

光ディスク装置用誤り訂正の方式およびLSIの開発

加藤 修 黒沢勝広 岩国 薫 新保正利
松下通信工業(株) 開発研究所 松下電器産業(株) 音響研究所

光ディスク装置用誤り訂正の訂正方式および訂正用LSIを開発した。誤り訂正符号はRS符号(Reed-Solomon符号)で、104バイトの情報シンボルに16バイトの検査シンボルを付加しており最大8重誤り訂正が可能である。訂正方式は、4重誤り以下を今回開発したLSIで実時間訂正し、5重誤り以上は外部の汎用プロセッサにより非実時間訂正を行うという方式である。誤りバイト数による処理の場合分けを行うためには、誤りバイト数を判定する必要がある。また4重誤り訂正としても、実時間訂正を行うためには復号アルゴリズムの高速化が必要である。本報告では、誤りバイト数の判定方法および4重誤り実時間訂正を可能とするための復号アルゴリズムの高速化を中心に報告する。

Development of Error Correction Method and LSI for Optical Disk Data Storage System

O. Kato K. Kurosawa K. Iwakuni M. Shimbo
Advanced Development Lab. Matsushita Acoustic Research Lab. Matsushita Electric Ind.
Communication Ind. 600 Saedo, Midori-ku, 1006 Kadoma, Kadoma-shi
Yokohama-shi

Error correction method and LSI for optical disk data storage system were developed. The error correction code is Reed-Solomon long distance code which consists of 104 data bytes and 16 redundancy bytes, and octuple or less error is correctable. In designing stage of LSI, we have developed practical error correction method in which quadruple or less error is corrected in realtime by the newly developed LSI and quintuple or more error is corrected in non-realtime by external processor.

This paper mainly describes the estimation of the number of error bytes and the high-speed decoding algorithm which enables quadruple realtime error correction.

1. はじめに

光ディスクは磁気ディスクに比べ、ビット密度が10倍程度高く低ビット価格であり、かつ媒体可換であることから、ファイル記憶の主力媒体としてその発展が待望されている。しかし、光ディスクでは、媒体ビットエラー率が 10^{-5} オーダーと悪いために、誤り訂正技術の適用が不可欠である。

今回、我々は、5.25インチ光ディスク装置の誤り訂正符号の主流である、RS符号(Reed-Solomon符号)のLDC(Long-Distance-Code)フォーマットに対する訂正用LSIを開発したので、ここに報告する。この符号は符号長が120バイトと長く、そのうち16バイトが検査シンボルであり、8バイト以下の誤り訂正が可能である。しかし、8バイトまでの誤り訂正を実行するLSIの実現は、回路規模および実時間訂正の可能性の点で非常に困難である。そこで我々は、誤り訂正方式として、1符号語に含まれる誤りバイト数を判定し、4重誤り以下の場合にはこれを専用の誤り訂正用LSIで実時間訂正し、5重誤り以上の場合には外部の汎用プロセッサを用いて非実時間訂正を行うという方式を考案した。

本報告では、誤りバイト数の判定方法および4重誤り実時間訂正を実現するための復号アルゴリズムの高速化を中心に報告する。

2. 誤り訂正符号とフォーマット

図1に、今回開発した誤り訂正用LSIが適用される誤り訂正のフォーマットを示す。誤り訂正符号はRS符号であり、1符号語は、104バイトの情報シンボルに対して16バイトの検査シンボルを付加した120バイトより成る。検査シンボルを生成する生成多項式 $G(X)$ は式(1)で与えられる。但し β は、有限体 $GF(2^8)$ 上の原始元であり、有限体上の演算を決定する法多項式 $m(X)$ は式(2)で与えられる¹⁾。

$$G(X) = \prod_{i=120}^{135} (X + \beta^{88i}) \quad (1)$$

$$m(X) = X^8 + X^5 + X^3 + X^2 + 1 \quad (2)$$

本誤り訂正符号は最小距離が17であり、120バイト中8バイト以下の誤りを訂正できる。また、図1に示すように、媒体上の連続シンボルが、異なる符号語に含まれるよう、インタリーブを行っている。

