} \ (Y, Z\in\mathbb{F}_{2^4}) \\ &= \{DEQ+(D+E)^2R\}^{-1}\{E\gamma+D\gamma^{16}\}. \quad (11) \end{aligned}$$

表 2, 図 3(a), 4(a) より，上記 2 つの逆元計算回路はともに，$R$ 倍算回路を通るパスはクリティカルパスにならず，$Q$ 倍算回路を通るパスがクリティカルパスになる．よって，計算時間の観点から言えば，$R$ は $\mathbb{F}_{2^4}$ 上の非零元ならばどれでもよく，$Q$ はその倍算に必要な処理時間が実質 0 となる 1 が望ましい．$Q = 1$ の場合，$\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路におけるクリティカルパスの遅延時間は $\mathbf{4T}_{\mathrm{AND}} + \mathbf{9T}_{\mathrm{XOR}}$ となる．このとき，回路内部における論理ゲートの総数は表 3 の通

りになる．

## 3 冗長表現基底

式(4)に着目すると，type–I ONB $\{\beta, \beta^2, \beta^{2^2}, \beta^{2^3}\}$ ではなく，それに $\{1\}$ を加えた集合 $\{\beta, \beta^2, \beta^{2^2}, \beta^{2^3}, 1\}$ で $\mathbb{F}_{2^4}$ 上の元 $M$ を表現すれば，積 $M = D \times E$ を求める際に $\check{m}_0 + \check{m}_4$，$\check{m}_1 + \check{m}_4$，$\check{m}_2 + \check{m}_4$，$\check{m}_3 + \check{m}_4$ を計算せずに済む．つまり，$\mathbb{F}_{2^4}$ 上の元を type–I ONB に替えてこの集合で表現すれば，$\mathbb{F}_{2^4}$ 上の乗算に掛かる時間を $1T_{\mathrm{XOR}}$ 削減できる．集合 $\{\beta, \beta^2, \beta^{2^2}, \beta^{2^3}, 1\}$ は多項式基底や正規基底のように $\mathbb{F}_{2^4}$ 上の元を一意的に表現できない．しかし，上述のように $\mathbb{F}_{2^4}$ 上の演算に必要な処理時間を削減できる．そこで，本稿ではこの集合を冗長表現基底（Redundantly Represented Basis: RRB）と呼び，以降，この基底を採用した際の $\mathbb{F}_{2^4}$ 上の四則演算の式および演算回路について考える．

### 3.1 $\mathbb{F}_{2^4}$ 上における四則演算の高速化

$D$ を引き続き $\mathbb{F}_{2^4}$ 上の元とし，今回は RRB で表現すると，次式のようになる．

$$D = \check{d}_0\beta + \check{d}_1\beta^2 + \check{d}_2\beta^{2^2} + \check{d}_3\beta^{2^3} + \check{d}_4 \ (\check{d}_j \in \mathbb{F}_2), \quad (12)$$

式(2a)より，RRB で表現された $D$ は次式のように正規基底表現へ簡単に変換できる．

$$D = d_0\beta + d_1\beta^2 + d_2\beta^{2^2} + d_3\beta^{2^3}, \quad (13a)$$

$$d_0 = \check{d}_0 + \check{d}_4, \quad d_1 = \check{d}_1 + \check{d}_4,$$
$$d_2 = \check{d}_2 + \check{d}_4, \quad d_3 = \check{d}_3 + \check{d}_4. \quad (13b)$$

ここで，$E$ を $\mathbb{F}_{2^4}$ 上の元とし，$D$ と同様に RRB で表現すると，次式のようになる．

$$E = \check{e}_0\beta + \check{e}_1\beta^2 + \check{e}_2\beta^{2^2} + \check{e}_3\beta^{2^3} + \check{d}_4 \ (\check{e}_j \in \mathbb{F}_2), \quad (14)$$

