解　説

誤り検出・訂正符号の応用†

藤原英二†† 金田重郎††

1. まえがき

誤り検出・訂正符号[1]は，計算機システムの高信頼化を図る上で重要な技術の一つである．

誤り検出・訂正符号は，元来，符号理論[1]をもとにディジタル通信分野で発達したものである．計算機システムへの本格的な応用の最初の例は，IBM 7030 システム (1961年) であるといわれ[24]，1ビット誤り訂正・2ビット誤り検出の機能を有するハミング符号 (主記憶用)，誤り検出を目的とした Modulo 3 剰余符号 (浮動小数点演算用)，パリティ検査符号が採用されている．

現在では，符号は論理回路，および主記憶装置，磁気ディスク装置等の記憶装置を中心に広く応用されている．この背景には主として次の要因が考えられる．

1) 計算機システムの大規模化と，それによる高信頼化への要求

2) LSI 技術等の進展による符号化・復号回路の経済化・高速化・高信頼化

3) LSI 高密度化に伴う信号レベルの低下，外部雑音・α線等の影響によるソフトエラー[25]の問題

4) 計算機への応用に適した符号の研究

また，誤り検出・訂正符号の採用に当っては，故障発生傾向に基づく最適な機能の符号を構成するとともに，経済性・性能への影響の観点から，付加検査ビット数，符号化・復号回路ゲート量，復号速度等について考慮しなければならない．

本解説は，このような観点から，計算機システムへの符号応用の現状について述べる．

2. 論理回路への応用

現状における論理回路への符号の実用例は，誤りの検出が主体である．たとえば，ALU を含むデータパス回路へのパリティ検査符号の応用や，加算回路，乗算回路への剰余符号の応用である[4]．また，セルフチェッキング論理回路の構成を目的に，前記符号以外に K out of N 符号，Berger 符号等の誤り検出符号の適用が研究されている[3]．

一方，誤り訂正符号としての応用は，ゲートへの入力を符号化し，入力中に誤りが含まれていてもゲート演算を行うと正しい出力を生成するフォールトトレラントゲートの提案がある[7]．また，近年，算術演算回路に適した訂正機能を有する算術符号 (Arithmetic Codes) の研究が活発である．

なお，論理回路への符号の実用化に当っては，回路の2重化，3重化による構成と比較して，優位性が高いことが必要である．

(1) 算術符号

この符号は，ハミング距離とは別の算術距離を定義して構成するもので，AN 符号が著名である[2]．AN 符号は，もとの数 N に A を乗じて AN なる数を作成して構成するもので，符号語として情報部と検査部が分離されない符号である．

N の範囲が $0 \sim m-1$ のとき m を情報域，A を生成数という．N を r 進表示したとき，A は r と互いに素となるように選ばれる．2個の数 N_1, $N_2(<m)$ の符号語 AN_1, AN_2 に対し，加法 (mod. M, $M=A\cdot m$) は，

$$(AN_1+AN_2)_{\mathrm{mod.}\,M}=A\{(N_1+N_2)_{\mathrm{mod.}\,m}\}$$

となり，本符号は本質的に算術演算に適する．与えられた語 L が符号語であれば，$L_{\mathrm{mod.}\,A}=0$ であり，誤って L' になると $L'_{\mathrm{mod.}\,A}=(L'-L)_{\mathrm{mod.}\,A}=E_{\mathrm{mod.}\,A}\neq 0$ となる．$E_{\mathrm{mod.}\,A}$ は誤り E のシンドローム* であり，対象とするすべての誤りに対して，このシンドロームがすべて異なれば訂正が可能である．

† Application of Error Correcting Codes for Increasing Computer System Reliability by Eiji FUJIWARA and Shigeo KANEDA (Musashino Electrical Communication Laboratory, N.T.T.).

