計算機システムに適用可能なBCH(or RS)符号復号器の一構成法

A Construction Method for Decoders of BCH (or RS) Codes for Computer Memory Systems

岡野博一　守川和夫　高浪五男
Hirokazu OKANO　Kazuo MORIKAWA　Itsuo TAKANAMI
徳山高専　山口大学
Tokuyama Technical College　Yamaguchi University

本論文において、誤り訂正と誤り検出を同時に行うBCH(or Reed Solomon)符号の復号法を示し、具体例として、1ビット誤り訂正/2，3，4ビット誤り検出BCH符号、1，2ビット誤り訂正/3ビット誤り検出BCH符号、さらに、1ディジット誤り訂正/2ディジット誤り検出RS符号、およびその倍長符号、1，2ディジット誤り訂正/3ディジット誤り検出RS符号の復号器を提案する。これらの復号器は高速復号を行うので、計算機メモリシステムへの適用が可能である。

1. まえがき

情報処理システムの高信頼度化の一手法として誤り訂正符号が実用されている。特にBCH符号[1][2]は高い誤り訂正能力を持つので、衛星通信、Audio, Video 等のディジタルシステムに用いられている。

そこで、誤りを訂正するBCH符号の復号器が種々提案されている[5][6]。しかし、誤り訂正符号を誤り訂正のみに用いると誤訂正を除くことができないので実用上問題が生じる。従ってある誤りビット数までは誤りを訂正し、それ以上の誤りは検出のみとすることが実用上有効であることが多い。このような符号の存在は良く知られているが、その具体的な復号法は明確に示されていない。

このような符号の復号器のとしては、1ビット誤り訂正/2ビット誤り検出(SEC/DED)符号[3]、1ディジット誤り訂正/2ディジット誤り検出(SbEc/DbEd)符号[4]が計算機の主記憶に用いられている。しかし、その他の1，2ビット誤り訂正/3ビット誤り検出(DEC/TED)符号および1ビット誤り訂正/2，3，4ビット誤り検出(SEC/QdED)符号のように、誤り訂正と誤り検出を同時に行うことのできるBCH符号の効率的な復号法は未だ示されていない。

本論文において、誤り訂正および検出を同時に行うBCH符号、Reed-Solomon(以下、RSと称す)符号の効率的な復号法を提案する。この復号法はシンドロームの関係よりなる判定式を用いて、誤りを訂正するか、検出するかを判定することを特徴としている。

なお、復号法の有効性について、判定式の証明は紙面の都合で　　示していないが[6]コンピュータ、シミュレーション結果を示し動作を確認している。また、具体的に、1ビット誤り訂正/2，3，4ビット誤り検出(SEC/QdED)符号、1，2ビット誤り訂正/3ビット誤り検出(DEC/TED)BCH符号、および1ディジット誤り訂正/2ディジット誤り検出(SbEC/DbED)RS符号とその倍長符号、1，2ディジット誤り訂正/3ディジット誤り検出(DbED/TbEC)RS符号について、判定式を証明するとともに、復号器の構成を示す。

これらの復号器について、ハードウェア量、復号遅延を評価し、LSI化が可能であり、

計算機メモリシステムへの実用が期待されることを示す。

2. BCH符号の復号法

まず、誤り訂正のみを行う場合のBCH符号の復号法について概説する[1][2]。BCH符号の生成多項式は $GF(q^m)$ 上の原始元を α、最小距離を d、任意の整数を r とすると、$\alpha^r, \alpha^{r+1}, \cdots, \alpha^{r+d-2}$ を根とする多項式である。

復号するには、まず、受信符号よりシンドロームを求めるが、誤りのみ依存するので次式となる。

$$S_j = \sum_{i=1}^{t} Y_i X_i^j \quad , \quad r \le j \le r+d-2 \qquad (1)$$

ここで、t は誤りの数、Y_i は誤りの大きさ、X_i は誤り位置数である。2元BCH符号のときは $q=2$ で Y_i は0か1である。そして、通常 $r=1$ として、$d=2t+1$、$S_{2n}=S_n^2$ の性質から、$S_1, S_3, \cdots, S_{2t-1}$ を復号に用いる。

次に誤り位置多項式の係数 σ_i とシンドローム S_j との間に次式が成立する。

$$S_j\sigma_t + S_{j+1}\sigma_{t-1} + \cdots + S_{j+t-1}\sigma_1 + S_{j+t} = 0 \qquad (2)$$

ここで、$r \le j \le r+t-1$

したがって、(2)式より σ_i $(1 \le i \le t)$ を求めて誤り位置多項式

$$x^t + \sigma_1 x^{t-1} + \cdots + \sigma_t = 0 \qquad (3)$$

を解き、誤り位置数を求めることによって復号する。

また Peterson の復号法[1]において、誤りビット数の判定のために次式を用いる。

$$M_\nu = \begin{vmatrix} S_r & S_{r+1} & \cdots & S_{r+\nu-1} \\ S_{r+1} & S_{r+2} & \cdots & S_{r+\nu} \\ \vdots & \vdots & & \vdots \\ S_{r+\nu-1} & S_{r+\nu} & \cdots & S_{r+2\nu-2} \end{vmatrix} \qquad (4)$$

ν 重誤りのときは $M_\nu \neq 0$ であり、$\nu-1$ 重以下の誤りのときは $M_\nu = 0$ となる。

さらに、2元の場合は次式を用いる。

$$M_\nu = \begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ S_2 & S_1 & 1 & \cdots & 0 \\ S_4 & S_3 & S_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ S_{2\nu-2} & S_{2\nu-3} & S_{2\nu-4} & \cdots & S_{\nu-1} \end{vmatrix} \qquad (5)$$

ν または $\nu-1$ 重誤りのとき $M_\nu \neq 0$、$\nu-2$ 重以下の誤りのとき $M_\nu = 0$ である。

なお、RS符号は $r=1$ ($r=0$ でも良い) $q=2^b$、$m=1$ としたBCH符号の特別の場合である。

3. 誤り訂正および検出を同時に行うBCH符号およびRS符号の復号法

3.1 判定式を用いた復号法

一般的に、t 重誤り訂正/$t+1, \cdots, t+k$ 誤り検出BCH(またはRS)符号の最小距離は図1のようになり次式で与えられる。

$$d = 2t + k + 1 \qquad (6)$$

したがって、式(1),(6)より $S_j (r \le j \le r+2t+k-1)$ を復号に用いることになる。そして、訂正のみを行うときは、$d=2t+1$ であるから、$S_{j'}$ $(r \le j' \le r+2t-1)$ を用いて誤りを訂正する。図2に t 重誤り訂正/$t+1, \cdots, t+k$ 誤り検出符号復号器の一般的構成を示す。JUDは誤りを訂正するか検出のみとするかを判定する回路であり、S_j 間の関係よりなる判定式が零(論理式のときは真)のときは誤りを訂正し、そうでないときは誤り検出のみとする。