式(4)の $D, E, M$ を式(13)に示すような変数変換の逆変換を行うことで，RRB を採用した $\mathbb{F}_{2^4}$ 上の乗算式が次式のように求まる（図 1(b)）．

$$M = \check{m}_0\beta + \check{m}_1\beta^2 + \check{m}_2\beta^{2^2} + \check{m}_3\beta^{2^3} + \check{m}_4, \quad (15)$$

$$\check{m}_0 = a_{\check{1},2} + a_{\check{0},4}, \quad \check{m}_1 = a_{\check{2},3} + a_{\check{1},4},$$
$$\check{m}_2 = a_{\check{0},3} + a_{\check{2},4}, \quad \check{m}_3 = a_{\check{0},1} + a_{\check{3},4},$$
$$\check{m}_4 = a_{\check{0},2} + a_{\check{1},3}, \quad \check{a}_{j,k} = (\check{d}_j + \check{d}_k)(\check{e}_j + \check{e}_k)$$

同様に，式(5), (6), (7) の $D, E, N_0, N_1, N_2, N_3, S, I$ を変数変換することで，RRB を採用した $\mathbb{F}_{2^4}$ 上における他の演算式が次式のように求まる（図 2(b)）．

$$N_0 = \check{n}_{0,0}\beta + \check{n}_{0,1}\beta^2 + \check{n}_{0,2}\beta^{2^2} + \check{n}_{0,3}\beta^{2^3} + \check{n}_4$$
$$= \check{d}_4\beta + \check{d}_0\beta^2 + \check{d}_3\beta^{2^2} + \check{d}_1\beta^{2^3} + \check{d}_2, \quad (16a)$$

$$N_1 = \check{n}_{1,0}\beta + \check{n}_{1,1}\beta^2 + \check{n}_{1,2}\beta^{2^2} + \check{n}_{1,3}\beta^{2^3} + \check{n}_4$$
$$= \check{b}_{2,4}\beta + \check{b}_{0,4}\beta^2 + \check{b}_{1,3}\beta^{2^2} + \check{b}_{0,1}\beta^{2^3} + \check{b}_{2,3}, \quad (16b)$$

$$N_2 = \check{n}_{2,0}\beta + \check{n}_{2,1}\beta^2 + \check{n}_{2,2}\beta^{2^2} + \check{n}_{2,3}\beta^{2^3} + \check{n}_4$$
$$= \check{b}_{1,4}\beta + \check{b}_{0,3}\beta^2 + \check{b}_{3,4}\beta^{2^2} + \check{b}_{1,2}\beta^{2^3} + \check{b}_{0,2}, \quad (16c)$$

$$N_3 = \check{n}_{3,0}\beta + \check{n}_{3,1}\beta^2 + \check{n}_{3,2}\beta^{2^2} + \check{n}_{3,3}\beta^{2^3} + \check{n}_4$$
$$= \check{b}_{0,3}\beta + \check{b}_{1,2}\beta^2 + \check{b}_{0,2}\beta^{2^2} + \check{b}_{3,4}\beta^{2^3} + \check{b}_{1,4}, \quad (16d)$$

$$(\check{n}_{j,k} \in \mathbb{F}_2) \quad (16e)$$

$$S = \check{d}_1\beta + \check{d}_2\beta^2 + \check{d}_3\beta^{2^2} + \check{d}_0\beta^{2^3} + \check{d}_4$$
$$= \check{s}_0\beta + \check{s}_1\beta^2 + \check{s}_2\beta^{2^2} + \check{s}_3\beta^{2^3} + \check{s}_4, \quad (17)$$

$$I = \check{i}_0\beta + \check{i}_1\beta^2 + \check{i}_2\beta^{2^2} + \check{i}_3\beta^{2^3} + \check{i}_4, \quad (18)$$

$$\check{i}_0 = \check{b}_{2,4} + \check{b}_{0,4}\check{b}_{1,4}\check{b}_{1,3}, \quad \check{i}_1 = \check{b}_{3,4} + \check{b}_{1,4}\check{b}_{2,4}\check{b}_{0,2},$$
$$\check{i}_2 = \check{b}_{0,4} + \check{b}_{2,4}\check{b}_{3,4}\check{b}_{1,3}, \quad \check{i}_3 = \check{b}_{1,4} + \check{b}_{3,4}\check{b}_{0,4}\check{b}_{0,2},$$
$$\check{i}_4 = \check{b}_{0,4}\check{b}_{2,4}\overline{\check{b}_{1,3}} + \check{b}_{1,4}\check{b}_{3,4}\overline{\check{b}_{0,2}}, \quad \check{b}_{j,k} = (\check{d}_j + \check{d}_k).$$