†† 日本電信電話公社武蔵野電気通信研究所

* シンドローム (Syndrome) は誤りの状態を示す情報であり，復号時には，この情報から具体的な誤り位置指摘を行う．

表-1 単一誤り訂正 AN 符号

A	m	$M=A\cdot m$
11	3	2^5+1
19	27	2^9+1
29	565	$2^{14}+1$
47	178481	$2^{23}-1$
59	9099507	$2^{29}+1$
67	128207979	$2^{33}+1$
79	6958934353	$2^{39}-1$
83	26494256091	$2^{41}+1$
101	5099523830125	$2^{50}+1$

表-1 は，単一誤り訂正 AN 符号 (BP 符号)[2] の $r=2$ に対する A, m, M を示す．ここで，A は検査部の大きさを，m は情報部の大きさを表わしている．この他，算術符号としては，単一バイト*誤り訂正 AN 符号[5]，組織符号** として構成した gAN 符号，多剰余符号等がある[2]．

(2) 組合せ符号

単一誤り訂正 AN 符号を，たとえば加算回路に適用した場合，訂正回路はもとの加算回路に対し 4～5 倍のゲート量を必要とし，実用的に問題がある．そこで，簡明な検査が行えるパリティ検査符号と算術演算に適する剰余符号を組み合わせ，両符号の特長を生かした組合せ符号 (Combination Code) の提案がある[6]．この符号の符号語を**図-1** に示す．k バイトからなる符号語 $\boldsymbol{X}$ の各バイト b ビット (X_i) に対して，1ビットのパリティ検査符号 (P_i) を付加し，全体として1バイトの low cost 剰余符号[4] $|\boldsymbol{X}|_z (z=2^b-1)$ を付加する．ここで，$|\boldsymbol{X}|_z$ は，$\boldsymbol{X}$ の z に対する剰余を示す．

演算回路 (ALU) に入力する情報は，$[\boldsymbol{A}\vdots P_A\vdots|\boldsymbol{A}|_z]$，$[\boldsymbol{B}\vdots P_B\vdots|\boldsymbol{B}|_z]$ であり，演算 ϕ*** を実行して結果 $[\boldsymbol{Y}\vdots P_Y]$ を出力する．ここで，$\boldsymbol{Y}$ は演算結果であり，P_Y は予知パリティビットである．訂正のための概念図を**図-2** に示す．剰余計算部は演算 ϕ に対応して剰余 $|\boldsymbol{A}|_z$, $|\boldsymbol{B}|_z$ から $\boldsymbol{Y}$ に対する剰余 $|\boldsymbol{Y}|_z$ を予知する．そこで，$[\boldsymbol{Y}\vdots P_Y\vdots|\boldsymbol{Y}|_z]$ が符号語であるかをデコーダ部で調べ，シンドローム (S_p, S_r) を作成する．ここで，S_p はパリティ検査結果のシンドロームであり，

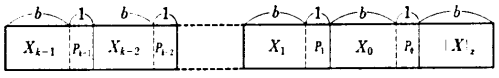

図-1 符号語 $\boldsymbol{X}$ のフォーマット

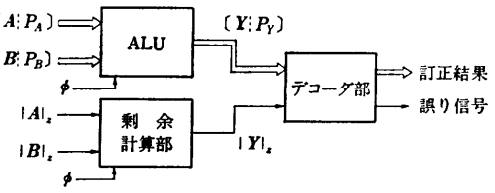

図-2 組合せ符号による訂正の概念図

S_r は剰余検査結果のシンドロームである．**図-1** に示すパリティ符号の付加方法より，S_p から誤りバイト位置を知る．S_r から誤りの大きさとバイト内誤り位置を知り，以上から $\boldsymbol{Y}$ の誤りを訂正する．本手法を，具体的に ALU に適用した場合，訂正回路はもとの ALU に対し，2倍程度のゲート量となる*．

(3) パリティを基本とする符号

主記憶装置等で実用化されているパリティを基本とする符号をそのまま ALU の誤り訂正に適用する提案がある[8]．これは，すべての算術論理演算結果を排他的論理和結果に変換して，その結果に対してパリティを基本とする符号を適用する考えである．この変換は，演算結果の誤りを伝搬させる観点から線形変換となる．本手法では，同一符号で主記憶と ALU の誤り訂正が可能であり，訂正回路はもとの ALU の 1.3 倍程度のゲート量増加で構成できる．