なお、RS符号の場合は $r=0$ とする。2元BCH符号の場合は $S_{2n}=S_n^2$ が成立するから、シンドロームの添字の最大値 $(r+d-2)$ が偶数となるように r を選ぶ。つまり、d が奇数のときは $r=1$、偶数のときは $r=0$ とする。

さて、判定式の算出法および証明は紙面の都合で省略し[8]、ここではいくつかの符号の判定式のみを示すこととする。

まず、(6)式を用いて算出されるBCH符号について、表1に t 重誤り訂正/$t+1$, $t+2$ 誤り検出符号の判定式、表2に t 重誤り訂正/$t+1$ 誤り検出符号の判定式を示す。また、RS符号の判

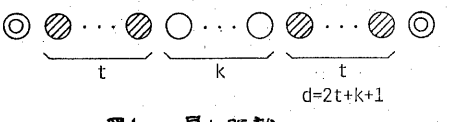


図1 最小距離

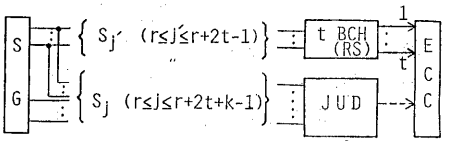


図2 復号器の一般的構成

表1 t重誤り訂正/t+1, t+2 誤り検出
BCH 符号の 判定式

符 号	d	判定式 B2Mν
DED (0/1,2)	3	$B2M_2=S_1$
SEC/TED (1,2/3)	5	$B2M_3=S_1^3+S_3$
DEC/QdED (1,2/3,4)	7	$B2M_4=S_3(S_1^3+S_3)+S_1(S_1^5+S_5)$
TEC/QnED (1,2,3/4,5)	9	$B2M_5=(S_1^3+S_3)(S_1^7+S_7)+S_5(S_1^5+S_5)+S_1S_3(S_3^2+S_1S_5)$

表3 t重誤り訂正/ t+1 誤り検出
RS 符号の判定式

符 号	d	判定式 R1Mν
SED (0/1)	2	$R1M_1=S_0$
SEC/DED (1/2)	4	$R1M_2=S_0S_2+S_1^2$
DEC/TED (1,2/3)	6	$R1M_3=S_0S_2S_4+S_2^3+S_1^2S_4+S_0S_3^2$
TEC/QdED (1,2,3/4)	8	$R1M_4=S_6\cdot R1M_3+S_3^2(S_2S_4+S_3^2)+S_4^2(S_0S_4+S_2^2)+S_5^2(S_0S_2+S_1^2)$

定式は(4)式を用いて算出する。表3にt重誤り訂正/t+1誤り検出符号、表4にt重誤り訂正/t+1, t+2 誤り検出符号の判定式を示す。

さらに、単一誤り訂正/2, 3, 4ディジット検出(SEC/QdED) RS 符号の判定式は次式となる。d=6である。

$$R3M_4=(S_0S_1+S_1^2=0)\ AND\ (S_2^3+S_0S_3^2=0)$$

表2 t重誤り訂正/ t+1 誤り検出
BCH 符号の判定式

符 号	d	判定式 B1Mν
SEC/DED (1/2)	4	$B1M_2=(1+S_0)\cdot B2M_2=(1+S_0)S_1$
DEC/TED (1,2/3)	6	$B1M_3=S_0\cdot B2M_3=S_0(S_1^3+S_3)$
TEC/QdED (1,2,3/4)	8	$B1M_4=(1+S_0)\cdot B2M_4=(1+S_0)\{S_3(S_1^3+S_3)+S_1(S_1^5+S_5)\}$

表4 t重誤り訂正/ t+1, t+2 誤り検出
RS 符号の判定式

符 号	d	判定式 R2Mν
SEC/TED (1/2,3)	5	$R2M_3=(S_0S_2+S_1^2=0)\ AND\ (S_0S_3^2+S_2^3=0)$
DEC/QdED (1,2/3,4)	7	$R2M_4=(S_0S_2S_4+S_2^3+S_1^2S_4+S_0S_3^2=0)\ AND\ \{(S_3^2(S_2S_4+S_3^2)+S_4^2(S_0S_4+S_2^2)+S_5^2(S_0S_2+S_1^2)=0\}$

$$AND\ \{S_3^2(S_2S_4+S_3^2)+S_4^2(S_0S_4+S_2^2)+S_5^2(S_0S_2+S_1^2)=0\} \quad (7)$$

なお、次に述べるBCH 符号の判定式は式(4),(5)からは算出されない。著者独自の方法による(8)。ここでは判定式のみを示す。

(i) 単一誤り訂正/2, 3, 4ビット誤り検出
(SEC/QdED) BCH 符号

$$BZ1=(S_1^3+S_3=0)\ AND\ (S_0S_1^3+S_3=0)$$

ただし、d=6 (8)

(ii) 単一誤り訂正/2, 3, 4, 5ビット誤り検出
(SEC/QnED) BCH 符号

$$BZ2=(S_1^3+S_3=0)\ AND\ (S_1^5+S_5=0)$$

ただし、d=7 (9)

(iii) 2重誤り訂正/3, 4, 5ビット誤り検出
(DEC/QnED) BCH 符号

$$BZ3=\{S_3(S_1^3+S_3)+S_1(S_1^5+S_5)=0\}\ AND\ \{S_0(S_1^3+S_3)=0\}\ AND\ (S_1\neq 0)$$

ただし、d=8. (10)

3.2 一般的な誤り訂正および検出を行うBCH(RSを含む)符号の復号法(7)