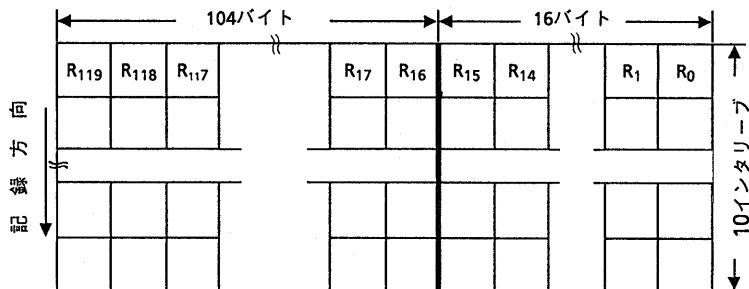


図1. 誤り訂正のフォーマット

3. 復号アルゴリズムとLSIの構成

RS符号の復号は、次の5つのステップに分けられる。

- (1) シンドロームの計算
- (2) 誤り位置多項式 $\sigma(X)$ と誤り数値多項式 $\eta(X)$ の導出
- (3) 誤り位置の求解($\sigma(X)=0$ の解を求める。)
- (4) 誤りパターンの計算
- (5) 誤り訂正の実行

上記5ステップのうち、(2)が最も計算時間の長くなるステップである。

(2)を行うアルゴリズムとしては、ユークリッドアルゴリズム等が知られている²⁾。(2)の処理の高速化が、復号処理全体の高速化において重要な点であるが、後述する高速化ユークリッドアルゴリズムによっても(2)の演算量は、 t を誤り訂正数として $8t^2+6t$ 回であり、 $t=8$ に対して実時間訂正を行うためには、(2)の演算を実行する有限体演算プロセッサを複数個もつLSI構成が必要で、推定40000~50000ゲートの回路規模となる。一方、図2に示すように、誤りバイト数が大きくなる程その発生確率は小さくなるので、例えば5重誤り以上は非実時間訂正としても全体に占める非実時間訂正の割合は非常に小さい。

これらの理由により、誤り訂正用LSIとしては4重誤り実時間訂正を行うものを開発し、5重誤り以上は外部の汎用プロセッサによる非実時間訂正を行うという方式を採用した。そのため上記復号ステップの(1)と(2)の間に、(1.5)誤りバイト数の判定というステップが追加となる。

図3にLSIのハード構成を示す。上記6ステップは、(1)がシンドロームジェネレータで、(1.5)、(2)および(4)が有限体演算プロセッサで、(3)が誤り位置求解回路で、(5)が訂正ユニットによって処理が実行される。