3. 記憶装置への応用

3.1 主記憶装置への応用

高速な読み書き動作を行う主記憶装置では，論理回路への応用と同様高速な復号が要求される．そのため，組合せ回路による並列復号手法を適用するとともに，符号構成に当っては，高速な復号が可能となるよう考慮しなければならない．また，高速な記憶素子は高価であるので，符号の冗長度ができるだけ小さいことが望ましい．

1ビット出力の構造を有する記憶素子を使用した装置では，故障による影響はビット単位に現われる．現状の 16 Kビット，64 Kビット素子は，1ビット出力構成が主流であるため，超大型機から小型機の主記憶装置に対して，1ビット誤り訂正・2ビット誤り検出 (SEC-DED**) 符号が広く使用されている．さらに一層の高信頼化を狙いとして，2ビット誤り訂正 (DEC***) 符号の適用に関する研究が行われている

* 本解説では，バイトは複数ビットの塊を意味する．
** 符号として情報部と検査部が区別できる符号をいう．
*** 演算 ϕ は加算，減算，シフト，論理和，論理積，排他的論理和など．

* 筆者らの試算による．
** Single bit Error Correcting-Double bit Error Detecting
*** Double bit Error Correcting

が，実用には至っていない．

一方，記憶素子の高集積化の進展にともない，複数ビット（バイト）出力の構造を有する記憶素子を用いた装置では，単一素子の故障が複数ビットの塊の誤り（バイト誤り）となって現われる比率が大きくなる．そのため，SEC-DED の機能を有して単一バイト誤りを検出する（SEC-DED-S*b*ED*）符号や，単一バイト誤り訂正（S*b*EC**）符号，単一バイト誤り訂正・二重バイト誤り検出（S*b*EC-D*b*ED***）符号等の研究が盛んである[9]．

（1）　1ビット誤り訂正・2ビット誤り検出（SEC-DED）符号

この符号を使用すれば，1ビット出力構成を有する1個の記憶素子故障は訂正でき，同時に2個の素子の故障は完全に検出できる．**図-3** は，本機能を有する簡単な符号と，それによる並列復号回路の例である．符号はパリティ検査行列（Hマトリクス****）で表現する．

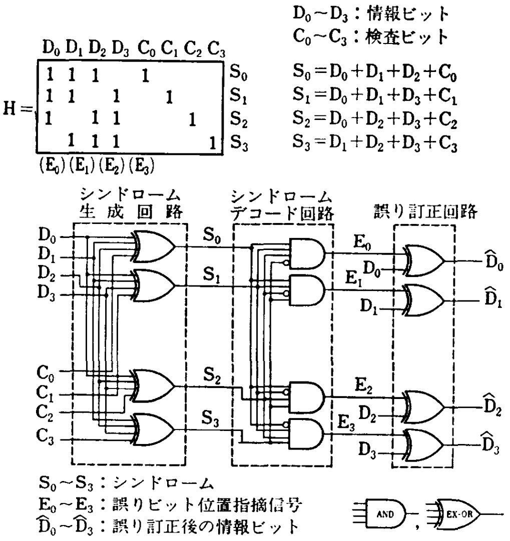

図-3　並列復号回路の構成例

この機能を有する符号として最も優れた符号は，**図-4** に示す奇数重み列 SEC-DED 符号である[10]．図は，情報ビット数 64 ビット，検査ビット数 8 ビットを有する H マトリクスの例である．この符号の特徴は，

1)　各列ベクトルは，相異なる奇数重みとする．

2)　1)の条件の下で，マトリクス中の '1' の数を最小にする．

3)　8ビット単位に各行ベクトルを垂直方向に巡回させる．

にあり，復号遅延，復号回路ゲート量の最小化を図っている．3)から，符号化・復号回路を8ビット単位のくり返し性を与えることができ，回路の LSI 化に都合がよい．また，この符号は，3ビット誤り検出能力が高い特長を有する*．

（2）　2ビット誤り訂正（DEC）符号[11]

この符号を使用すれば，同時に2ビットのランダムな誤りを訂正できる．本符号に対する研究は，BCH 符号[1] 等の検査ビット数が比較的少ない符号を用いて符号化・復号回路を簡単化，高速化する研究と，検査ビット数はある程度多くても高速で簡単な多数決復号可能な符号の研究がある．後者の符号としては，MA 符号，1段直交可能符号，直交ラテン方陣符号が提案されている．また，BCH 符号と多数決復号可能な符

