# 誤り訂正符号のVLSIプロセッサ上での機構化および VLSI検査への応用に関する研究

1988年1月

岩 崎 一 彦

# 内容梗概

本論文は，修正ハミング符号およびファイヤ符号のＶＬＳＩプロセッサ上での機構化，ならびに，ハミング符号およびリード・ソロモン符号のＶＬＳＩ検査法への応用に関するものである。本論文は７章から構成されている。各章の概要を以下に示す。

第１章では，本研究の一般的背景，研究経過を述べる。

第２章では，修正ハミング符号の２つの構成法について論じる。１つは，非零要素の総数が最小であるようなパリティ検査行列を持つ修正ハミング符号のうちで，３重誤りの訂正確率，４重誤りの見逃し確率が最小の符号に関する構成法である。特に実用性の高い（72,64）修正ハミング符号に対して，計算機を用いて，最適な符号を求めた。計算にあたっては，パリティ検査行列中でより重みの小さい列ベクトルが固定できることを利用し，調べるべき場合の数を減らした。

また，修正ハミング符号のもう一つの構成法としてディジタル化されたデータなどの上位ビットの誤り検出率を高くするような符号を構成する。ここでは，パリティ検査行列の性質を利用して，ビット位置毎の誤り検出率を定式化する。いくつかの符号長に対し，計算機を用いて，良好な符号を見いだした。また，特定ビットの誤り検出率と符号全体の誤り検出率はトレード・オフの関係にあることを示した。

第３章では，ＶＬＳＩプロセッサに適したファイヤ符号およびバースト誤り訂正巡回符号の復号法を提案する。提案する復号法は，ハードウェアが比較的少なくてよい。また，短縮化前の符号長がレコード長よりも十分に長い ―― 通常の応用では成り立つ ―― 場合，復号時間の面でも，他の高速復号法と同じオーダまで高速化することができる。また，一部の設定値を変更することにより，種々の短縮化に対して使用できるという特徴も持つ。この性質は，マイコン周辺ＬＳＩのように１個のチップで複数のレコード長を扱う場合に適しており，ＲＯＭなどの規則論理素子で実現される。

さらに，提案した復号法を具体的にＶＬＳＩプロセッサへ適用した。適用した符号は32ビットファイヤ符号であり，適用したプロセッサはハードディスクコントローラＶＬＳＩである。このＶＬＳＩチップは，16ビットマイコン周辺ＬＳＩとして開発されたものである。提案した復号法を実行する誤り訂正部は，16ビットＣＲＣ機能および誤り訂正用マイクロプログラムを含め，約6700トランジスタとなった。これは，ハードディスクコントローラＶＬＳＩの約５％を占める。また，シリアルインターフェイスＶＬＳＩの設計法につ

いても述べる。

第4章はＬＳＩ検査，特にシグナチャ検査法に関するものである。まず，多入力シグナチャレジスタ（ＭＩＳＲ）に対してシグナチャＳ（ｘ）を定式化する。次に，G等価という考え方を提案し，原始多項式に基づくＭＩＳＲの2重ビット誤りの検出率を求める。その結果，2重ビット誤り検出率が100％とならないことを示す。さらに，行列の縮退化という考え方を提案し，この考え方とハミング符号の重み3，4の符号語数から，ＭＩＳＲの3，4重ビット誤り検出率を計算する。同時に,拡大ハミング符号生成多項式に基づくＭＩＳＲの1～4重ビット誤り検出率を求め，2重ビット誤り検出率が100％とならないことを示す。

2重ビット誤り検出率を100％にする一つの方法として，逆2重化ＭＩＳＲというシグナチャ回路を提案する。この回路は，シフト方向が互いに逆方向である2組のＭＩＳＲを用いる方法である。特に，拡大ハミング符号生成多項式に基づく逆2重化ＭＩＳＲを用いると，1～3重誤りをすべて検出できることを示す。

第5章では，d重シンボル誤りを検出できるシグナチャ回路 —— 多重化MISR —— を提案する。提案したシグナチャ回路では，入力検査系列をＧＦ（$2^m$）上のシンボルとみなし，リード・ソロモン符号を適用した。また，リード・ソロモン符号の重み分布から，単一および多重化ＭＩＳＲの多重シンボル誤り検出率を求める。さらに，提案する多重化ＭＩＳＲの32ビットマイクロプロセッサへの応用についても述べる。

また，検査時間を削減する一つの方法として，ビット幅の圧縮を行なうシグナチャ回路を提案する。このシグナチャ回路は，ランダム誤り訂正符号と多重化ＭＩＳＲに基づくものである。この方法は，水平形マイクロプログラムＲＯＭのシグナチャ検査などに有効である。

第6章では，リード・ソロモン符号の重み分布を利用して，単一および多重化ＭＩＳＲ回路の誤り見逃し率（エイリアス確率）を解析する。その結果，ＭＩＳＲではエイリアス誤りは滑らかに収束する，ことを示す。すなわち，単一入力シグナチャレジスタ（ＳＩＳＲ）で見られるような変動はない。同時に，ＭＩＳＲおよび多重化ＭＩＳＲのエイリアス誤りは，選択する原始多項式に依存しないことを示す。

さらに，ＳＩＳＲのエイリアス確率を解析する。この解析にあたっては，まず双対符号（最大長系列符号）の重み分布を，計算機を用いて求めた。さらに，MacWilliams の等式により短縮化ハミング符号の重み分布を求め，ＳＩＳＲの解析をおこなった。その結果，

ＳＩＳＲのエイリアス確率は，選択する多項式によって，異なることを確認した。また，Ｔ．Ｗ．Ｗｉｌｌｉａｍｓの解析と同様に，エイリアス確率の過渡的振る舞いに変動があることを確認した。

第７章では，本研究の内容をまとめる。

# 誤り訂正符号のＶＬＳＩプロセッサ上での機構化およびＶＬＳＩ検査への応用に関する研究

## 目　　次

# 第1章　序　　論

## 1.1　序　　論

誤り訂正符号を用いた技術は，情報工学における基礎技術としてその発展に大きく貢献してきた［１－３］。近年，半導体技術の進歩に伴い，より低いコストで，より多くの素子が１チップに集積化できるようになった。すなわち，より複雑な機能 ―― 従来は諦めていた ―― を１チップ化することが経済的に見合うようになってきた。このことは，情報工学に多くの変化をもたらした。この一つの例として，誤り訂正符号の集積化が挙げられる。

近年，誤り訂正符号の実用化がめざましく，大型の計算機システム［４，５］のみならず，パーソナルコンピュータ，コンパクトディスクプレーヤー，ディジタルＶＴＲ(Video Tape Recorder)などにも，広く応用されている。用いられる符号も，初期の頃は，ハミング符号［６－８］やCRC（Cyclic Redundancy Check）符号といった，機構化に必要なハードウェア量の比較的少ないものであった。その後，ファイヤ符号やリード・ソロモン符号など，より誤り訂正能力の高い，それだけハードウェア量が余分に必要な符号が実用化されるようになった。例えば，フロピィディスクでは，16ビットＣＲＣ符号が用いられているし，ハードディスクではより高い信頼性を目的として，バースト誤り検出能力のあるファイヤ符号が用いられている。さらに，コンパクトディスクプレーヤやディジタルＶＴＲでは，より複雑なリード・ソロモン符号が用いられている［９，10］。

筆者は，日立製作所中央研究所における担当業務の一環として，ＶＬＳＩ（Very Large Scale Integrated circuit）プロセッサの研究開発に携わり，その中でも特に，誤り訂正符号のＶＬＳＩ上での機構化ならびにＶＬＳＩプロセッサの検査法への応用に取り組んできた。筆者は，まず，16ビットマイコン周辺ＬＳＩ（ＨＤＣ：Hard Disk Controller）の開発に従事した［11］。このとき，ファイヤ符号の符号化／復号化回路を集積化する必要があった。多大のハードウェア（チップ面積）を必要とせず，マイコンシステムで受け入れられる程度に復号化も早い，復号アルゴリズムの開発が必要であった。そこで，ディスクシステムのレコード長は，通常，ファイヤ符号の符号長よりも十分短いことを利用して，ＶＬＳＩ向き復号法を提案し，集積化した［12］。同時に，ＶＬＳＩプロセッサの設計法，検査法にも興味を持った［13，14］。

また，当時，高集積化が進む中で，より複雑なハードウェアをオンチップ化したが，検

査コストの増大が問題となりつつあった[15, 16]。これは，外部ピンの数が，集積度程増えておらず，検査回路の助けなしでは，検査が困難になったことに起因する。同時に，検査パターンの作成も，困難になりつつあった。

これを解決する一つの方法として，シグナチャ解析と呼ばれる自己検査法が提案されていた[17]。この方法は，もともと，プリント基板の検査および故障箇所の特定を目的として実用化されたものである[18－20]。シグナチャ回路としてm個の入力を持つ線形帰還シフトレジスタが，しばしば用いられる。この回路は，Multiple Input Signature Register（ＭＩＳＲ）と呼ばれる。

このような状況のもとで，ＨＤＣの開発にあたり，メンバーの一人からシグナチャ回路の提案があり，集積化することにした。しかし，当時，ＭＩＳＲに関して誤りの見逃し（エイリアス誤り）条件が定式化されておらず，誤り検出率も知られていなかった。筆者は，この回路に強い興味を持ち，その解析に取り組んだ。そこで，回路の線形性に着目し，ＭＩＳＲにおけるエイリアス誤りの発生条件を定式化し，1～4重ビット誤りの検出率を計算した。さらに，3重ビット誤りの検出ができる新しい回路を提案した[21]。この論文は，高い評価を得ることができ，昭和60年度電子通信学会（現電子情報通信学会）論文賞を受賞した。この時期，シグナチャ回路に関し，多くの解析／提案がなされるようになった[21－26]。

その後，32ビットマイクロプロセッサの開発を担当することになった。検査法は，より一層大きな問題となっていた。その一つの方策として，シグナチャ回路の誤り検出能力をさらに改善する必要があった。そこで，検査パターンがガロア体上のシンボルとみなせることに着目し，リード・ソロモン符号に基づくd重化ＭＩＳＲ回路を提案した。

さらに，単一入力シグナチャ回路の過渡的特性に興味が持たれていた。T. W. Williamsらは，Z変換を用いてこれを解析し，1986年ＩＴＣ（国際検査会議）の論文賞を受賞した[29, 30]。筆者もこれに興味を持ち，シグナチャ回路の過渡的なエイリアス確率が，いくつかの誤り訂正符号の重み分布[31, 32]と密接な関係があることに着目し，その性質を明らかにした[33]。

本論文は7章から構成されている。各章の概要を以下に示す。

第2章は，修正ハミング符号の構成法に関するものである。特に，3重および4重誤りの検出率改善に関して，2通りの構成法を提案する。提案する考え方に基づき，計算機を

用いて，いくつかの最適／良好な符号を示す。

第３章では，ファイヤ符号および巡回符号のＶＬＳＩ向き復号法を提案する。符号長に比べ情報点数が十分に短い ―― 通常の応用は当てはまる ―― 場合，ハードウェアも少なく，復号時間も短いような方法を提案する。これを16ビットマイコン周辺ＬＳＩへ応用し，ＶＬＳＩプロセッサでの有用性を明らかにする。

第４章は，シグナチャ解析法における誤り検出率の解析に関するものである。特に，原始多項式に基づくMISRの1～4重ビット誤り検出率，および拡大ハミング符号生成多項式に基づくＭＩＳＲの１～4重ビット誤り検出率を定式化する。誤り検出率の解析にあたっては，いくつかの新しい考え方を提案し，同時にハミング符号および拡大ハミング符号の性質を適用した。

第５章では，ｄシンボル誤りを検出できるシグナチャ回路 ―― 多重化ＭＩＳＲ ――を提案する。これは，シンボル距離ｄ＋１のリード・ソロモン符号に基づくものである。このシグナチャ回路の32ビットマイコンＶＬＳＩへの応用についても述べる。また，ビット幅の圧縮をおこなうシグナチャ回路も提案する。

第６章は，シグナチャ検査におけるエイリアス確率と符号の重み分布について考察したものである。多入力シグナチャ回路について，リード・ソロモン符号の重み分布を用いて，過渡的なエイリアス確率を，代数的に求めた。単一入力シグナチャ回路については，短縮化ハミング符号の重み分布を，計算機を用いて計算し，過渡的なエイリアス確率を示した。

第７章では，本研究の内容をまとめる。

## １２　参考文献


[１] 嵩，都倉，岩垂，稲垣，“符号理論,”コロナ社，1975年.

[２] W. W. Peterson and E. J. Weldon, Jr., “Error-correcting codes,” 2nd Edition, MIT Press, 1972.

[３] 宮川，岩垂，今井，“符号理論,”昭晃堂，1973年.

[４] M. Y. Hsiao, “A class of optimal minimum odd-weight-column SEC-DED codes,” IBM J. R&D, vol. 14, pp. 395-401, July 1970.

[５] A. M. Patel, “Optimal rectangular code for high density magnetic tapes,” IBM J. R&D, vol. 18, pp. 579-588, Nov. 1974.

[6] 安積，嵩，“最適な修正ハミング符号について”，信学論（A），vol.58-A, no. 6, pp. 325-330, 1975年6月.

[7] 岩崎，山村，嵩，“Hsiaoの意味で最適な（72, 64）修正ハミング符号,” 信学論（A），vol.61-A, no.3, pp. 270-271, 1978年3月.

[8] 岩崎，萩原，“ランダム誤り訂正符号の一短縮化法”信学論（A），vol. J68-A, no.1, pp. 100-101, 1985年1月.

[9] 今井秀樹，“誤り訂正符号化技術,” 電子通信学会誌，vol.67, no. 10, pp. 1094-1099, 1984年 10月

[10] 土居，今井，泉田，三田，“リードソロモン符号に対する軟判定復号法,” 信学論（A），vol.J70-A, no.9, pp. 1340-1345, 1987年 9月

[11] T. Funabashi, K. Iwasaki, K.Minorikawa, H. Yonezawa and T.Cantrell, “2 micron CMOS VLSI hard disk controller integrates data buffer and error correction,” WESCON, 24/3, pp.1-7, U.S.A., 1984.

[12] 岩崎，船橋，上野，“短縮化ファイヤ符号の復号法とＶＬＳＩプロセッサへの応用,” 信学論（D），vol.J68-D, no. 12, pp. 1998-2006, 1985年12月.

[13] N. Yamaguchi, T. Funabashi, K. Iwasaki, T. Shimura, Y. Hagiwara and K. Minorikawa, “A self-testing method for modular structured logic VLSIs,” Int. conf. CAD, pp. 99-101, U.S.A., Jan. 1984.

[14] K. Iwasaki, N. Yamaguchi, T. Shimura, Y. Hagiwara, T. Funabashi and K. Minorikawa, “Design methodology for reconfigurable module-structured VLSI,” Custom integrated circuits conf., pp.464-467, U.S.A., May1985.

[15] T.W. Williams and K. P. Parker, “Design for testability - a survey,” IEEE Trans. Comput., vol.C-31, no.1, pp. 2-15, Jan. 1982.

[16] R. M. Sedmak, “Built-in self-test: Pass or fail ?” IEEE Design & Test of Comput., vol. 2, no. 2, pp. 17-19, Apr. 1985.

[17] E. J. Mc Cluskey, “Built-in self-test techniques,” IEEE Design & Test of Comput., vol. 2, no. 2, pp. 21-28, Apr. 1985.

[18] R. A. Frohwerk, “Signature analysis: A new digital field service method,” HP Journal, pp. 2-8, May 1977.

[19] B. Könemann, J. Mucha and G. Zwiehoff, "Built-in test for complex digital integrated circuits," IEEE J. Solid-State Circuits, vol. SC-15, no. 3, pp. 315-319, June 1980.

[20] J. E. Smith, "Measures of the effectiveness of fault signature analysis," IEEE Trans. Comput., vol. C-29, no. 6, pp. 510-514, June 1980.

[21] 岩崎, 山口, 萩原, "ＬＳＩ自己検査用シグナチャ回路の誤り検出率の解析と新回路方式の提案," 信学論（Ｄ）, vol. J68-D, no. 5, pp. 1063-1070, 1985年5月.

[22] S. Z. Hassan and E. J. McCluskey, "Increased fault coverage through multiple signatures," Dig. FTCS-14, pp. 354-359, June 1984.

[23] W. C. Carter, "Improved parallel signature checkers/analyzers," Dig. FTCS-16, pp. 416-421, June 1986.

[24] R. David, "Signature analysis for multiple-output circuits," IEEE Trans. Comput., vol. C-35, no. 9, pp. 830-837, Sept. 1986.

[25] S. K. Jain and C. E. Stroud, "Built-in self testing of embedded memories," IEEE Design & Test of Comput., vol. 3, no. 5, pp. 27-37, Oct. 1986.

[26] M. Karpovsky and P. Nagvajara, "Optimal time and space compression of test responses for VLSI devices," Proc. ITC, pp. 523-529, Sept. 1987.

[27] K. Iwasaki, N. Yamaguchi and T. Nishimukai, "Analysis and proposal of signature circuit for LSI testing," Proc. ITC, pp. 517-522, Sept. 1987.

[28] K. Iwasaki, "Analysis and proposal of signature circuits for LSI testing," to appear in IEEE Trans. CAD special issue on testing, Jan. 1988.

[29] T. W. Williams, W. Daehn, M. Gruetzner and C. W. Starke, "Comparison of aliasing errors for primitive and non-primitive polynomials," Proc. ITC, pp. 282-288, Sept. 1986.

[30] A. Ivanov and V. Agarwal, "On a fast method to monitor the behaviour of signature analysis registers," Proc. ITC, pp. 645-655, Sept. 1987.

[31] 藤原, 嵩, S. Lin, "イーサネットで用いられる短縮巡回ハミング符号の誤り検出能力," 信学論（Ａ）, vol. J69-A, no. 6, pp. 731-735, 1986年6月.

[32] V. プレス, "符号理論," ワイリー・ジャパン社, 1984年.

[33] K. Iwasaki, "Aliasing probabilities and weight distributions of several codes," submitted to FTCS, 1988.

# 第2章　修正ハミング符号のいくつかの構成法

## 2.1　まえがき

拡大ハミング符号は距離4の符号として知られており[ 1 , 2 ]，単一誤りを訂正し，2重誤りを検出する。実用に際しては，情報点数が2の累乗となるように短縮化される。短縮化された拡大ハミング符号を修正ハミング符号と呼ぶ[ 3 ]。短縮の仕方には自由度があり，以下の事項を考慮する必要がある。機構化に必要なハードウェア量，多重誤り検出率などが挙げられる。あるいは，音声・画像などのデータをディジタル化して通信／記憶する場合，上位ビットの誤り検出率が高いような符号を構成することが望ましい。

これらの要求事項に対し，M. Y. Hsiao は機構化に必要なハードウェア量を少なくすることを目的として，非零要素の総数が最小であるようなパリティ検査行列を持つ修正ハミング符号を構成した[ 3 ]。同時に，3重誤りの誤訂正（ミスコレクト）確率および4重誤りの見逃し確率は，重み4の符号語数$N_4$に依存することを示した。安積，嵩は，$N_4$が小さい修正ハミング符号を構成した[ 4 ]。このほか，メモリ検査用に，オール1を符号語として持つような修正ハミング符号が，計算機を用いて構成されている[ 5 ]。また，修正ハミング符号の誤り訂正回路に自己検査機能を追加する方法も提案されている[ 6, 7 ]。

本章では，修正ハミング符号の2つの機成法について論じる。1つは，非零要素の総数が最小であるようなパリティ検査行列を持つ修正ハミング符号のうちで，3重誤りの誤訂正確率，4重誤りの見逃し確率が最小の符号に関するものである[ 8 ]。このような修正ハミング符号をHsiaoの意味で最適な修正ハミング符号と呼ぶ。特に実用性の高い（72, 64）修正ハミング符号に対して，計算機を用いて，Hsiaoの意味で最適な符号を求めた。ここで，符号長n，情報点数kの符号を（n，k）符号と表す。計算にあたっては，パリティ検査行列中でより重みの小さい列ベクトルが固定できることを利用し，調べるべき場合の数を減らした。

もう一つは，ディジタル化されたデータなどの上位ビットの誤り検出率を高くするような修正ハミング符号に関するものである[ 9 ]。パリティ検査行列の性質を利用して，ビット位置毎の誤り検出率を定式化する。いくつかの符号長に対し，計算機を用いて，良好な符号を見いだした。また，特定ビットの誤り検出率と符号全体の誤り検出率はトレード・オフの関係にあることを示す。

## 2.2　3および4重誤り検出率の高い修正ハミング符号の一構成法

### 2.2.1　調べるべきパリティ検査行列の発生

符号長72，情報点数64の（72，64）修正ハミング符号に関し，Hsiaoの意味で最適な修正ハミング符号を求める。非零要素の総数が最小のパリティ検査行列Hを次式のように表す。

$$〔\,H\,〕 \;=\; 〔\,H_1\ H_2\,〕 \qquad (2-1)$$

ここで，$H_1$は長さが8，かつ重みが1および3の列ベクトル全体からなる$8\times64$の行列である。ベクトルの重みは，そのベクトルに含まれる1の総数と定義する。$H_2$は長さが8，かつ重みが5の異なる8個の列ベクトルからなる$8\times8$の行列である。$H_1$，$H_2$は共に2元の行列である。

問題は，重み4の符号語数$N_4$を最小にするような$H_2$行列を求めることにある。可能な$H_2$行列の数は，${}_{56}C_8 \simeq 1.4\times10^9$と非常に大きい。しかし$H_1$行列の性質から$N_4$は$H_2$行列の行および列の置換に対して不変である。この性質を利用することにより，実際に調べるべき$H_2$行列の数は大きく減少する。2つの$H_2$行列が行および列の置換により等しくなるとき，等価であると定義する。等価な$H_2$行列のうち一つを代表として選べば，代表の数は近似的に${}_{56}C_8 / 8! = 3.5\times10^4$となる。置換の性質をすべて利用して等価な$H_2$行列のうちから一つだけを選ぶことは必ずしも容易ではなく，プログラムの作成上あるいは実行上からも得策とはいえない。

比較的能率の良い$H_2$行列の発生法について説明する。次の（条件1），（条件2）を満たすように$H_2$行列の各列を第1列，第2列，……と第8列まで発生させる。第i列までを既に選んだとして，この行列をGとする。Gの第a行から第b行までを任意に置換して得られる行列をG′とする。G′をGに変換するような列の置換が存在するとき，第（i＋1）列の第a成分から第b成分〔a，b〕を置換可能な範囲と呼ぶ。ここでは，連続した範囲の置換のみを考える。

（条件1）　各列を2進数（上の成分を下位桁）とみなし，昇べき順に第1列，第2列，……と並べる。

（条件2）　置換可能な範囲内で，1はできるだけ上方に置く。

第１列の置換可能な範囲は，第１列の要素の置換のみを考えればよいから〔１，８〕である。よって，（条件２）より重み５のベクトルのうち（１１１１１０００）$^{T}$ だけを考えればよい。ここで，$\mathbf{a}^{T}$ はベクトル$\mathbf{a}$の転置ベクトルを表す。この結果，第２列の置換可能な範囲は〔１，５〕〔６，８〕となる。それぞれの範囲にはいる１の数を考慮すると，条件２より（１１１１０１００）$^{T}$，（１１１００１１０）$^{T}$，（１１０００１１１）$^{T}$ の３通りだけを考えればよい。いずれも（条件１）を満たしている。以下同様な操作を第８列までおこなう。図2.1に，$H_2$ 行列の選択を例示する。図中〕は次に選択する列の置換可能な範囲を示す。第３列の第５，６成分は置換可能である。理由は，第１，２列のそれぞれ第５，６成分を置換し，さらに第１，２列を置換すると元の行列に戻るからである。

置換可能な範囲としてできるだけ広いものを採用するのが望ましい。しかし，そのための手間と $H_2$ 行列の発生個数の兼ね合いから，最大の範囲を求めることは必ずしも得策ではない。実際には，第６列以降では既に選んだ部分行列を不変に保つような行の置換のみを考慮した。この結果，調べるべき $H_2$ 行列の数を約 $7 \times 10^{5}$ まで抑えることができた。

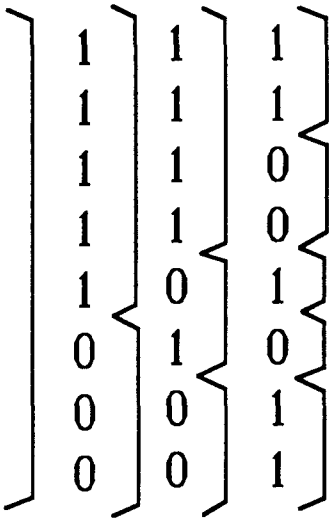



図 2.1　列ベクトルの選択例

### 2.2.2　重み４の符号語数の計算

行列 $H_1$，$H_2$ が与えられたとき，それぞれからｉ，ｊ個の列ベクトルを選びそれらのＧＦ（２）上の和ベクトルが零ベクトルとなるような列ベクトルの選び方の数を $A_{ij}$ と表す。ここでＧＦ（２）は位数２の有限体を表す。$H_2$ 行列のｊ個の列ベクトルの和の重み

がwとなるようなj個の列ベクトルの選び方の数を$B_j(w)$と表す。$N_4$はパリティ検査行列中の4個の列ベクトルの和が零ベクトルとなる4個の列ベクトルの選び方に等しいので，次式が成立する。

$$N_4 = \sum_{i=0}^{4} A_{i\ 4-i} \qquad (2-2)$$

ここで式（2－1）の性質により$A_{40}$，$A_{31}$は共に定数となり，それぞれ5376，2560と求まる。このほかの$A_{ij}$は，$H_2$行列に属するj個の列ベクトルの和の重みで分類することにより，次のように表される。

$$\begin{aligned} A_{22} &= 22\,B_2(2) + 16\,B_2(4) + 10\,B_2(6) \\ A_{13} &= B_3(1) + B_3(3) \\ A_{04} &= B_4(0) \end{aligned} \qquad (2-3)$$

ここで式（2－3）の$B_2(2)$の係数は，$H_1$行列から2個の列ベクトルを選び，その和が重み2の与えられたパターンとなるような選び方の数であり，$1+6+{}_6C_2=22$となる。同様に，$B_2(4)$，$B_2(6)$の数は$H_1$行列のある2つの列ベクトルの和がそれぞれ重み4および6のある与えられたパターンとなるような選び方の数であり，それぞれ$4+4\times{}_4C_2/2=16$，${}_6C_3/2=10$となる。また，

$$B_2(2)+B_2(4)+B_2(6) = {}_8C_2 \qquad (2-4)$$

より式（2－2）は次のように表される。

$$N_4 = 8216 + 12\,B_2(2) + 6\,B_2(4) + B_3(1) + B_3(3) + B_4(0) \qquad (2-5)$$

与えられた任意のパリティ検査行列から$N_4$を求めるには，双対符号の重み分布を利用する方法もあるが［4］，本節での問題の場合，式（2－5）から求める方が能率がよい。プログラムはFORTRANで作成した。

### 2.2.3 結　果

Hsiaoの意味で最適な，すなわちパリティ検査行列中の非零要素の総数が最小のもののうちで重み4の符号語数$N_4$が最小の（72，64）修正ハミング符号は，$N_4=8392$とな

ることがわかった。このような符号のうち，互いに等価でない $H_2$ 行列の $B_j(w)$ の分布が異なることからわかる。得られた3つの符号の $H_2$ 行列の等価な組の中から，それぞれ対称であるものを選び図2.2に示す。行列の対称性と列の重みがすべて5であることより，得られた $H_2$ 行列の行の重みもすべて5である。このうち，(a)の $H_2$ 行列はHsiaoの例の一つ[ 3 ]と等価であり，$H_2$ 行列の行または列を巡回置換したものとなっている。図2.2で示す $H_2$ 行列は，図2.1で示される発生法とは必ずしも一致しない。

パリティ検査行列中の非零要素の総数が最小との条件がなければ，$N_4 = 8175$ であるような(72, 64)修正ハミング符号が求められている[ 4 ]。

| (a) | (b) | (c) |
|---|---|---|
| 01111001 | 11111000 | 00011111 |
| 11110010 | 11110100 | 00110111 |
| 11100101 | 11101010 | 01100111 |
| 11001011 | 11010101 | 11001011 |
| 10010111 | 10101011 | 10011101 |
| 00101111 | 01010111 | 11101010 |
| 01011110 | 00101111 | 11110100 |
| 10111100 | 00011111 | 11111000 |