復号処理のシーケンスは、図4に示すように、10インタリーブを単位としたパイプライン処理方法として、処理の高速化を図った。

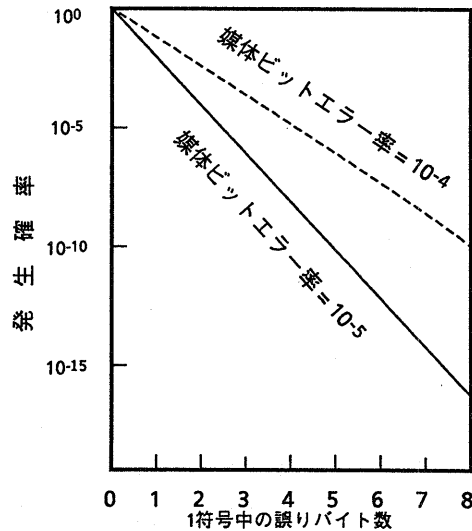


図2. 誤りバイト数 対 発生確率

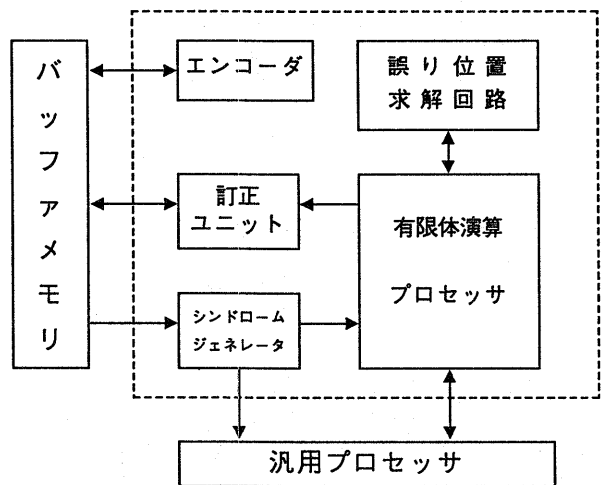


図3. LSIハード構成図

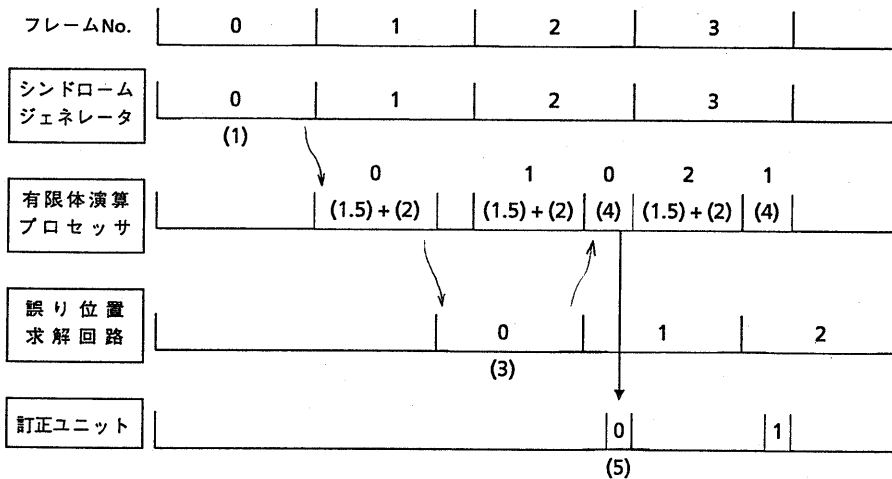


図4. 復号処理のシーケンス (()内は復号ステップ番号)

4. 誤りバイト数の判定方法

4-1 誤りバイト数判定行列式

式(3)に、1符号語に含まれる誤りバイト数が4バイト以下か5バイト以上かを判定するための行列式を示す。この行列式Sは誤りバイト数が4バイト以下ならば $S=0$ 、5バイトならば $S \neq 0$ という特徴を持つ。問題は誤りバイト数が6バイト以上の場合で、理論計算およびシミュレーションにより、1/255の確率で $S=0$ となることがわかった。このため $S=0$ の場合に誤りバイト数Kが $K \leq 4$ と判定すると、 $K \geq 6$ に対して1/255の確率で誤判定することになる。

$$S = \begin{vmatrix} S_0 & S_1 & S_2 & S_3 & S_4 \\ S_1 & S_2 & S_3 & S_4 & S_5 \\ S_2 & S_3 & S_4 & S_5 & S_6 \\ S_3 & S_4 & S_5 & S_6 & S_7 \\ S_4 & S_5 & S_6 & S_7 & S_8 \end{vmatrix} \quad (3)$$

但し式(3)の行列式の要素は、式(4)で与えられるシンδροームである。

$$S_j = \sum_{i=0}^{119} R_i \cdot \beta^{88(j+120)i} \quad (4)$$

4-2 誤りバイト数誤判定の問題点

誤りバイト数Kが $K \geq 6$ に対して、 $K \leq 4$ と誤判定した場合、その符号語に対しては、4重誤り訂正処理が行われる。したがって $6 \leq K \leq 8$ の場合には、本符号の最大訂正バイト数以下であるにもかかわらず、誤った訂正が行われることになる。図2より、6重誤りが発生する確率は媒体ビットエラー率を 10^{-4} と悪く見積ったとすると、 10^{-7} 程度であるので、 $1/255 \times 10^{-7} \approx 4 \times 10^{-10}$ の確率で誤訂正が行われることになり、誤り訂正後のビットエラー率の仕様値である 10^{-12} を満足できない。そこで、4-3.に述べる対策を行った。

4-3. 誤判定確率の低減方法

誤りバイト数の誤判定確率を低減して上記問題点を解決するために、次の2つの対策を行った。

(i) 誤りバイト数を判定するのに、式(3)のSと式(5)のTの2種類の行列式を用いて、 $S=0$ かつ $T=0$ の場合にのみ $K \leq 4$ と判断することにする。理論計算およびシミュレーションにより、SとTの値は独立であり、 $K \geq 6$ に対して $S=0$ かつ $T=0$ となる確率は $(1/255)^2 \approx 1.5 \times 10^{-5}$ であることが確かめられた。

$$T = \begin{vmatrix} S_7 & S_8 & S_9 & S_{10} & S_{11} \\ S_8 & S_9 & S_{10} & S_{11} & S_{12} \\ S_9 & S_{10} & S_{11} & S_{12} & S_{13} \\ S_{10} & S_{11} & S_{12} & S_{13} & S_{14} \\ S_{11} & S_{12} & S_{13} & S_{14} & S_{15} \end{vmatrix} \quad (5)$$