以上で示した RRB による各演算に必要な処理時間をまとめると，表 2 の通りになる．この表が示すように，正規基底に比べて RRB は乗算と各種倍算の処理時間を $1T_{\mathrm{XOR}}$ 削減できる．

### 3.2 計算効率の良い $\mathbb{F}_{(2^4)^2}$ 上の逆元計算

$\mathbb{F}_{2^4}$ 上の RRB による四則演算を $\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路に適用すると，図 3(b)，4(b) の通りになる．これらの逆元計算回路の入力になっている $\mathbb{F}_{2^4}$ 上の元 $D, E$ は，RRB ではなく正規基底で表現されている．さらに言えば，式(2b)より次式が満たされるため，入力 $D, E$ は多項式基底で表現されていても構わない．

$$\{\beta, \beta^2, \beta^{2^2}, \beta^{2^3}, 1\} = \{1, \beta, \beta^2, \beta^3, \beta^4\}. \quad (19)$$

一方で，逆元回路の出力になっている $\mathbb{F}_{2^4}$ 上の元 $Y, Z$ は，RRB で表現されているため，5×2–bit 出力となっている．この問題については，[10] に示す手法を用いれば解決できる．

逆元計算に必要な処理時間の観点から言えば，表 2 より $Q$ は $1, \beta, \beta^2, \beta^{2^2}, \beta^{2^3}$ のうちのどれかが望ましい．一方で，$R$ は $\mathbb{F}_{2^4}$ 上の非零元であ

表 1: $\mathbb{F}_{2^4}$ 上の共役元

| | $\beta$ | $\beta+\beta^2$ | $\beta+\beta^{2^2}$ | $\beta+\beta^2+\beta^{2^2}$ |
|---|---|---|---|---|
| 共役元 | $\beta^2$, $\beta^{2^2}$, $\beta^{2^3}$ | $\beta^2+\beta^{2^2}$, $\beta^{2^2}+\beta^{2^3}$, $\beta^{2^3}+\beta$ | $\beta^2+\beta^{2^3}$ | $\beta^2+\beta^{2^2}+\beta^{2^3}$, $\beta^{2^2}+\beta^{2^3}+\beta$, $\beta^{2^3}+\beta+\beta^{2^2}$ |

表 2: $\mathbb{F}_{2^4}$ 上の各演算におけるクリティカルパスの遅延時間

| 採用基底 | 乗算 | 自乗算 | 逆元計算 | $\beta$ 倍算 | $(\beta+\beta^2)$ 倍算 | $(\beta+\beta^{2^2})$ 倍算 | $(\beta+\beta^2+\beta^{2^2})$ 倍算 |
|---|---|---|---|---|---|---|---|
| type–I ONB | $(1,3)$ | $(0,0)$ | $(2,2)^{\dagger\star}$ | $(0,1)$ | $(0,2)$ | $(0,2)$ | $(0,2)$ |
| RRB | $(1,2)$ | $(0,0)$ | $(2,2)^{\dagger}$ | $(0,0)$ | $(0,1)$ | $(0,1)$ | $(0,1)$ |

† $(j,k)$ は $jT_{\mathrm{AND}}+kT_{\mathrm{XOR}}$ を示す. ‡ XNOR 演算のクリティカルパスにおける遅延時間を $T_{\mathrm{XOR}}$ としている.
⋆ $T_{\mathrm{AND}} \geq T_{\mathrm{XOR}}$ の場合を示している. $T_{\mathrm{AND}} \leq T_{\mathrm{XOR}}$ の場合は $(1,3)$ となる.