図2.2　Hsiao の意味で最適な(72，64)修正ハミング符号の$H_2$ 行列

## 2.3　特定ビットの誤り検出率の高い修正ハミング符号の一構成法

### 2.3.1　第ｉビットを含んだ３，４重誤りの検出率

修正ハミング符号に対して，第ｉビットを含んだ３重誤りの検出率を$PD_3{}^i$，第ｉビットを含んだ４重誤りの検出率を$PD_4{}^i$と表す。符号長ｎの修正ハミング符号のパリティ検査行列Ｈを次のように表す。

$$H=[\,h_0,\ h_1,\ \cdots\cdots,\ h_{n-1}\,] \qquad (2-6)$$

パリティ検査行列Ｈにおいて，列ベクトル$h_i$（$0 \leqq i \leqq n-1$）を含みｊ個の列ベクトルの和が零ベクトルとなるような列ベクトルの選び方の組み合わせの数を$N_j{}^i$と表す。

ここで，第ｉビットを含む３重誤りの検出率を計算する。第ｉビットを含む３重誤りの誤りパターンを$\mathbf{e}_i$とする。ある符号語$\mathbf{a}$に対し，$\mathbf{a}+\mathbf{e}_i$を別の符号語$\mathbf{b}$に誤訂正（ミスコレクト）することは，図2.3に示すように，$\mathbf{a}+\mathbf{e}_i$と符号語$\mathbf{b}$の距離が１となったこ

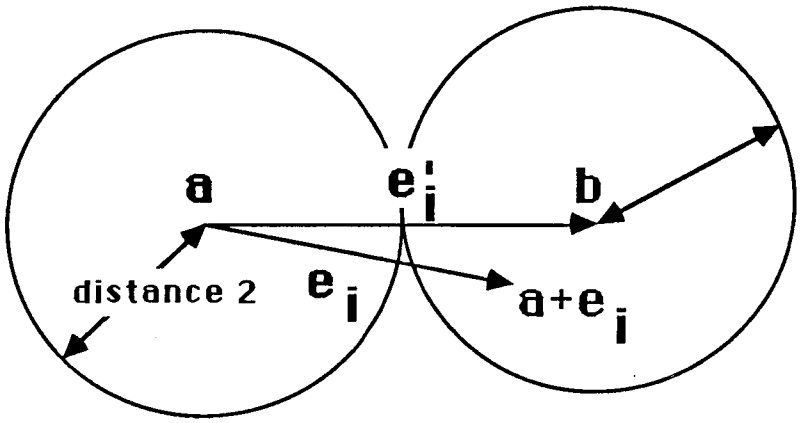



図 2.3　修正ハミング符号の符号語と誤り検出／見逃し／誤訂正

とを意味する。すなわち，第ｉビットが１かつ重み４のベクトル$\mathbf{e}'_i$が存在し，$\mathbf{a}+\mathbf{e}'_i=\mathbf{b}$が成立することに等しい。このような誤りパターン$\mathbf{e}_i$の選び方の数は，第ｉビットが１のため，$\mathbf{e}'_i$の第ｉビット以外の３個の１から２個を選ぶ場合の数${}_3C_2$に等しい。符号語$\mathbf{a}$に対して符号語$\mathbf{b}$の選び方は$N_4{}^i$個存在する。また，第ｉビットを含んだ３重誤りが生じる場合の数は，第ｉビットが１のため，（ｎ－１）個から２個を選ぶ場合の数${}_{n-1}C_2$に等しい。よって，次式が成立する。

$$PD_3{}^i = 1 - {}_3C_2 \times N_4{}^i / {}_{n-1}C_2 \qquad (2-7)$$

次に，第ｉビットを含む４重誤りの検出率を計算する。第ｉビットを含んだ４重誤りを見逃すことは，図2.3より，重みが４かつ第ｉビットが１であるような符号語$\mathbf{e}_i'$が存在することを意味する。このような誤りパターン$\mathbf{e}_i'$の選び方の数は$N_4{}^i$に等しい。第ｉビットを含んだ４重誤りが生じる場合の数は，第ｉビットが１のため，（ｎ－１）個から３個を選ぶ場合の数${}_{n-1}C_3$に等しい。よって次式が成立する。

$$PD_4{}^i = 1 - N_4{}^i / {}_{n-1}C_3 \qquad (2-8)$$

第ｉビットを含んだ３重および４重誤りの検出確率は，いずれも$N_4{}^i$，すなわちパリティ検査行列Ｈの第ｉ列を含みｊ個の列ベクトルの和が零ベクトルとなるような列ベクトルの選び方の数に依存する。

### 2.3.2　パリティ検査行列の構成および良好な修正ハミング符号

式（２－７）および（２－８）の右辺に現われる$N_4{}^i$は，短縮化した符号ではビット毎に異なる値を持つ。言い換えると，短縮化の仕方によって，各ｉについての$N_4{}^i$が異なってくる。したがって，誤り検出率を高めたいビットについては，$N_4{}^i$が小さな値を持つように短縮化すればよい。つまり，特定の列ベクルトについて$N_4{}^i$が小さな値を持つパリティ検査行列の構成法に帰着される。

以下に，（１８，１２）修正ハミング符号の例を示す。パリティ検査行列Ｈを次式のように選ぶ。

$$H = \begin{bmatrix} 1\,0\,1\,1\,1\,1\,1\,1\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ 1\,1\,1\,1\,0\,0\,0\,1\,1\,1\,0\,1\,0\,1\,0\,0\,0\,0 \\ 1\,1\,1\,0\,1\,1\,1\,0\,1\,0\,1\,1\,0\,0\,1\,0\,0\,0 \\ 0\,1\,0\,1\,1\,0\,1\,0\,0\,1\,1\,1\,1\,0\,0\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,1\,0 \\ 1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix} \qquad (2-9)$$

この時，重み４の符号語数$N_4$は１２６となり，$N_4{}^i$の分布は次のようになる。

$$N_4{}^i = (14,\ 30,\ 31,\ 31,\ 31,\ 31,\ 14,\ 30,\ 31,\ 31,\ 31,\ 31,\ 13,\ 31,\ 30,\ 31,\ 31,\ 14) \qquad (2-10)$$

このときの$PD_3{}^i$および$PD_4{}^i$の計算結果を表2.1に示す。

表 2.1　ビット位置 i を含んで3，4重誤りが生じた場合の誤り検出率 $PD_3^i$，$PD_4^i$ の計算例，パリティ検査行列が式（2－9）のとき

| $PD_3^i$ | $PD_4^i$ | ビット位置 i（$0 \leqq i \leqq 17$） |
|---|---|---|
| 0.691 | 0.979 | 0, 6, 17 |
| 0.338 | 0.956 | 1, 7, 14 |
| 0.316 | 0.954 | 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16 |

いくつかの符号長の修正ハミング符号について，$N_4^i$ の分布を大型コンピュータを用いて求めた。その結果，良好な修正ハミング符号を見つけることができた。ここで良好な（n，k）符号とは，符号語中に k／2 ビットのデータが 2 個含まれるものと仮定し，データ部（$0 \leqq i \leqq k-1$）の $N_4^i$ のうち最小のものと2番目に小さいものとの和（SM2と表す）が小さいものをいう。これらを表2.2に示す。パリティ検査行列Hは，列ベクトルを2進数とみなして表した。重み4の符号語数 $N_4$ の欄には，最良の $N_4$ の値［4］をブラケット内に示した。

表 2.2　誤り検出率がビット毎に異なる，良好な修正ハミング符号の構成例

| (n,k) | SM2 | $N_4$ | | パリティ検査行列および $N_4^i$ |
|---|---|---|---|---|
| (22,16) | 80 | 260 (250) | H | 38 7 41 11 44 13 47 50 52 21 22 55 25 59 61 31 1 2 4 8 16 32 |
| | | | $N_4^i$ | 50 40 50 50 40 50 50 50 50 40 50 40 50 50 50 50 40 50 50 40 50 50 |
| (39,32) | 212 | 1506 (1335) | H | 67 69 70 73 74 76 81 82 84 87 88 91 93 94 31 97 98 100 103 104 107 109 110 47 112 115 117 118 121 122 124 127 1 2 4 8 16 32 64 |
| | | | $N_4^i$ | 167 167 167 167 167 167 167 167 167 167 167 167 167 167 106 167 167 167 167 167 167 167 106 166 168 168 168 168 168 168 166 105 105 105 105 106 106 166 |
| (55,48) | 802 | 5598 (5589) | H | 67 69 70 73 74 76 79 81 82 84 22 87 88 25 91 93 94 97 98 35 100 37 38 103 104 41 42 107 44 109 110 47 112 49 50 115 52 117 118 55 56 121 122 59 124 61 62 127 1 2 4 8 16 32 64 |
| | | | $N_4^i$ | 408 408 408 408 408 408 408 408 408 408 401 408 408 401 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 408 401 401 401 401 402 408 408 |
| (72,64) | 814 | 8401 (8175) | H | 131 133 134 7 137 138 11 140 13 14 145 146 19 148 21 22 152 25 26 28 161 162 35 164 37 38 168 41 42 44 176 49 50 52 56 191 193 194 67 196 69 70 200 73 74 76 208 81 82 84 88 224 97 98 100 104 239 112 115 117 118 121 122 124 1 2 4 8 16 32 64 128 |
| | | | $N_4^i$ | 442 442 442 462 442 442 462 442 462 462 451 451 473 451 473 473 451 473 473 473 453 453 475 453 475 475 453 475 475 475 463 481 481 481 481 407 488 463 463 463 463 463 491 491 491 491 496 500 496 476 |

表2.2に示した良好な符号の $N_4$ は，最良の場合と比べて，0.2～20%程度大きくなっている。短縮化の自由度が大きいほど最良の場合との差が大きい。$N_4$ が大きくなれば符号としての3，4重誤り検出率は下がる。つまり，符号全体としての検出率と特定ビットの検出率はトレード・オフの関係にある。

プログラムは，アセンブラ（HITAC－Mシリーズ）とFORTRANをリンクさせて実行した。

## 2.4 まとめ

本章では，修正ハミング符号の２つの構成法を提案した。１つは，非零要素の総数が最小であるようなパリティ検査行列を持つ修正ハミング符号のうちで，３重誤りの誤訂正確率，４重誤りの見逃し確率が最小の符号に関する構成法である。特に実用性の高い（７２，６４）修正ハミング符号に対して，計算機を用いて，このような符号を求めた。計算にあたっては，パリティ検査行列中でより重みの小さい列ベクトルが固定できることを利用し，調べるべき場合の数を減らした。

また，ディジタル化されたデータなどの上位ビットの誤り検出率を高くするような修正ハミング符号を構成した。まず，パリティ検査行列の性質を利用して，ビット位置毎の誤り検出率を定式化した。いくつかの符号長に対し，計算機を用いて，良好な符号を見いだした。また，特定ビットの誤り検出率と符号全体の誤り検出率はトレード・オフの関係にあることを示した。

## 2.5 参考文献


［１］ 嵩，都倉，岩垂，稲垣，“符号理論,”コロナ社，１９７５年．

［２］ W.W.Peterson and E.J.Weldon, Jr.,“Error-correcting codes,” 2nd Edition, MIT Press, 1972.

［３］ M.Y.Hsiao,“A class of optimal minimum odd-weight column SEC-DED codes,”IBM J. R & D, vol.14,pp.395-401, July 1970.

［４］ 安積，嵩，“最適な修正ハミング符号について，”信学論（Ａ），vol.58-A, no.6,pp.325-330,1975年６月．

［５］ C.E.Sundberg,“Properties of transparent shortened codes for memories with stack-at faults,”IEEE Trans. Comput.,vol.C-28,no.9,pp.686-690, Sept.1979.

［６］ 伊藤，中道，“自己検査可能な主記憶用奇数重み列ＳＥＣ－ＤＥＤ符号誤り検査訂正回路の構成，”信学論（Ｄ），vol.J66-D,no.9,pp.1070-1077,1983年９月．

［７］ 新森，宝田，古賀，“誤り検査訂正可能なメモリシステムのセルフチェッキング化の一方法，”信学論（Ｄ），vol.J69-D,no.2,pp.148-158,1986年２月．

［８］ 岩崎，山村，嵩，“Hsiaoの意味で最適な（７２，６４）修正ハミング符号，”信学論（Ａ），vol.61-A,no.3,pp.270-271,1978年３月．

［９］　岩崎，萩原，“ランダム誤り訂正符号の一短縮化法，”信学論（Ａ），vol J68-A,no.1,pp.100-101,1985年１月．

# 第3章　ファイヤ符号および巡回符号のVLSI向き復号法

## 3.1　まえがき

誤り訂正符号の一つの重要な応用として，ハードディスクが挙げられる。ハードディスクでは，データの信頼性向上を目的として，バースト誤り訂正符号がしばしば用いられる。バースト誤り訂正機能をハードディスクのコントロール用ＶＬＳＩチップに実現するためには，次のことを考慮する必要がある。①ハードウェア規模（チップ面積），②復号時間，③使用される頻度，④設計（論理，回路，レイアウト）の容易さ，などが挙げられる。さらに，１個のＶＬＳＩチップで複数の情報点数（レコード長）に対応するためには，⑤拡張性のある復号アルゴリズムを採用する必要がある。

本章では，まず，バースト誤り訂正符号としてファイヤ符号（生成多項式$G(x)=(x^c+1)\times p(x)$）を考える［１－３］。ファイヤ符号を実用化する場合，短縮化して使用されるのが一般的である。通常，短縮化前の符号長ｎはレコード長（ＲＬバイト）より十分長く，符号語の先頭部分はオール０とみなしてよい。このオール０の部分の省略法は現在までにいくつか提案されている［４，５］。ここでは，除多項式が$p(x)$である線形帰還シフトレジスタ（LFSR：Linear Feedback Shift Register）の内容は，$p(x)$の周期ｅだけシフトすると元のパターンに戻ることに着目し，オール０パターンのうちｅの倍数回のシフトを省略して復号の高速化を図る[6]。この復号法は，ハードウェアが比較的少なく（シフトレジスタ＋カウンタ），通常の応用（符号長ｎ≫レコード長ＲＬ）では復号時間も短い。また，設計の容易さ，拡張性の面でもすぐれている。さらに，この復号法を，一般のバースト誤り訂正巡回符号へ拡張する。

提案する復号法の応用対象として，32ビットファイヤ符号を選び，Hard Disk Controller（HDC）へ応用した［７，８］。HDCは１６ビットマイクロコンピュータの周辺ＬＳＩとして開発されたものである。マイクロコンピュータ周辺ＬＳＩに誤り訂正機能を集積化する場合，前期①～⑤の各項目を考慮して設計する必要があり，ここで提案する復号法が適している。具体的適用の結果，HDCにおける誤り訂正機能の素子数は約６７００トランジスタとなった。これは，HDC全体（１２９kトランジスタ）の5.2％に過ぎない。このようにコンパクトな設計が可能となった。

また，周辺ＬＳＩでは拡張性のあるマイクロアーキテクチャが必要である。このような設計法［９，１０］，検査法［11］についても言及する。

## 3.2　ファイヤ符号のＶＬＳＩ向き復号法の提案

### 3.2.1　ファイヤ符号と誤り訂正モデル

ファイヤ符号は，次式の多項式を生成多項式とする巡回符号である[1]。

$$G(x)=(x^c+1)\,p(x) \qquad (3-1)$$

ここで，$p(x)$はGF(2)上の$\delta$次の既約多項式である。$p(x)$の周期をｅとする。ただし，ｅはｃを割り切らない。この符号の符号長ｎは，ｃとｅの最小公倍数ＬＣＭ(ｃ，ｅ)である。この符号の検査点の数は(ｃ＋$\delta$)，情報点数ｋは(ｎ－ｃ－$\delta$)となる。もしも，$c \geqq b+d-1$および，$\delta \geqq b$という条件が満たされれば，この符号は長さｂまでのバースト誤りを１個訂正できる。あるいは，長さｄ(＞ｂ)以下で，ｂより大きいバースト誤りを検出できる[2]。

ファイヤ符号を実際に使用する場合，符号長ｎはｎ′に，情報点数ｋはｋ′に短縮化して使用されるのが普通である。短縮化しても検査点の数(ｎ′－ｋ′)は(ｃ＋$\delta$)に等しい。

図3.1はＶＬＳＩチップと誤り訂正モデルについて示した図である。ＶＬＳＩチップ上にはファイヤ符号のエンコーダおよびデコーダが集積化されている。データの書き込みおよび読み出しは以下のようにおこなわれる。ファイヤ符号エンコーダは，書き込みデータに検査ビットを付加し，送信符号$F(x)$を生成する。送信符号$F(x)$はファイヤ符号の符号語であり，多項式$(x^c+1)$および$p(x)$で整除される。記憶装置は，磁気的手段あるいは電気的手段により$F(x)$を記憶する。データの読み出し時に，ＶＬＳＩチップは記憶装置から受信符号$F^*(x)$を受け取る。このとき，雑音やメディアの欠陥などにより，$F(x) \neq F^*(x)$となることがある。ＶＬＳＩ内のファイヤ符号デコーダは，誤り訂正能力の範囲内で，$F^*(x)$から$F(x)$を復元する。

ファイヤ符号の復号化の手順を図3.1にしたがって説明する。まず，ファイヤ符号デコーダは次式にしたがって，シンドローム$S_0(x)$，$S_1(x)$を計算する。

$$\begin{aligned} S_0(x) &\equiv F^*(x) \mod x^c+1 \\ S_1(x) &\equiv F^*(x) \mod p(x) \end{aligned} \qquad (3-2)$$

受信符号$F^*(x)$の(ｉ＋１)ビット目からバースト誤り$B(x)$を含んでいる場合，次式が成り立つ。ただし，先頭を第１ビットとする。

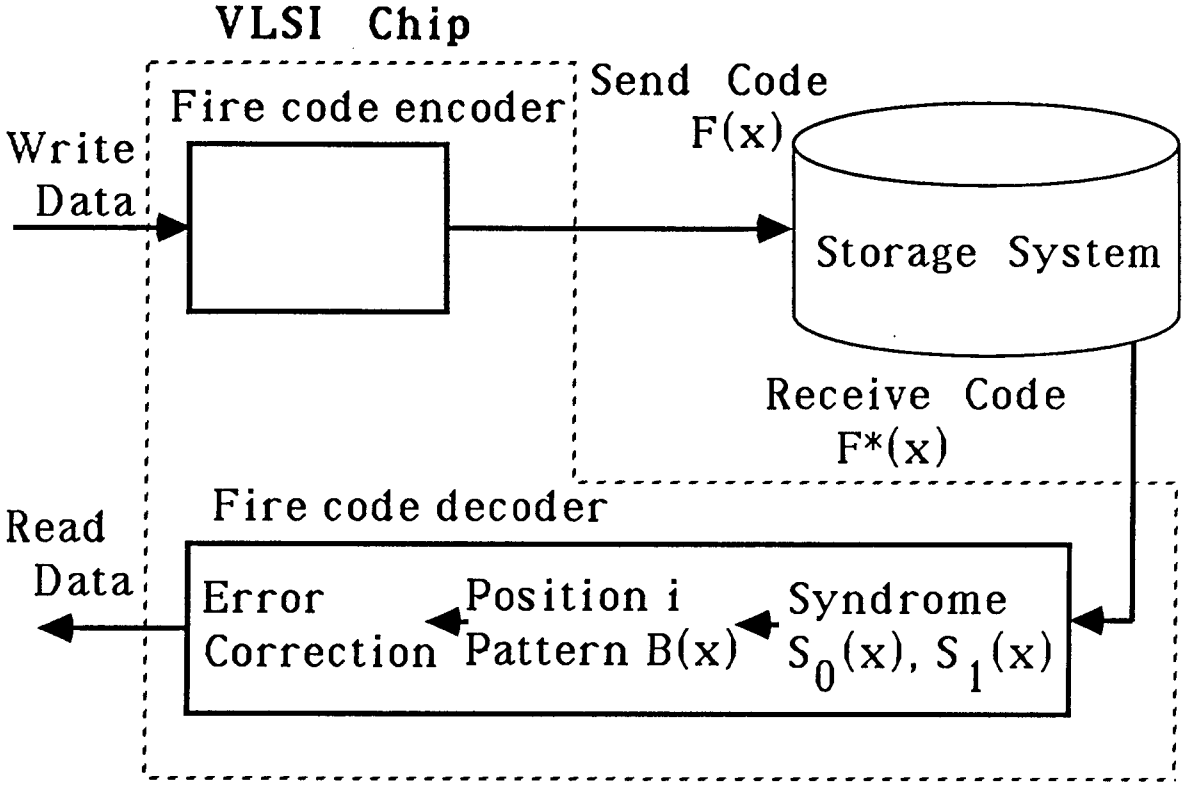



図 3.1　VLSIチップと誤り訂正モデル

$$S_0(x) \equiv x^i B(x) \mod x^c + 1$$
$$S_1(x) \equiv x^i B(x) \mod p(x) \qquad (3-3)$$

次にファイヤ符号デコーダは，$S_0(x)$，$S_1(x)$から，誤り位置 i と誤りパターン B(x)を求める。この計算アルゴリズムは，3.2.2 および 3.2.3 で述べる。受信符号 $F^*(x)$，誤り位置 i，誤りパターン B(x)がわかれば，元のデータ F(x)が復元できる。式（３－２）の計算は，通常のLFSRで実現できる。ここでは，（$x^c+1$）を除多項式とするLFSRをLFSR0，p(x)を除多項式をするLFSRをLFSR1と表すことにする。

### 3.2.2　従来のファイヤ符号復号法

従来のファイヤ符号復号法について簡単に説明する。（n′，k′）に短縮化されたファイヤ符号の復号法として，次の３通りの方法が知られている。

（Ｄ１）　巡回符号の性質を利用した方法［3］

（Ｄ２）　シンドロームの線形性を利用してシフト回数を減少させる方法［4］

（Ｄ３）　“中国人の剰余定理”を用いる方法[５]

上記（Ｄ１）の方法は，式（３－２）にしたがってシンドロームを計算した後，LFSR0とＬＦＳＲ1を同時にシフトして，ｉとＢ(x)を求める方法である。この方法によると，最大シフト回数は｛ｎ－（ｃ＋$\delta$）｝回であり，必要なハードウェアは（c+$\delta$)ビットのLFSR「log n┐†ビットカウンタである。この方法はハードウェアが少なく，復号手順が短縮化を考慮しなくてよいという利点がある。しかし，先頭部分のオール０を省略しておらず，復号時間が長いという欠点がある。

（Ｄ２）の方法は，シンドロームを計算する際に，同時に$x^{n-n'}$を乗算してしまう方法である。この後，ＬＦＳＲ0とＬＦＳＲ1を同時にシフトさせ，ｉとＢ(x)を求める。この方法は，シフト回数を符号長$n'$にまで減少させることができるため，復号時間の短縮が可能である。しかし，ＬＦＳＲ0およびＬＦＳＲ1の内部の接続が，短縮化された符号長$n'$に依存して異なる。このため，複数の短縮化した符号長に対応するためには，接続変更用の論理回路が必要となる。接続変更用の回路は，Read Only Memory（ROM），Programmable Logic Array（PLA）あるいはレジスタなどの規則的な論理素子では実現が困難である。よって，ＶＬＳＩへの適用[１２，１３]という観点から，論理設計，レイアウト設計，回路遅延の面で必ずしも有利な方法ではない。必要なハードウェアは，接続変更用の論理回路に加え，ｃビットおよび$\delta$ビットのＬＦＳＲ，「log $n'$┐ビットカウンタである。

（Ｄ３）の復号方法は，“中国人の剰余定理”を用いてシフト回数を大幅に減らす方法である。シフト回数は最大（ｃ＋ｅ）回である。誤り位置ｉを求めるために，次の計算をおこなう必要がある。

$$i \equiv A_0 \times e \times r_0 + A_1 \times c \times r_1 \mod n \qquad (3-4)$$

ここで，$r_0$，$r_1$はそれぞれＬＦＳＲ0，ＬＦＳＲ1のシフト回数であり，$0 \leqq r_0 < c$，$0 \leqq r_1 < e$を満たす。また，$A_0$，$A_1$は次式を満たす定数である。

$$A_0 \times e + A_1 \times c \equiv 1 \mod n \qquad (3-5)$$

$A_0$および$A_1$は生成多項式によって一意に決まる値であり，前もって計算しておくことが可能である。この方法では，式（３－４）を計算する際に「log ｎ┐ビットの乗除算機能が必要であり，ハードウェア（チップ面積）が大きくなってしまう。

---

†「ａ┐はａの切り上げを表す。

### 3.2.3 ファイヤ符号のVLSI向き復号法

ここではVLSI向きの復号法を提案する。すなわち，符号の情報点数が小さい場合には，それに対応してシフト回数を減らすことが可能で，かつハードウェア量も少ないファイヤ符号の復号法である。この復号法は，複数の短縮化した符号長に対しても，ハードウェアの変更なしにシフト回数の減少が可能である。また，ROM，レジスタなどの規則的な論理素子で実現するのに適している。

式（3－1）で示した多項式を生成多項式とするファイヤ符号の符号語の先頭の（n－n′）ビットはオール0とみなしてよい。この部分のシフト回数を省略することにより，誤り位置iの計算時間を大幅に短くすることができる。

このため次の各点に注目した。

(1) 多項式（$x^c+1$）の周期はcである。これは，LFSR0をc回シフトすると，元のパターンに戻ることを意味する。

(2) 多項式p(x)の周期はeである。これは，LFSR1をe回シフトすると，元のパターンに戻ることを意味する。

(3) ファイヤ符号の符号語F(x)をeビットごとのブロックに分割すると，先頭に近いブロックはオール0データを含む。

以上3項目より，符号語F(x)の先頭（n－n′）ビットのオール0データのうち，eのj倍（jは正整数）のシフトを省略することは，LFSR0のみをある整数P（計算式は後述）回シフトすることに相当する。理由は以下の通りである。シンドローム$S_0(x)$，$S_1(x)$に$x^{je}$を乗算して，それぞれ（$x^c+1$），p(x)で除算したときの剰余を考える。$\beta$は$je \geqq \beta c$を満たす最大の整数とすると，次式が成り立つ。

$$\begin{aligned} x^{je}S_0(x) &\equiv x^{je-\beta c}S_0(x) \mod x^c+1 \\ x^{je}S_1(x) &\equiv S_1(x) \mod p(x) \end{aligned} \qquad (3-6)$$

$\beta$は，j，c，eが与えられたとき，計算によって求めることができる。したがって，前もってP＝（$je-\beta c$）を計算によって求めておき，$x^P$を$S_0(x)$に乗算して（$x^c+1$）で除算すれば，$S_0(x)$，$S_1(x)$のおのおのに$x^{je}$を乗算し，それぞれ（$x^c+1$），p(x)で除算したことと同じ結果になる。

図3.2は，ファイヤ符号の短縮化した符号語を示した図である。図3.2に示すように，巡回符号の性質を用いた復号法（前述（D1））では｛n－（k＋c＋$\delta$）｝回の余分な

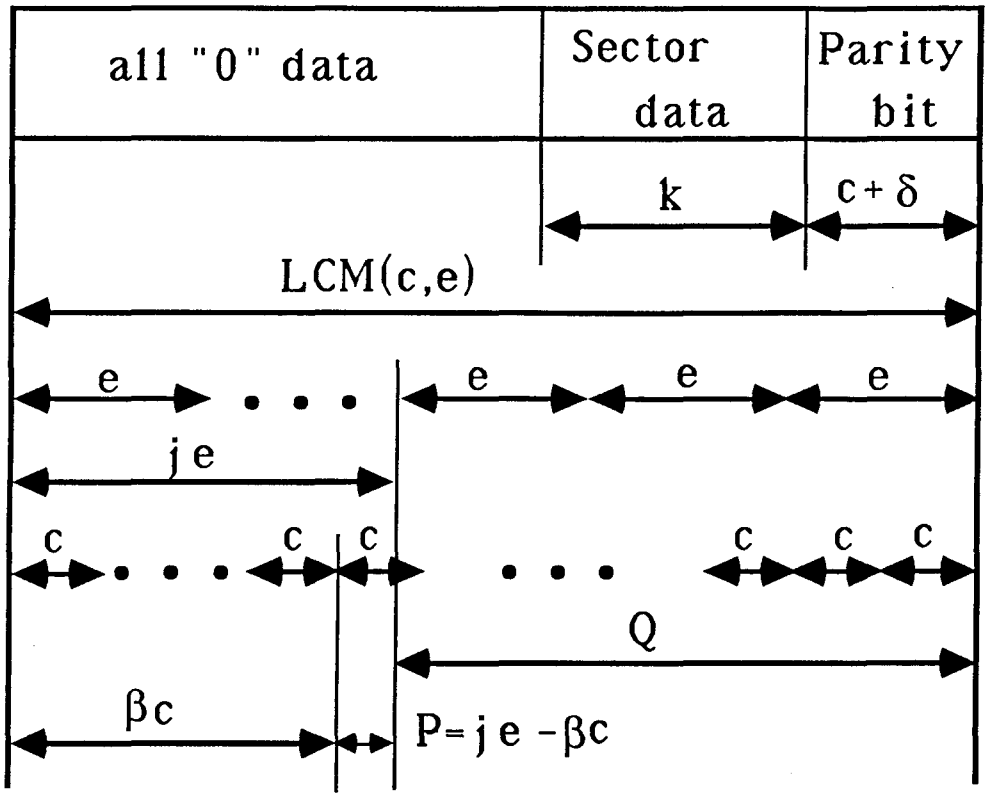



図 3.2　短縮化ファイヤ符号の符号語

シフトが必要となる。そして，この間に p(x)の周期 e が j 回存在する。ここで，次式が成り立つ。

$$\begin{aligned} j &= \lfloor \{ n - ( k + c + \delta ) \} / e \rfloor \dagger \\ &= n / e - \lceil ( k + c + \delta ) / e \rceil \end{aligned} \qquad ( 3 - 7 )$$