前述した復号法は判定式を用いるので並列に復号できる利点があるが、誤りの数が増加すると判定式、したがって、判定回路が複雑になる。

ここでは、最も一般的な誤り訂正および検出を行うBCH(RSを含む)符号の復号法について述べる。まず、dは(6)式を満たすものとする。このとき、S_j $(r \leq j \leq r+d-2=r+2t+k-1)$ を算出し $S_{j'}$ $(r \leq j' \leq r+t-1)$ を用いても重以下の誤りを訂正する。そして訂正された符号よりもう一度 S_j を求め、S_j が全て零であれば正しい訂正とし、そうでなければ誤訂正がなされたとして誤りの検出のみとする。

この復号法は誤りの数が多くなっても、シンドローム生成回路を付加するだけで簡単ではあるが、復号遅延時間が大きくなる欠点を有する。

4. シミュレーションによる復号性能

前項までに述べた各種の符号の復号性能を表5、表6に示す。表において、判定式による誤り訂正(または誤訂正)の割合を上段に、誤り検出の割合を下段に示す。( )内は誤り訂正回路内での訂正不能を検出に含めた場合の復号器全体の誤訂正と検出の割合を示す。設計能力以内の誤り訂正および検出は判定式で100%実行されており、それ以上の誤りに対してもかなりの検出能力を有する。

なお、シミュレーションはパソコンを用い、誤りを乱数により一万回変化させて行った。

また、3.2項の復号法によるシミュレーションの結果も同様だったので省略した。

5. ROMを用いた復号器の構成例

前項までに述べた判定式を用いた誤り訂正および検出を行うBCH(RS)符号のうち、計算機メモリシステムに適用可能と思われる符号の復号器のいくつかの具体的構成例を示す。

5.1 構成要素について

$GF(2^m)$ 上のガロア体の元の基本演算回路を示す。ここで述べる方法は文献(5)の引用である。元はm次以下の多項式で表現(ベクトル表現と称す)する方法と原始元αのべき乗で表現(指数表現)する方法がある。ここでは主として後者を用いる。まず、図3に乗算回路を示す。A1Dは零元検出回路、ZOは零元出力回路、mFAはmビット全加算器である。OCCは出力補正回路であり、MOD 2^m-1 を行う。図4に除算回路を示す。1Cは1の補数器であり、被除数の指数から除数の指数を減ずるためのものである。なお、除数が零元のとき除算エ

表5 BCH符号の復号性能

符号	誤り発生数 (bit) 1	2	3	4	5	6
SEC/DED	100.00	0.00	100.00	1.80		
			(95.97)	(0.00)		
	0.00	100.00	0.00	98.20		
			(4.03)	(100.00)		
SEC/TED	100.00	0.00	0.00	1.56	1.77	
				(1.56)	(1.76)	
	0.00	100.00	100.00	98.44	98.23	
				(98.44)	(98.24)	
SEC/QdED	100.00	0.00	0.00	0.00	1.77	0.10
					(1.76)	(0.00)
	0.00	100.00	100.00	100.00	98.23	99.90
					(98.24)	(100.00)
DEC/TED	100.00	100.00	0.00	100.00	1.77	
				(45.67)	(1.76)	
	0.00	0.00	100.00	0.00	98.23	
				(54.33)	(98.24)	
DEC/QdED	100.00	100.00	0.00	0.00	2.17	1.73
					(1.62)	(1.05)
	0.00	0.00	100.00	100.00	97.83	98.27
					(98.38)	(98.95)
DEC/QnED	100.00	100.00	0.00	0.00	0.00	1.63
						(1.05)
	0.00	0.00	100.00	100.00	100.00	98.37
						(98.95)
TEC/QdED	100.00	100.00	100.00	0.00	100.00	1.73
					(23.77)	(1.63)
	0.00	0.00	0.00	100.00	0.00	98.27
					(76.23)	(98.37)

上段：誤り訂正(%)，下段：誤り検出(%)

表6 RS符号の復号性能

符号	誤り発生数 (digit) 1	2	3	4	5	6
SEC/DED	100.00	0.00	2.12	1.59		
			(1.84)	(1.49)		
	0.00	100.00	97.88	98.41		
			(98.16)	(98.51)		
SEC/TED	100.00	0.00	0.00	0.01	0.03	
				(0.01)	(0.03)	
	0.00	100.00	100.00	99.99	99.97	
				(99.99)	(99.97)	
SEC/QdED	100.00	0.00	0.00	0.00	0.00	0.00
					(0.00)	(0.00)
	0.00	100.00	100.00	100.00	100.00	100.00
					(100.00)	(100.00)
DEC/TED	100.00	100.00	0.00	1.63	1.27	
				(0.69)	(0.59)	
	0.00	0.00	100.00	98.37	98.73	
				(99.31)	(99.41)	
DEC/QdED	100.00	100.00	0.00	0.00	0.02	0.01
					(0.01)	(0.01)
	0.00	0.00	100.00	100.00	99.98	99.99
					(99.99)	(99.99)

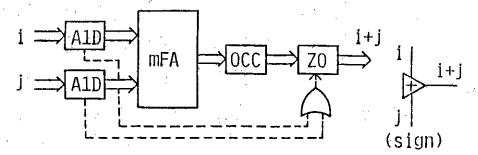


図3 乗算回路

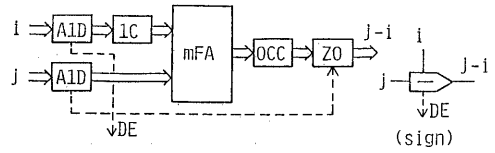


図4 除算回路

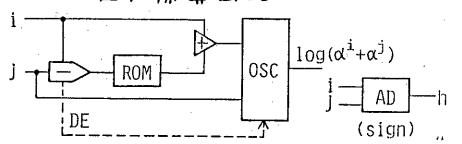


図5 加法回路

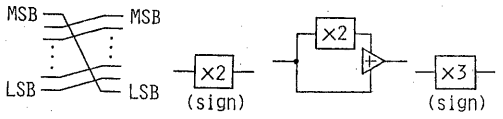


図6 2乗回路と3乗回路

ラー信号DEを出力する。つぎに、図5に加法回路を示す。$\alpha^i+\alpha^j=\alpha^i(1+\alpha^{j-i})$ より $\log\alpha(1+\alpha^{j-i})$ を得るROMテーブルを用いている。OSCはα^iが零元のときにjを出力するための方のである。図6に累乗回路を示す。2乗の場合は1ビット分上位方向にローテートシフトすれば良い。3乗の場合はもとの元の指数と2乗回路の出力を加算すれば良い。なお、図3の乗算回路で零元処理を行わないときは、A1D、ZOを省くことができる。この場合の乗算回路をMUで表わすこととする。同様にこのとき、図4の除算回路はMUと1C(1の補数器)で構成できる。