* Single bit Error Correcting-Double bit Error Detecting-Single *b* bit-byte Error Detecting

** Single *b* bit-byte Error Correcting

*** Single *b* bit-byte Error Correcting-Double *b* bit-byte Error Detecting

**** マトリクス中の行ベクトルパターンに応じて，検査ビット，シンドロームの生成を行い，列ベクトルパターンに従ってデコードを行う．マトリクス中の '1' の数が少ないほど，復号遅延，回路ゲート量は小さくなる．

* SEC-DED 符号は 100% の3ビット誤り検出は保証していない．したがって，この符号では，同一検査ビット数のもとで3ビット誤り検出能力の高い符号が優れた符号である．

情報ビット																																																																検査ビット							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	0	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1																							1					1	1	1		1		1	1	1			1	1	1	1			1			1	1		1			1		1	1							
1		1			1		1	1	1	1	1	1	1	1	1																									1						1	1	1		1		1	1	1			1	1	1	1			1			1		1						
1	1			1			1	1		1			1		1	1	1	1	1	1	1	1	1																							1						1	1	1		1		1	1	1			1	1			1					
	1	1	1			1	1	1	1			1			1	1		1			1		1	1	1	1	1	1	1	1	1																							1						1	1	1		1				1				
			1	1	1		1		1	1	1			1	1	1	1			1			1	1		1			1		1	1	1	1	1	1	1	1	1																							1						1			
						1					1	1	1		1		1	1	1			1	1	1	1			1			1	1		1			1		1	1	1	1	1	1	1	1	1																					1		
														1					1	1	1		1		1	1	1			1	1	1	1			1			1	1		1			1		1	1	1	1	1	1	1	1	1															1	
																						1					1	1	1		1		1	1	1			1	1	1	1			1			1	1		1			1		1	1	1	1	1	1	1	1	1								1

図-4　(72, 64)* SEC-DED 符号（空白は '0'）

* (n, k) は符号長が n ビットで，情報ビット数が k ビットであることを示す．したがって，検査ビット数は $(n-k)$ ビットである．

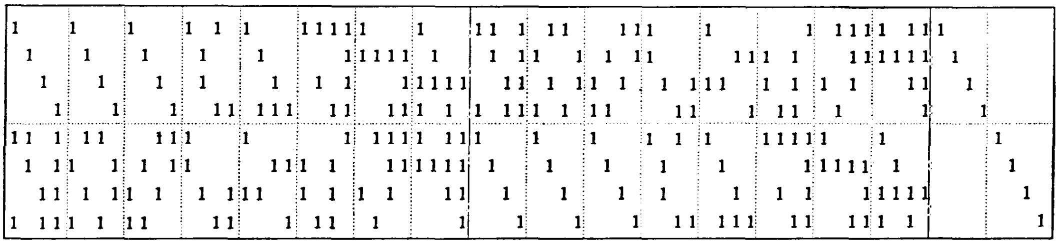

図-5 (72, 64) SEC-DED-S4ED 符号

表-2 DEC 符号と検査ビット数

符号		$k=32$	$k=64$	$k=128$
DEC-BCH 符号		12	14	16
中間的符号		14～16	15～19	19～23
多数決復号可能な符号	1段直交可能符号	16～19	22～25	31～37
	MA・符号	23	31	40
	直交ラテン方陣符号	24	32	48

(注) k: 情報ビット数 (ビット)

号の中間的符号の提案もある．これらの符号の検査ビット数を**表-2** に示す．復号回路のゲート量は，BCH 符号を用いた場合で，SEC-DED 符号に比較して3倍程度となり，多数決復号可能な符号の場合には，1～1.5 倍程度となる．

（3） バイト誤り検出符号 (SEC-DED-SbED 符号)

本符号は，SEC-DED の機能を有し，さらに単一バイト誤り検出 (SbED) の機能を有する符号である．本符号は，SEC-DED 符号と比較して検査ビット数，符号化・復号回路ゲート量が同程度で実現できる点で実用的である．