一方，$x^{je}$ を $S_0(x)$ に乗算して $( x^c + 1 )$ で除算すると，剰余 $S'_0(x)$ を生じる。すなわち，次式が成り立つ。

$$\begin{aligned} S'_0(x) &\equiv x^{je} S_0(x) \mod x^c + 1 \\ &\equiv x^{je-\beta c} S_0(x) \mod x^c + 1 \\ &\equiv x^p S_0(x) \mod x^c + 1 \end{aligned} \qquad ( 3 - 8 )$$

† $\lfloor b \rfloor$ は b の切り捨てを表す。

ここで，Pは，図3.2で示すようにjeをcで除算したときの剰余である。次式が成り立つ。

$$\begin{aligned} P &\equiv je \mod c \\ &\equiv \{ n/e - \lceil (k+c+\delta)/e \rceil \} \times e \mod c \\ &\equiv c - \{ \lceil (k+c+\delta)/e \rceil \times e \mod c \} \end{aligned} \qquad (3-9)$$

したがって，式（３－９）によってPを計算しておき，ＬＦＳＲ0をＰ回シフトすれば，ＬＦＳＲ0，ＬＦＳＲ1を同時にje回シフトしたことと等価になる。

以上まとめると，式（３－１）を生成多項式とするファイヤ符号の短縮化符号に対し，次の３ステップによって誤り位置iおよび誤りパターンB(x)の計算ができる。

［ステップ１］　受信符号$F^*(x)$から，式（3－2）にしたがって，シンドローム$S_0(x)$，$S_1(x)$を計算する。

［ステップ２］　ＬＦＳＲ0のみをＰ回シフトし，$S'_0(x)$を求める。このステップは，$S_0(x)$，$S_1(x)$に$x^{je}$を乗算し，それぞれ$(x^c+1)$，p(x)で除算したことに相当し，次式で表される。

$$\begin{aligned} S'_0(x) &\equiv x^{je} S_0(x) \mod x^e + 1 \\ &\equiv x^p S_0(x) \mod x^c + 1 \\ S_1(x) &\equiv x^{je} S_1(x) \mod p(x) \\ &\equiv S_1(x) \mod p(x) \end{aligned} \qquad (3-10)$$

［ステップ３］　式（３－１０）に$x^i$を乗算する。すなわち，次式の操作を行なう。

$$\begin{aligned} B_0(x) &\equiv x^i S'_0(x) \mod x^c + 1 \\ B_1(x) &\equiv x^i S_1(x) \mod p(x) \end{aligned} \qquad (3-11)$$

このステップで，$B_0(x)$と$B_1(x)$ 等しくなったとき，その時のiが誤り位置を表す。同時に，$B_0(x) = B_1(x)$が誤りパターンB(x)を表す。誤り位置iは，短縮化しない符号語の先頭からjeビットのオフセットを持つ。ステップ３において，シフト回数iが所定のシフトオーバ回数Qを越えても$B_0(x)$と$B_1(x)$が一致しなかったとき，訂正不能な誤りが発生したと判断する。Qの値は，図3.2にも示すように，次式で表される。

$$Q = \lceil (k+c+\delta)/e \rceil \times e \qquad (3-12)$$

以上３ステップにより，受信符号$F^*(x)$ から，誤り位置iと誤りパターンB(x)が計算で

きる。提案した復号法による最大シフト回数scは，ステップ2による最大（c－1）回のシフトと，ステップ3による最大シフト回数（シフトオーバ回数）を加えたものである。次式が成立する。

$$sc = c - 1 + \lceil (k + c + \delta) / e \rceil \times e \qquad (3-13)$$

### 3.2.4 例　題

次式の多項式を生成多項式とするファイヤ符号について，3.2.3の復号法を適用した例を示す。次式の多項式を考える。

$$G(x) = (x^{21} + 1)(x^{11} + x^2 + 1) \qquad (3-14)$$

多項式（$x^{11} + x^2 + 1$）は，原始多項式であり，周期$e = (2^{11} - 1)$である。ここで，符号長nは，次式で表される。

$$\begin{aligned} n &= LCM(21, 2047) \\ &= 42987 \end{aligned} \qquad (3-15)$$

検査部の長さは，32ビットである。このファイヤ符号は，長さ11ビットまでのバースト誤りを1個訂正できる。

この符号の復号用LFSR回路の例を図3.3に示す。このLFSR回路は，“中国人の剰余定理”を用いる復号法と同じ回路である。図3.3は，LFSR0とLFSR1からなっている。LFSR0は，21段のローテート回路であり，多項式（$x^{21} + 1$）の除算を行なう。11段のLFSR1は，多項式（$x^{11} + x^2 + 1$）の除算を行なう回路である。LFSR0, LSFR1とも11段目から受信符号$F^*(x)$を入力している。

図3.3において，一致判定回路COMPは，LFSR0の下位11ビットとLFSR1がビット毎に一致しているかどうか判定する回路である。ゼロ検出回路0－DETは，LFSR0の上位10ビットがオール0かどうか判定する回路である。

ステップ2におけるシフト回数Pおよびステップ3におけるシフトオーバ回数Qは，$c = 21$，$\delta = 11$，$e = 2047$をそれぞれ式（3－9），（3－12）に代入して，

$$\begin{aligned} P &= 21 - \{\lceil (k + 32) / 2047 \rceil \times 2047 \mod 21\} \\ Q &= \lceil (k + 32) / 2047 \rceil \times 2047 \end{aligned} \qquad (3-16)$$

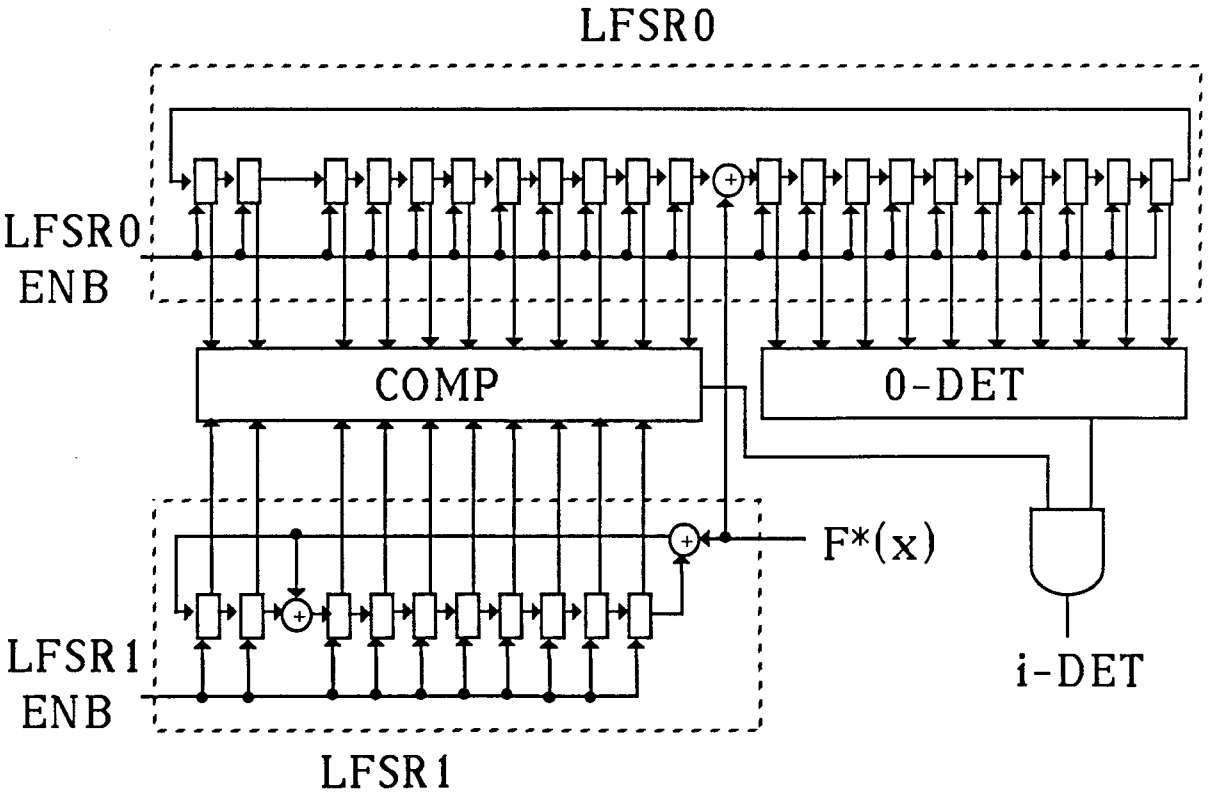



図 3.3　ファイア符号復号用フィードバックシフトレジスタ回路の例
$G(x)=(x^{21}+1)(x^{11}+x^{2}+1)$

と求めることができる。

ファイヤ符号をディスクシステムに応用する場合，ディスクのレコード長ＲＬは，１２８バイト，２５６バイト，……であることが多い。このとき，ｋ＝ＲＬ×８ビットとなる。いくつかのレコード長について，式（３－１６）を用いて，シフト回数Ｐ（ステップ２）とシフトオーバ回数Ｑ（ステップ３）を計算した例を，表3.1に示す。

表 3.1　提案する復号法におけるシフト回数Ｐ（ステップ２）とシフトオーバ回数Ｑ（ステップ３）の計算例。$G(x)=(X^{21}+1)(X^{11}+X^{2}+1)$

| ＲＬ | Ｐ | Ｑ |
|---|---|---|
| １２８バイト | １１ | ２０４７ |
| ２５６ | １ | ４０９４ |
| ５１２ | １２ | ６１４１ |
| １０２４ | １３ | １０２３５ |
| ２０４８ | １５ | １８４２３ |
| ４０９６ | １９ | ３４７９９ |

## 3.3　一般のバースト誤り訂正巡回符号への拡張

### 3.3.1　復号アルゴリズム

ここでは，ファイヤ符号の復号法を一般のバースト誤り訂正巡回符号へ拡張する。次式の多項式を生成多項式とするバースト誤り訂正符号について考える。

$$G(x)=(x^c+1)\prod_{j=1}^{h} p_j(x) \qquad (3-17)$$

ここで，$p_j(x)$は次数$\delta_j$，周期$e_j$の異なる既約多項式とする。ただし，$e_j$はcを割り切らない。符号長nは，$n=LCM(c, e_1, e_2, \cdots\cdots, e_h)$となる。式(3－17)で生成される符号は，長さd以下のバースト誤りを検出し，長さb以下かつ$p_j(x)$と互いに素な，バースト誤りを1個訂正できる[4]。ただし，$c \geqq b+d-1$，$b \leqq \delta_1+\delta_2+\cdots\cdots+\delta_h$とする。式（３－１７）の多項式の次数をmとすると，$m=\delta_1+\delta_2+\cdots\cdots+\delta_h$となる。式（３－１７）で生成されるバースト誤り訂正符号に対し，従来の復号法は，ファイヤ符号と同様，以下の方法が知られている。

（Ｄ1′）　巡回符号の性質を利用した方法[3]

（Ｄ2′）　シンドロームの線形性を利用してシフト回数を減少させる方法[4]

（Ｄ3′）　“中国人の剰余定理”を用いる方法[5]

これらの復号法は，ファイヤ符号に対する復号法と同様の長所，欠点を持つ。長所，欠点については省略するが，いずれの方法も3.2.2項（Ｄ１）～（Ｄ３）と同じ理由により，必ずしもＶＬＳＩ化に適しているとは言えない。

そこで，3.2.3項で示したファイヤ符号に対する新しい復号法と同様の特徴を持つ復号法を提案する。すなわち，ＬＦＳＲ0（除多項式（$x^c+1$）），LFSRj（除多項式$p_j(x)$）に対し，それぞれ以下のシフトが省略できることを利用する。

$$\begin{aligned} &n-\lceil(k+m)/c\rceil\times c \\ &n-\lceil(k+m)/e_j\rceil\times e_j \end{aligned} \qquad (3-18)$$

ファイヤ符号との違いは，周期$e_j$の選び方に自由度があることである。$e_1, e_2 \cdots\cdots, e_h$のうち$e_h$を選び，多項式$p_h(x)$を基準にして他のＬＦＳＲのシフトをおこなうと考えても一般性を失わない。式（３－１７）を生成多項式とする巡回符号の復号手順は，ファイヤ符号の場合と同様，次の３ステップからなる。

［ステップ１］　受信符号$F^{*}(x)$から，シンドローム$S_0(x)$，$S_1(x)$，……，$S_h(x)$を計算する。次式で表される。

$$\begin{aligned} S_0(x) &\equiv F^{*}(x) \mod x^{c}+1 \\ S_1(x) &\equiv F^{*}(x) \mod p_1(x) \\ &\cdots\cdots \\ S_h(x) &\equiv F^{*}(x) \mod p_h(x) \end{aligned} \qquad (3-19)$$

この計算は，ＬＦＳＲ０，ＬＦＳＲ１，……，ＬＦＳＲｈへ$F^{*}(x)$を入力することによっておこなわれる。

［ステップ２］　ＬＦＳＲｈを基準とし，ＬＦＳＲ０，ＬＦＳＲ１，……，ＬＦＳＲ(ｈ−１)をそれぞれ，$P_0$，$P_1$，……，$P_{h-1}$回シフトし，$S'_0(x)$，$S'_1(x)$，……，$S'_{h-1}(x)$を求める。各々のシフト回数は，3.2.3のシフト回数Ｐの導出と同様に考えて，以下のように表される。

$$\begin{aligned} P_0 &= c-\{\lceil(k+m)/e_h\rceil\times e_h \mod c\} \\ P_1 &= e_1-\{\lceil(k+m)/e_h\rceil\times e_h \mod e_1\} \\ &\cdots\cdots \\ P_{h-1} &= e_{h-1}-\{\lceil(k+m)/e_h\rceil\times e_h \mod e_{h-1}\} \end{aligned} \qquad (3-20)$$

このステップでは，ＬＦＳＲ０，ＬＦＳＲ１，……，ＬＳＦＲｈを，それぞれ，$\lceil(k+m)/e_h\rceil\times e_h$回シフトしたことに相当する。

［ステップ３］　ＬＦＳＲ０，ＬＦＳＲ１，……，ＬＳＦＲｈを同時にシフトし，誤り位置ｉと誤りパターンＢ(x)を計算する。すなわち，次式の操作を行なう。

$$\begin{aligned} B_0(x) &\equiv x^{p_0+i}S'_0(x) \mod x^{c}+1 \\ B_1(x) &\equiv x^{p_1+i}S'_1(x) \mod p_1(x) \\ &\cdots\cdots \\ B_{h-1}(x) &\equiv x^{p_{h-1}+i}S'_{h-1}(x) \mod p_{h-1}(x) \\ B_h(x) &\equiv x^{i}S_h(x) \mod p_h(x) \end{aligned} \qquad (3-21)$$

ステップ３において，シフト回数ｉが所定のシフトオーバ回数Ｑを越えても，$B_0(x)$，$B_1(x)$，……，$B_h(x)$が一致しなかったとき，訂正不能な誤りが発生したと判断する。この場合，

Qの値は，3.2.3の式（３－１２）と同様に考えて，次式で表される。

$$Q = \lceil (k+m) / e_h \rceil \times e_h \qquad (3-22)$$

以上の３ステップにより，式（３－１７）を生成多項式とする巡回符号に対し，受信符号$F^*(x)$から，誤り位置ｉと誤りパターン$B(x)$が計算できる。

### 3.3.2 例　　題

ここでは，一般のバースト誤り訂正巡回符号の例を挙げて，提案した復号法を具体的に説明する。次式で表される多項式を考える。

$$\begin{aligned} G(x) = &(x^{22}+1) \\ &\times (x^{12}+x^{11}+\cdots\cdots+x+1) \\ &\times (x^{11}+x^{9}+x^{7}+x^{6}+x^{5}+x+1) \\ &\times (x^{11}+x^{7}+x^{6}+x+1) \end{aligned} \qquad (3-23)$$

式（３－２３）の右辺に現われる４個の多項式を順に，$p_0(x)$，$p_1(x)$，$p_2(x)$，$p_3(x)$ と表す。$p_0(x)$，$p_1(x)$，$p_2(x)$，$p_3(x)$の周期は，それぞれ，２２，１３，２３，８９である[2]。符号長は，n＝LCM（２２，１３，２３，８９）＝５８５４２ である。この符号は，長さ１１ビットまでのバースト誤りを１個訂正できる。特に，$p_2(x)$はゴーレイ符号[2]を生成する多項式として知られている。

図3.4は，式（３－２３）を生成多項式とする巡回符号の復号回路を例示した図である。一致判定回路COMPは，LFSR0の下位１１ビットLFSR1の下位１１ビット，LFSR1の下位１１ビットとLFSR2の下位ビット，LFSR2の下位１１ビットとLFSR3，がビット毎に一致しているかどうか判定する。ゼロ検出回路0－DETは，LFSR0 の上位１１ビットとLFSR0の上位１ビットがすべて０かどうか判定する回路である。３個の一致判定回路とゼロ検出回路がすべてアサートされたとき，その時のシフト回数が誤り位置ｉを，各LFSRの内容が誤りパターン$B(x)$を表す。

ステップ２において，シフトの基準多項式を$p_3(x)$にしたときのシフト回数$P_0$，$P_1$，$P_2$およびステップ３におけるシフトオーバ回数Qの値は，式（３－２０），（３－２２）にそれぞれｃ＝２２，ｍ＝５６，$e_1$＝１３，$e_2$＝２３，$e_3$＝８９，ｋ＝ＲＬ×８を代入して求めることができる。ＲＬはレコード長を表す。いくつかのレコード長について，

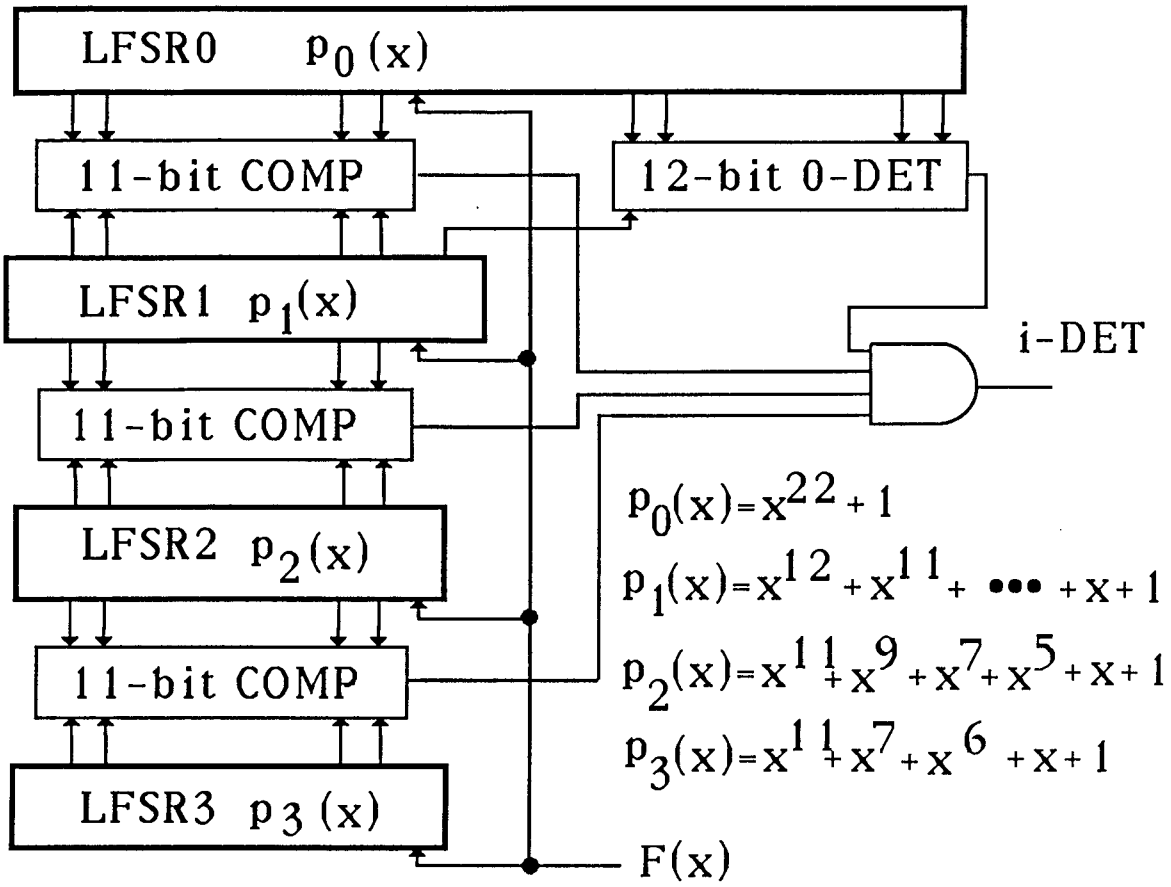



図 3.4　巡回符号復号回路の例
$G(x)=P_0(x)\times P_1(x)\times P_2(x)\times P_3(x)$

$P_0$, $P_1$, $P_2$, Qを計算した例を表3.2に示す。

表 3.2　ステップ2におけるシフト回数 $p_0$, $p_1$, $p_2$ とステップ3におけるシフトオーバ回数Qの計算例。生成多項式が式(3－23)，ステップ2におけるシフトの基準多項式を $p_3(x)$ とした場合。

| R | $P_0$ | $P_1$ | $P_2$ | Q |
|---|---|---|---|---|
| 128バイト | 9 | 0 | 16 | 1157 |
| 256 | 20 | 9 | 3 | 2136 |
| 512 | 19 | 3 | 3 | 4183 |
| 1024 | 17 | 4 | 3 | 8277 |
| 2048 | 13 | 6 | 3 | 16465 |
| 4096 | 5 | 11 | 3 | 32841 |

## 3.4 ＶＬＳＩプロセッサ（16ビットマイコン周辺ＬＳＩ）への応用

### 3.4.1 ハードディスクコントローラＶＬＳＩへの応用

提案したファイヤ符号のＶＬＳＩ向き復号法を，ハードディスクコントローラ（HDC）チップへ応用した。このＶＬＳＩプロセッサは，１６ビットマイクロプロセッサ用周辺ＬＳＩとして開発された[７，８]。ＨＤＣ内部には，ＥＣＳＰ（Error Correcting Satellite Processor）と呼ばれる誤り制御部があり，３２ビットファイヤ符号の符号化／復号化，および１６ビットＣＲＣ（Cyclic Redundancy Check）コードの生成／検査をおこなう。ファイヤ符号生成多項式は，3.2.4の式（３－１４）と同じく，次式で表されるものを用いた。

$$G(x)=(x^{21}+1)(x^{11}+x^{2}+1) \qquad (3-24)$$

図3.5に，ＨＤＣのチップ写真を示す。写真中の白線内が，ＥＣＳＰに相当する部分である。ＨＤＣは２５６バイトのデータバファを２面内蔵しており，レコード長ＲＬが256バイトに対しては，マイクロプログラムを用いてチップ内部にて誤り訂正をおこなうことができる。

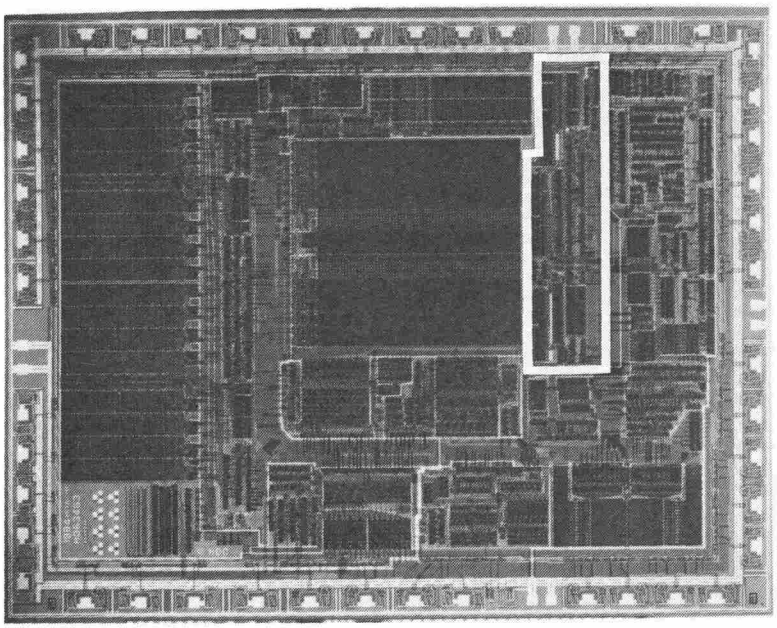


図3.5　16ビットマイコン周辺ＬＳＩ（ハードディスクコントローラ）のチップ写真

表3.3には，ＨＤＣならびにＥＣＳＰのハードウェア諸元を示す。ＥＣＳＰの素子数，チップ面積は，ＣＲＣ機能を含んだものである。ＨＤＣの総素子数は１２９Ｋトランジスタであり，チップ面積は６２.３㎟である。このうちＥＣＳＰは素子数3625トランジスタ（２.８％），チップ面積2.５２㎟（４.０％）を占める。また，誤り訂正に必要なマイクロプログラムのステップ数は，１３０ステップ（２４ビット幅）である。ＥＣＳＰとマイクロプログラムを加えた総素子数は6745トランジスタ（５.２％）である。製造プロセスは２μmＣＭＯＳ（Complementary Metal Oxide Semiconductor）である。

表3.3　ハードディスクコントローラおよび誤り制御部（ECSP）のハードウェア諸元

| 項　　目 | ＥＣＳＰ | 誤り訂正<br>マイクロプログラム | ＥＣＳＰ＋<br>マイクロプログラム | ＨＤＣ |
|---|---|---|---|---|
| 素子数<br>（％） | 3625Tr.<br>（2.8％） | 3120Tr.<br>（2.4％） | 6745Tr.<br>（5.2％） | 129KTr. |
| チップ面積<br>（％） | 2.52㎟<br>（4.0％） | 0.58㎟<br>（0.9％） | 3.10㎟<br>（4.9％） | 62.3㎟ |
| ステップ数<br>（％） | － | 130<br>（6.3％） | － | 2048 |
| プロセス | ２μmＣＭＯＳ | | | |

### 3.4.2　シリアルインターフェイスＶＬＳＩの設計法

マイコン周辺ＬＳＩでは，比較的類似した複数のＬＳＩを開発しなければならないことがある。たとえば，ハードディスクコントローラ，フロッピィディスクコントローラ，ネットワークコントローラ等が挙げられる。これらはいずれも，シリアルデータを扱うＬＳＩである。これらのＬＳＩに対し，個別に開発したのでは，期間あるいはコストの面で得策ではない。そこで，共通的に適用できるマイクロアーキテクチャを構築し，ＨＤＣを最初の適用品として応用した[９，１０]。

図3.6に，シリアルインターフェイスＶＬＳＩに共通的に適用できるマイクロアーキテクチャを示す。このマイクロアーキテクチャの特徴は，１個のKernel Processor(KP)と複数のSatellite Processor(ＳＰ)が階層的に接続されている点にある。ＫＰは入出力装置の状態監視や低速ポートの制御などをおこなう部分である。ＫＰはマイクロプログラムで制御されるため汎用性が高く，別の被制御装置に対しても，マイクロプログラムの変更だけで対処できる。