5.2 復号器の構成例

5.2.1 単一誤り訂正/2, 3, 4ビット誤り検出(SEC/QdED)BCH符号の復号器

SEC/QdED BCH符号の復号について述べる。さて、単一誤りのときの誤り位置多項式は次式となる。

$$x+S_1=0 \qquad (11)$$

また、判定式は(8)式、$BZ1$ であるが次式

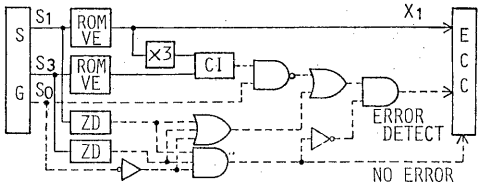


図7 SEC/QdED BCH符号復号器

のように変形できる。

$$BZ1'=(S_1^3+S_3=0)\ AND\ (S_0=1) \qquad (12)$$

したがって、SEC/QdED BCH符号復号器は図7となる。SGは受信符号よりシンドロームをベクトル表現で算出する回路。ROM(VE)は元をベクトル表現から指数表現に変換するROMである。X_1は誤り位置数である。ECCは誤りを訂正する回路。ZDはベクトル表現の零元検出回路。C1は一致検出回路である。シンドロームがall 0'sのとき誤りなし、一部のシンドロームが零か、$BZ1'$が偽のとき誤り検出とする。

5.2.2 2重誤り訂正/3ビット誤り検出(DEC/TED)BCH符号の復号器

DEC/TED BCH符号の復号について述べる。まず、2重以下の誤りの訂正原理について述べる[5]。単一誤りのとき $x=S_1$ である。2重誤りのとき誤り位置多項式は次式となる。

$$x^2+S_1x+\frac{S_1^3+S_3}{S_1}=0 \qquad (13)$$

(13)式に $x=S_1y$ を代入して次式を得る。

$$y^2+y+1+\frac{S_3}{S_1^3}=0 \qquad (14)$$

したがって、$C_i=S_3/S_1^3$ に対して(14)式の根y_1, y_2を格納した表を用いれば(13)式の根は $x_i=S_1y_i$ $(i=1,2)$ となる。

つぎに、誤り判定式は表2より $B1M_3=S_0(S_1^3+S_3)$ である。ここで、2重以下の誤りのとき$S_1\neq0$であり、$GF(2^m)$が1の立方根を持たないとき$S_3\neq0$である。よって、このとき零元操作が不要となるので図1の回路の代わりにMUを用いる。また、除算はMUへの除数の入力が1の補数となるようにする。図8にDEC/TED BCH符号復号器を示す。誤り訂正動作は、MU2の出力で$C_i=S_3/S_1^3$を求め、C_iでROMを索表し、y_1, y_2を求めればMU3, 4の出力が誤り位置数X_1, X_2となる[5]。根がないときROMの出力はall 1'S

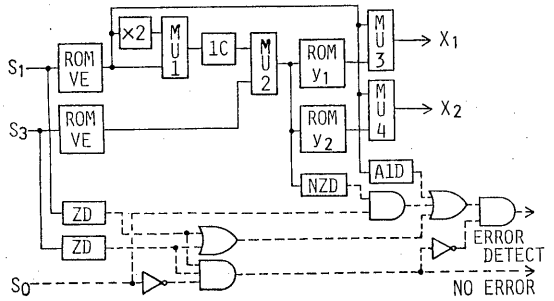


図8 DEC/TED BCH 符号復号器

(指数表現の零元)を出力するので、A1Dで検出して、誤り検出とする。また、$S_1^3+S_3\neq 0$は、$S_3/S_1^3\neq\alpha^0$として、NZDで検出し、$S_0=1$とANDをとって判定式が非零であることを検出し、誤り検出としている。

なお、$GF(2^m)$が1の立方根を持つときは$S_3=0$となり得るので、S_3のZDを除き、$S_3=0$ならば、MU2の出力をall 1'sとしてROM(γ_1,γ_2)の出力が1の立方根となるようにすれば良い。

5.2.3 単一誤り訂正/2ディジット誤り検出(SbEC/DbED) RS符号の復号器

SbEC/DbED RS符号の復号について述べる。まず、単一誤りのとき、誤り位置数X_1、誤りの大きさY_1は次式となる。

$$X_1 = S_1/S_0 \qquad (15)$$

$$Y_1 = S_0 \qquad (16)$$

また、判定式は表3より、$R1M_2=S_0S_2+S_1^2$である。したがって、前項と同様にしてSbEC/DbED RS符号の復号器は図9に示すものとなる。単一誤りのときシンドロームは非零であり、零元操作は不要である。

5.2.4 2重誤り訂正/3ディジット誤り検出(DbEC/TbED) RS符号の復号器

DbEC/TbED RS符号の復号について述べる。単一誤りのときは前項と同じである。2ディジット誤りのときは、誤り位置多項式は次式となる。

$$x^2+\sigma_1 x+\sigma_2=0 \qquad (17)$$

ただし、$\sigma_1=\dfrac{S_0S_3+S_1S_2}{S_0S_2+S_1^2}$, $\sigma_2=\dfrac{S_1S_3+S_2^2}{S_0S_2+S_1^2}$

(17)式は(13)式と同様に$x=\sigma_1 y$とおき

$$y^2+y+\sigma_2/\sigma_1^2=0 \qquad (18)$$

として、$C_i=\sigma_2/\sigma_1^2$に対する根のテーブルを用いて解くことができる。(17)式の根をX_1, X_2とすると誤りの大きさY_1, Y_2は次式となる。

$$Y_1=(X_2S_0+S_1)/\sigma_1$$
$$Y_2=(X_1S_0+S_1)/\sigma_1 \qquad (19)$$

また、判定式は表3より、$R1M_3=S_0S_2S_4+S_2^3+S_1^2S_4+S_0S_3^2$となるので、DbEC/TbED RS符号復号器は図10のように構成される。零元操作は必要である。