本符号に対する理論的構成では，バイト長 $b=4$ ビット，情報ビット数 32, 64, 128 ビットに対し，検査ビット数は SEC-DED 符号と比較して1ビットずつ多くなる[12]．しかし，計算機を用いて符号を探索する手法を用いると，同一検査ビット数で構成できる符号が得られる[13]．**図-5** はこの手法を用いて構成した (72, 64) SEC-DED-S4ED 符号の例であり，くり返し性2を有する．

SEC-DED-SbED 符号では，固定故障が存在するとき，そのバイト位置を示すポインタ* を与えれば，単一バイト誤り訂正が可能となる[12]．

（4） バイト誤り訂正符号 (SbEC, SbEC-DbED 符号)

* 固定故障の位置をあらかじめ調べておき，その位置を示す情報をポインタとして記憶しておく．

バイト出力記憶素子を使用した装置の高信頼化には，ここで示すバイト誤り訂正符号が最適である．しかし，必要とする検査ビット数は，バイト誤り検出符号と比較して多くなる．

b ビットの単一バイト誤り訂正の機能を有する SbEC 符号の H マトリクスは，ガロア体 GF(2^b) 上の元を要素として互いに線形独立な列ベクトルにより構成することができる．GF (2^b) 上の元を GF (2) 上で表現するため，{0, 1} で表現した b 次の正則行列 T を定義し，T のべき乗の要素で GF(2^b) 上のすべての元を表現する[14]．

Hマトリクスの各列ベクトルの要素の和 (GF(2^b) 上の和) を単位元 I に等しくさせた奇数重み列 SbEC 符号が提案されている[15]．この符号は，GF (2) 上で表現したHマトリクスの列ベクトルが常に奇数重みとなり，しかも2バイト間誤り検出能力が高い特徴を有する*．

単一バイト誤り訂正だけでなく，さらに完全な2バイト誤り検出を行うためには SbEC-DbED 符号を使用する必要がある．この符号では，GF(2^b) 上の元で表わしたHマトリクスの任意の3個以下の列ベクトルが線形独立でなければならない．任意のバイト長，符号長に対して能率の良い符号構成が提案されている[16]．**図-6** に，$b=4$ ビット，情報ビット数 128 ビットに対する符号例を示す．この符号は，くり返し性4を有し，しかもHマトリクス中の '1' の数を最少にした符号の一つである．なお，最近，一部の商用機の記憶装置に (80, 64) S4EC-D4ED 符号が導入された報告がある[17]．

以上のバイト誤り検出・訂正符号を中心に主要な符号に対する検査ビット数を**表-3** に，また復号回路ゲート量，復号遅延の関係を**図-7** に示す．

* SbEC 符号では，同時に2バイト誤りを完全に検出する能力は有さない．この符号では，同一検査ビット数のもとで2バイト間誤り検出能力の高い符号が優れた符号である．

$I\ I\ I\ I\ I\ I\ I\ I$		$T^3\ T^2\ T^1\ I\ T^{14}\ T^{13}\ T^{12}\ T^{11}$	$T^{11}\ T^{12}\ T^{13}\ T^{14}\ I\ T^1\ T^2\ T^3$	I
$T^{11}\ T^{12}\ T^{13}\ T^{14}\ I\ T^1\ T^2\ T^3$	$I\ I\ I\ I\ I\ I\ I\ I$		$T^3\ T^2\ T^1\ I\ T^{14}\ T^{13}\ T^{12}\ T^{11}$	I
$T^3\ T^2\ T^1\ I\ T^{14}\ T^{13}\ T^{12}\ T^{11}$	$T^{11}\ T^{12}\ T^{13}\ T^{14}\ I\ T^1\ T^2\ T^3$	$I\ I\ I\ I\ I\ I\ I\ I$		I
	$T^3\ T^2\ T^1\ I\ T^{14}\ T^{13}\ T^{12}\ T^{11}$	$T^{11}\ T^{12}\ T^{13}\ T^{14}\ I\ T^1\ T^2\ T^3$	$I\ I\ I\ I\ I\ I\ I\ I$	I

$$\left(T=\begin{bmatrix}0001\\1001\\0100\\0010\end{bmatrix},\ I=\begin{bmatrix}1000\\0100\\0010\\0001\end{bmatrix},\ 空白=\begin{bmatrix}0000\\0000\\0000\\0000\end{bmatrix}\right)\quad (b=4)$$