ＳＰは，高速（例えば２０Mb／Ｓ）のシリアルデータの変復調，直列／並列変換などを

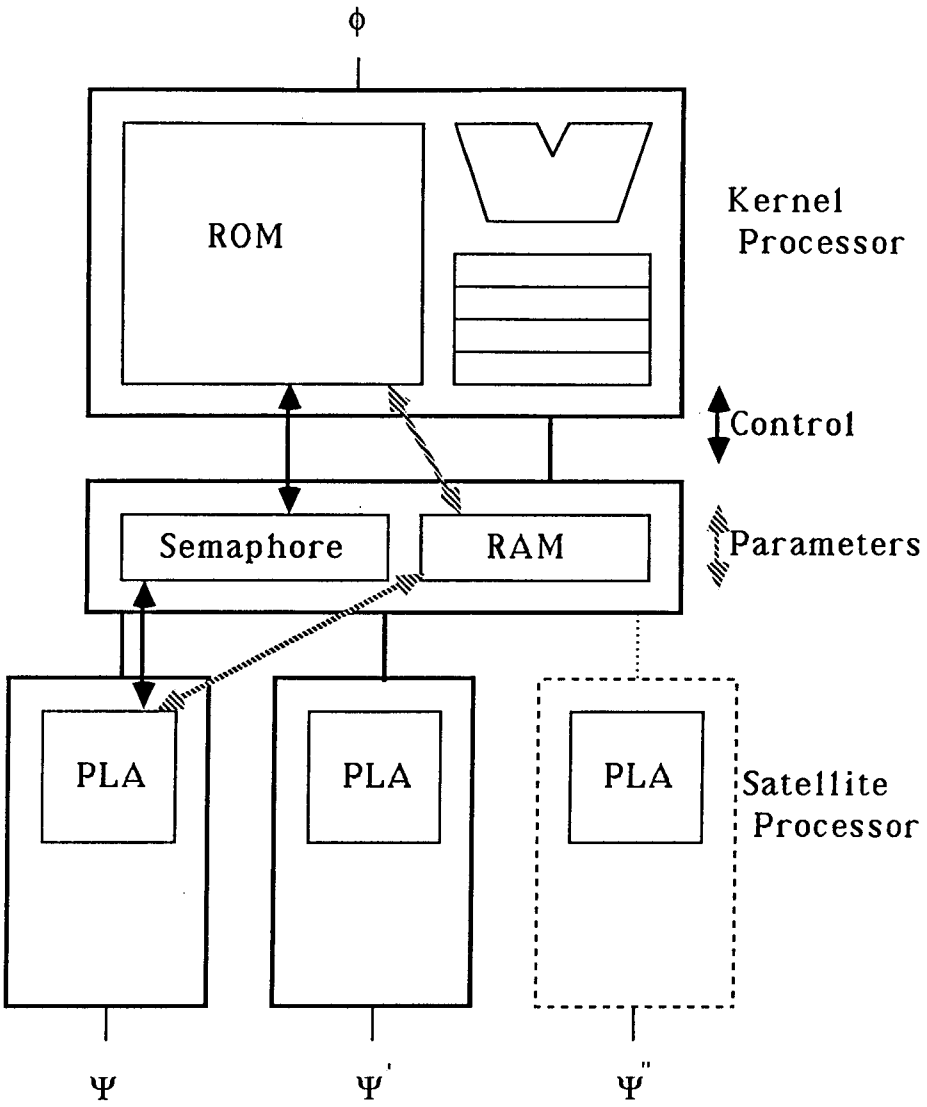



図３．６　シリアルインターフェイスＶＬＳＩの共通マイクロマーキテクチャ

おこなう部分である。この部分は，ＰＬＡによって制御されている。被制御装置に依存して設計し直す必要がある。

また，ＫＰとＳＰは異なるクロックで動作するため，制御およびパラメータの受渡しを工夫した。すなわち，パラメータ（データ長など）をスタチックメモリに格納し，制御信号（起動／終了など）は，クロック同期回路（Semaphore）を通しておこなうようにした。このようなＫＰ－ＳＰインターフェイスを提供することにより，ＫＰとＳＰは，互いに他の詳細を知らなくても動作が可能となる。また，別の設計者が並行してＫＰとＳＰを設計できるという効果もあった。以上のようなマイクロアーキテクチャをマイコン周辺ＬＳＩ――ＨＤＣ――へ適用した。ＨＤＣのチップ写真は図3.5に示した。

ＨＤＣの検査回路について簡単に述べる。ＨＤＣでは，検査コストの低減を目的として，階層構造と内部バスを利用した検査法を考案した［１１］。検査回路のハードウェア量はチップ全体の素子数に対し，約７％である。また，内部バスにシグナチャ回路を設けた。

## 3.5 まとめ

本章では，ＶＬＳＩプロセッサに適したファイヤ符号およびバースト誤り訂正巡回符号の復号法を提案した。提案した復号法は，ハードウェアが比較的少なくてよい。また，短縮化前の符号長がレコード長よりも十分に長い場合 —— 通常の応用では成り立つ ——，復号時間の面でも，他の高速復号法と同じオーダまで高速化することができた。また，一部の設定値を変更することにより，種々の短縮化に対して使用できるという特徴も持つ。この性質は，マイコン周辺ＬＳＩのように１個のチップで複数のレコード長を扱う場合に適しており，ＲＯＭなどの規則論理素子で実現される。

さらに，提案した復号法を具体的にＶＬＳＩプロセッサへ適用した。適用した符号は32ビットファイヤ符号であり，適用したプロセッサはハードディスクコントローラＶＬＳＩである。このＶＬＳＩチップは，１６ビットマイコン周辺ＬＳＩとして開発されたものである。提案した復号法を実行する誤り訂正部は，１６ビットＣＲＣ機能および誤り訂正用マイクロプログラムを含め，約6700トランジスタとなった。これは，ハードディスクコントローラＶＬＳＩの約５％を占める。また，シリアルインターフェイスＶＬＳＩの設計法についても述べた。

## 3.6 参考文献


[1] 嵩，都倉，岩垂，稲垣，“符号理論”コロナ社，1975年．

[2] 宮川，岩垂，今井，“符号理論，”昭晃堂，1973年．

[3] S.Lin,“An Introduction to error correcting codes,”Prentice-Hall, 1970.

[4] S.Lin and D.J.Costello,“Error control coding,”Prentice-Hall,1983.

[5] R.T.Chien,“Burst-correcting codes with high speed decoding, IEEE Trans. Inf. Theory,vol.IT-5,no.1,pp.109-113,Jan.1969.

[6] 岩崎，船橋，上野，“短縮化ファイヤ符号の復号法とＶＬＳＩプロセッサへの応用，”信学論(D),vol.J68-D,no.12,pp.1998-2006,1985年12月．

[7] T.Funabashi,K.Iwasaki,K.Minorikawa,H.Yonezawa and T.Cantrell,“2 micron CMOS VLSI hard disk controller integrates data buffer and error correction,”WESCON, 24/3, pp.1-7, U.S.A., 1984.

[8] T.Funabashi,J.Tatezaki and K.Iwasaki.“CMOS hard disk controller

with data buffer and error correction," Hitachi review, vol.33, no. 5, pp. 251−254, Oct. 1984.

[9] K.Iwasaki,N.Yamaguchi,T.Shimura,Y.Hagiwara,T.Funabashi and K.Minorikawa, "Design methodology for reconfigurable module-structured VLSI,"Custom integrated circuits conf,pp.464−467, U.S.A.,May 1985.

[10] 岩崎，船橋，山口，志村，御法川，館崎，"階層構造ＶＬＳＩ設計法とハードディスクコントローラ（ＨＤＣ）への応用，"電子通信学会半導体トランジスタ研究会, SSD84−56,1984年9月.

[11] N.Yamaguchi,T.Funabashi,K.Iwasaki,T.Shimura and K.Minorikawa, "A self−testing method for modular structured logic VLSIs,"Int. conf.CAD,pp.99−101,Jan.1984.

[12] C.E.Mead and L.Conway,"Introduction to VLSI systems," Addison−Wesley,1980.

[13] S.Muroga,"VLSI system design," John-Wiley, 1982.

# 第4章　シグナチャ検査法の誤り検出率の解析とランダム誤り訂正符号

## 4.1　まえがき

半導体集積度の向上と共に，1チップに集積できる素子数は飛躍的に増加している。一方，外部ピンの数はさほど増えていない。このような状況のもとで，論理回路の検査法への関心が高まっている[1,2]。

ＬＳＩ回路の論理検査では，Multiple Input Signature Register(MISR)と呼ばれる多入力線形帰還シフトレジスタを，パターン圧縮器として用いた自己検査法が提案されている[3,4]。ＭＩＳＲはパターン圧縮に伴い，パターンに含まれる様々の誤りを見逃す[5,6]。この見逃し率を$2^{-m}$から$2^{-2m}$（mはＭＩＳＲの段数）へ向上させる方法も提案されている[7]。しかし，ランダムな多重誤り検出率に関しては，理論的解析が十分ではなかった。すなわち，誤り見逃しの例はいくつも知られていたが，一般的条件は定式化されていなかった。

本章では，まずＭＩＳＲに対し，回路の線形性を利用して，シグナチャS(x)を定式化する。次に，G等価という考え方を提案し，原始多項式に基づくＭＩＳＲの２重ビット誤りの検出率を求める。その結果，２重ビット誤り検出率が１００％とならないことを示す。さらに，行列の縮退化という考え方を提案する。この考え方とハミング符号の重み３，４の符号語数から，原始多項式に基づくＭＩＳＲの３，４重ビット誤り検出率を計算する。同時に，拡大ハミング符号生成多項式に基づくＭＩＳＲの１～４重ビット誤り検出率を求め，２重ビット誤り検出率が１００％とならないことを示す[8,9]。

２重ビット誤り検出率を１００％にする一つの方法として，逆２重化ＭＩＳＲというシグナチャ回路を提案する。この回路は，シフト方向が互いに逆方向である２組のＭＩＳＲを用いる方法である。２重ビット誤りが互いに打ち消し合う条件が，逆方向シフトの２つのＭＩＳＲを用いると，同時には生じないことを利用したものである。特に，拡大ハミング符号生成多項式に基づく逆２重化ＭＩＳＲを用いると，１～３重ビット誤りをすべて検出できることを示す[8,9]。

## 4．2　検査モデルとＭＩＳＲの諸性質

### 4．2．1　検査モデル

本章では，図4.1で示されるような論理回路の検査モデルを扱う。被検査回路は，検査パターンにしたがって，m個のビット出力を同時にパターン圧縮回路に送出する。これらのビット出力の長さをnとし，それぞれのビット時系列をm個の（n－1）次多項式 $a_0(x), a_1(x), \cdots\cdots, a_{m-1}(x)$ で表すことにする。すなわち，多項式 $a_i(x)$ はビットの時系列 $a_{i0}, a_{i1}, \cdots\cdots, a_{in-1}$ の多項式表現であり，次式で表される。

$$a_i(x) = a_{i0} + a_{i1}\,x + \cdots\cdots + a_{in-1}\,x^{n-1} \qquad (4-1)$$

パターン圧縮回路へは，$a_{in-1}, a_{in-2}, \cdots\cdots, a_{i0}$ の順で入力される。パターン圧縮回路（ＭＩＳＲ等）は，$a_0(x), a_1(x), \cdots\cdots, a_{m-1}(x)$ をmビットに圧縮し，その結果をシグナチャ多項式S(x)として故障判定回路へ出力する。

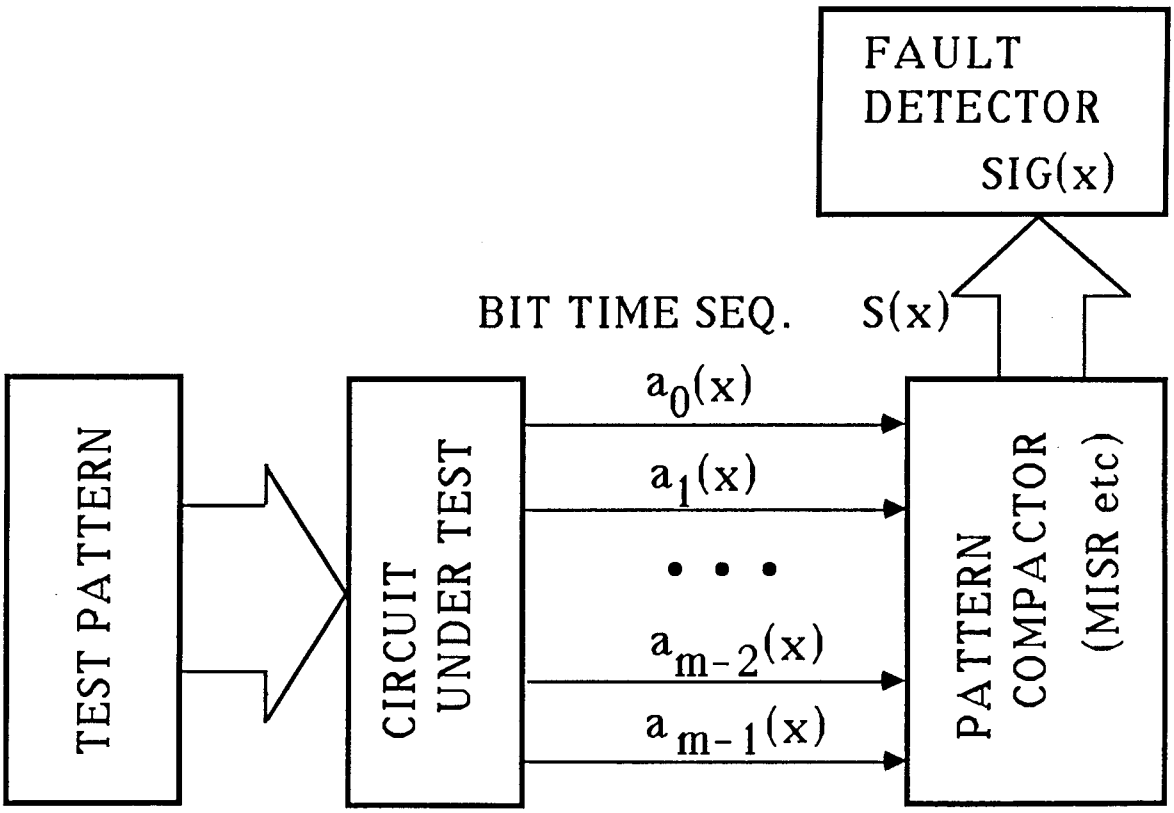



図4．1　論理回路の検査モデル

故障判定回路では，シグナチャS(x)と，被検査回路に故障がない場合のシグナチャSIG(x)とを比較し，S(x)＝SIG(x)のとき故障なし，S(x)≠SIG(x)のとき故障ありと判断する。

本章では，$a_0(x)$, $a_1(x)$, ……, $a_{m-1}(x)$に含まれる誤りはランダムビット誤りと仮定する。また，パターン圧縮回路，故障判定回路には故障がないものと仮定する。

### 4・2・2　MISRとシグナチャ

MISRは，図4.2で示すように，m段のシフトレジスタ，m個の排他的論理和ゲート，（m－1）個の係数器からなる線形帰還シフトレジスタである。MISRに対してシグナチャS(x)を定義する。シグナチャS(x)とは，入力検査多項式$a_0(x)$，$a_1(x)$，……，$a_{m-1}(x)$をMISRへ入力し終えたときのMISRの内容と定義する[2，5]。以下に，MISRが線形順序回路であることを利用して，シグナチャS(x)を定式化する。

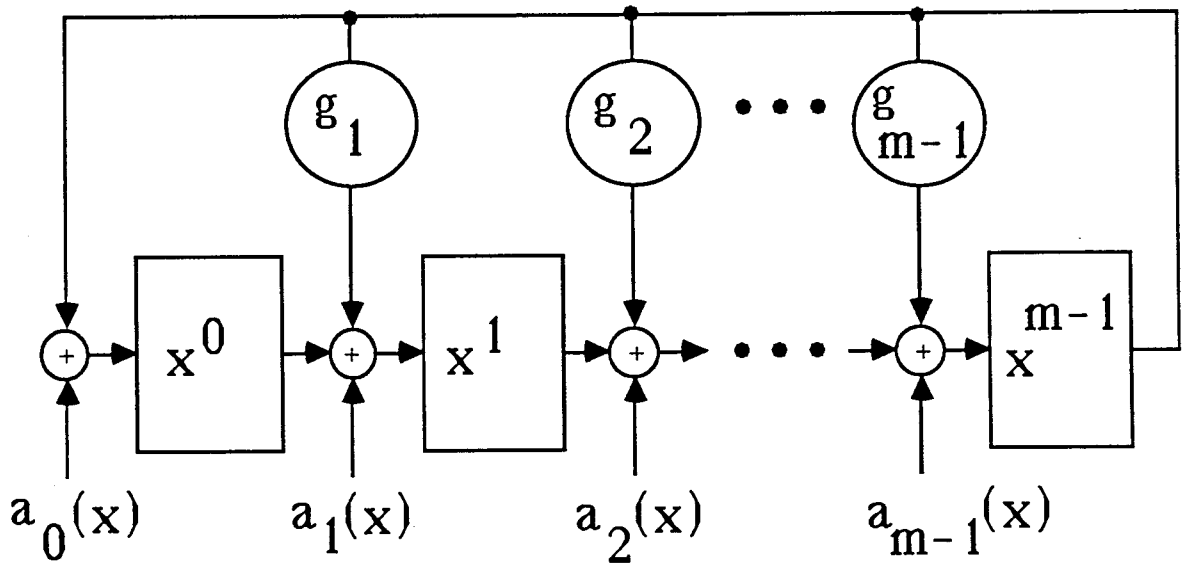



図4．2　多入力シグナチャレジスタ（MISR）回路

まず，m個の入力検査多項式$a_0(x)$，$a_1(x)$，……，$a_{m-1}(x)$のうち$a_i(x)$だけが非零多項式 —— 他の多項式はすべて0 —— であると仮定する。このとき，図4.3で示されるような線形帰還シフトレジスタとなる。これは除算回路としてよく知られた回路であり[6，7]，除多項式G(x)は，次式で表される。

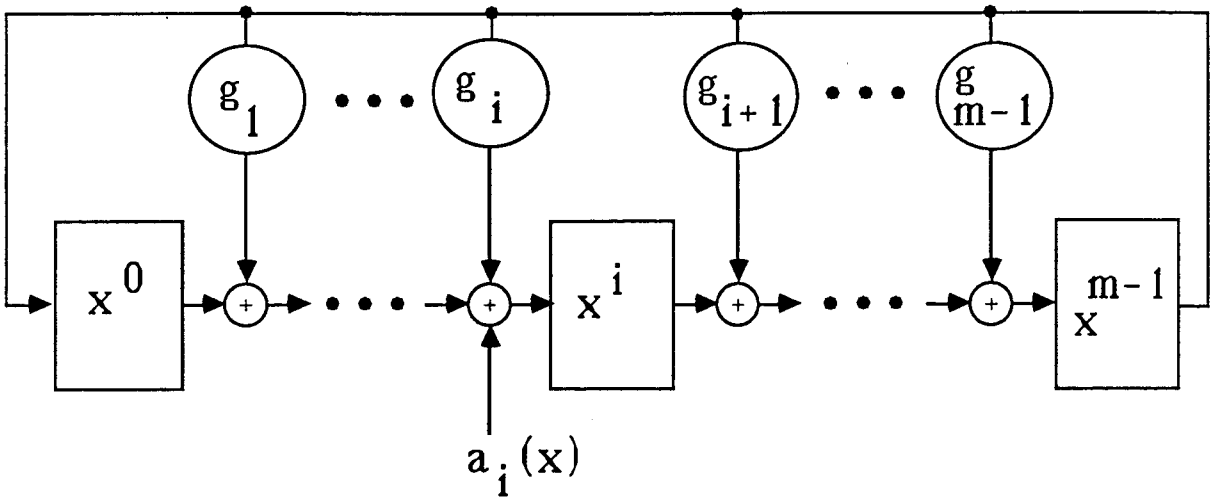



図４．３　$a_i(x)$のみが非零多項式であるときのＭＩＳＲ

$$G(x) = g_0 + g_1 x + \cdots + g_{m-1} x^{m-1} + g_m x^m \qquad (4-2)$$

ただし，$g_m = g_0 = 1$ である。$a_i(x)$ は i 段目のシフトレジスタに入力されるため，被除多項式は，次式で表される。

$$x^i a_i(x) \qquad (4-3)$$

入力検査多項式$a_0(x)$, $a_1(x)$, …… , $a_{m-1}(x)$のうち$a_i(x)$ だけが非零多項式とした場合のシグナチャを$S_i(x)$ と表すと，次式が成立する。

$$S_i(x) \equiv x^i a_i(x) \mod G(x) \qquad (4-4)$$

各々の入力検査多項式が非零多項式とした場合，シグナチャ S(x)は式（4−4）を用いて求めることができる。すなわち，ＭＩＳＲは $a_i(x)$ に対して線形であるから，$S_0(x)$, $S_1(x)$ , …… , $S_{m-1}(x)$の線形和が S(x)と一致する。このとき，次式が成立する。

$$\begin{aligned} S(x) &= S_0(x) + S_1(x) + \cdots\cdots + S_{m-1}(x) \\ &\equiv a_0(x) + x a_1(x) + \cdots\cdots + x^{m-1} a_{m-1}(x) \mod G(x) \end{aligned} \qquad (4-5)$$

### ４．２．３　検査シンドロームと誤り検出

検査シンドローム Y(x)を定義する。被検査回路に故障が含まれていない場合のシグナチ

ャをＳＩＧ(x)と表す。被検査回路のシグナチャをS(x)とするとき，

$$Y(x) \equiv SIG(x) + S(x) \mod G(x) \qquad (4-6)$$

と定義する。このとき，故障検出回路は次のように判定する。

Y(x)＝０のとき　：　被検査回路に故障なし

≠０のとき　：　被検査回路に故障あり

ここで，被検査回路に故障があってもY(x)＝０となる場合があり，これについて本章で議論する。

ここまでは，入力検査パターンを多項式の組として表してきた。このビットパターンをｍ×ｎ次の行列として表現することも可能であり，次式のように表す。

$$A = \begin{bmatrix} a_{00} & a_{01} & & a_{0n-1} \\ a_{10} & & & \\ & \cdots & & \\ a_{m-10} & & & a_{m-1n-1} \end{bmatrix} \qquad (4-7)$$

多項式表現も行列表現も同じものの別な表し方であり，扱いやすいほうを使用する。被検査回路に故障が含まれないときの入力検査パターンを特に行列ＥＦＡと表し，行列ＥＦＡの第ｉ行をｅｆａｉ(x)と表す。行列Ａの重みとは，行列Ａに含まれる１の個数と定義し，w(A)と表す。w(EFA⊕A)＝ｔのとき，Ａにｔ重誤りが生じたと定義する。ただし，Ａ⊕ＥＦＡは行列Ａと行列ＥＦＡの成分毎の排他的論理和を表す。

誤り行列ＥをＥ＝ＥＦＡ⊕Ａと定義する。行列Ｅの第ｉ行をｅ (x) と表すとき，次式が成立する。

$$e_i(x) = efa_i(x) + a_i(x) \qquad (4-8)$$

式(4－5)，(4－6)，(4－8)より，次式が成立する。

$$Y(x) \equiv e_0(x) + x e_1(x) + \cdots + x^{m-1} e_{m-1}(x) \mod G(x) \qquad (4-9)$$

ここで，ｔ重誤り検出率$PD_t$，ｔ重誤り見逃し率$PM_t$を定義する。w(EFA⊕A)＝ｔかつY(x)≠０となる確率を$PD_t$，w(EFA⊕A)＝ｔかつY(x)＝０となる確率を$PM_t$と定義する。このとき次式が成立する。

$$PD_t + PM_t = 1 \qquad (4-10)$$

### 4．2．4　G等価とその諸性質

多項式G(x)の周期［6］をnと表す。G(x)がm次の原始多項式の場合，$n=2^m-1$である。m×n次の入力検査パターン行列Aの成分に対し，G等価という性質を定義する。行列Aの2成分$a_{kl}$，$a_{k'l'}$が次の条件，

（条件1）　$k+l \equiv k'+l' \quad \mathrm{mod}\ n$　　（4－11）

を満たすとき，$a_{kl}$，$a_{k'l'}$はG等価であると定義する。また，$a_{kl}$，$a_{k'l'}$をG等価対と呼ぶ。誤り行列Eの成分$e_{kl}$，$e_{k'l'}$についても同様にG等価の性質を定義する。G等価対の例を図4.4に示す。図4.4に示すように，G等価対は従対角線（右上から左下）の方向に存在している。G等価の関係にある2成分はつぎの補題で示されるような性質を持つ。

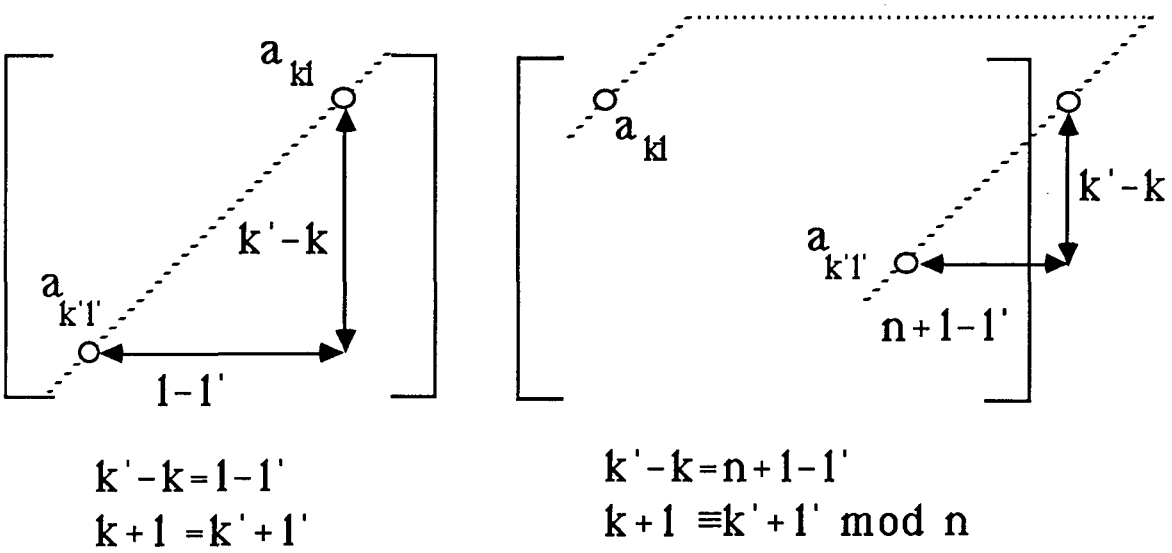



図4．4　G等価対の例

〔補題4.1〕

行列Aの異なる2成分$a_{kl}$，$a_{k'l'}$がG等価であるとき，行列Aのなかで$a_{kl}$のみが非零成分としたときのシグナチャS(x)と$a_{k'l'}$のみが非零成分としたときのシグナチャS′(x)とは等しくなる。

（証明）

行列Aのなかで$a_{kl}$のみが非零成分とする。このとき，入力検査多項式$a_0(x)$，$a_1(x)$，……，$a_{m-1}(x)$のうち$a_k(x)$のみが非零多項式となり，

$$a_k(x) = x^l \qquad (4-12)$$

となる。$a_k(x)$はＭＩＳＲのｋ段目に入力されるから，シグナチャ$S(x)$は，式（４－５）より，

$$S(x) \equiv x^{k+l} \mod G(x) \qquad (4-13)$$

となる。同様に，$a_{k'l'}$のみが非零成分としたとき次式が成立する。

$$S'(x) \equiv x^{k'+l'} \mod G(x) \qquad (4-14)$$

いま（条件１）が成立しているので，次式が成り立つ。

$$k + l = b \times n + k' + l' \qquad (4-15)$$

ただし，ｂ＝－１，０，１のいずれかである。また，$G(x)$はｍ次の原始多項式なので周期ｎに対し，

$$x^n \equiv 1 \mod G(x) \qquad (4-16)$$

である。よって，

$$\begin{aligned} S(x) &\equiv x^{k+l} && \mod G(x) \\ &\equiv x^{b\times n+k'+l'} && \mod G(x) \\ &\equiv S'(x) && \mod G(x) \end{aligned} \qquad (4-17)$$