σ_1の分母($S_0S_2+S_1^2$)が零のときDEエラーとなる。このとき、単一誤りなのでSWを上側とする。動作は前項までの回路と同様なので

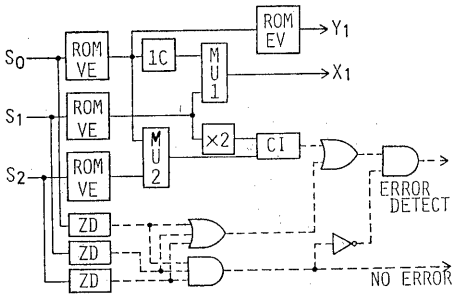


図9 SbEC/DbED RS符号復号器

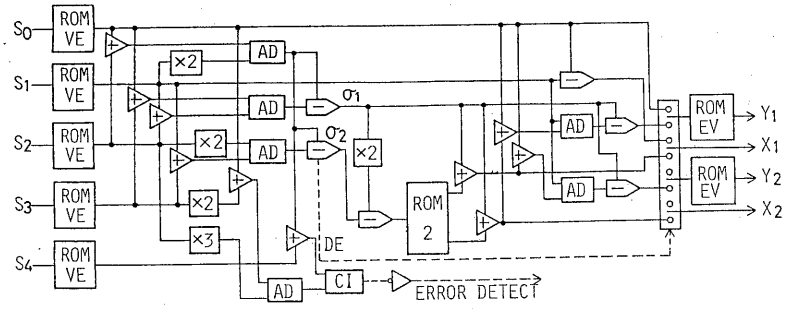


図10 DbEC/TbED RS符号復号器

表7 構成要素の評価

要素	Time (ns)	ゲート数	チップ数
乗算	62 (54)	167 (154)	9 (5)
除算	70	174	11
加法	176	367	23*
2乗	0	0	0
3乗	62	167	9

*加法回路はROM1個を含む。

詳細は省略する。

5.3 ROMを用いた復号器の評価

GF(2^8)を用い、情報128ビットの短縮化符号としたときの構成要素と前項に示した復号器の評価を行う。ただし、通常のTTLで構成したものとし、4入力AND、OR、2入力XORを1ゲートと数える。また1入力Inverterを各々1ゲートと数える。またROMの遅延時間は35nsとして計算した。構成要素の評価を表7に示す。乗算の( )内は零元操作を行わない場合、すなわちMUの評価である。また、表8に復号器の評価を示す。評価した値はシンドローム生成回路と訂正を実行する部分を省いた値である。したがって、復号器全体の遅延時間はこの両者の遅れ(約100ns)を加えた値となる(5)。なお、( )内は誤り訂正および検出の判定回路の評価であり、復号遅延時間は大きい方の値となる。2重誤り訂正以上の場合、判定回路の復号遅延時間は無視でき、ハードウェア量も復号器全体の約15%以下である。

また、DEC/TED RS符号の場合、情報128ビットであれば、GF(2^5)を用いれば良いから、さらに簡単な回路で構成できる。

6. 復号器の修正

前項までに述べた復号器をROMを用いないで構成する。また、SbEC/DbED RS符号の符号長を2倍に拡張する。これらの修正された復号器は計算機メモリシステムにより適合するであろう。

6.1 ROMを用いないBCH符号復号器

6.1.1 Hマトリックスについて

いま、例としてGF(2^4)の2重誤り訂正BCH符号のHマトリックスを考える。

$$H_1=\begin{bmatrix}\alpha^{14} & \alpha^{13} & \alpha^{12} & \cdots & \alpha^{i} & \cdots & \alpha^{2} & \alpha & 1\\ \alpha^{12} & \alpha^{9} & \alpha^{6} & \cdots & \alpha^{3i} & \cdots & \alpha^{6} & \alpha^{3} & 1\end{bmatrix} \quad (20)$$

(20)式をベクトル表現すると次式となる。

$$H_2=\begin{bmatrix}1 & 1 & 1 & \cdots & \alpha^{i} & \cdots & 0 & 0 & 0\\ 0 & 1 & 1 & \cdots & & \cdots & 1 & 0 & 0\\ 0 & 0 & 1 & \cdots & & \cdots & 0 & 1 & 0\\ 1 & 0 & 1 & \cdots & & \cdots & 0 & 0 & 1\\ 1 & 1 & 1 & \cdots & \alpha^{3i} & \cdots & 1 & 0 & 0\\ 1 & 0 & 1 & \cdots & & \cdots & 0 & 1 & 0\\ 1 & 1 & 0 & \cdots & & \cdots & 0 & 0 & 1\\ 1 & 0 & 0 & \cdots & & \cdots & 0 & 0 & 1\end{bmatrix} \quad (21)$$

このとき、生成多項式は次式となる。

$$g(X)=(X^4+X+1)(X^4+X^3+X^2+X+1)$$
$$=X^8+X^7+X^6+X^4+1 \quad (22)$$

さて、$X^{n-1}=X^{14}$ を $g(X)$ で割った剰余の多項式の係数を最左列におく。同様に X^i を $g(X)$ で割った剰余多項式の係数を α^i の列におく。このHマトリックスを次式に示す。

$$H_3=\begin{bmatrix}1 & 0 & 0 & \cdots & \alpha^{i} & \cdots & 0 & 0 & 0\\ 1 & 1 & 0 & \cdots & & \cdots & 0 & 0 & 0\\ 1 & 1 & 1 & \cdots & & \cdots & 0 & 0 & 0\\ 0 & 1 & 1 & \cdots & & \cdots & 0 & 0 & 0\\ 1 & 0 & 1 & \cdots & \alpha^{3i} & \cdots & 0 & 0 & 0\\ 1 & 0 & 1 & \cdots & & \cdots & 1 & 0 & 0\\ 0 & 1 & 0 & \cdots & & \cdots & 1 & 0 & 0\\ 0 & 0 & 0 & \cdots & & \cdots & 0 & 0 & 1\end{bmatrix} \quad (24)$$

表8 ROMを用いた復号器の評価

符号	Time(ns)	ゲート数	ROM数	チップ数	検査ビット数
SEC/QdED BCH	35 (137)	196 (196)	2 (1)	19 (18)	17
DEC/TED BCH	240 (—)	639 (15)	4 (0)	33 (7)	17
SEC/DED RS	97 (121)	345 (183)	3 (0)	25 (14)	24
DEC/TED RS	832 (—)	5,206 (887)	14 (2)	323 (55)	40

もちろん、H_1, H_2, H_3 は等価である。したがって、符号化の際にはH_3 を用い、復号化の際にH_2 を用いると効率が良い。チェックビットおよびシンドロームの算出はHマトリックスの1のあるビットのEXOR Treeを求めれば良い。