図-6 (144, 128) S4EC-D4ED 符号

表-3 代表的な主記憶用符号と検査ビット数

符号		検査ビット数（ビット） $k=32$	$k=64$	$k=128$	記事
1ビット誤り訂正・2ビット誤り検出符号 (SEC-DED 符号)		7	8	9	1ビット出力記憶素子用符号
1ビット誤り訂正・2ビット誤り検出 単一バイト誤り検出符号 (SEC-DED-SbED 符号)	$b=4$	7	8	9	複数ビット出力素子用符号．検査ビット数が小さく，経済性にすぐれている．ただし，2ビット以上のバイト誤りは訂正できない．
	$=8$	10	11	12	
単一バイト誤り訂正符号 (SbEC 符号)	$b=2$	8	8	10	複数ビット出力素子用符号．検査ビット数は奇数重み列 SbEC 符号のものを示した．
	$=4$	8	12	12	
	$=8$	16	16	16	
単一バイト誤り訂正・二重バイト誤り検出符号 (SbEC-DbED 符号)	$b=2$	10	12	12	複数ビット出力素子用符号．2個の記憶素子故障を完全に検出することが要求される場合に適する．
	$=4$	12	16	16	
	$=8$	24	24	24	

(注) b: バイト長（ビット），k: 情報ビット数（ビット）．

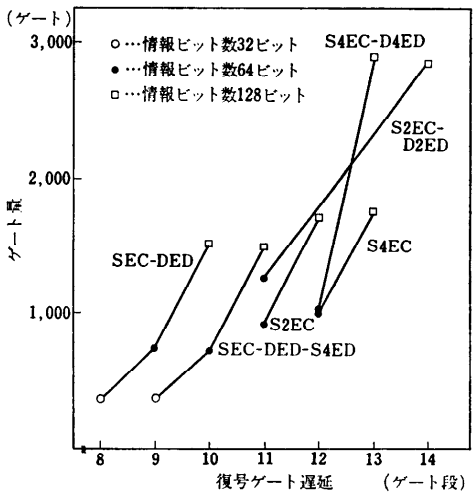

図-7 復号回路遅延とゲート量*

3.2 ファイル記憶装置**への応用

ファイル記憶装置における誤りは，媒体の欠陥，ゴミ等によって，バースト状に発生することが多い．そのため，ファイル記憶装置にはバースト誤りを検出・訂正する符号が必要となる．ファイル記憶装置用符号では，主記憶装置の場合のような高速な符号化・復号は必ずしも必要なく，巡回シフトレジスタ等による逐次的な符号化・復号を行うことが多い．

（1） 磁気テープ記憶装置

従来，磁気テープ記憶装置では，トラックに並行した方向に巡回符号[1]を，トラックと垂直な方向にパリティ検査符号を組み合わせてバースト状の誤り検出・訂正を行う手法を採用していた．近年における 6250 BPI (Bit Per Inch) 磁気テープ装置では，$b=8$ ビットの SbEC 符号（符号長 72 ビット，情報ビット数 56 ビット）を採用している[18]．符号語は，9個のバイトに分けて，それぞれ9個のトラックに並列に記録する．こうすれば，9トラック中の1トラックに生じたバースト誤りは，各々の符号語にとっては単一バイト誤りとなるため，SbEC 符号で訂正可能となる．さらに，読出し信号波形等から誤りの生じているトラックを示すエラーポインタを作成し，このポインタを使用することにより，最大2トラックまでの誤りを訂正すること