表 3: $\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路内部における論理ゲートの総数

| 採用基底 | | $\mathbb{F}_{2^4}$ 構成用 | |
|---|---|---|---|
| | | 正規基底 | RRB |
| $\mathbb{F}_{2^4}$ の 2 次 | 多項式基底 | $(42,86^{\ddagger},2)$ | $(42,82^{\ddagger},2)$ |
| 逐次拡大用 | 正規基底 | $(42,78^{\ddagger},2)$ | $(42,74^{\ddagger},2)$ |

† $(j,k,l)$ は $j$AND + $k$XOR + $l$XNOR を示す.
‡ さらに, $Q$ 倍算内部の論理ゲート数が加わる.

ればどの元でも構わない. この $Q$ と $R$ の組のうち, $\{1,\gamma\}$ および $\{\gamma,\gamma^{16}\}$ が $\mathbb{F}_{(2^4)^2}$ 上で 1 次独立となる組が実際に利用可能な $Q$ と $R$ になる. $Q=1,\beta,\beta^2,\beta^{2^2},\beta^{2^3}$ の場合, $\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路におけるクリティカルパスの遅延時間は, 正規基底を採用した際より $2T_{\mathrm{XOR}}$ 削減され, $\mathbf{4T_{AND}+7T_{XOR}}$ となる. このとき, $\mathbb{F}_{2^4}$ 上の逆元計算回路内部における論理ゲートの総数を表 3 に示すように 4XOR 削減できる.

## 4 結論

本稿では, AES の SubBytes 変換回路で利用できる逐次拡大体 $\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路について考え, その回路の高速化を目的として, RRB による $\mathbb{F}_{2^4}$ 上の四則演算について提案を行った. この RRB を採用することで, 回路規模を増大させることなく, $\mathbb{F}_{(2^4)^2}$ 上の逆元計算の処理時間を $2T_{\mathrm{XOR}}$ 削減可能なことを示した.

## 参考文献


[1] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," FIPS publication 197, "http://csrc.nist.gov/encryption/aes/", 2001.

[2] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," EUROCRYPTO'93, LNCS 765, pp. 386–397, 1994.

[3] S. Morioka and A. Satoh, "An Optimized S–box Circuit Architecture for Low Power AES Design," Workshop on Cryptographic Hardware and Embedded Systems (CHES2002), LNCS 2523, pp. 172–186, Springer–Verlag, 2003.

[4] D. Canright, "A Very Compact S-Box for AES," Workshop on Cryptographic Hardware and Embedded Systems (CHES2005), LNCS 3659, pp. 441–455, Springer–Verlag, 2005.

[5] Y. Nogami, K. Nekado, T. Toyota, N. Hongo, and Y. Morikawa, "Mixed Bases for Efficient Inversion in $\mathbb{F}_{((2^2)^2)^2}$ and Conversion Matrices of SubBytes of AES," Workshop on Cryptographic Hardware and Embedded Systems (CHES2010), LNCS 6225, pp. 234–247, Springer–Verlag, 2010.

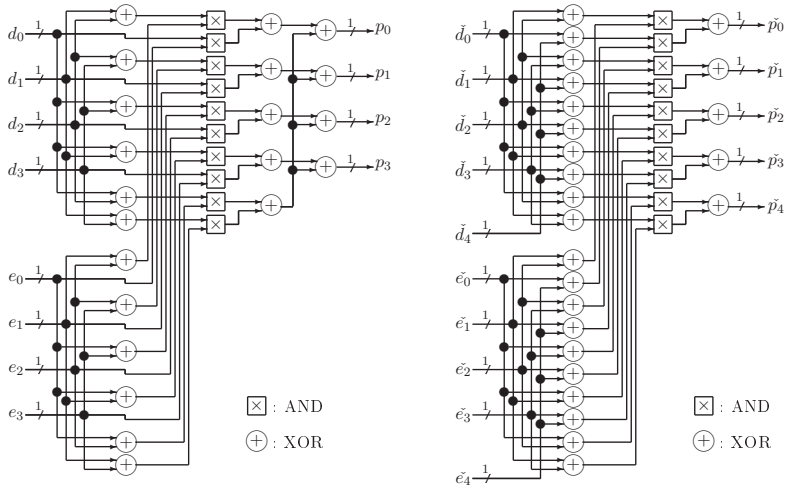
[6] Y. Jeon, Y. Kim, and D. Lee, "A Compact Memory-free Architecture for the AES Algorithm Using Resource Sharing Methods," Journal of Circuits, Systems, and Computers, Vol. 19, No. 5, pp. 1109–1130, 2010.