6.1.2 復号器

前章で述べた SEC/QdED, DEC/TED BCH符号復号器をROMを用いないで構成する。ガロア体の元はベクトル表現で行う。

まず、基本演算回路について述べる。2乗回路はMOD2の性質を用いて簡単に構成できる。具体的として、GF(2^4)の元を用いると

$$(A_3\alpha^3+A_2\alpha^2+A_1\alpha+A_0)^2 = A_3\alpha^3+(A_3+A_1)\alpha^2+A_2\alpha+(A_2+A_0) \quad (25)$$

つぎに、任意の元とある定まった元の乗算を考える。例として、$X\alpha^8$ を考える。

$$(A_3\alpha^3+A_2\alpha^2+A_1\alpha+A_0)\times\alpha^8 = A_3\begin{bmatrix}\alpha^3\\ \alpha^2\\ \alpha\\ 0\end{bmatrix}+A_2\begin{bmatrix}0\\ \alpha^2\\ \alpha\\ 1\end{bmatrix}+A_1\begin{bmatrix}\alpha^3\\ 0\\ \alpha\\ 0\end{bmatrix}+A_0\begin{bmatrix}0\\ \alpha^2\\ 0\\ 1\end{bmatrix} \quad (26)$$

ここで、(26)式に対応した次の行列を考える。

$$T_{\alpha^8}=\begin{bmatrix}1&0&1&0\\ 1&1&0&1\\ 1&1&1&0\\ 0&0&0&1\end{bmatrix} \quad (27)$$

(27)式をα^8の変換行列と呼ぶ。変換行列の上側、横にA_3, A_2, A_1, A_0を並べると最上段のEXOR Treeは積のα^3の係数となり、次段のEXOR Tree はα^2の係数となる。結局、ある定まった元の乗算はEXOR Tree で容易に実現できることになる。また、ベクトル表現の元同志の乗算は図11 のように構成される。もちろん、3乗回路は2乗回路と乗算回路を組合わせれば良い。

さて、上記基本回路によって構成したROMを用いない SEC/QdED BCH符号復号器を図12に示す。A(ho)~A(hn-1)はHマトリックスの各列とシンドロームの一致を判定するためのものである。もし、2, 3, 4ビット誤りであればORの出力は零(L)である。

つぎに、図13 に同様なDEC/TED BCH

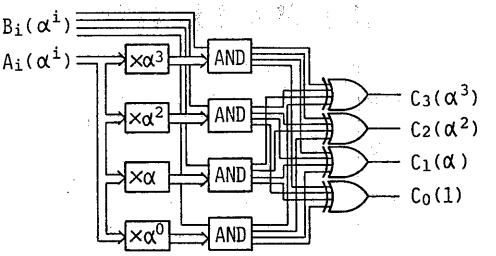


図11 $(A_3\alpha^3+A_2\alpha^2+A_1\alpha+A_0)\cdot(B_3\alpha^3+B_2\alpha^2+B_1\alpha+B_0)$
$= C_3\alpha^3+C_2\alpha^2+C_1\alpha+C_0$ の乗算回路

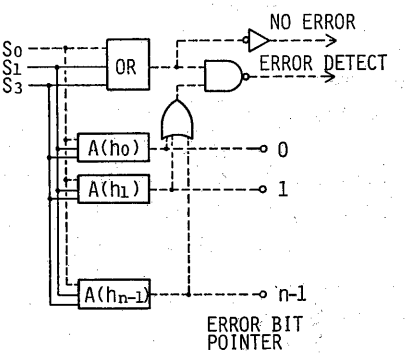


図12 ROMを用いない SEC/QdED BCH
符号復号器

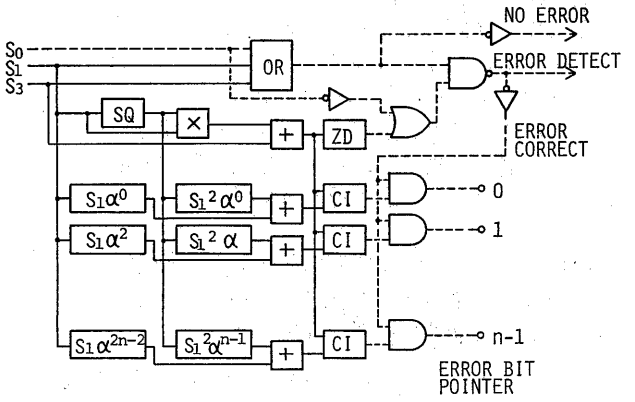


図13 ROMを用いない DEC/TED
BCH符号復号器

符号復号器を示す。誤り位置多項式の根を求めるために、方程式に全ての元を代入し並列に根か否かを確かめる方法を用いている。除算回路を用いないで、$S_1X^2+S_1X=S_1^3+S_3$ の形で根を

求めている。

6.2 SbEC/DbED RS 復号器の修正

6.2.1 ROMを用いないSbEC/DbED RS符号復号器

本復号器を図14に示す。シンドロームの導出回路は前項の変換行列によるEXOR Treeを用いれば容易に行える。

誤り位置の判定は

$S_0\alpha^i = S_1$ AND $S_2 = \alpha^i S_1$

で行っている。これは $\alpha^i \neq 0$ のとき $Z = S_1^2 + S_0 S_2$ と等価である。

6.2.2 倍長SbEC/DbED RS符号復号器

まず、倍長SbEC/DbED RS符号のHマトリックスを求める。

通常のSbEC/DbED RS符号に単位行列を付加し、最上段がall 1'sとall'0の行を加えた次式のH_1マトリックスは$d=4$、すなわち、やはりSbEC/DbEDの能力を有する(4)。

GF(2^3)を例にとると、

$$H_1 = \left[\begin{array}{ccc|ccc|cccc} 1 & 1 & \cdots & 1 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1 & \cdots & 1 & 1 & 0 & 0 & 1 & \cdots & 1 & 1 & 0 \\ \alpha^6 & \alpha^5 & \cdots & 1 & 0 & 1 & \alpha^6 & \alpha^5 & \cdots & 1 & 0 & 1 \\ \alpha^5 & \alpha^3 & \cdots & 1 & 0 & 0 & \alpha^5 & \alpha^3 & \cdots & 1 & 0 & 0 \end{array}\right] \quad (28)$$