* 4入力 OR, NOR ゲート換算．2入力 EX-OR ゲートは，2ゲート，1ゲート段に換算．

** 本文では，磁気テープ記憶装置，磁気ディスク記憶装置等の磁気記録媒体を有する補助記憶装置（二次記憶装置）をファイル記憶装置と呼ぶ．

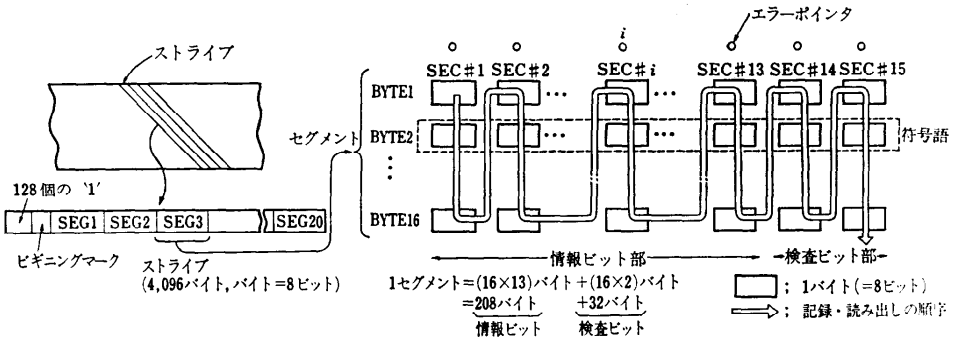

図-8 超大容量記憶装置 (MSS) の誤り検出・訂正方式

ができる.

(2) 超大容量記憶装置 (MSS)

超大容量記憶装置 (MSS) では，符号として $b=8$ ビットの $SbEC$ 符号（符号長 120 ビット，情報ビット数 104 ビット）を用い，さらに図-8 に示すように，16 個の符号語をトラック方向にインタリーブ* している[19]．このため，同一符号語に属する情報は 16 バイトごとに分散することになり，最大 16 バイトまでのバースト誤りを訂正できることになる．また，エラーポインタを使用することにより，240 バイト中，最大 32 バイトまでの誤りを訂正する能力を有する．

以上述べたように，磁気テープ媒体を用いた記憶装置では，$SbEC$ 符号を採用し，これにエラーポインタやインタリーブを併用して訂正能力を強化する手法が主流となっている．研究レベルでは，さらに多重のバイト誤り訂正を行う符号の適用や，算術符号の適用等が提案されている[20]．

(3) 磁気ディスク記憶装置

磁気ディスク記憶装置では，バースト誤り検出・訂正符号の一種である Fire 符号[21]を主として用いている．磁気ディスク用符号の詳細は一般に公表されないことが多いが，100 MB（メガバイト）磁気ディスク記憶装置では媒体互換を取る必要上，その生成多項式は知られており，次のような多項式となっている．

$$g(x)=(x^{22}+1)(x^{11}+x^7+x^6+x+1)(x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1)(x^{11}+x^9+x^7+x^6+x^5+x+1)$$

この符号は，検査ビット数 56 ビットで，22 ビットまでのバースト誤りを検出し，11 ビットまでのバースト誤りを訂正する．

さらに高記録密度化を図った装置では，媒体欠陥回避手法（ディフェクト・スキップ）を採用することにより，前述の装置より誤り検出・訂正能力を低下させた Fire 符号を用いている．

最近，従来の Fire 符号に代って，インタリーブ手法を併用したバイト誤り訂正符号の適用が提案されている[22),24)]．

4. あとがき

誤り検出・訂正符号の計算機システムへの応用として，論理回路，記憶装置への応用についてのべた．

計算機システムへの高信頼化要求が高まる中で，誤り検出・訂正符号の重要性は増々大きくなるものと考えられる．特に，近年における LSI 技術の進展は，符号応用における経済的負担を相対的に小さくしており，適用の容易性が増大しているといえよう．一方，最近，符号は高信頼化のためだけでなく，その符号間距離の概念を利用して，テスト用検査パターン，連想処理[23]等への応用が提案されており，この点からも今後の幅広い応用が期待される．

参考文献

1) Peterson, W. W. and Weldon, E. J. Jr.: Error-Correcting Codes, MIT Press, Massachusetts (1972).
2) Rao, T. R. N.: Error Coding for Arithmetic Processors, Academic Press, New York (1974).
3) Wakerly, J.: Error Detecting Codes, Self-Checking Circuits and Applications, North-Holland, New York (1978).
4) Sellers, Jr. F. F., Hsiao, M. Y. and Bearnson, L. W.: Error Detecting Logic for Digital Computers, McGraw-Hill (1968).