となる。

Ｇ等価対の一例を示し，シグナチャ$S(x)=0$となる様子を説明する。

［例4.1］

行列Ａにおいて成分$a_{01}$と$a_{10}$はＧ等価である。実際，成分$a_{01}$と$a_{10}$のみが非零である場合，すなわち，

$$A = \left(\begin{array}{cc|c} 0 & 1 & \\ 1 & 0 & 0 \\ \hline 0 & & 0 \end{array}\right)$$

のときを考える。これをＭＩＳＲへ入力したならば，$a_{01}$と$a_{10}$は０段目と１段目の間の排他的論理和回路により打ち消しあい，シグナチャ$S(x)$は０となる。

（例終わり）

次に，２重誤り検出率を計算するために必要な補題を示す。

［補題 4.2 ］

行列Ａをｍ×ｎ次の行列とする。行列ＡにＧ等価対を１組だけ指定する組み合わせの数は次式で表される。

$$m(m-1)n/2 \qquad (4-18)$$

（証明）

行列Ａの成分の数はｍ×ｎである。各成分について従対角線方向に（ｍ－１）個のＧ等価対を持つ。単純な乗算ｍ×ｎ×（ｍ－１）では数え上げが２回づつおこなわれるため，行列ＡにＧ等価対を１組だけ指定する組み合わせの数は式（４－１８）で与えられる。

（証明終わり）

次に，４重誤り検出率を計算するために必要な補題を示す。

［補題 4.3 ］

行列Ａをｍ×ｎ次の行列とする。行列Ａから２組のＧ等価対を選び出す組み合わせの数は次式で表される。ただし，$m \geqq 4$ とする。

$$1/24 \times m(m-1)n\{(m-2)(m-3)+3m(m-1)(n-1)\} \qquad (4-19)$$

（証明）

最初のＧ等価対の選び出し方の総数は，補題 4.2 より，$1/2 \times m(m-1)n$ である。２組目のＧ等価対の選び方を次の２つの場合に分けて考える。

(a)　最初のＧ等価対と２番目のＧ等価対がＧ等価である場合

最初のＧ等価対によって従対角線上に既に２個の成分が指定されている。２番目のＧ等価対を選ぶ場合の数は，残りの（ｍ－２）個から２個選ぶ場合の数 ${}_{m-2}C_2$ に等しい。ただし，$m \geqq 4$ とする。４個の成分は互いにＧ等価の関係にあり，${}_4C_2=6$回だけ重複して数え上げられる。

(b)　最初のＧ等価対と２番目のＧ等価対がＧ等価でない場合

行列Ａの成分のうち，最初のＧ等価対とＧ等価の関係にない成分の数は$m(n-1)$個であり，各成分について（ｍ－１）個のＧ等価成分の選択ができる。この場合，最初のＧ

等価対と２番目のＧ等価対はそれぞれ２重に数えられる。以上，(a)，(b)より次式が成り立つ。

$$1/2\times m(m-1)n\{1/6\times {}_{m-2}C_2+m(n-1)(m-1)(1/2)^2\} \qquad (4-20)$$
$$=1/24\times m(m-1)n\{(m-2)(m-3)+3m(m-1)(n-1)\}$$

（証明終わり）

### 4．2．5　行列の縮退化

３重以上の誤り検出率を考えるために，誤り行列Ｅの縮退化という考え方を定義する。ｍ×ｎ次の行列Ｅにおいて，第０，１，……，（ｍ－２）行に含まれる任意の非零成分$e_{ij}$に対し，第（ｍ－１）行の成分のうちで$e_{ij}$とＧ等価な成分$e_{m-1j'}$が１個だけ存在する。すべての$e_{ij}$に対し，次式の操作を施すことを行列ＥのＧ縮退化と定義する。

$$\begin{aligned} e_{m-1j'} &\leftarrow e_{m-1j'}+e_{ij} \\ e_{ij} &\leftarrow 0 \end{aligned} \qquad (4-21)$$

このときの第（ｍ－１）行の多項式表現をＤＧ(x)と表す。補題4.1と式（４－９），(4－21）から行列Ｅによる検査シンドロームとＤＧ(x)×$x^{m-1}$による検査シンドロームは等しい。すなわち，次式が成立する。

$$Y(x)\equiv DG(x)\times x^{m-1} \mod G(x) \qquad (4-22)$$

図4.5に行列ＥのＧ縮退化の例を示す。縮退化した行列の最下行がＤＧ(x)となる。次の補題はＧ縮退化の定義からただちに導かれる。

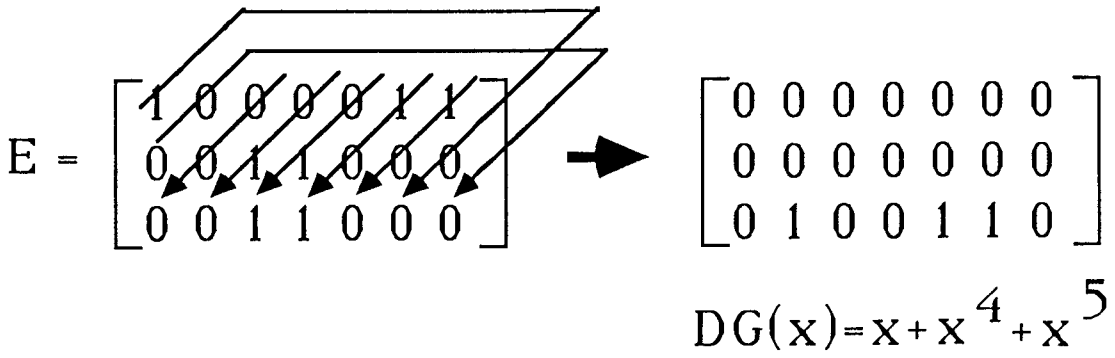



図４．５　行列ＥのＧ縮退化の例

［補題 4.4 ］

行列Eの重みをwとし，EのG縮退化多項式DG(x)の項数がwに等しい —— EはG等価な非零成分を含まない —— とする。このときDG(x)を生じるような行列Eの組み合わせの数は$m^w$である。

（証明略）

## 4.3 ＭＩＳＲの2,3および4重誤り検出率の解析

### 4.3.1 原始多項式に基づくＭＩＳＲ

最も基本的なＭＩＳＲとして，Ｇ(Ｘ)がｍ次の原始多項式である場合を考える。このときＧ(Ｘ)は符号長($2^m-1$)のハミング符号を生成する。ハミング符号は重み２以下の符号語を含まないので，Ｇ(Ｘ)は次数($2^m-1$)以下の単項および２項の多項式を整除しない。符号語の重みとは，符号語に含まれる１の個数である。ＭＩＳＲに対する入力ビット系列の長さを($2^m-1$)として，以下単一，２重，３重，４重ビット誤りについて検出率を求める。

#### 4.3.1.1 単一誤りが生じた場合

$$e_{ij} = \begin{cases} 1 & (i, j)=(k_0, l_0) \\ 0 & \text{上記以外の}(i, j) \end{cases}$$

とする。このとき，式(4-9)より

$$Y(X)\equiv X^{k_0+l_0} \mod G(X) \qquad (4-23)$$

となる。Ｇ(ｘ)は単項多項式を整除しないので，

$$Y(X)\not\equiv 0 \mod G(X)$$

が成立し，単一誤りを検出する。

#### 4.3.1.2 ２重誤りが生じた場合

$$e_{ij} = \begin{cases} 1 & (i, j)=(k_0, l_0)\ (k_1, l_1) \\ 0 & \text{上記以外の}(i, j) \end{cases}$$

とする。このとき，

$$Y(X)\equiv X^{k_0+l_0}+X^{k_1+l_1} \mod G(X) \qquad (4-24)$$

となる。次の２つの場合に分ける。

(a) ２個の非零成分$e_{ij}$がＧ等価対の場合

Ｇ等価対の定義から，

$$k_0+l_0\equiv k_1+l_1 \mod 2^m-1$$

である。よって，

$$Y(X)\equiv 0 \mod G(X)$$

となり，この２重誤りを見逃してしまう。

(b)　２個の非零成分$e_{ij}$がG等価対でない場合

検査シンドローム$Y(X)$ $(\equiv X^{m-1} \times DG(X) \mod G(X))$ は２項を含む多項式となる。$G(X)$は次数$n$（$=2^m-1$）以下の２項多項式を整除しないから，

$$Y(X) \not\equiv 0 \mod G(X)$$

が成り立ち，この２重誤りを検出する。

以上(a)，(b)よりG等価の位置に生じた２重誤りを見逃してしまう。$m \times n$次の行列Eのなかに G等価を１組だけ選択する場合の数は，補題4.2より，$1/2\,m(m-1)n$である。２重誤りが生じる場合の数は，${}_{mn}C_2$である。よって，２重誤り見逃し率$PM_2$は，

$$PM_2 = 1/2 \times m(m-1)n / {}_{mn}C_2 \qquad (4-25)$$

となる。$n=2^m-1$を代入し次式を得る。

$$PM_2 = 1/2 \times m(m-1)(2^m-1) / {}_{m(2^m-1)}C_2 \qquad (4-26)$$

4.3.1.3　３重誤りが生じた場合

$$e_{ij} = \begin{cases} 1 & (i,j)=(k_0,l_0)\ (k_1,l_1)\ (k_2,l_2) \\ 0 & \text{上記以外の}(i,j) \end{cases}$$

とする。このとき，

$$Y(X) \equiv X^{k_0+l_0} + X^{k_1+l_1} + X^{k_2+l_2} \mod G(X) \qquad (4-27)$$

となる。次の２つの場合に分ける。

(a)　G等価対を含む場合

$(i,j)=(k_0,l_0)(k_1,l_1)$がG等価対とみなしても一般性を失わない。

式(4-27)は，

$$Y(X) \equiv X^{k_2+l_2} \mod G(X)$$
$$\not\equiv 0 \mod G(X)$$

となり，この３重誤りを検出する。

(b)　G等価対を含まない場合

式(4-27)は３項多項式となる。検査シンドローム$Y(X)$ $(\equiv DG(X) \times X^{m-1} \mod G(X))$がG(X)で整除されるということは，Y(X)がG(X)の倍数であることに等しい。すなわち，G(X)が生成するハミング符号をCとするとY(X)∈Cと同値である。G(X)の周期はY(X)の最高次数よりも１だけ大きいので，式(4-27)が割り切れる場

合の数は，符号Cに含まれる重み3の符号語数に等しい。

G(X)が生成する符号語のうち重み3の符号語数は，符号Cのパリティ検査行列の列ベクトルのうち3個の列ベクトルの和が0となるような組み合わせの数に等しい[8]。ハミング符号では，2個の列ベクトルを任意に選んだとき，3個の列ベクトルの和が0となるように残りの列ベクトルを選ぶ方法は，一意に決定される。2個の列ベクトルを選ぶ場合の数は${}_nC_2$であるが，重み3の符号語は3重に数え上げられる。よって，重み3の符号語数は，

$$1/3 \times {}_nC_2 \qquad (4-28)$$

となる。3項多項式を生成するG縮退化前の誤り行列Eの数は，補題4.4と式(4-28)より，

$$m^3 \times 1/3 \times {}_nC_2 \qquad (4-29)$$

となる。式(4-29)は，式(4-27)の右辺が0となる場合の数を表し，このとき3重誤りを見逃してしまう。

以上(a)，(b)より，3重誤りを見逃す場合の数は，式(4-29)で表される数だけである。3重誤りが生じる場合の数は，${}_{mn}C_3$であり，次式が成立する。

$$PM_3 = 1/3 \times m^3 \times {}_nC_2 / {}_{mn}C_3 \qquad (4-30)$$

となる。$n = 2^m - 1$を代入し次式を得る。

$$PM_3 = 1/3 \times m^3 \times {}_{(2^m-1)}C_2 / {}_{m(2^m-1)}C_3 \qquad (4-31)$$

4.3.1.4　4重誤りが生じた場合

$$e_{ij} = \begin{cases} 1 & (i, j) = (k_0, l_0)(k_1, l_1)(k_2, l_2)(k_3, l_3) \\ 0 & \text{上記以外の}(i, j) \end{cases}$$

とする。このとき，

$$Y(X) \equiv X^{k_0+l_0} + X^{k_1+l_1} + X^{k_2+l_2} + X^{k_3+l_3} \mod G(X) \qquad (4-32)$$

となる。式(4-32)は，0，2，4項のいずれかを含む多項式となる。

(a)　G等価対が2組含まれる場合

式(4-32)の右辺は0となり，この誤りを見逃してしまう。このような4成分の場合の数は，補題4.3で与えられる。

(b)　G等価対が1組だけ含まれる場合

式(4-32)の右辺は2項を含む。4.3.1.2(b)と同じ理由でこの誤りを検出する。

(c)　G等価対を含まない場合

検査シンドローム$Y(X)$ $(\equiv DG(X)\times X^{m-1} \mod G(X))$は4項を含む。 $DG(X)$が$G(X)$で割り切れる場合の数は，$G(X)$が生成するハミング符号に含まれる重み4の符号語数に等しい。$G(X)$が生成する符号語のうち重み4の符号語数は，符号Cのパリティ検査行列の列ベクトルのうち4個の列ベクトルの和が0となるような組み合わせの数に等しい[8]。すなわち，3個の列ベクトルを選んで一次独立である場合の数を求めることに等しい。3個の列ベクトルの選び方は，${}_nC_3$通り存在し，このうち，${}_nC_2$通りは3個の列ベクトルが一次従属となる。よって，重み4の符号語数は，

$$1/4\times({}_nC_3-{}_nC_2) \tag{4-33}$$

となる。補題4.4および式(4-33)より，式(4-32)が$G(x)$で割り切れるような4成分の選び方は，

$$m^4\times 1/4\times({}_nC_3-{}_nC_2) \tag{4-34}$$

となる。この場合には，4重誤りを見逃してしまう。以上(a)，(b)，(c)より，4重誤りを見逃し率$PM_4$を求めることができる。4重誤りが生じる場合の数は，${}_{mn}C_4$である。$n=2^m-1$を代入し次式を得る。

$$\begin{aligned}PM_4=[\,&1/24\times m(m-1)(2^m-1)\{(m-2)(m-3)\\&+3m(m-1)(2^m-2)\}+1/4\times m^4\times\{{}_{(2^m-1)}C_3\\&-{}_{(2^m-1)}C_2\}]/{}_{m(2^m-1)}C_4\end{aligned} \tag{4-35}$$

4.3.1項のまとめとして，$m$次の原始多項式$G(X)$に基づくMISRの誤り検出率を表4.1にまとめる。表4.1から，単一誤りは必ず検出するが，2重以上の誤りについては，100%は検出できないことがわかる。

表4.1　G(x)がm次の原始多項式の場合のMISRの誤り検出率

| | |
|---|---|
| $PD_1$ | 1 |
| $PD_2$ | $1-1/2\times m(m-1)(2^m-1)/{}_{m(2^m-1)}C_2$ |
| $PD_3$ | $1-1/3\times m^3\times{}_{(2_m-1)}C_2/{}_{m(2^m-1)}C_3$ |
| $PD_4$ | $1-[1/24\times m(m-1)(2^m-1)\{(m-2)(m-3)+3m(m-1)(2^m-2)\}+1/4\times m^4\times\{{}_{(2^m-1)}C_3-{}_{(2^m-1)}C_2\}]/{}_{m(2^m-1)}C_4$ |

### 4.3.2　拡大ハミング符号生成多項式に基づくＭＩＳＲ

ここでは，ＭＩＳＲに用いられる多項式として，次式で表される多項式について考える［１０］。

$$G(X)=(1+X)G'(X) \qquad (4-36)$$

ここで，$G'(X)$は$(m-1)$次の原始多項式とする。$G'(X)$の周期は$(2^{m-1}-1)$である。$G(x)$はｍ次の多項式となり，$(1+X)$，$G'(X)$ともに重根を持たないので，$G(X)$の周期は$(2^{m-1}-1)$となる［１１］。$G(X)$は拡大ハミング符号を生成する。拡大ハミング符号は重み３以下の符号語を含まないので，$G(x)$は次数$(2^{m-1}-1)$以下の単項，２項，３項多項式を整除しない。

以下，式(4-36)に基づくＭＩＳＲの誤り検出率について議論する。4.3.1項では，次数がｍ，周期ｎが$(2^{m}-1)$の原始多項式$G(X)$を用いた。ここでは，$G(X)$の次数は同じくｍであるが，周期ｎは$(2^{m-1}-1)$となる。4.3.1における$G$等価に関する議論は，$n=(2^{m-1}-1)$とすることにより，そのまま成り立つ。

#### 4.3.2.1　単一誤りが生じた場合

4.3.1.1の議論が成立し，$PD_1=1$となる。

#### 4.3.2.2　２重誤りが生じた場合

4.3.1.2の議論が成立し，式(4-25)に$n=(2^{m-1}-1)$を代入して次式を得る。

$$PM_2=1/2\times m(m-1)(2^{m-1}-1)/{}_{m(2^{m-1}-1)}C_2 \qquad (4-37)$$

#### 4.3.2.3　３重誤りが生じた場合

検査シンドローム$Y(X)$$(\equiv DG(X)\times X^{m-1} \mod G(X))$は，次数$(2^{m-1}-1)$以下でかつ単項又は３項を含む多項式となる。$G(X)$はこのような多項式を整除しないので，３重誤りをすべて検出する。

#### 4.3.2.4　４重誤りが生じた場合

検査シンドローム$Y(X)$は，０，２，４項からなる多項式となる。次の3通りに分ける。

(a)　G等価対が２組含まれる場合

検査シンドローム$Y(X)\equiv 0 \mod G(X)$，となりこの４重誤りを見逃してしまう。

場合の数は補題4.3で与えられる。

(b)　G等価対が1組だけ含まれる場合

検査シンドロームY(X)は次数が($2^{m-1}-1$)以下の2項を含む多項式となる。G(X)はこのような多項式を整除しないので，この4重誤りを検出する。

(c)　G等価対が含まれない場合

検査シンドロームY(X)は4項を含む多項式となる。Y(X)がG(X)で割り切れる場合の数は，G(X)が生成する拡大ハミング符号に含まれる重み4の符号語数に等しい。これは，拡大ハミング符号のパリティ検査行列の列ベクトルのうち4個の列ベクトルの和が0となるような組み合わせの数に等しい[12]。式(4-28)の導出と同様に考えると，符号長nの拡大ハミング符号に含まれる重み4の符号語数は，

$$1/4 \times {}_nC_3 \qquad (4-38)$$

となる。4項多項式を生成するG縮退化前の行列Eの数は，補題4.4と式(4-38)より，

$$m^4 \times 1/4 \times {}_nC_3 \qquad (4-39)$$

となる。この場合4重誤りを見逃してしまう。以上(a)，(b)，(c)より，4重誤りを見逃し率$PM_4$を求める。検査入力パターンに4重誤りが生じる場合の数は，${}_{mn}C_4$であり，$n=2^{m-1}-1$を代入し次式を得る。

$$PM_4=[1/24\times m(m-1)(2^{m-1}-1)\{(m-2)(m-3)+3m(m-1)(2^{m-1}-2)\}+1/4\times m^4\times {}_{(2^{m-1}-1)}C_3\}]/{}_{m(2^{m-1}-1)}C_4 \qquad (4-40)$$

4.3.2項のまとめとして，拡大ハミング符号生成多項式（式(4-36)）に基づくMISRの誤り検出率を表4.2にまとめる。表4.2から，単一および3重誤りは必ず検出するが，2，4重誤りについては，100%は検出できないことがわかる。

表4.2　次数mの拡大ハミング符号生成多項式に基づくMISRの誤り検出率

| | |
|---|---|
| $PD_1$ | 1 |
| $PD_2$ | $1-1/2\times m(m-1)(2^{m-1}-1)/{}_{m(2^{m-1}-1)}C_2$ |
| $PD_3$ | 1 |
| $PD_4$ | $1-[1/24\times m(m-1)(2^{m-1}-1)\{(m-2)(m-3)+3m(m-1)(2^{m-1}-2)\}+1/4\times m^4\times {}_{(2^{m-1}-1)}C_3]/{}_{m(2^m-1)}C_4$ |

## 4.4　逆２重化ＭＩＳＲの提案

### 4.4.1　逆２重化ＭＩＳＲ

ＭＩＳＲの誤り検出率は，4.3節で示したように，多重誤りに対して１００％に達しないことがある。特に２重誤り検出率が１００％に達しないことは望ましくない。多重誤り検出率を向上させる方法も提案されているが，２重誤り検出率は１００％に達しない[４]。

ここでは，２重誤りを１００％検出できるようなシグナチャ回路 ―― 逆２重化ＭＩＳＲ ―― を提案する。同時に，３重誤り検出率も理論的に求める。逆２重化ＭＩＳＲという回路を説明する。この回路は，図4.6に示すように，２個のＬＦＳＲからなり，かつそれぞれのシフト方向が逆であるところに特徴がある。磁気テープシステムでは，逆方向のＬＦＳＲを設けて誤り訂正を行なっている例もある[１３] 図4.6において，ＭＩＳＲＧ／ＭＩＳＲＨはそれぞれｍ次の多項式Ｇ(Ｘ)／Ｈ(Ｘ)に基づくＬＦＳＲであり，それぞれシグナチャＳＧ(Ｘ)／ＳＨ(Ｘ)を生成する。4.2節の式(4-5)と同様，

$$SG(X)\equiv a_0(X)+Xa_1(X)+\cdots+X^{m-1}a_{m-1}(X) \mod G(X) \qquad (4\text{-}41)$$

となる。また，ＭＩＳＲＨのシフト方向が逆であることより，次式が成立する。

$$SH(X)\equiv X^{m-1}a_0(X)+X^{m-2}a_1(X)+\cdots+a_{m-1}(X) \mod H(X) \qquad (4\text{-}42)$$

4.2節と同様に，ＭＩＳＲＧ／ＭＩＳＲＨによる検査シンドロームＹＧ(Ｘ)／ＹＨ(Ｘ)を次の通り定義する。

$$YG(X)\equiv e_0(X)+Xe_1(X)+\cdots+X^{m-1}e_{m-1}(X) \mod G(X) \qquad (4\text{-}43)$$

$$YH(X)\equiv X^{m-1}e_0(X)+X^{m-2}e_1(X)+\cdots+e_{m-1}(X) \mod G(X)$$

故障判定回路は，ＹＧ(Ｘ)＝ＹＨ(Ｘ)＝０のとき誤りなし，ＹＧ(Ｘ)≠０またはＹＨ(Ｘ)≠０のとき誤りありとみなす。

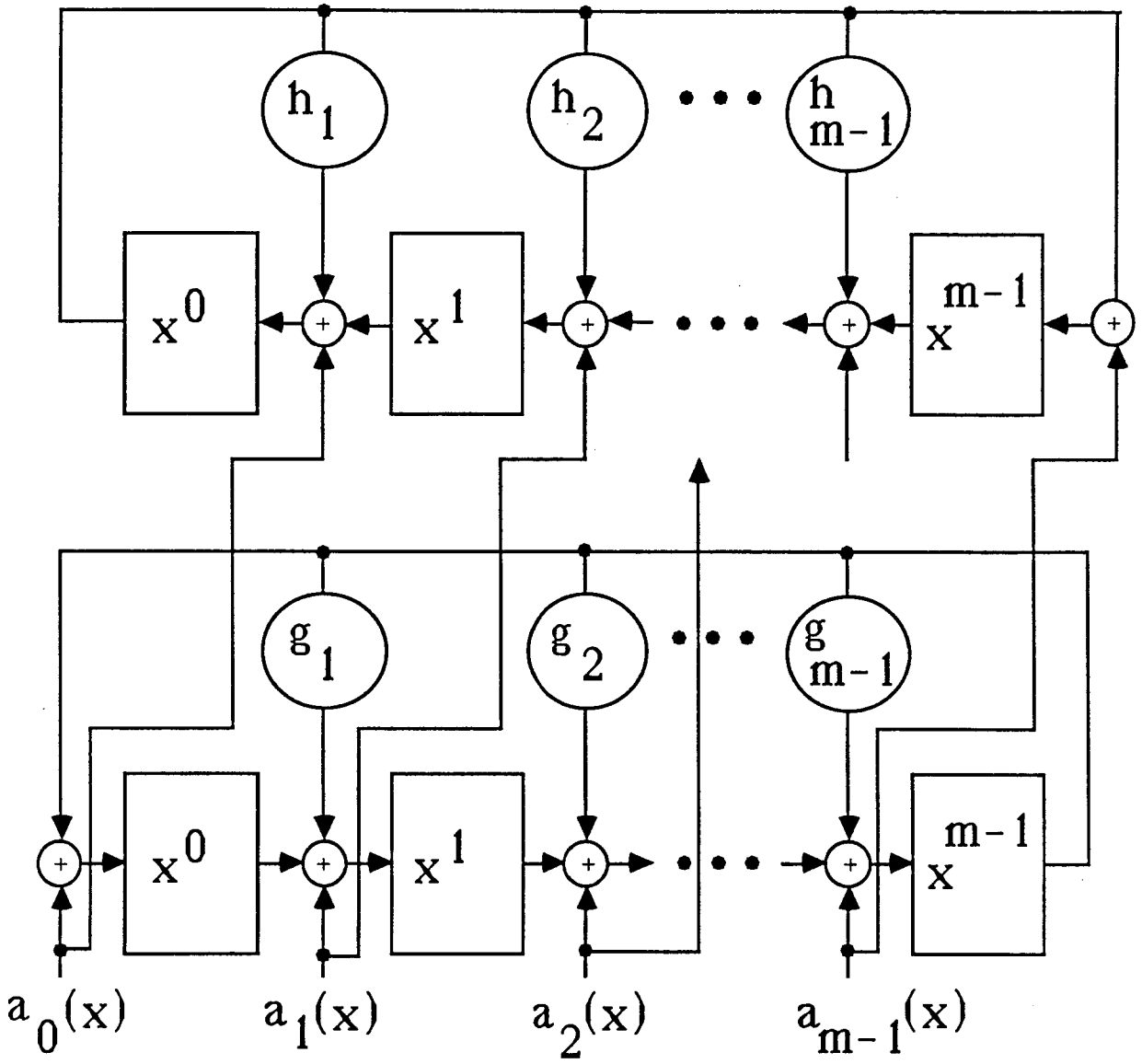



図4.6　逆2重化ＭＩＳＲ回路

### 4.4.2　H等価とその諸性質

入力検査パターン行列Aの成分に対し，H等価という性質を定義する。行列Aの2成分 $a_{kl}$，$a_{k'l'}$ が次の条件，

（条件2）　$k-l \equiv k'-l' \mod n$　　（4－44）

を満たすとき，$a_{kl}$，$a_{k'l'}$ はH等価であると定義する。また，$a_{kl}$，$a_{k'l'}$ をH等価対と呼ぶ。ただし，nは多項式H（x）の周期を表す。H等価対の例を図4.7に示す。図4.7に示すように，H等価対は主対角線（左上から右下）の方向に存在している。