このH_1マトリックスが単位行列を持つように、最上段の行を次段に加え、2列を除き$(1000)^t$を後に廻すと次式を得る。

$$H_2 = \left[\begin{array}{ccc|ccc|cccc} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & 0 & 1 & 0 & 0 \\ \alpha^6 & \alpha^5 & \cdots & 1 & \alpha^6 & \alpha^5 & \cdots & 1 & 0 & 0 & 1 & 0 \\ \alpha^5 & \alpha^3 & \cdots & 1 & \alpha^5 & \alpha^3 & \cdots & 1 & 0 & 0 & 0 & 1 \end{array}\right] \quad (29)$$

A　　B　　$C_1\ C_2\ C_3\ C_4$

なお、このH_2マトリックスで求めたシンドロームC_1, C_2, C_3, C_4はH_1マトリックスで考えたとすれば、$S_0 = C_1 + C_2$, $S_1 = C_3$, $S_2 = C_4$とすれば良い。

したがって、H_2の倍長SbEC/DbED RS符号の復号アルゴリズムは図15となる。

$C_1 \neq 0 \wedge C_2 \neq 0$であれば2重誤りである。また、チェックビットの誤りは$C_1 \sim C_4$のうち1つが非零であることを判定しておけば、後は単一ブロック内の誤りを判定すれば良い。これはSbEC/DbED RS符号と同じ$Z = S_1^2 + S_0 S_2$を用いれば良い。さらにC_1でAブロックの誤りかBブロックの誤りかを判定する。

図16にROMを用いた倍長SbEC/DbED RS符号の復号器、図17にROMを用いない倍長SbEC/DbED RS符号復号器を示す。

なお、倍長の原理は文献(4)による。

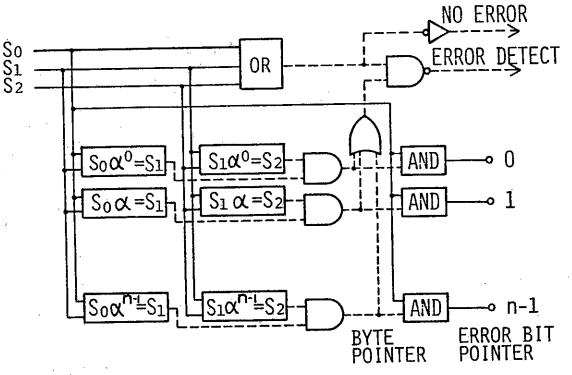


図14　ROMを用いないSbEC/DbED RS符号復号器

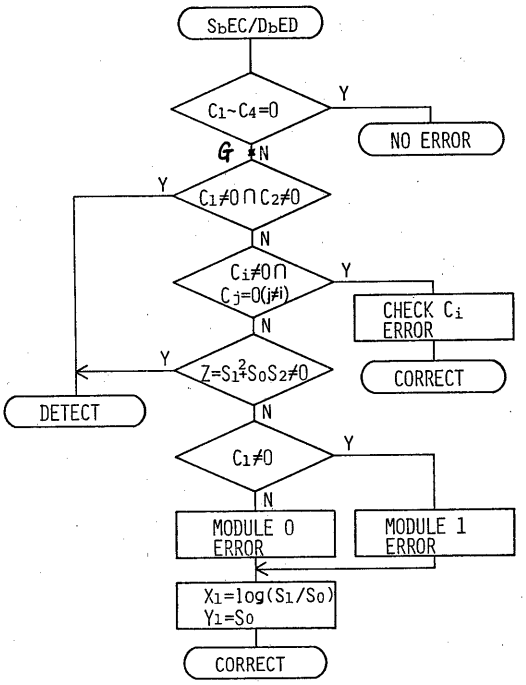


図15　倍長SbEC/DbED RS符号の復号アルゴリズム

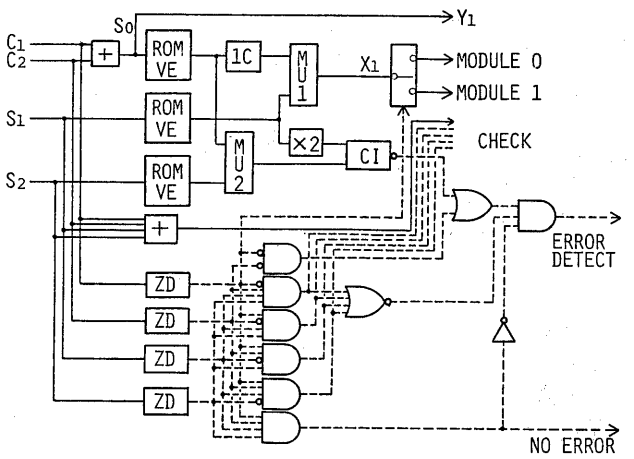


図16 ROMを用いた倍長 SbEC/DbED RS符号復号器

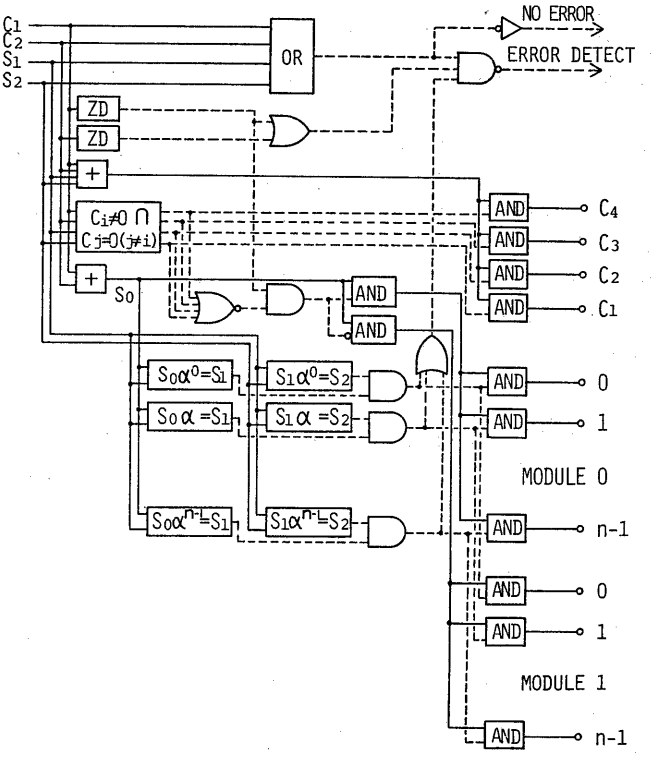


図17 ROMを用いない倍長 SbEC/DbED RS符号復号器

表9　ROMを用いない復号器の評価

符号	シンドローム生成部		復号部		復号器合計		検査ビット数
	段数	ゲート数	段数	ゲート数	段数	ゲート数	
SEC/QdED BCH	7	1,248	4	954	11	2,202	15
DEC/TED BCH	7	1,248	16	10,024	23	11,272	15
S4EC/D4ED RS	6	656	5	397	11	1,053	16