* 複数個の符号語に対し，まず各符号語の 1 バイト目を記録し，次に各符号語の 2 バイト目を記録し，…と続け，一つの符号語をはなれた位置に分散させる手法をインタリーブという．

5) Neumann, P. G. and Rao, T. R. N.: Error Correcting Codes for Byte-organized Arithmetic Processors, IEEE Trans. Comput. Vol. C-24, No. 3, pp. 226-232 (1975).
6) Rao, T. R. N. and Reinheimer, H. J.: Fault-Tolerant Modularized Arithmetic Logic Units, National Computer Conference, pp. 703-710 (1977).
7) 武藤, 池田：フォールトトレラントゲートの提案, 情報処理, Vol. 21, No. 5, pp. 445-454 (1980).
8) Fujiwara, E. and Haruta, K.: Fault-Tolerant Arithmetic Logic Unit Using Parity-Based Codes, Trans. IECE Japan, Vol. E 64, No. 10, pp. 653-660 (1981).
9) 藤原, 金田：主記憶用誤り検出・訂正符号, 研究実用化報告, Vol. 30, No. 4, pp. 151-180 (1981).
10) Hsiao, M. Y.: A Class of Optimal Minimum Odd-Weight-Column SEC-DED Codes, IBM J. Res. Develop., Vol. 14, No. 4, pp. 374-401 (1970).
11) 今井：情報伝達の符号化と処理B, 昭和55年度電気四学会連合大会, 32-6 (1980).
12) Fujiwara, E.: Error Control for Byte-per-Package Organized Memory System, Trans. IECE Japan, Vol. E 63, No. 2, pp. 98-103(1980).
13) 金田：符号化データの誤り検出訂正方式, 特願昭 56-176686 (1981).
14) Bossen, D. C.: *b*-Adjacent Error Correction, IBM J. Res. Develop., Vol. 14, No. 4, pp. 402-408 (1970).
15) Fujiwara, E.: Odd-Weight-Column *b*-Adjacent Error Correcting Codes, Trans. IECE Japan, Vol. E 61, No. 10, pp. 781-787 (1978).
16) Kaneda, S. and Fujiwara, E.: Single Byte Error Correcting-Double Byte Error Detecting Codes for Memory Systems, FTCS-10, pp. 41-46 (1981).
17) 平栗, 上野, 槌本：3階層メモリ方式や高密度化技術により性能向上を図った大型コンピュータシステム M-380/382, 日経エレクトロニクス, No. 276, pp. 176-199 (1981).
18) Patel, A. M. and Hong, S. J.: Optimal Rectangular Code for High Density Magnetic Tapes, IBM J. Res. Develop., Vol. 18, No. 6, pp. 579-588 (1974).
19) Patel, A. M.: Error Recovery Scheme for the IBM 3850 Mass Storage System, IBM J. Res. Develop., Vol. 24, No. 1, pp. 32-42 (1980).
20) Goto, M.: Rates of Unidirectional 2-Column Errors Detectable by Arithmetic Codes, FTCS-10, pp. 21-25 (1980).
21) Chien, R. T.: Burst-Correcting Codes with High-Speed Decoding, IEEE Trans. Information Theory, Vol. IT-15, No. 1, pp. 109-113 (1969).
22) ホッジス, シューブル, シェイファー：誤り訂正検査バイト生成装置, 特開昭 55-9296 (1980).
23) 市川, 坂村, 諸隈, 相磯：連想アレイプロセッサ ARES, 信学論 (D), Vol. 61-D, No. 10, pp. 743-750 (1978).
24) Hsiao, M. Y., Carter, W. C., Thomas, J. W. and Stringfellow, W. R.: Reliability, Availability and Serviceability of IBM Computer Systems: A Quarter Century of Progress, IBM J. Res. Develop., Vol. 25, No. 5, pp. 453-465 (1981).
25) Bossen, D. C. and Hsiao, M. Y.: A System Solution to the Memory Soft Error Problem, IBM J. Res. Develop., Vol. 24, No. 3, pp. 390-397 (1980).

(昭和56年12月2日受付)