(ii) $K \leq 4$ と判定された場合、その符号語はLSIによる4重誤り訂正処理が行われるが、実際の誤りバイト数が $K \geq 5$ の場合に対しては、その訂正処理の過程で得られる誤り位置多項式 $\sigma(X)$ の次数と $\sigma(X)=0$ の解の個数が、約 $1-1.5 \times 10^{-3} = 0.9985$ の確率で一致しないことが、シミュレーションの結果わかった。 $K \geq 5$ に対して、 $\sigma(X)$ の次数と $\sigma(X)=0$ の解の個数が一致するのは、その $K \geq 5$ の符号語に対するシンドローム $S_0 \sim S_7$ (4重誤り訂正処理では $S_0 \sim S_7$ の8つのシンドロームのみを使用する。)の値が、ある $K \leq 4$ の符号語に対する $S_0 \sim S_7$ と一致してしまう場合であり、その確率が 1.5×10^{-3} である。この性質を利用して、4重誤り訂正処理の過程で、 $\sigma(X)$ の次数と $\sigma(X)=0$ の解の個数が一致しないことがわかった段階で、LSIによる誤り訂正処理は中断して、外部の汎用プロセッサを用いた8重誤り非実時間訂正処理を開始するようにする。

(i)および(ii)の対策により、本来訂正可能な $6 \leq K \leq 8$ の誤りを含む符号語が、LSIによる4重誤り訂正処理によって誤訂正となってしまう確率は、 $(1/255)^2 \times 1.5 \times 10^{-3} \times 10^{-7} \approx 2.3 \times 10^{-15}$ となり、誤り訂正後のビットエラー率の仕様値である 10^{-12} よりも十分小さくすることができる。

5. 復号アルゴリズムの高速化

光ディスクの回転数は2400rpmであるが、将来3000rpmまで高速化されても、本LSIが適用できるようにするためには、1セクタ(図1に示す記憶単位)当たりの復号時間を1.1ms以内とする必要がある。復号処理のシーケンスは図4に示すようなパイプライン処理方式としたが、図4で1フレームの時間は93 μ sであり、LSIの動作クロック周波数は4.5MHzとしたので、1フレームは420クロック(1クロックが、有限体演算プロセッサの1命令実行に相当する。)となる。この復号処理シーケンスにおいて、(1.5)、(2)および(4)の処理を1フレーム内に収めることが、処理時間的に最も困難なところである。(1.5)、(2)および(4)の処理を実行する有限体演算プロセッサの構成を図5に示す。処理の高速化を図るために、次のようにした。

(i) 有限体演算プロセッサの演算処理部に、有限体上の演算をハード的に実行する有限体加算器および有限体乗算器を設ける。

(ii) 1命令で多くの演算操作(例えばRAMに格納された2つの有限体の積をとり、その積にレジスタにラッチされていた有限体を加算する。)が行えるように命令体系を強化する。