6.3 ROMを用いない復号器の評価

前項で述べたROMを用いない復号器の評価を表9に示す。

情報ビット数が64の場合を評価している。SEC/QdED BCH および SbEC/DbED RS（ただし、b=4）符号の復号器はゲート段数、ゲート量とも少なく、LSI化すれば主記憶に充分適用できる。なお、SbEC/DbED RS符号は文献(4)とほぼ同じ結果である。

DEC/TED BCH符号復号器は2重誤り訂正をベースにしているので、段数、ゲート量とも大きくなっている。（XORは2ゲートとした）

上記の復号器はハードウエア量からみて充分LSI化が可能であり、主記憶に適用可能であろう。

7. あとがき

SEC/QdED BCH符号、DEC/TED BCH符号、SbEC/DbED RS 符号、DbEC/TbED RS符号 の復号器 の構成を示した。ROMを用いた方が簡単に構成できる利点があるが、ROMを用いないでより高速に復号することも可能であることを明らかにした。

なお、DEC/TED BCH符号復号器は文献(9)の手法、特に逆元の算出、あるいは文献(2)によるベクトル表現による2次方程式の解法を用いる等すれば、ゲート数を減少することができるので今後改良して行きたい。

なお、(29)式の符号は富士通KKの特許と同じであることが判ったが、復号器において、A, Bブロックの誤り位置判定部分を共用している部分が異なる。（未だ特許を確認していないが）

また、この場合 (144, 128) S4EC/S4ED RS 符号は構成できないので、この符号の構成法、および復号法を付録に付す。

謝辞　日ごろ御指導、御助言頂く、広島大学工学部市川忠男教授に深謝します。また、有益な御助言を頂いた、横浜国立大学工学部今井秀樹助教授、武蔵野電気通信研究所、藤原英二博士、金田重郎調査員に深謝します。

文献

(1) W.W.Peterson and E.J.Weldon, "Error Correcting Codes", 2nd Edition, p.p. 269-309, MIT Press, Mass. (1992).

(2) E.R.Berlekamp, "Algebraic Coding Theory", McGraw-Hill Book Co. New York, 1968.

(3) M.Y. Hsiao, "A Class of Optimal Minimum Odd-Weight-Column SEC-DED Codes, IBM. J. Res. Develop., 14, No.4, pp.395~401, 1970.

(4) S. Kaneda and E. Fujiwara, "Single Byte Error Correcting-Double Byte Error Detecting Codes for Memory Systems", IEEE Trans. Comput., C-31, No.7, pp.596~602.

(5) 山岸、今井、"ROMを用いたBCH符号の復号器の一構成法"、信学論、J63-D、12、pp.1034-1041（昭55-12）.

(6) 岡野、"ROMを用いたBCH符号復号器の改良"、信学論、Vol.J67-D, 3, pp.359-366（昭59-3）

(7) 今井、私信

(8) 岡野、守川、"誤り訂正および検出を行うBCH符号復号器の一構成法"、信学論、投稿中

(9) 今井、上柳、"2重誤り訂正BCH符号の並列復号器について"、信学論D. Vol. J60-D, 9, p.p.761-762（昭52-9）.

付録

倍長S_4EC/S_4ED RS符号による (144,128) 符号の構成とその復号法

上記符号の構成法を示す。

(28)式より(29)式を求める際に省略した2列を(29)式のチェックビットの前に並べて次式のHマトリックスを得る。

$$H_3=\left[\begin{array}{cccc|cccc|c|c|cccc} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & 1 & 1 & 0 & 1 & 0 \\ \alpha^b & \alpha^5 & \cdots & 1 & \alpha^b & \alpha^5 & \cdots & 1 & 1 & 0 & 0 & 1 & 0 \\ \alpha^5 & \alpha^3 & \cdots & 1 & \alpha^5 & \alpha^3 & \cdots & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \quad (A1)$$

A　B　K_1K_2　$C_1C_2C_3C_4$

このH_3は$d=4$であることは明らか。

復号は(29)式のHマトリックスの符号と同様に行えば良いが、K_1, K_2の信号については、チェックビットと同様に特別に単一誤りを検出する。したがって(A1)式のH_3による復号アルゴリズムは図15のGの個所に付図1を挿入すれば良い。

この場合の復号器は図16, 17の回路に付図1の機能を付加すれば良いので構成図は省略する。

なお、およそのゲート量は表9のS_4EC/D_4ED RS符号の2倍程度となり、文献(4)の復号器よりゲート量はやや少なくなる。

また、(29)式のH_2による最長符号長は、

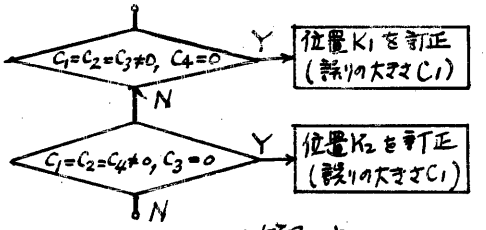
$$\underbrace{(2^b-1)\times b\times 2}_{\text{情報ビット}}+\underbrace{4b}_{\text{チェックビット}} \quad (A1)$$

となり、$b=4$のとき(136, 120)符号までしか構成できない。しかし、(A1)式のH_3による符号の最長符号長は

$$\underbrace{[(2^b-1)+3]\times b\times 2-4b}_{\text{情報ビット}}+\underbrace{4b}_{\text{チェックビット}} \quad (A2)$$

となり、$b=4$のとき(144, 128)符号までが構成できる。(GF(2⁴)上で構成する。)

なお、図15の復号アルゴリズムでは、Zが零か否かの判定と誤り位置の算出を別々に行っているが、ROMを用いない復号器においては、この両方を同時に行うようになっている。



付図1. (144, 128) 符号とするために図15に加える処理