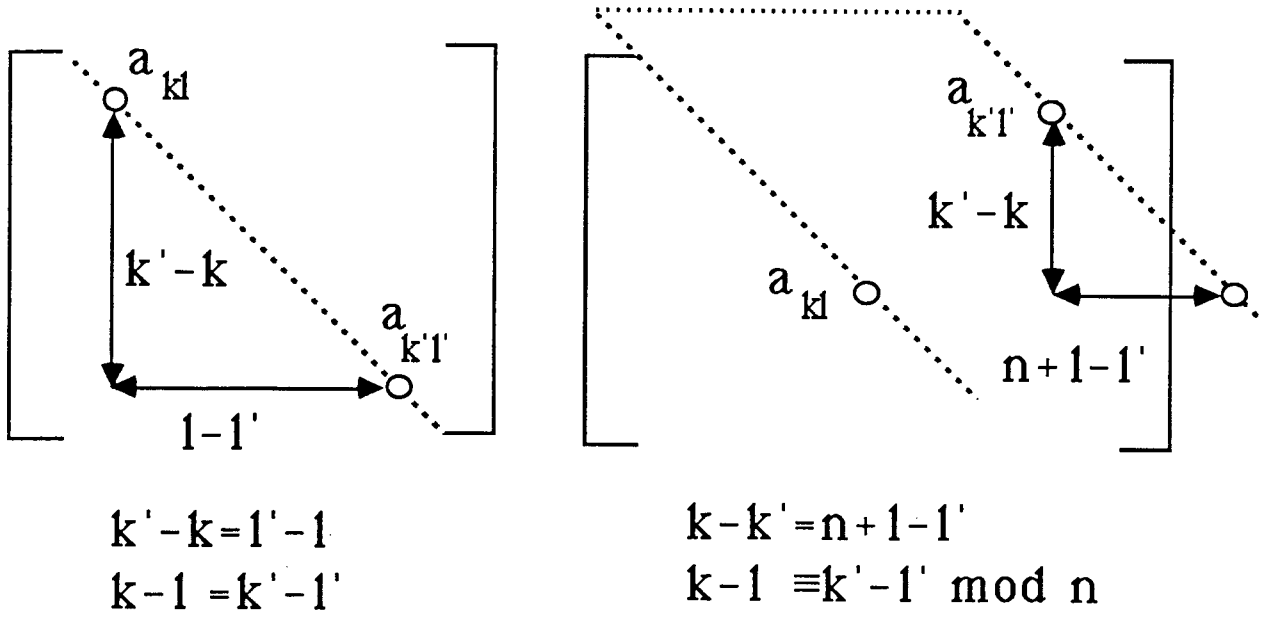



図4.7　H等価対の例

図4.4で示したG等価対の例および図4.7のH等価対の例からも明らかなように行列Aの異なる2成分が，同時にG等価かつH等価であることはない。

補題4.1，4.2，4.3において，G等価をH等価，（条件1）を（条件2）に置き換えても，補題は成立する。また，H等価の関係を用いて誤り行列Eを第（m－1）へ縮退化させることをH縮退化と呼び，このときの第（m－1）行の多項式表現をDH(X)と表す。補題4.4において，G縮退化をH縮退化，DG(X)をDH(X)に置き換えても，補題は成立する。

### 4.4.3　原始多項式に基づく逆2重化MISRの誤り検出率

最も基本的な逆2重化MISRとして，G(X)，H(X)が共にm次の原始多項式の場合について考える。G(X)，H(X)の周期は共に（$2^m-1$）となり，符号長（$2^m-1$）のハミング符号を生成する。入力検査パターンの長さを（$2^m-1$）として，以下逆2重化MISRに対する単一，2重，3重ビット誤りの検出率を求める。

#### 4.4.3.1　単一誤りが生じた場合

4.3.1.1と同じ理由により，

$$YG(X) \not\equiv 0 \mod G(X)$$

$$YH(X) \not\equiv 0 \mod G(X)$$

となり，単一誤りを検出する。

4.4.3.2　2重誤りが生じた場合

4.3.1.2と同じ理由により，YG(X)，YH(X)はそれぞれ0項または2項を含む。ただし，2成分$e_{k1}$，$e_{k'1'}$が同時に（条件1），（条件2）を満たすことはないから，YG(X)，YH(X)の少なくとも一方は非零となる。よって，2重誤りをすべて検出する。

4.4.3.3　3重誤りが生じた場合

3重誤りを見逃す場合の数は，YG(X)がG(X)で割り切れて，かつYH(X)がH(X)で割り切れる場合の数に等しい。$YG(X) \equiv 0 \mod G(X)$となる確率は，4.3.1.3の式(4-30)より，

$$1/3 \times m^3 \times {}_nC_2 / {}_{mn}C_3 \qquad (4-45)$$

である。また，$YH(X) \equiv 0 \mod H(X)$となる確率も，この式に等しい。よって，

YG(X)=YH(X)=0となる確率は，$n = 2^m - 1$と置いて，近似的に次式の通りとなる。

$$\{ 1/3 \times m^3 \times {}_{(2^m-1)}C_2 / {}_{m(2^m-1)}C_3 \}^2 \qquad (4-46)$$

4.4.3項のまとめとして，逆2重化MISRに用いる多項式G(X)，H(X)が共に原始多項式とした場合の，誤り検出率を表4.3にまとめる。

表4.3　逆2重化MISRの誤り検出率，G(x)，H(x)が共に原始多項式の場合

| | |
|---|---|
| $PD_1$ | 1 |
| $PD_2$ | 1 |
| $PD_3$ | $1-\{ 1/3 \times m^3 \times {}_{(2^m-1)}C_2 / {}_{m(2^m-1)}C_3 \}^2$ |

4.4.4　拡大ハミング符号生成多項式に基づく逆2重化MISRの誤り検出率

G(X)，H(X)がともに拡大ハミング符号を生成する多項式の場合について，逆2重化MISRの誤り検出率を求める。このとき，$G(X) = (1+X)G'(X)$，$H(X) = (1+X)H'(X)$であり，G'(X)、H'(X)はともに(m-1)次の原始多項式である。G(X)，

$H'(X)$の周期は，4.3.2項と同じ理由により，$(2^{m-1}-1)$である。以下，入力検査パターンの長さを$(2^{m-1}-1)$として，単一，2重ビット誤りの検出率を求める。

単一誤りが生じた場合は，4.3.2.1と同じ理由により，これを検出する。

2重誤りが生じた場合は，4.4.2.2と同じ理由により，これを検出する。

3重誤りが生じた場合は，4.3.2.3と同じ理由により，これを検出する。

以上，逆2重化MISRの多項式$G(X)$，$H(X)$として拡大ハミング符号生成多項式を選んだ場合，単一，2重，3重誤りをすべて検出する。

## 4.5 まとめ

本章では，まずＭＩＳＲに対してシグナチャＳ(Ｘ)を定式化した。次に，Ｇ等価という考え方を提案し，２重ビット誤りの検出率を求めた。その結果，２重ビット誤り検出率が１００％とならないことを示した。さらに，行列の縮退化という考え方を提案し，この考え方とハミング符号の重み３，４の符号語数から，ＭＩＳＲの３，４重誤り検出率を計算した。同時に，拡大ハミング符号生成多項式に基づくＭＩＳＲの１～４重ビット誤り検出率を求め，２重ビット誤り検出率が１００％とならないことを示した。

２重ビット誤り検出率を１００％にする一つの方法として，逆２重化ＭＩＳＲというシグナチャ回路を提案した。この回路は，シフト方向が互いに逆方向である２組のＭＩＳＲを用いる方法である。特に，拡大ハミング符号生成多項式に基づく逆２重化ＭＩＳＲを用いると，１～３重誤りをすべて検出できることを示した。

## 4.6 参考文献


[1] T.W.Williams and K.P.Parker, "Design for testability - a survey," IEEE Trans. Comput., vol. C-31, no.1, pp. 2-15, Jan. 1982.

[2] T.W.Williams, "VLSI testing," IEEE Computer, vol.17, no.10, pp.126-136, Oct.1984.

[3] B.Könemann, J.Mucha and G.Zwiehoff, "Built-in test for complex digital integrated circuits," IEEE J. Solid-State Circuits, vol. SC-15, no.3, pp.315-319, June 1980.

[4] J.E.Smith, "Measures of the effectiveness of fault signature analysis," IEEE Trans. Comput., vol.C-29, no.6, pp.510-514, June 1980.

[5] E.J.McCluskey, "Built-in self-test techniques," IEEE Desigh & Test of Comput., vol. 2, no.2, pp.21-28, Apr.1985.

[6] S.K.Jain and C.E.Stroud, "Built-in self testing of embedded memories," IEEE Design & Test of Comput., vol.3, no.5, pp.27-37, Oct.1986.

[7] S.Z.Hassan and E.J.McCluskey, "Increased fault coverage through multiple signatures," Dig. FTCS-14, pp.354-359, June 1984.

[8] 岩崎，山口，萩原，"ＬＳＩ自己検査用シグナチャ回路の誤り検出率の解析と新回路方式の提案，"信学論(Ｄ)，vol.J68-D, no.5, pp.1063-1070, 1985年5月.

[9] K.Iwasaki,N.Yamaguchi and T.Nishimukai, "Analysis and proposal of signature circuit for LSI testing," Proc. ITC, pp.517-522, Sept.1987.

[10] 嵩, 都倉, 岩垂, 稲垣, "符号理論" コロナ社, 1975年.

[11] 宮川, 岩垂, 今井, "符号理論," 昭晃堂, 1973年.

[12] W.W.Peterson and E.J.Weldon,Jr., "Error-correcting codes," 2nd Edition MIT Press,1972.

[13] A.M.Patel, "Optimal rectangular code for high density magnetic tapes," IBM J.R&D, vol. 18, pp 579-588, Nov.1974.

# 第5章　リード・ソロモン符号に基づく多重シンボル誤り検出シグナチャ回路

## 5.1　まえがき

ＬＳＩ検査手法の一つとして，ＭＩＳＲを用いたシグナチャ検査法が使われている[1,2]。さらに，誤り検出率を向上させる方法がいくつか提案されている[３－５]。しかし，ｄ重までの多重誤りを検出できるようなシグナチャ回路は見いだされていなかった。

前章では，ＭＩＳＲの1～4重ビット誤り検出率を求め，3重ビット誤りまでを検出できるシグナチャ回路を提案した。本章では，ＭＩＳＲに別の考え方を適用することにより，ｄ重シンボル誤りを検出できるシグナチャ回路 —— 多重化ＭＩＳＲ —— を提案する。提案するシグナチャ回路では，入力検査系列をＧＦ($2^m$)上のシンボルとみなし，リード・ソロモン符号が適用できることを利用する。前章で提案した逆2重化ＭＩＳＲ方式は，リード・ソロモン符号に基づく多重化ＭＩＳＲの一つと一致することを示す。また，リード・ソロモン符号の重み分布から，単一および多重化ＭＩＳＲの多重シンボル誤り検出率を求める。さらに，提案する多重化ＭＩＳＲの32ビットマイクロプロセッサへの応用についても述べる[6,7]。

また，検査時間を削減する一つの方法として，ビット幅の圧縮をおこなうシグナチャ回路を提案する。このシグナチャ回路は，ランダム誤り訂正符号と多重化ＭＩＳＲに基づくものである。この方法は次のような場合に有効である。通常のシグナチャ回路は幅がｍビット，長さがｎビットのテストパターンをｍビットのシグナチャに圧縮する。一方，ある種のマイクロプロセッサでは幅が100ビット以上のマイクロプログラムを使っている。ＬＳＩの内部バスを利用した検査法[８]という観点から見ると，シグナチャ回路のビット数は内部バスの幅と同じことが望ましい。このような応用では，マイクロプログラムＲＯＭの出力は，シグナチャ回路に，一度には，入力できない。この結果，テスト時間の増大を招いてしまう。そこで，ビット幅を圧縮した後，シグナチャ回路に入力すれば，この問題を解決できる。ビット幅を圧縮する一つの方法は，既に提案されている[９]。ここでは，より一般的なシグナチャ回路を提案する[6,10]。

## 5.2　d重シンボル誤り検出可能なシグナチャ回路の提案

### 5.2.1　LSI検査モデル

第4章では，パターン圧縮器への入力パターンは2元の時系列として扱ってきた。一方，このパターンは，図5.1に示すように，ガロア体GF($2^m$)上のシンボルの系列と見なすこともできる。すなわち，次式のように考えることができる。

$$\mathbf{a}_k = (a_{0k}, a_{1k}, \cdots, a_{m-1k}) \in GF(2^m) \qquad (5-1)$$

入力検査パターンは次式のように表すことができる。

$$\mathbf{a}(x) = \mathbf{a}_0 + \mathbf{a}_1 x + \cdots\cdots + \mathbf{a}_{n-1} x^{n-1}$$
$$\mathbf{a}_k \in GF(2^m) \qquad (5-2)$$

シンボル$\mathbf{a}_k$は検査パターン行列Aのk番目の列を表す。このため，行列Aは次のように表すことができる。

$$A = (\mathbf{a}_0, \mathbf{a}_1, \cdots\cdots, \mathbf{a}_{n-1}) \qquad (5-3)$$

次に，図5.1の検査モデルに基づき，シグナチャS(x)を定式化する。まず，シグナチャ回路として第4章の図4.2で示されるような単一MISRについて考える。この回路は，(x−α)除算回路と見なすことができる[11]。ここで，αはGF($2^m$)の原始元を表す。すなわち，単一MISRに基づくシグナチャ解析は，多項式$\mathbf{a}(x)$を(x−α)で割り算することに等しい。したがってシグナチャS(x)は，次のように計算される。

$$S(x) \equiv \mathbf{a}(x) \bmod (x-\alpha)$$
$$= \mathbf{a}(\alpha) \quad (\text{剰余定理による}) \qquad (5-4)$$

検査シンドロームY(x)も同様に計算できる。誤り行列Eに対し，k番目の列を$\mathbf{e}_k$と表す。誤りパターンは，次のように表すことができる。

$$\mathbf{e}(x) = \mathbf{e}_0 + \mathbf{e}_1 x + \cdots\cdots + \mathbf{e}_{n-1} x^{n-1}$$
$$\mathbf{e}_k \in GF(2^m) \qquad (5-5)$$

したがって，次式が成り立つ。

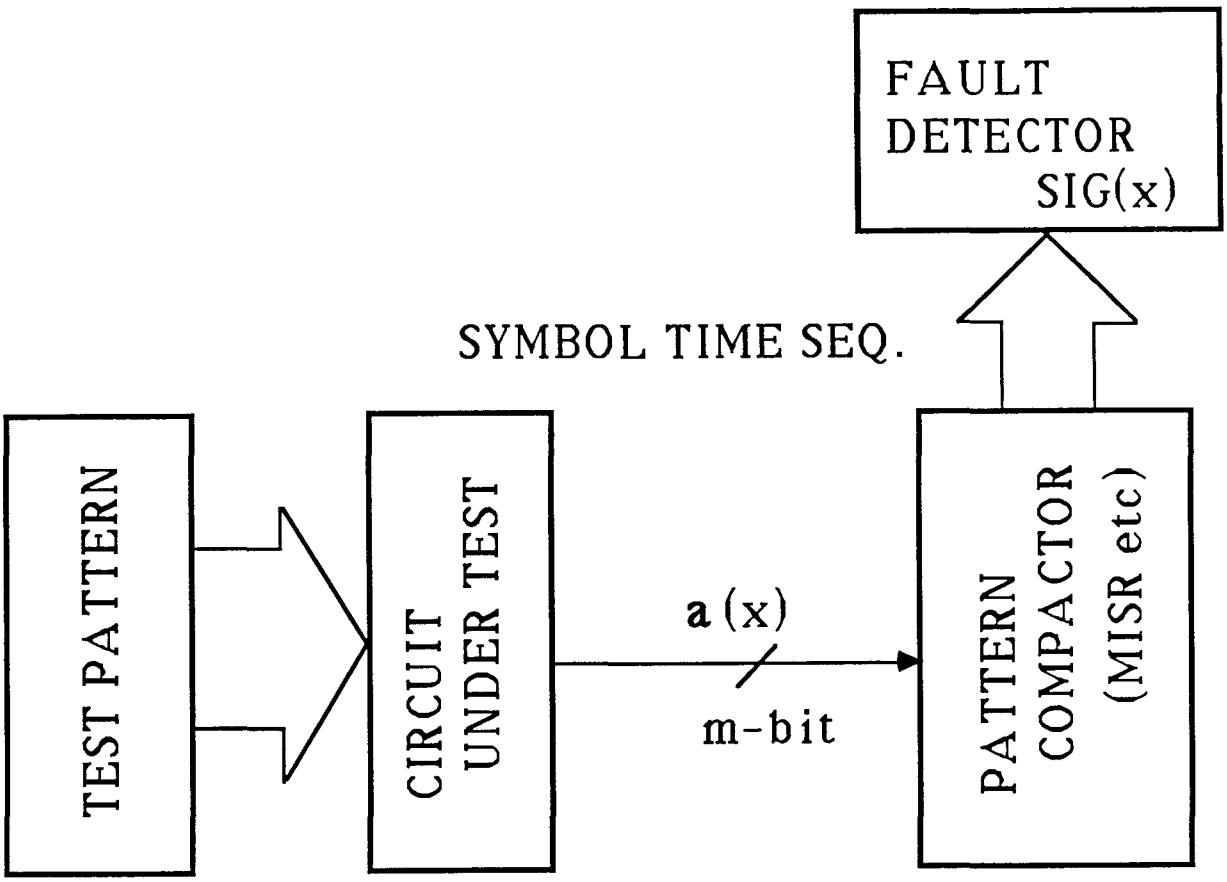



図 5.1　論理回路の検査モデル，検査パターンをGF( $2^m$ ) 上のシンボルとみなしたとき

$$Y(x) \equiv \mathbf{e}(x) \bmod (x-\alpha) = \mathbf{e}(\alpha) \quad (\text{剰余定理による}) \qquad (5-6)$$

### 5.2.2　d重化MISRの提案

式( 5－4 )をもう一度考える。この式は次式で定義されるリード・ソロモン符号のシンドローム計算と等しい。

$$G(x)=(x-\alpha) \qquad (5-7)$$

この符号のパリティシンボル数は1であり，最小シンボル距離は2である［11］。したがって，単一シンボル誤りは検出するが，2重以上のシンボル誤りは必ずしも検出できな

い。すなわち，単一シンボル誤り $\mathbf{e}_j$ に対して，

$$\begin{aligned} Y(x) &\equiv \mathbf{e}_j x^j \mod (x-\alpha) \\ &\not\equiv 0 \quad \mod (x-\alpha) \end{aligned} \qquad (5-8)$$

となり，この誤りを検出する。2重シンボル誤り $\mathbf{e}_j$, $\mathbf{e}_j'$ に対しては，

$$\begin{aligned} Y(x) &\equiv \mathbf{e}_j x^j + \mathbf{e}_j' x^{j'} \mod (x-\alpha) \\ &\not\equiv 0 \text{ または} \equiv 0 \quad \mod (x-\alpha) \end{aligned} \qquad (5-9)$$

となり，これらの誤りは必ずしも検出されない。単一MISRが検出できない2重シンボル誤りの位置の例として，第4章のG等価対が挙げられる。誤り検出能力を高めるためには，最小シンボル距離がd＋1であるようなリード・ソロモン符号を用いればよい。次の補題が成立する。

［補題 5.1 ］

$\alpha$ をGF($2^m$)の原始元とする。次のd個のMISR，すなわち($x-\alpha^b$)，($x-\alpha^{b+1}$)，……,($x-\alpha^{b+d-1}$) 除算回路をシグナチャ回路として用いるとき，dシンボルまでの誤りが検出される。

（証明）

次式を生成多項式とするリード・ソロモン符号の最小シンボル距離はd＋1である。

$$G(x)=(x-\alpha^b)(x-\alpha^{b+1})\cdots\cdots(x-\alpha^{b+d-1}) \qquad (5-10)$$

したがってdシンボルまでの誤りが検出できる。

（証明終わり）

補題5.1で示されるようなd個のMISRの組をd重化MISRと呼ぶ。d重化MISRに対し，シグナチャは次のd組の式で表される。

$$\begin{aligned} &S_b(x) \equiv \mathbf{a}(x) \mod (x-\alpha^b) \\ &S_{b+1}(x) \equiv \mathbf{a}(x) \mod (x-\alpha^{b+1}) \\ &\cdots\cdots \\ &S_{b+d-1}(x) \equiv \mathbf{a}(x) \mod (x-\alpha^{b+d-1}) \end{aligned} \qquad (5-11)$$

それぞれのシグナチャの計算は，$(x-\alpha^{b})$，$(x-\alpha^{b+1})$，$(x-\alpha^{b+d-1})$除算回路で実行される。

一例として，次式で表されるリード・ソロモン符号について考える。

$$G(x)=(x-\alpha^{0})(x-\alpha) \qquad (5-12)$$

このとき，検査シンドロームは次のようになる。

$$\begin{aligned} Y_0(x) &\equiv \mathbf{e}(x) \mod (x-\alpha^{0}) \\ Y_1(x) &\equiv \mathbf{e}(x) \mod (x-\alpha) \end{aligned} \qquad (5-13)$$

この符号の最小シンボル距離は3であるから，単一および2重シンボル誤りは検出できる。すなわち，次式が成立する。

$$\begin{aligned} &(\mathbf{e}_j x^{j} \mod (x-\alpha^{0})(x-\alpha)) \neq 0 \\ &(\mathbf{e}_j x^{j}+e_{j'} x^{j'} \mod (x-\alpha^{0})(x-\alpha)) \neq 0 \end{aligned} \qquad (5-14)$$

もう一つの例として，

$$G(x)=(x-\alpha^{-1})(x-\alpha) \qquad (5-15)$$

を生成多項式としてもつリード・ソロモン符号を考える。この符号の最小シンボル距離は3である。このとき，シグナチャの計算は，$(x-\alpha^{-1})$および$(x-\alpha)$除算回路によっておこなわれる。この2つの除算回路は，シフト方向が互いに逆であり，第4章で提案した逆2重化MISRに等価となる。

さらに，誤り検出能力を高めるためには，次式で示されるような多項式を用いればよい。

$$\begin{aligned} G(x) &= (x-\alpha^{-1})(x-\alpha^{0})(x-\alpha^{1}) \\ G(x) &= (x-\alpha^{0})(x-\alpha^{1})(x-\alpha^{2})(x-\alpha^{3}) \\ &\cdots\cdots \end{aligned} \qquad (5-16)$$

式(5－16)の多項式で定義されるリード・ソロモン符号の最小シンボル距離は，それぞれ4,5,……である。したがって，それぞれ3,4,……重シンボル誤り検出シグナチャ回路として使用できる。

最小シンボル距離d＋1のリード・ソロモン符号に基づくMISRに対して，d＋1重以

上のシンボル誤りの検出確率は，リード・ソロモン符号のシンボルに関する重み分布によって導かれる。リード・ソロモン符号は，最大距離分離符号であり，シンボル重みの分布公式は次式で表される[11]。

$$A_j = {}_nC_j \times \sum_{h=0}^{j-1-(n-k)} (-1)^h \times {}_jC_h \times (q^{j-h-(n-k)-1}) \qquad (5-17)$$

長さnの符号語に対し，j個のシンボル誤りが生じる場合の数は，${}_nC_j \times (q-1)^j$であるので，j重シンボル誤りの検出確率$PD_j$は次式の通りとなる。

$$PD_j = 1 - A_j / {}_nC_j \times (q-1)^j \qquad (5-18)$$

例えば，m次の単一MISRの2重シンボル誤り検出確率$PD_2$は，$j=2$，$n-k=1$，$q=2^m$を代して，次のように求められる。

$$PD_2 = 1 - 1/(2^m - 1) \qquad (5-19)$$

MISRの多重ビット誤り検出率は，リード・ソロモン符号の2元に関する重み分布に依存する。現在までのところ，リード・ソロモン符号の2元重み分布は，比較的簡単な生成多項式について解析がおこなわれている[15]。一般の場合については未解決の問題である。

以上，シグナチャ多項式と誤り検出能力の関係について表5.1にまとめる。

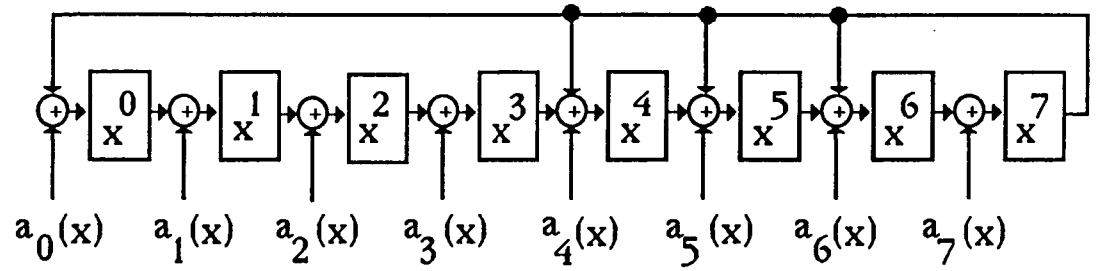
**表5.1　シグナチャ多項式と誤り検出率**

| シグナチャ多項式 | 誤り検出率 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ランダム誤り | | | | シンボル誤り | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| $(x-\alpha)$ | 1.0 | * | * | * | 1.0 | *** | *** | *** |
| $(x-\alpha^{-1})(x-\alpha)$ | 1.0 | 1.0 | ** | ** | 1.0 | 1.0 | *** | *** |
| $(x-\alpha^{-1})(x-\alpha^0)(x-\alpha)$ | 1.0 | 1.0 | 1.0 | ** | 1.0 | 1.0 | 1.0 | *** |
| $(x-\alpha^{-1})(x-\alpha^0)(x-\alpha)(x-\alpha^2)$ | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

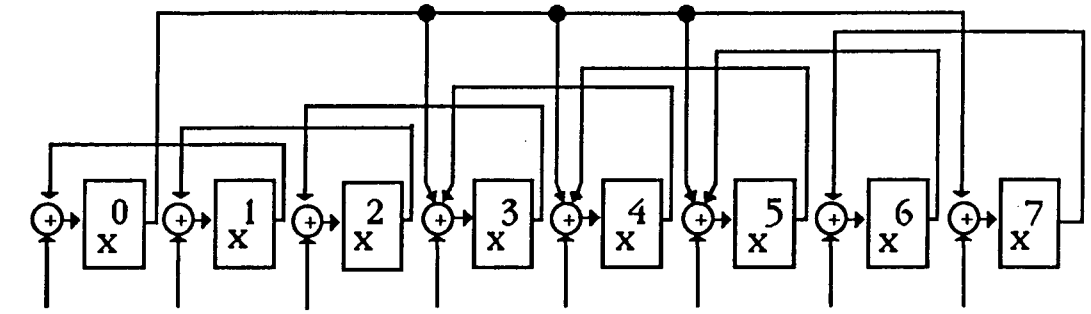
* 本論文第4章で計算した

** リード・ソロモン符号の2元重み分布による

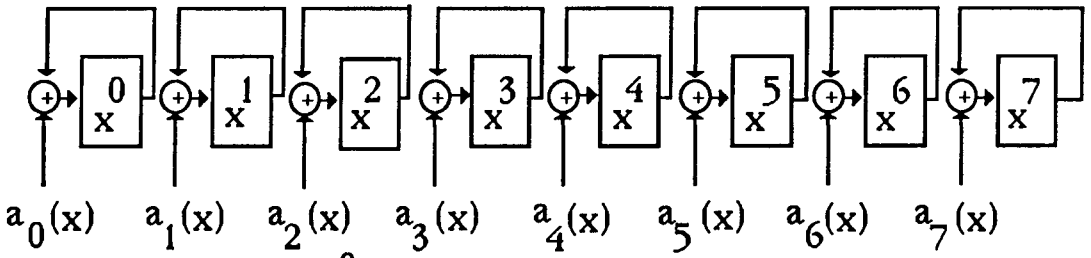
*** リード・ソロモン符号のシンボル重み分布による。本章式(5-17)(5-18)を用い計算できる



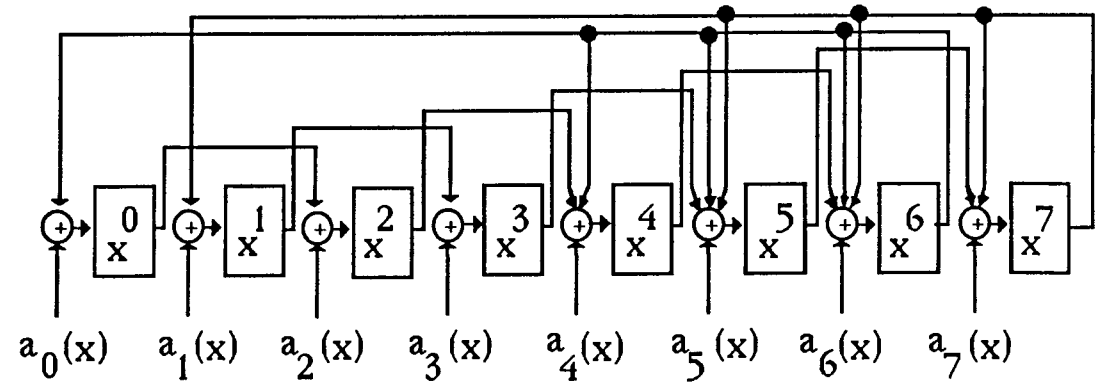

(a) $(x-\alpha)$ division circuit




(b) $(x-\alpha^{-1})$ division circuit




(c) $(x-\alpha^{0})$ division circuit




(d) $(x-\alpha^{2})$ division circuit

図 5.2　4種類のＭＩＳＲから成るシグナチャ回路の構成例

$\alpha$ は $1+x^4+x^5+x^6+x^8=0$ の根

### 5.2.3　多重化ＭＩＳＲの例

以下に示される８次の原始多項式を考える。

$$G(x)=1+x^4+x^5+x^6+x^8 \qquad (5-20)$$

このとき，検査パターンはGF($2^8$)上のシンボルとなる。図5.2(a)に($x-\alpha$)除算回路を示す。この回路は通常のＭＩＳＲ回路と等しい。($x-\alpha^{-1}$)除算回路は，図5.2(b)に示されるように，($x-\alpha$)除算回路とシフト方向が逆のシフトレジスタとして構成される。これら２個のＭＩＳＲがシグナチャ回路として用いられるならば，単一および２重シンボル誤りが検出できる。($x-\alpha^0$)除算回路は，図5.2(c)に示されるように，ビット位置毎の奇偶検査回路である。図5.2(a), (b), (c)で示される３個のＭＩＳＲをシグナチャ回路として用いられるならば，単一，２重および３重シンボル誤りが検出できる。さらに，図5.2で示される４個のＭＩＳＲをシグナチャ回路として用いるならば，４重シンボルまでの誤りが検出できる。