(iii) (2)の処理($\sigma(X)$ および $\eta(X)$ の導出)のアルゴリズムを改善し、演算量を低減する。

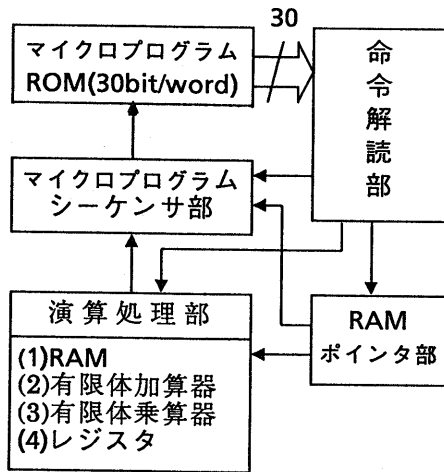


図5.有限体演算プロセッサの構成図

(iii)について詳述する。 $\sigma(X)$ および $\eta(X)$ の導出のアルゴリズムには、ユークリッドの互除法を利用するユークリッドアルゴリズムが知られている。さらにユークリッドアルゴリズムにおいて、繰り返して計算を行う必要のある多項式間の除算 $(A(X) \div B(X))$ の商および剰余を求める。)を、 $A(X)$ 、 $B(X)$ にそれぞれ $b(B(X))$ の最大次係数および $a(A(X))$ の最大次係数)を乗じて加算することに置換えたアルゴリズムが文献2)に報告されている。図6にそのアルゴリズムを示す。本LSIでは、有限体演算プロセッサの構成が比較的容易となる文献2)のアルゴリズムを採用し、さらに演算量を低減し復号の高速化を図るため、次の2点を改善した。

- ① 図7に示すように、 a/b および b/a を一方の多項式のみに乗じて加算するようにする。このようにしても、最終的に得られる $\sigma(X)$ および $\eta(X)$ は同じであることを証明した。(証明は省略)
- ② $A(X) \leftarrow A(X) + a/b \cdot B(X) \cdot X^{\deg A(X) - \deg B(X)}$ の計算は $A(X)$ および $B(X)$ の i 次の係数を A_i 、 B_i として

$$\begin{array}{rcl}
 A_8 & \leftarrow A_8 & + a/b \cdot B_8 + \deg B - \deg A \\
 & \vdots & \\
 A_{\deg A + 1} & \leftarrow A_{\deg A + 1} & + a/b \cdot B_{\deg B + 1} \\
 A_{\deg A} & \leftarrow A_{\deg A} & + a/b \cdot B_{\deg B} \\
 & \vdots & \\
 A_{\deg A - \deg B} & \leftarrow A_{\deg A - \deg B} & + a/b \cdot B_0
 \end{array}$$

のように、同じ次数の係数間の加算によって行われる。しかしこれらの $9 - \deg A + \deg B$ 個の式のうち、上の $8 - \deg A$ 個の式は $0 + 0$ の計算である。そこで、 A_i 、 B_i が格納されているRAMの、 $A_{\deg A}$ および $B_{\deg B}$ が格納されている番地から、係数間の加算を開始することで、無駄な $0 + 0$ の計算を省略した。

$B(X) \leftarrow B(X) + b/a \cdot A(X) \cdot X^{\deg B(X) - \deg A(X)}$ の計算も同様である。

$L(X) \leftarrow L(X) + a/b \cdot M(X) \cdot X^{\deg A(X) - \deg B(X)}$ の計算は $L(X)$ および $M(X)$ の i 次の係数を L_i, M_i として

L_8	$\leftarrow L_8$	$+ a/b \cdot M_8 + \deg B - \deg A$
L_7	$\leftarrow L_7$	$+ a/b \cdot M_7 + \deg B - \deg A$
\vdots	\vdots	\vdots
$L_{\deg M + \deg A - \deg B}$	$\leftarrow L_{\deg M + \deg A - \deg B}$	$+ a/b \cdot M_{\deg M}$
\vdots	\vdots	\vdots
$L_{\deg A - \deg B}$	$\leftarrow L_{\deg A - \deg B}$	$+ a/b \cdot M_0$

のように、同じ次数の係数間の加算によって行われる。しかしこれらの $8 - \deg A + \deg B$ 個の式のうち、 $p = \text{MAX}(\deg M + \deg A - \deg B, \deg L)$ として、上の $8 - p$ 個の式は $0 + 0$ の計算である。

したがって、 $A(X)$ の変換計算の場合と同様に L_p および $M_{p + \deg B - \deg A}$ が格納されている番地から、係数間の加算を開始することで、 $0 + 0$ の計算を省略することができる。しかしそのためには、 $\deg L, \deg M$ および p を求める必要があり、かえって演算量が増えてしまう。そこで、ユークリッドアルゴリズムの過程における $\deg A, \deg B, \deg L, \deg M$ 間の関係を検討し、式(6)および式(7)の関係式を見出した。

$\deg A \geq \deg B$ ならば

$$\text{MAX}(\deg M + \deg A - \deg B, \deg L) \leq 8 - \deg B \tag{6}$$

$\deg A < \deg B$ ならば

$$\text{MAX}(\deg L + \deg B - \deg A, \deg M) \leq 8 - \deg A \tag{7}$$

式(6)を用いて $L_{8 - \deg B}$ および $M_{8 - \deg A}$ が格納されている番地から、係数間の加算を開始することで、 $\deg L, \deg M$ および p を求めることなく、 $0 + 0$ の計算をほとんど省略することができる。