[7] R. Mullin, I. Onyszchuk, S. Vanstone, and R.Wilson, "Optimal Normal Bases in GF($p^n$)," Discrete Applied Math., Vol. 22, pp. 149–161, 1988.

[8] Y. Nogami, A. Saito, and Y. Morikawa, "Finite Extension Field with Modulus of All-One Polynomial and Representation of Its Elements for Fast Arithmetic Operations," IEICE Trans., Vol. E86–A, No. 9, pp. 2376–2387, 2003.

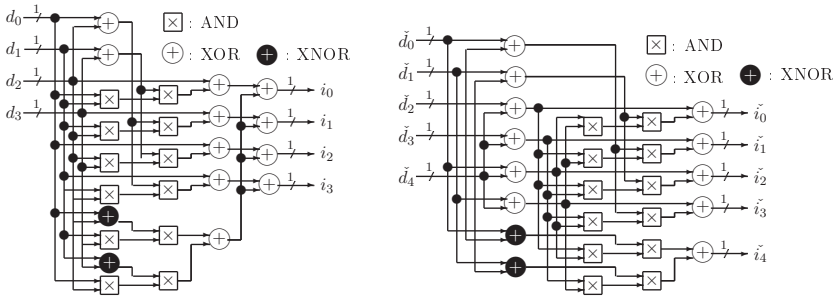
[9] T. Itoh and S. Tsujii, "A Fast Algorithm for Computing Multiplicative Inverse in GF($2^m$) using Normal Basis," Inf. Comput., Vol. 78, pp. 171–177, 1988.

[10] 根角健太, 野上保之, 森岡恵理, "$\mathbb{F}_{(2^4)^2}$ 上の複雑混合基底による基底変換を用いた AES の SubBytes 変換," コンピュータセキュリティシンポジウム 2011 (CSS2011), 2011.



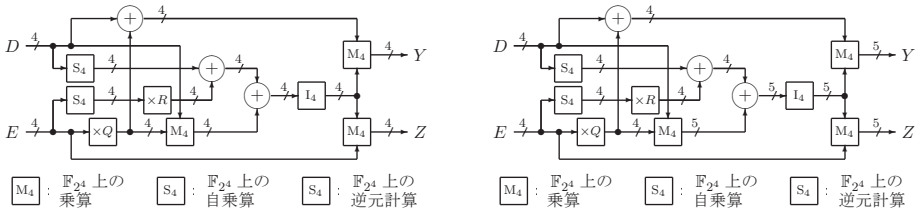

(a) 正規基底採用時（式 (4) ）　　(b) RRB 採用時（式 (15) ）

図 1: $\mathbb{F}_{2^4}$ 上の乗算回路



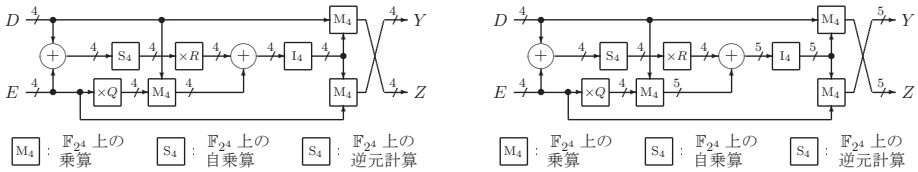

(a) 正規基底採用時（式 (7) ）　　(b) RRB 採用時（式 (18) ）

図 2: $\mathbb{F}_{2^4}$ 上の逆元計算回路




(a) $\mathbb{F}_{2^4}$ 上の正規基底採用時　　(b) $\mathbb{F}_{2^4}$ 上の RRB 採用時

図 3: 多項式底を採用した $\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路（式 (9) ）




(a) $\mathbb{F}_{2^4}$ 上の正規基底採用時　　(b) $\mathbb{F}_{2^4}$ 上の RRB 採用時

図 4: 正規基底を採用した $\mathbb{F}_{(2^4)^2}$ 上の逆元計算回路（式 (11) ）