## 5.3　ビット幅圧縮シグナチャ回路の提案

### 5.3.1　検査モデル

この節では，図5.3で示されるような検査モデルについて考える。このモデルでは，幅mビット，長さnビットの検査パターンが，幅rビット，長さnビットの2元パターンに圧縮される。この圧縮されたパターンが多重化MISRに入力される。ビット幅圧縮回路は，ランダム誤り訂正符号に基づいており[10, 16]，多重化MISRは前節で提案したものである。

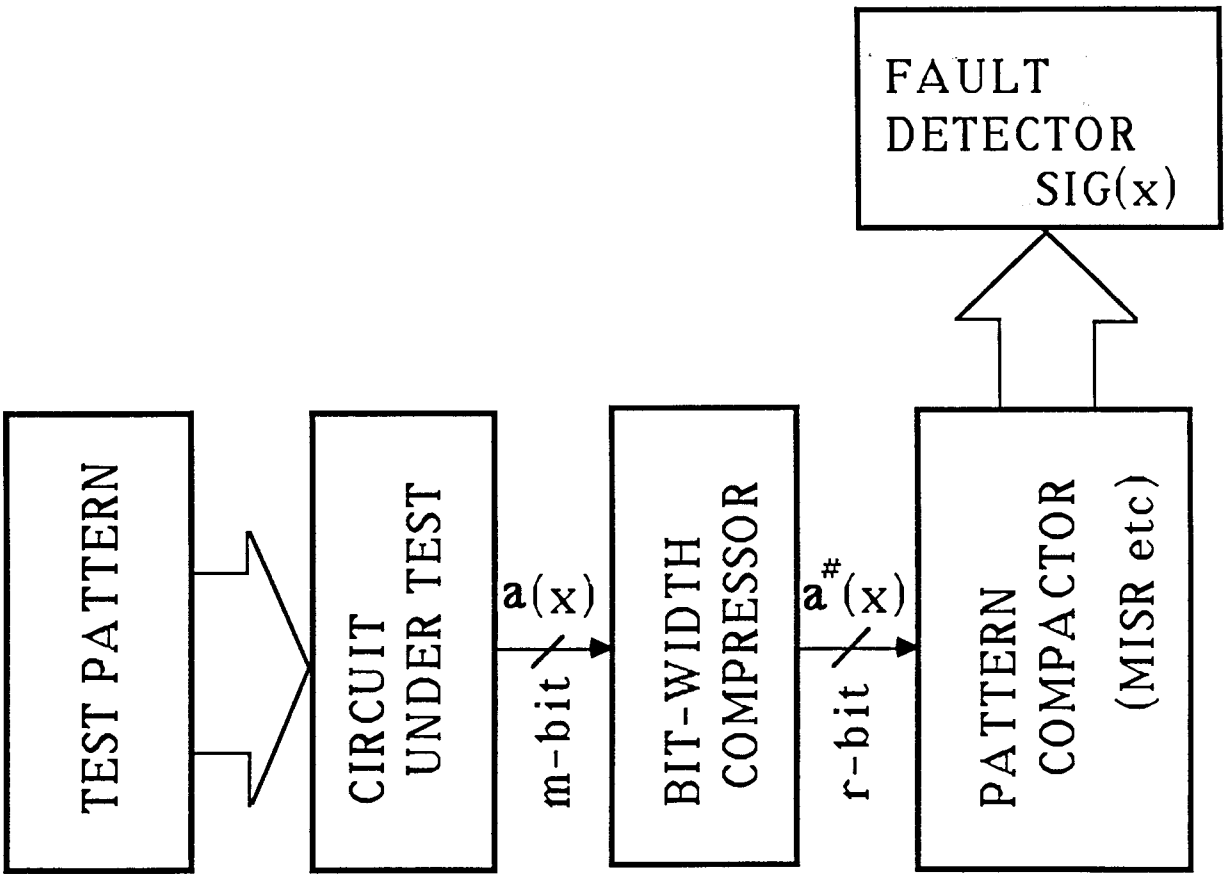



図5.3　ビット幅圧縮をおこなうシグナチャ回路の検査モデル

タイムスロットkにおいて，幅mビットのベクトル $\mathbf{a}_k$

$$\mathbf{a}_k=(a_{0k}, a_{1k}, \cdots\cdots, a_{m-1k}) \qquad (5-21)$$

はrビットのベクトル $\mathbf{a}^{\#}_k$ に圧縮される。

$$\mathbf{a}^{\#}_{k}=(\mathbf{a}^{\#}_{0k},\ \mathbf{a}^{\#}_{1k},\ \cdots\cdots,\ \mathbf{a}^{\#}_{m-1k}) \qquad (5-22)$$

したがって，ビット幅圧縮回路の出力を多項式$\mathbf{a}^{\#}(x)$と表すとき，次式が成立する。

$$\mathbf{a}^{\#}(x)=\mathbf{a}^{\#}_{0}+\mathbf{a}^{\#}_{1}x+\cdots\cdots+\mathbf{a}^{\#}_{n-1}x^{n-1}$$
$$\mathbf{a}^{\#}_{k}\in GF(2^{r}) \qquad (5-23)$$

ベクトル$\mathbf{a}_{k}$が単一ビット誤りを含む場合でも，ベクトル$\mathbf{a}^{\#}_{k}$はｒビットまでの誤りを含むことがある。式（5－23）はシンボルの系列を表しており，多重化ＭＩＳＲが適用できる。検査パターンに含まれる誤りパターンを次式で表す。

$$\mathbf{e}^{\#}(x)=\mathbf{e}^{\#}_{0}+\mathbf{e}^{\#}_{1}x+\cdots\cdots+\mathbf{e}^{\#}_{n-1}x^{n-1}$$
$$\mathbf{e}^{\#}_{k}\in GF(2^{r}) \qquad (5-24)$$

ここで，$\mathbf{e}^{\#}_{k}$はビット幅圧縮回路の出力に含まれる，タイムスロットｋにおける，誤りパターンを表す。この検査モデルにおける検査シンドロームは，（ｄ－１）重化ＭＩＳＲを用いるとし，次にように表される。

$$\begin{aligned}
&Y_{b}(x)\equiv\mathbf{e}^{\#}(x)\ \bmod(x-\alpha^{b})\\
&Y_{b+1}(x)\equiv\mathbf{e}^{\#}(x)\ \bmod(x-\alpha^{b+1})\\
&\cdots\cdots\\
&Y_{b+d-2}(x)\equiv\mathbf{e}^{\#}(x)\ \bmod(x-\alpha^{b+d-2})
\end{aligned} \qquad (5-25)$$

### 5.3.2　ビット幅圧縮シグナチャ回路の誤り検出率

符号長ｍ，情報点数（ｍ－ｒ）の線形符号を考える。この符号のパリティ検査行列Ｈは次のように表される。

$$H=\begin{bmatrix} h_{00} & h_{01} & \cdots\cdots & h_{0m-1}\\ h_{10} & & & \\ & & \cdots\cdots & \\ h_{r-10} & & \cdots\cdots & h_{r-1m-1} \end{bmatrix} \qquad (5-26)$$

次式が成立するように，ビット幅圧縮回路を構成することができる。

$$\mathbf{a}^{\#}_{k} = \mathbf{a}_{k} H^{T} \qquad (5-27)$$

ここで，$H^{T}$は行列Hの転置行列を表す。すなわち，ビット幅圧縮回路は次式のように構成することができる。

$$\begin{aligned} a^{\#}_{0k} &= a_{0k} h_{00} \oplus a_{1k} h_{01} \oplus \cdots\cdots \oplus a_{m-1k} h_{0m-1} \\ &\cdots\cdots \\ a^{\#}_{r-1k} &= a_{0k} h_{r-10} \oplus a_{1k} h_{r-11} \oplus \cdots\cdots \oplus a_{m-1k} h_{r-1m-1} \end{aligned} \qquad (5-28)$$

この場合，$\mathbf{a}^{\#}_{k}$は符号語$\mathbf{a}_{k}$のシンドロームと考えることができる。式（5－26）のパリティ検査行列により生成される符号の最小ハミング距離をdとする。このとき，$\mathbf{a}_{k}$に含まれる（d－1）個までのビット誤りが検出可能である。すなわち，$\mathbf{e}^{\#}_{k}$に含まれる1の数が（d－1）以下のとき次式が成り立つ。

$$\mathbf{e}^{\#}_{k} = \mathbf{e}_{k} H^{T} \neq 0 \qquad (5-29)$$

式（5－25）（5－29）より次の補題が成り立つ。

［補題 5.2］

最小ハミング距離がdであるようなランダム誤り訂正符号に基づくビット幅圧縮回路と（d－1）重化MISRからなるシグナチャ回路を考える。このシグナチャ回路は検査パターンに含まれる（d－1）ビットまでの誤りを検出する。

（証明略）

### 5.3.3 ビット幅圧縮シグナチャ回路の構成例

一例として，ビット幅が136ビットであるようなマイクロプログラムROMのシグナチャ解析を考える。単一および2重ビット誤りを検出するものとする。補題5.2より，ビット幅圧縮回路は，最小ハミング距離が3であるハミング符号に基づく必要がある。同時に2重化MISRが必要である。このシグナチャ回路によって，2重ビットまでの誤りを検出できる。

ビット幅圧縮回路は136ビット幅のパターンを8ビットにまで圧縮することができる。また，ハードウェアの面からは，式（5－26）のパリティ検査行列Hに含まれる1の総数

は少ないほうが望ましい。次の3通りの圧縮幅について考える。

(1) 8ビット幅への圧縮

パリティ検査行列Hは，255個の非零ベクトルから選ばれた136個の列ベクトルからなっている。重み1，2，3，4の非零ベクトルは，それぞれ8，28，56，70個存在する。重み1，2，3の非零ベクトルすべてと，重み4の非零ベクトル44個を選択する場合，機構化に必要な排他的論理和回路の総数は次式で表される。

$$1\times 8+2\times 28+3\times 36+4\times 44-8=400$$

(2) 16ビット幅への圧縮

重み1，2の非零ベクトルは，それぞれ16，120個存在する。これらをパリティ検査行列の列ベクトルとして選択すればよい。この場合，

$$1\times 16+2\times 120-16=240$$

個の排他的論理和回路が必要である。

(3) 32ビット幅への圧縮

重み1，2の非零ベクトルは，それぞれ32，1496個存在する。重み1の非零ベクトルと，重み2の非零ベクトルを104個選択する場合，機構化に必要な排他的論理和回路の総数は次式で表される。

$$1\times 32+2\times 104-32=208$$

それぞれの場合について，8ビット，16ビットまたは32ビットの2重化MISRが必要である。シグナチャ回路を構成するために必要なゲート数の例を表5.2にまとめる。排他的論理和回路は2ゲートからなるものとし，MISRの一段は入力の排他的論理和回路を含め10ゲートからなるものとして計算した。また，簡単化のため帰還用の排他的論理回路は無視した。表5.2に示されるように，ハードウェアの面からは16ビット幅への圧縮がが望ましい。しかし，VLSIチップが32ビットの内部バスをもつ場合，32ビットへの圧縮が現実的である。いずれの圧縮幅を採用しても，圧縮しない場合に比べ，検査時間は大幅に短縮される。

表5.2　ビット幅圧縮をおこなうシグナチャ回路の機構化に要するハードウェア量の例

| 圧縮幅 | EOR回路数 | MISRの次数 | 総ゲート数 |
|---|---|---|---|
| 8 | 400 | 8×2 | 960 |
| 16 | 240 | 16×2 | 800 |
| 32 | 208 | 32×2 | 1056 |
| 136<br>（圧縮しない） | 0 | 136×2 | 2720 |

## 5.4 VLSIプロセッサ（32ビットCPU）への応用

本章で提案した多重化MISRを32ビットマイクロプロセッサVLSIへ応用した。次の3種類の多項式を用いた。

(1) 32ビット原始多項式

(2) 32ビットファイヤ符号生成多項式

(3) 8ビットリード・ソロモン符号に基づく多重化MISR

- $(x-\alpha^{-1})$ 除算回路
- $(x-\alpha^{0})$ 除算回路
- $(x-\alpha)$ 除算回路
- $(x-\alpha^{2})$ 除算回路

いずれの多項式に基づいてシグナチャ解析をおこなうかは，内部レジスタ（テストモードレジスタ）で指定される。このレジスタへは外部ピンの制御によりスキャン・インができる。多重誤りの検出をおこなうためには，複数回のシグナチャ解析が必要である。上記シグナチャ回路はチップ内のテストバスに接続されている。

図5.4に，VLSIのチップ写真を示す。総トランジスタ数は，約73万トランジスタである。このうちシグナチャ回路は約2000トランジスタ（0.27％）を占めるに過ぎない。なお，自動配置配線を用いたレイアウトをおこなっているため，どの部分がシグナチャ回路であるかの特定はできない。

なお，このVLSIはTRON（The Real-time Operating system Nucleus）アーキテクチャ[15]に基づく32ビットマイクロプロセッサである。

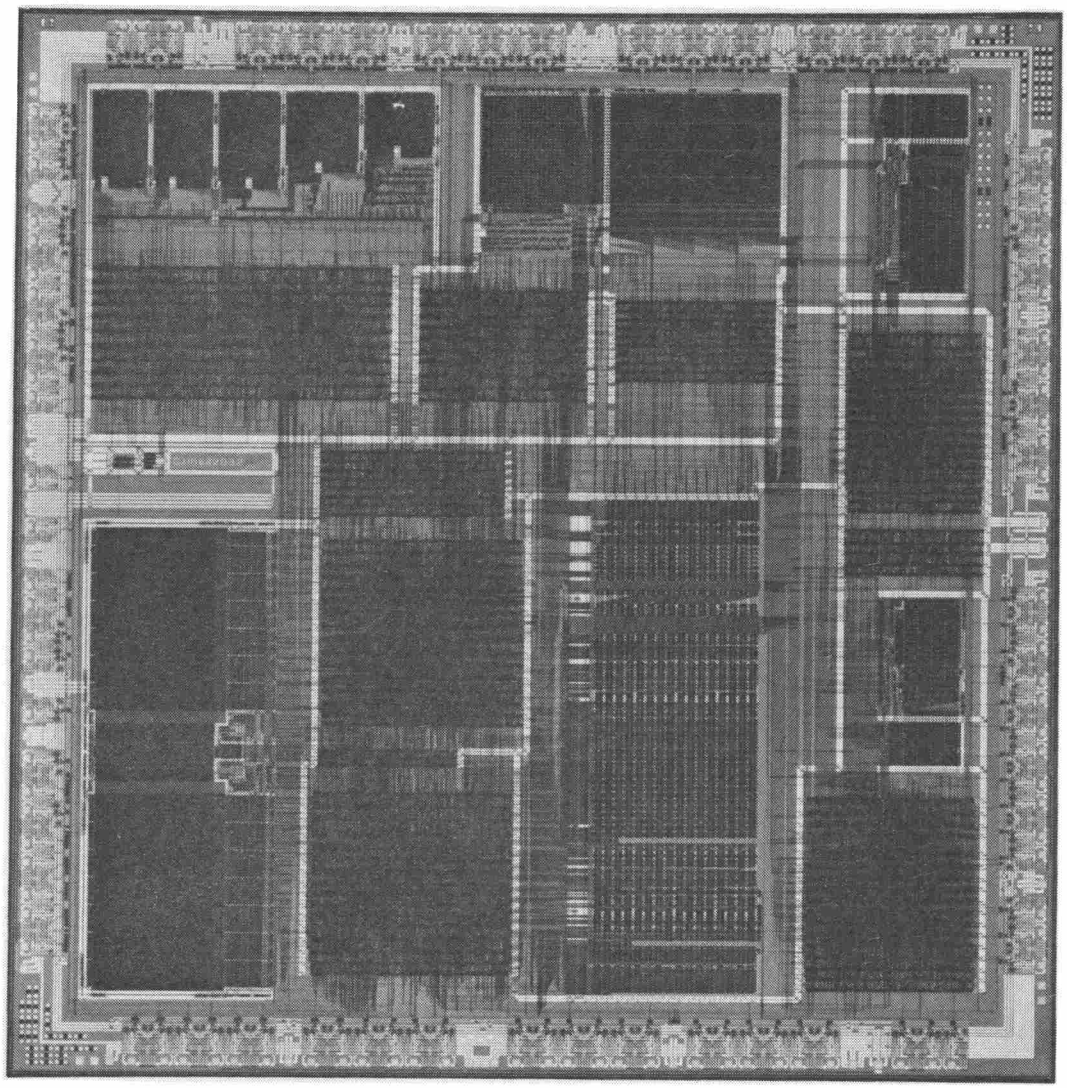


図 5.4　32 ビットマイクロプロセッサのチップ写真

## 5.5 まとめ

本章では，d重シンボル誤りを検出できるシグナチャ回路——多重化MISR ——を提案した。提案したシグナチャ回路では，入力検査系列をGF($2^m$)上のシンボルとみなし，リード・ソロモン符号を適用した。また，リード・ソロモン符号の重み分布から，単一および多重化MISRの多重シンボル誤り検出率を求めた。さらに，提案する多重化MISRの32ビットマイクロプロセッサへの応用についても述べた。

また，検査時間を削減する一つの方法として，ビット幅の圧縮をおこなうシグナチャ回路を提案した。このシグナチャ回路は，ランダム誤り訂正符号と多重化MISRに基づくものである。この方法は，水平形マイクロプログラムROMのシグナチャ検査などに有効である。ビット幅を圧縮する一つの方法は，既に提案されている。ここでは，より一般的なシグナチャ回路を提案した。

## 5.6 参考文献


[1] T. W. Williams, "VLSI testing," IEEE Computer, vol. 17, no. 10, pp. 126-136, Oct. 1984.

[2] S. K. Jain and C. E. Stroud, "Built-in self testing of embedded memories," IEEE Design & Test of Comput., vol. 3, no. 5, pp. 27-37, Oct. 1986.

[3] S. Z. Hassan and E. J. McCluskey, "Increased fault coverage through multiple signatures," Dig. FTCS-14, pp. 354-359, June 1984.

[4] W. C. Carter, "Improved parallel signature checkers/analyzers," Dig. FTCS-16, pp. 416-421, June 1986.

[5] M. Karpovsky and P. Nagvajara, "Optimal time and space compression of test responses for VLSI devices," Proc. ITC, pp. 523-529, Sept. 1987.

[6] K. Iwasaki, N. Yamaguchi and T. Nishimukai, "Analysis and proposal of signature circuit for LSI testing," Proc. ITC, pp. 517-522, Sept. 1987.

[7] K. Iwasaki, "Analysis and proposal of signature circuits for LSI testing," to appear in IEEE Trans. CAD special issue on

testing, Jan. 1988.

[8] N. Yamaguchi, T. Fuanbashi, K. Iwasaki, T. Shimura, Y. Hagiwara and K. Minorikawa, "A self-testing method for modular structured logic VLSIs," Int. conf. CAD, pp.99-101, U.S.A., Jan. 1984.

[9] R. David, "Signature analysis for multiple-output circuits," IEEE Trans. Comput., vol. C-35, no. 9, pp. 830-837, Sept. 1986.

[10] 岩崎, 荒川, 山口, 三科, "シグナチャ解析用回路の別の一提案," 電子情報通信学会フォールト・トレラント・システム研究会, FTS87-3, 1987年5月.

[11] W.W. Peterson and E.J. Weldon, Jr., "Error-correcting codes," 2nd Edition, MIT Press, 1972.

[12] V. プレス, "符号理論," ワイリー・ジャパン社, 1984年.

[13] T. Kasami and S. Lin, "The binary weight distribution of the extended $(2^m, 2^m-4)$ code of Reed-Solomon code over $GF(2^m)$ with generator polynomial $(x-\alpha)(x-\alpha^2)(x-\alpha^3)$," in Dig. First Colloqium on Coding Theory, USSR, Aug. 1986; also to appear in J. Linear Algebra & Its Appl.

[14] 嵩, S. Lin, "$GF(2^m)$ 上の線形符号の2元重み分布について," 電子通信学会情報理論研究会, IT86-100, 1987年1月.

[15] K. Sakamura, "Architecture of the TRON VLSI CPU," IEEE Micro, vol. 7, no. 2, pp. 17-31, Apr. 1987.

# 第6章　シグナチャ検査法のエイリアス確率と符号の重み分布

## 6.1　まえがき

ＬＳＩ検査法の一つとして，シグナチャ検査法が提案され［1］,応用されている［2］。シグナチャ検査法の一つの問題点は，エイリアス誤りである。すなわち，被検査回路の出力パターンに含まれる誤りを必ずしもすべては検出できない，ことである。本章では，シグナチャ解析におけるエイリアス確率（誤り見逃し率），特に過渡的な性質を解析する。

単一入力シグナチャレジスタ（ＳＩＳＲ：　Single　Input　Signature　Register）を用いたシグナチャ解析について，T. W. Williams らは，マルコフ連鎖の確率過程とみなし，Ｚ変換を用いてエイリアス誤りを解析した［3, 4］。そこでは，原始多項式と（$1+x^k$）多項式を比較し，漸近的なエイリアス確率はどちらも $1/2^k$ であることを示した。同時に，原始多項式に基づくシグナチャ回路の方が，収束が早いことを明らかにした。この功績により，１９８６年ＩＴＣ（国際検査会議）論文賞を獲得した。また，A. Ivanov らは，線形変換の手法を用い，（$1+x^k$）多項式について，ＳＩＳＲの正確なエイリアス確率を計算した［5］。原始多項式に基づく単一および多重化ＭＩＳＲ［6］については，エイリアス誤りの過渡的性質は知られていなかった。

本章では，リード・ソロモン符号の重み分布から，単一および多重化ＭＩＳＲのエイリアス誤りを解析する［7, 8］。その結果，単一および多重化ＭＩＳＲでは，エイリアス確率は選択する原始多項式に依存しないことを示す。同時に，ＳＩＳＲで見られるような変動［3］もないことを示す。これは，リード・ソロモン符号の重み分布が，短縮化した場合も含め，滑らかであるからと考えられる。

また，計算機により短縮化ハミング符号の重み分布を求め，ＳＩＳＲのエイリアス誤りの性質を示す。その結果，ＳＩＳＲのエイリアス確率については，選択する原始多項式によって，過渡的性質が異なることを示す。また，T. W. Williams［3］とほぼ同様に，エイリアス確率に変動が見られることを確認した。これは，短縮化ハミング符号の重み分布が，滑らかでないからと考えられる。

一般に，符号の重み分布は，特定の符号あるいは特殊な符号長のみしか知られていない。リード・ソロモン符号は，短縮化した場合も含め，重み分布が知られている［9］。短縮化ハミング符号では，計算機を用いていくつかの計算がおこなわれている［10, 11］。

## 6.2　MISRのエイリアス確率とリード・ソロモン符号の重み分布

### 6.2.1　検査モデル

m入力MISRは，mビット入力の線形帰還シフトレジスタである。再度，例を図6.1に示す。多重化MISRでは，同じ原始多項式に基づく複数のMISR（除数が互いに異なる）が用いられる[6]。本章では，原始多項式に基づく単一および多重化MISRを扱う。

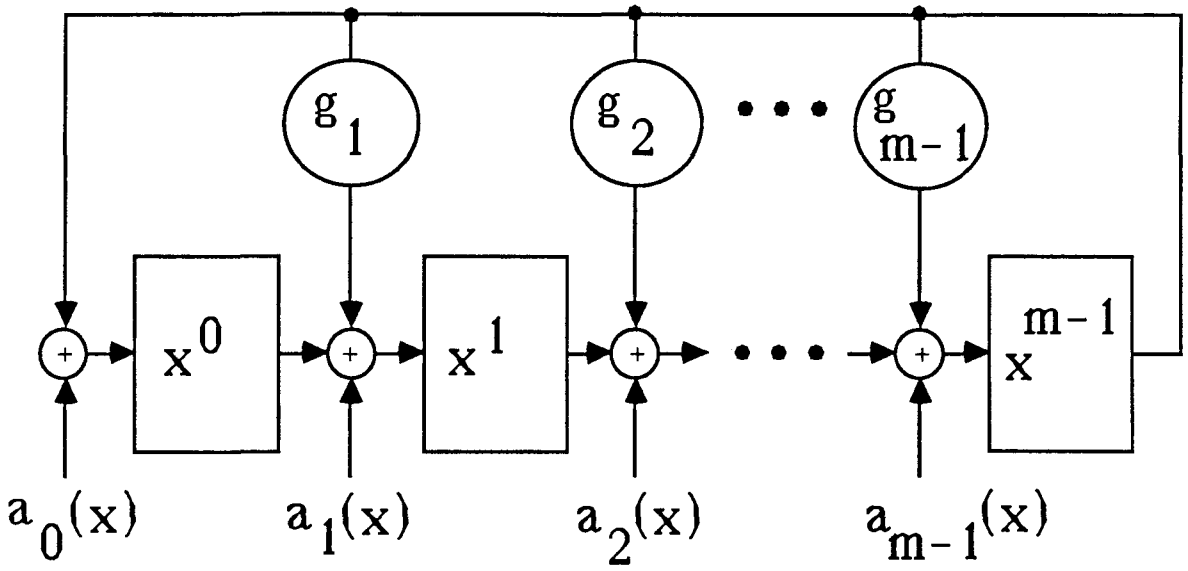



図6.1　多入力シグナチャレジスタ（MISR）回路

図6.1において，$a_i(x)$は，入力検査パターンのビット列を表す。一方，タイムスロットkにおけるmビット入力$\mathbf{a}_k$は，第5章と同様，GF($2^m$)上のシンボルとみなせる。この時，長さnの入力パターンを$\mathbf{a}(x)$，誤りパターンを$\mathbf{e}(x)$とすると，それぞれGF($2^m$)上のシンボルを係数とするn次の多項式で表される。すなわち，次式が成立する。

$$\begin{aligned}&\mathbf{a}(x)=\mathbf{a}_0+\mathbf{a}_1 x+\cdots+\mathbf{a}_{n-1}x^{n-1}\\&\mathbf{e}(x)=\mathbf{e}_0+\mathbf{e}_1 x+\cdots+\mathbf{e}_{n-1}x^{n-1}\\&\mathbf{a}_k,\ \mathbf{e}_k \in GF(2^m)\end{aligned}\qquad(6-1)$$

シンボル誤りが発生する確率，すなわち$\mathbf{e}_k \neq 0$である確率をpで表す。　GF($2^m$)には，$2^m-1$個の非零シンボルが存在する。それぞれの誤りパターンの発生確率は一様であると仮定する。このとき，各々の非零シンボルの発生確率は，$p/(2^m-1)$となる。

### 6.2.2 MISRのエイリアス誤りとリード・ソロモン符号

原始多項式に基づく単一およびd重化MISRに対して，シグナチャの計算はシンボル距離d＋1のリード・ソロモン符号のシンドローム計算に等しい[6]。シンボル距離d＋1のリード・ソロモン符号は，次の多項式で生成される。

$$G(x)=(x-\alpha^{b})(x-\alpha^{b+1})\cdots(x-\alpha^{b+d-1}) \qquad (6-2)$$

この多項式に基づくMISRのシグナチャ$S_b(x), S_{b+1}(x), \cdots, S_{b+d-1}(x)$は，次のように表される。

$$\begin{aligned} S_b(x) &\equiv \mathbf{a}(x) \mod (x-\alpha^{b}) \\ S_{b+1}(x) &\equiv \mathbf{a}(x) \mod (x-\alpha^{b+1}) \\ &\cdots \\ S_{b+d-1}(x) &\equiv \mathbf{a}(x) \mod (x-\alpha^{b+d-1}) \end{aligned} \qquad (6-3)$$