$M(X) \leftarrow M(X) + b/a \cdot L(X) \cdot X^{\deg B(X) - \deg A(X)}$ の計算も同様である。(式(7)を適用)

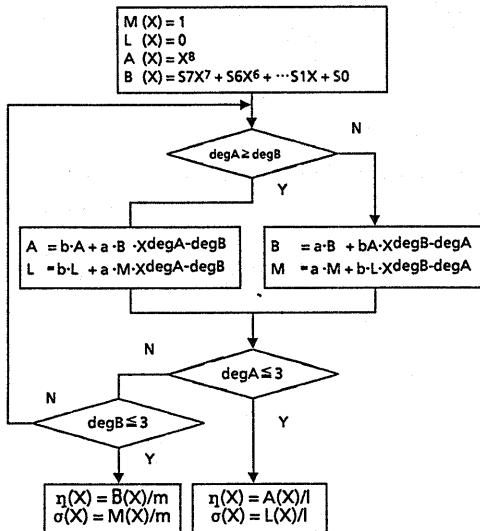


図6. 文献2)のユークリッドアルゴリズム

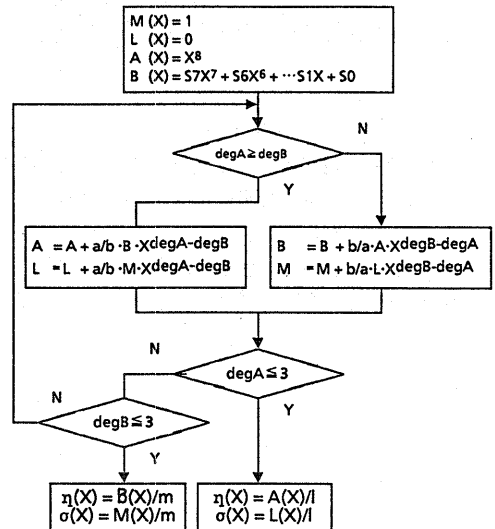


図7. 高速化ユークリッドアルゴリズム

以上の①および②により、ユークリッドアルゴリズムでの最大演算量は、 t を最大誤り訂正数として、 $18t^2 + 6t$ 回から $8t^2 + 6t$ 回に低減された。今回の場合は $t=4$ なので、312回から152回への低減となる。

(i)~(iii)の高速化により、有限体演算プロセッサによる復号処理を、表1に示すように、合計で420命令以内とすることができ、4重誤り実時間訂正が可能となった。

表1. マイクロプログラムステップ数

処理名	最長ステップ数
(1.5) 誤りバイト数の判定	118
(2) $\sigma(X), \eta(X)$ の導出	211
(4) 誤りパターンの計算	65
合計	394

表2. LSI諸元

訂正能力	4重誤り以下を実時間訂正 5重誤り以上は外部マイコンに 訂正処理を依頼
プロセス	CMOSスタンダードセル
動作クロック	4.5MHz
パッケージ	120ピンフラットパッケージ
ゲート数	18000ゲート
消費電力	1700 mW max

6. むすび

誤りバイト数を判定して、 $K \leq 4$ の場合はLSIで4重誤り実時間訂正処理を、 $K \geq 5$ の場合には汎用プロセッサで8重誤り非実時間訂正処理を行う方式、およびLSIを開発した。表2にLSIの諸元を示す。

謝辞 本研究開発で御指導頂いた、松下通信工業(株)開発研究所・利光所長、渡辺部長、松下電器産業(株)音響研究所・山元部長、菊池室長に深謝致します。

7. 参考文献

1)ANSI X3B11-86-153

ENCODING-SPECIFICATION FOR X3B11 APPROVED EDAC CODE FOR $5\frac{1}{4}$ INCH OPTICAL STORAGE DEVICES, sbmitted to X3B11, 4-Nov.-86 NEAL GLOVER, DATA SYSTEMS TECHNOLOGY CORP.

2)木村、今井、土肥:シストリックアルゴリズムに基づくReed-Solomon符号の復号器の構成法、信学技報 IT84-43(1985)