ここで，$\alpha$はGF($2^m$)の原始元を表す。エイリアス誤りが生じる必要十分条件は，シグナチャ$S_0(x)$, $S_1(x)$, …, $S_{b+d-1}(x)$がすべて期待値と一致するときである。すなわち，$\mathbf{e}(x)$が式(6-2)で生成されるリード・ソロモン符号の符号語となる場合である。この条件は次式で表される。

$$\mathbf{e}(x)\equiv 0 \mod (x-\alpha^{b})(x-\alpha^{b+1})\cdots(x-\alpha^{b+d-1}) \qquad (6-4)$$

ここで，符号長nのリード・ソロモン符号に含まれるシンボル重みjの符号語数をA(n, j)と表す。このとき，j箇所にシンボル誤りが発生して，これを見逃す確率は，

$$(p/2^m-1)^{j}\times(1-p)^{n-j}\times A(n,j) \qquad (6-5)$$

となる。よって，検査パターン長nに対するエイリアス確率Pal(n)は，次式で表される。

$$Pal(n)=\sum_{j=2}^{n} A(n,j)(p/2^m-1)^{j}(1-p)^{n-j} \qquad (6-6)$$

リード・ソロモン符号は，最大距離分離符号として知られており，重み分布は，短縮化した場合も含め，次式で表される[9]。

$$A(n,j) = {}_{n}C_{j} \sum_{h=0}^{j-1-(n-k)} (-1)^{h} {}_{j}C_{h} \left(q^{j-h-(n-k)} - 1\right) \qquad (6-7)$$

m次のd重化MISRでは，パリティシンボル数（MISRの個数）がdであるから，（n－k）＝dである。また，各シンボルはGF($2^m$)上の元であるから，q＝$2^m$である。よって，式（6－7）は次式のように表される。

$$Pal(n) = \sum_{j=d+1}^{n} \{ {}_{n}C_{j} \sum_{h=0}^{j-1-d} (-1)^{h} {}_{j}C_{h} (2^{m(j-h-d)} - 1) \} (p/2^m - 1)^{j} (1-p)^{n-j} \qquad (6-8)$$

m次の単一MISRでは，（n－k）＝1より，エイリアス確率Pal（n）は次式のように表される。

$$Pal(n) = \sum_{j=2}^{n} \{ {}_{n}C_{j} \sum_{h=0}^{j-2} (-1)^{h} {}_{j}C_{h} (2^{m(j-h-1)} - 1) \} (p/2^m - 1)^{j} (1-p)^{n-j} \qquad (6-9)$$

式（6－8）および（6－9）は，任意の原始多項式について成立する。したがって，単一および多重化MISRのエイリアス確率は選択する原始多項式に依存しない。これは，リード・ソロモン符号の重み分布が選択する原始多項式に依存しないためである。

### 6.2.3 計算例

例として，8次の原始多項式に基づく単一MISRのエイリアス確率の計算例を，図6.2に示す。シンボルの誤り確率p＝0.8とp＝0.2についてプロットした。最終的なエイリアス確率は1／256へ収束している。SISRの場合のような，変動は見られない。

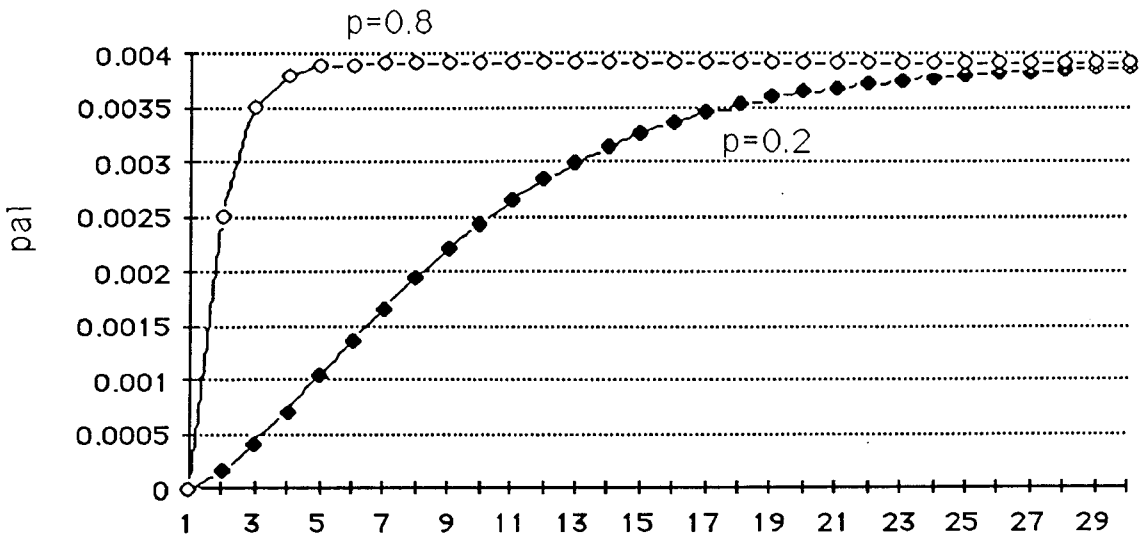



図 6.2　8 次の原始多項式に基づく単一ＭＩＳＲのエイリアス確率

8 次の原始多項式に基づく 2 重化 ＭＩＳＲ のエイリアス確率の計算例を，図 6.3 に示す。漸近値は（１／２５６）$^2$ へ収束し，その値はＭＩＳＲと異なる。しかし，過渡的性質すなわち滑らかに収束，は類似している。ＳＩＳＲ の場合のような，振動は見られない。単一および多重化ＭＩＳＲ においてエイリアス確率に変動がみられないことは，リード・ソロモン符号の重み分布が，短縮した場合も含め，滑らかであることによる，と思われる。

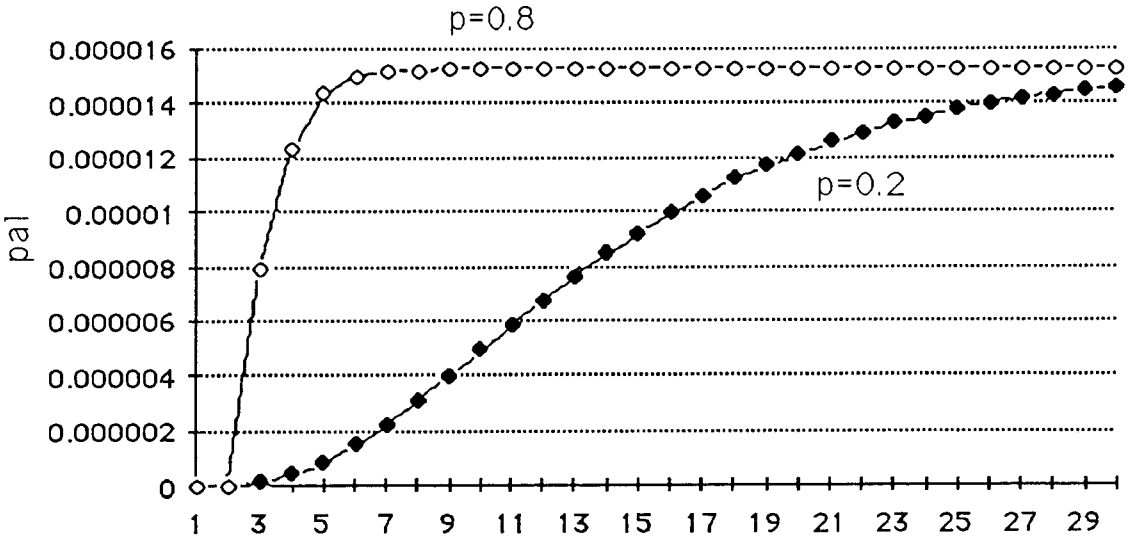



図 6.3　8 次の原始多項式に基づく 2 重化ＭＩＳＲのエイリアス確率

誤り確率 $p = 0.8$ という値は，原理的には意味があるが，実用上は頻繁には生じないと考えて差しつかえない。例えば，8入力ANDゲートの出力に1固定故障が生じた場合，$p = 255/256$ となる。LSI検査では，単一の故障により，上記のようにpの値が大きくなることがあり，注意が必要である。しかし，一般の多入力多出力回路では，pは比較的小さいと考えられる。

## 6.3　ＳＩＳＲのエイリアス確率と短縮化ハミング符号の重み分布

### 6.3.1　検査モデル

図6.4で示されるような，単一入力の線形帰還シフトレジスタ［3］をＳＩＳＲと呼ぶ。本章では，原始多項式に基づくＳＩＳＲを考える。

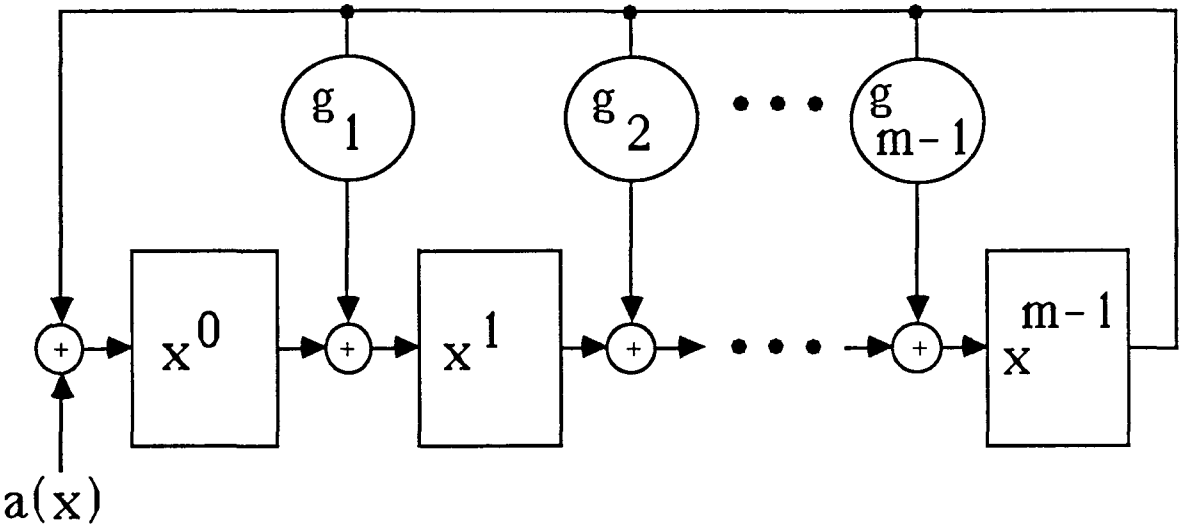



図6.4　単一入力シグナチャレジスタ（ＳＩＳＲ）回路

この回路に対し，長さｎの入力検査パターンは，ビット時系列であり，ｎ次の多項式ａ（ｘ）で表される。同様に，誤りパターンｅ(ｘ)もｎ次の多項式で表される。このとき，次式が成立する。

$$a(x) = a_0(x) + a_1(x)x + \cdots + a_{n-1}x^{n-1}$$
$$e(x) = e_0(x) + e_1(x)x + \cdots + e_{n-1}x^{n-1} \qquad (6-10)$$
$$a_k,\ e_k \in GF(2)$$

また，ａ(ｘ)の各ビットにビット誤りが生じる確率，すなわち$e_k \neq 0$である確率をｐと表す。

### 6.3.2　ＳＩＳＲのエイリアス誤りと短縮化ハミング符号

原始多項式に基づくＳＩＳＲに対し，シグナチャＳ(ｘ)の計算は，短縮化ハミング符号のシンドローム計算に等しく，次式が成立する［7］。

$$S(x) \equiv a(x) \mod G(x) \qquad (6-11)$$

エイリアス誤りが生じる必要十分条件は，S(x)が期待値と一致するときである。すなわち，誤りパターンe(x)がハミング符号の符号語となる場合である。この条件は次式で表される。

$$e(x) \equiv 0 \mod G(x) \qquad (6-12)$$

ここで，A(n, j)を，長さnの短縮化ハミング符号に含まれる2元重みjの符号語数とする。このとき，j箇所にビット誤りが発生して，これを見逃す確率は，

$$p^j \times (1-p)^{n-j} \times A(n, j) \qquad (6-13)$$

となる。したがって，エイリアス確率Pal(n)は，次式で表される。

$$Pal(n) = \sum_{j=3}^{n} A(n, j)\, p^j (1-p)^{n-j} \qquad (6-14)$$

G(x)が原始多項式のとき，ハミング符号は重み2の符号語を含まないため，A(n, 2)＝0である。

短縮化ハミング符号に対しては，A(n, j)を求める一般的な代数手法は，現在までのところ，知られていない。計算機を用いて求めることはできるが，特に大きなnに対しては，工夫を要する[10, 11]。

### 6.3.3 計算例

2通りの原始多項式を例にとり，SISRのエイリアス確率を示す。エイリアス確率を計算するためには，式(6−14)でも示されるように，短縮化ハミング符号の重み分布を求める必要がある。このためには，MacWilliamsの等式[12]を用いることが有効である。MacWilliamsの等式を次式に示す。

$$\sum_{i=0}^{n} B(n,i) \times {}_iC_r = 2^{k-r} \sum_{j=0}^{n} A(n,j) \times {}_{n-j}C_{n-r} \times (-1)^j \qquad (6-15)$$

ここで，B(n, i)は双対符号の重み分布を表す。この等式を用いると，まず双対符号の重み分布B(n, i)を求め，次いで元の符号の重み分布A(n, j)を計算することができる。特に，双対符号の重み分布が容易に求められるとき有効である。

nビットに短縮化されたハミング符号の双対符号は，nビットに短縮化された最大長系列符号である[13]。m次の原始多項式が生成する最大長系列符号の符号語数は$2^m$であ

り，元の短縮化されたハミング符号の符号語数$2^{n-m}$よりもはるかに少ない。最大長系列符号の重み分布B(n, i)は，原始多項式が生成する最大長系列に含まれる，nビット長系列の重み分布として求められる。計算されたB(n, i)とMacWilliamsの等式を用いて，短縮化ハミング符号の重み分布A(n, i)を求めた。

次の2通りの原始多項式を例にとり，エイリアス確率を計算した。結果を図5.6および図5.7に示す。p＝0.8とp＝0.2についてプロットした。

$$G(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^7 + x^8$$
$$G(x) = 1 + x^4 + x^5 + x^6 + x^8 \qquad (6-16)$$

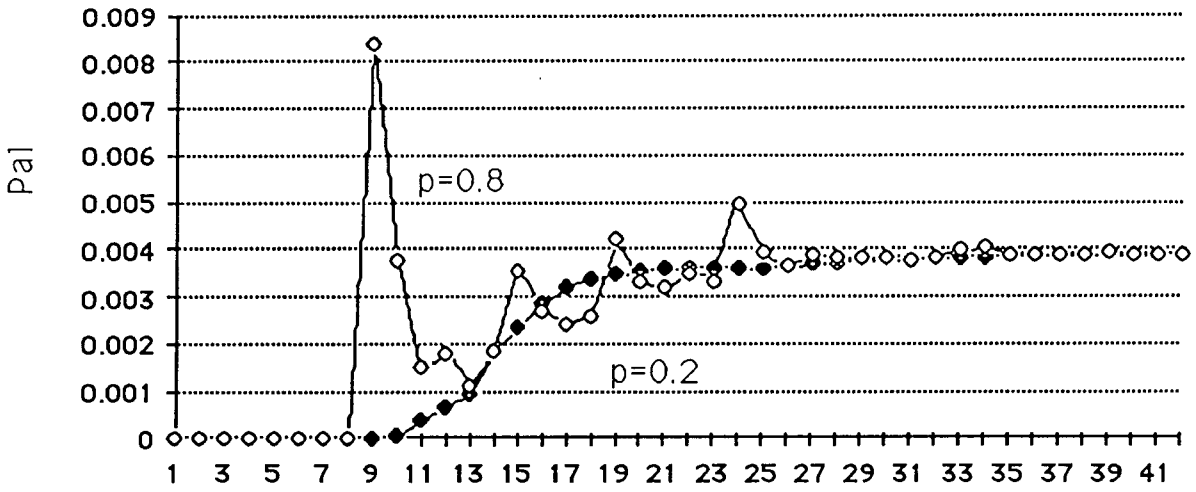



図6.5　原始多項式$G(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^7 + x^8$に基づくSISRのエイリアス確率

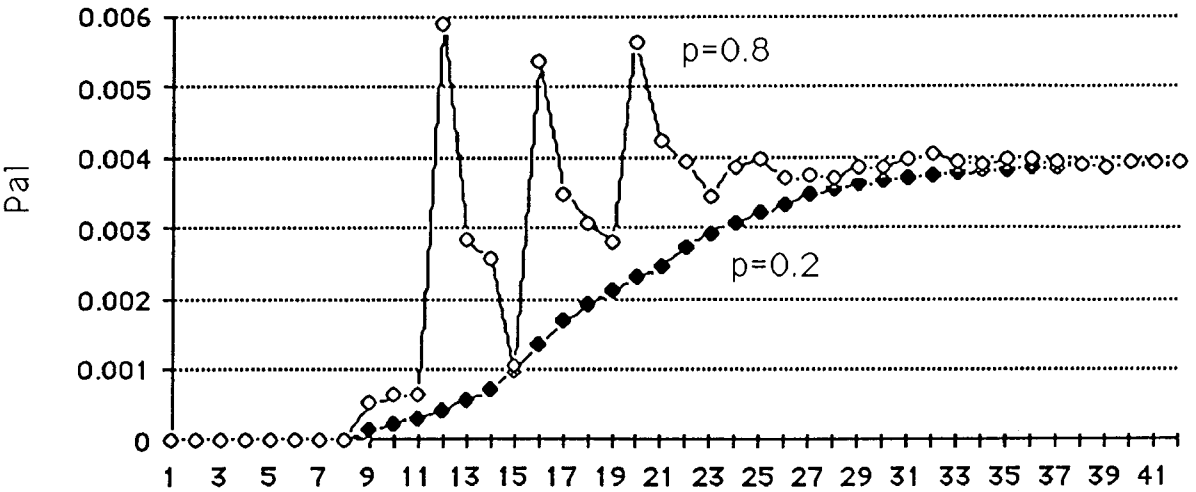



図6.6　原始多項式$G(x) = 1 + x^4 + x^5 + x^6 + x^8$に基づくSISRのエイリアス確率

これらの図では，最終値は1／256へ収束しているものの，T. W. Williams［3］と同じような変動が見られる。これは，短縮化ハミング符号では，符号の重み分布が滑らかでないから，と考えられる。また，選択する原始多項式によって，エイリアス誤りの過渡的性質が異なることがわかる。これは，短縮化ハミング符号では，符号毎に短縮化した場合の重み分布が異なるから，と考えられる。

## 6.4 まとめ

本章では，まず，リード・ソロモン符号の重み分布を利用して，単一および多重化MISR回路の誤り見逃し率（エイリアス確率）を解析した。その結果，MISRではエイリアス誤りは滑らかに収束する，ことを示した。すなわちSISRで見られるような変動はない。同時に，MISRおよび多重化MISRのエイリアス誤りは，選択する原始多項式に依存しないことを示した。

次に，SISRのエイリアス確率を解析した。この解析にあたっては，まずハミング符号の双対符号（最大長系列符号）の重み分布を，計算機を用いて求めた。さらに，MacWilliamsの等式により短縮化ハミング符号の重み分布を求め，SISRの解析をおこなった。その結果，SISRのエイリアス確率は，選択する多項式によって，異なることを確認した。T.W.Williamsの解析と同様に，エイリアス確率の過渡的振る舞いに変動があることを確認した。

## 6.5 参考文献


[1] E.J.McCluskey, "Built-in self-test techniques," IEEE Design & Test of Comput., vol.2, no.2, pp.21-28, Apr. 1985.

[2] S.K.Jain and C.E.Stroud, "Built-in self testing of embedded memories," IEEE Design & Test of Comput., vol.3, no.5, pp.27-37, Oct. 1986.

[3] T.W.Williams, W.Daehn, M.Gruetzner and C.W.Starke, "Comparison of aliasing errors for primitive and non-primitive polynomials," Proc. ITC, pp. 282-288, Sept. 1986.

[4] T.W.Williams, W.Daehn, M.Gruetzner and C.W.Starke, "Aliaing errors in signature register," IEEE Design & Test of Comput., vol.4, no.2, pp. 39-45, Apr. 1987.

[5] A.Ivanov and V.Agarwal, "On a fast method to monitor the behaviour of signature analysis registers, "Proc. ITC, pp. 645-655, Sept. 1987.

[6] K.Iwasaki, N.Yamaguchi and T.Nishimukai, "Analysis and proposal of signature circuit for LSI testing, "Proc. ITC, pp. 517-522, Sept. 1987.

[7] 岩崎，高木，"シグナチャの誤り検出率と符号の重み分布，" 電子情報通信学会フ

ォールト・トレラント・システム研究会, FTS 87-19, 1987年11月.

[8] K. Iwasaki, "Aliasing probabilities and weight distributions of several codes," submitted to FTCS, 1988.

[9] W.W. Peterson and E.J. Weldon, Jr., "Error-correcting codes," 2nd Edition, MIT Press, 1972.

[10] T. Fujiwara, T. Kasami, A. Kitai and S. Lin, "On the undetected error probability for shortened Hamming codes," IEEE Trans. Commun., vol. COM-33, no. 6, pp. 570-574, June 1985.

[11] 藤原, 嵩, S. Lin, "イーサネットで用いられる短縮巡回ハミング符号の誤り検出能力," 信学論(A), vol. J69-A, no. 6, pp. 731-735, 1986年6月.

[12] V.プレス, "符号理論," ワイリー・ジャパン社, 1984年.

[13] 嵩, 都倉, 岩垂, 稲垣, "符号理論," コロナ社, 1975年.

# 第7章　結　論

ここで，本論文をまとめる。

第2章では，修正ハミング符号の2つの構成法について論じた。1つは，非零要素の総数が最小であるようなパリティ検査行列を持つ修正ハミング符号のうちで，3重誤りの誤訂正確率，4重誤りの見逃し確率が最小の符号に関する構成法である。特に実用性の高い（72,64）修正ハミング符号に対して，計算機を用いて，最適な符号を求めた。計算にあたっては，パリティ検査行列中でより重みの小さい列ベクトルが固定できることを利用し，調べるべき場合の数を減らした。

また，記憶データの上位ビットの誤り検出率を高くするような修正ハミング符号を構成した。まず，パリティ検査行列の性質を利用して，ビット位置毎の誤り検出率を定式化した。いくつかの符号長に対し，計算機を用いて，良好な符号を見いだした。また，特定ビットの誤り検出率と符号全体の誤り検出率はトレード・オフの関係にあることを示した。

第3章では，VLSIプロセッサに適したファイヤ符号およびバースト誤り訂正巡回符号の復号法を提案した。提案した復号法は，ハードウェアが比較的少なくてよい。また，短縮化前の符号長がレコード長よりも十分に長い —— 通常の応用では成り立つ —— 場合，復号時間の面でも，他の高速復号法と同程度まで高速化することができた。また，一部の設定値を変更することにより，種々の短縮化に対して使用できるという特徴も持つ。この性質は，マイコン周辺LSIのように1個のチップで複数のレコード長を扱う場合に適しており，ROMなどの規則論理素子で実現される。

さらに，提案した復号法を具体的にVLSIプロセッサへ適用した。適用した符号は32ビットファイヤ符号であり，適用したプロセッサはハードディスクコントローラVLSIである。このVLSIチップは，16ビットマイコン周辺LSIとして開発されたものである。提案した復号法を実行する誤り訂正部は，16ビットCRC機能および誤り訂正用マイクロプログラムを含め，約6700トランジスタとなった。これは，ハードディスクコントローラVLSIの約5％を占める。また，シリアルインターフェイスVLSIの設計法についても述べた。第4章はVLSIのシグナチャ検査法に関するものである。まず，MISR

第4章はVLSIのシグナチャ検査法に関するものである。まず，MISRに対してシグナチャS(x)を定式化した。次に，G等価という考え方を提案し，2重ビット誤りの検出率を求めた。その結果，2重ビット誤り検出率を求めた。その結果，2重ビット誤り検出率

が１００％とならないことを示した。さらに，行列の縮退化という考え方を提案し，この考え方とハミング符号の重み３，４の符号語数から，ＭＩＳＲの３，４重ビット誤り検出率を計算した。同時に，拡大ハミング符号生成多項式に基づくＭＩＳＲの１～４重ビット誤り検出率を求めた。

２重ビット誤り検出率を１００％にする一つの方法として，逆２重化ＭＩＳＲというシグナチャ回路を提案した。この回路は，シフト方向が互いに逆方向である２組のＭＩＳＲを用いる方法である。特に，拡大ハミング符号生成多項式に基づく逆２重化ＭＩＳＲを用いると，１～３重ビット誤りをすべて検出できることを示した。

第５章では，ｄ重シンボル誤りを検出できるシグナチャ回路 ―― 多重化ＭＩＳＲ ―― を提案した。提案したシグナチャ回路では，入力検査系列をＧＦ($2^m$)上のシンボルとみなし，リード・ソロモン符号を適用した。また，リード・ソロモン符号の重み分布から，単一および多重化ＭＩＳＲの多重シンボル誤り検出率を求めた。さらに，提案した多重化ＭＩＳＲの３２ビットマイクロプロセッサへの応用についても述べた。

また，検査時間を削減する一つの方法として，ビット幅の圧縮をおこなうシグナチャ回路を提案した。このシグナチャ回路は，ランダム誤り訂正符号と多重化ＭＩＳＲに基づくものである。この方法は，水平形マイクロプログラムＲＯＭのシグナチャ検査などに有効である。

第６章では，まず，リード・ソロモン符号の重み分布を利用して，単一および多重化ＭＩＳＲ回路の誤り見逃し率（エイリアス確率）を解析した。その結果，ＭＩＳＲではエイリアス誤りは滑らかに収束する，ことを示した。すなわちＳＩＳＲで見られるような変動はない。同時に，ＭＩＳＲおよび多重化ＭＩＳＲのエイリアス誤りは，選択する原始多項式に依存しないことを示した。

次に，ＳＩＳＲのエイリアス確率を解析した。この解析にあたっては，まず双対符号（最大長系列符号）の重み分布を，計算機を用いて求めた。さらに，Mac Williamsの等式により短縮化ハミング符号の重み分布を求め，ＳＩＳＲの解析をおこなった。その結果，ＳＩＳＲのエイリアス確率は，選択する多項式によって，異なることを確認した。また，T. W. Williamsの解析と同様に，エイリアス確率の過渡的振る舞いに変動があることを確認した。

# 謝　辞

大阪大学基礎工学部嵩忠雄教授には，本研究をまとめるにあたり，懇切なるご指導ならびにご助言をいただき，また筆者が大阪大学基礎工学部および大学院に在学中より指導され，研究生活の端緒に導かれました。ここに心からの感謝の意を表します。また。本研究をまとめるにあたりご助言と励ましをいただき，また在学中よりご指導賜った大阪大学基礎工学部藤澤俊男教授，都倉信樹教授，鳥居宏次教授，谷口健一教授に厚く感謝いたします。

本研究は，主として日立製作所中央研究所における研究業務の一環として担当したものであります。この間，本研究の機会を与えていただき，またご指導，ご鞭達いただいた久保征治博士（もと日立製作所中央研究所第7部長），浅井彰二郎博士（日立製作所中央研究所ＵＬＳＩ総合研究センター長），増原利明博士（日立製作所中央研究所 ＵＬＳＩ総合研究センター第7部長）に心からお礼申し上げます。

本研究の過程で，電子情報通信学会フォールト・トレラント・システム研究会を通じご指導賜った広島大学総合科学部樹下行三教授，明治大学工学部藤原秀雄助教授に深謝します。また，在学中より熱心なご討論をいただいた神戸商船大学計測工学教室山村三朗助教授に感謝します。

本論文の執筆にあたり江尻正員博士（日立製作所中央研究所主幹研究長）には懇切なご指導と激励をいただきました。また，本研究を進めるにあたり，日立製作所中央研究所西向井忠彦主任研究員，同萩原吉宗主任研究員，同山口昇主任研究員には有益な討論，ご協力をいただきました。さらには，１６および３２ビットマイクロプロセッサの設計にあたり，日立製作所武蔵工場御法川和夫主任技師，同稲吉秀夫主任技師，同上野達彰主任技師，同船橋恒男グループリーダー，日立製作所茂原工場妻鹿真幸技師には懇切なご指導をいただきました。これらの方々に心から感謝いたします。

本研究の遂行，本論文の執筆は，以上の方々をはじめとする関係各位の多くの方々により支えられてきました。ここに深甚なる感謝の意を表します。